Red Hat Enterprise Linux 6

導入ガイド

デプロイメント、設定、および管理

Last Updated: 2023-03-02

Red Hat Enterprise Linux 6 導入ガイド

デプロイメント、設定、および管理

Enter your first name here. Enter your surname here.
Enter your organisation's name here. Enter your organisational division here.
Enter your email address here.

法律上の通知

概要

デプロイメントガイドでは、のデプロイメント、設定、および管理の関連情報を説明します。本書は、システムに関する基本的な理解があるシステム管理者を対象としています。

目次

パート I. システムの基本設定

ここでは、キーボード設定、日付と時刻の設定、ユーザーとグループの管理、権限の取得など基本的なシステム管理タスクを取り上げます。

第1章 キーボードの設定

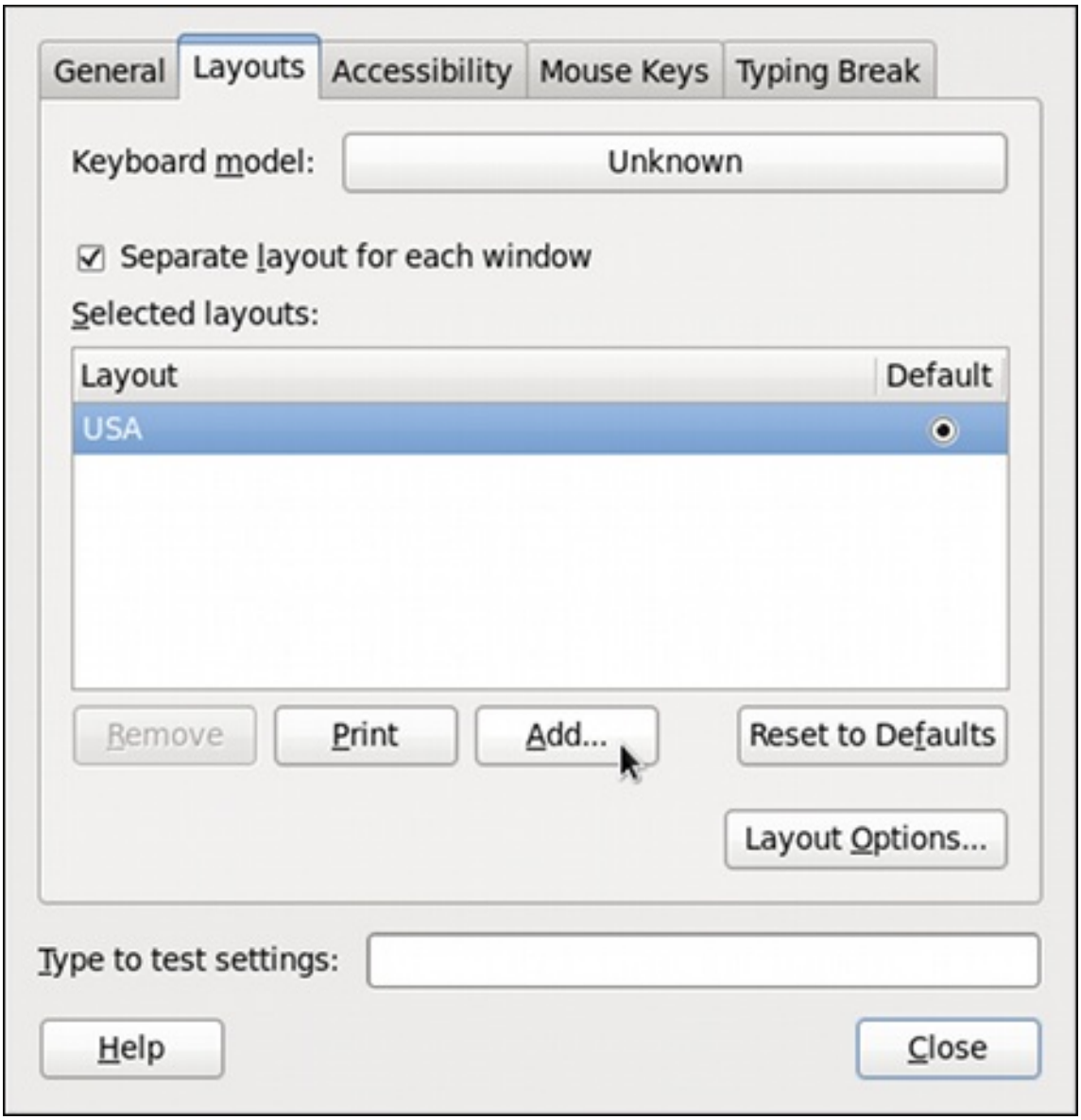
本章では、キーボードレイアウトを変更する方法と、パネルに Keyboard Indicator アプレットを追加する方法を説明します。また、強制的に休憩を設定するオプション、それによるメリットとデメリットについても取り上げます。

1.1. キーボードレイアウトの変更

インストールプログラムにより、ご使用のシステム用にキーボードのレイアウトが設定されています。しかし、デフォルト設定が必ずしも現在のニーズに合うとは限りません。インストール後に別のキーボードレイアウトを設定するには、キーボード設定 ツール **を使用** します。

Keyboard Layout Preferences を開くには、パネルから System → Preferences → Keyboard を選択し、**Layouts** タブをクリックします。

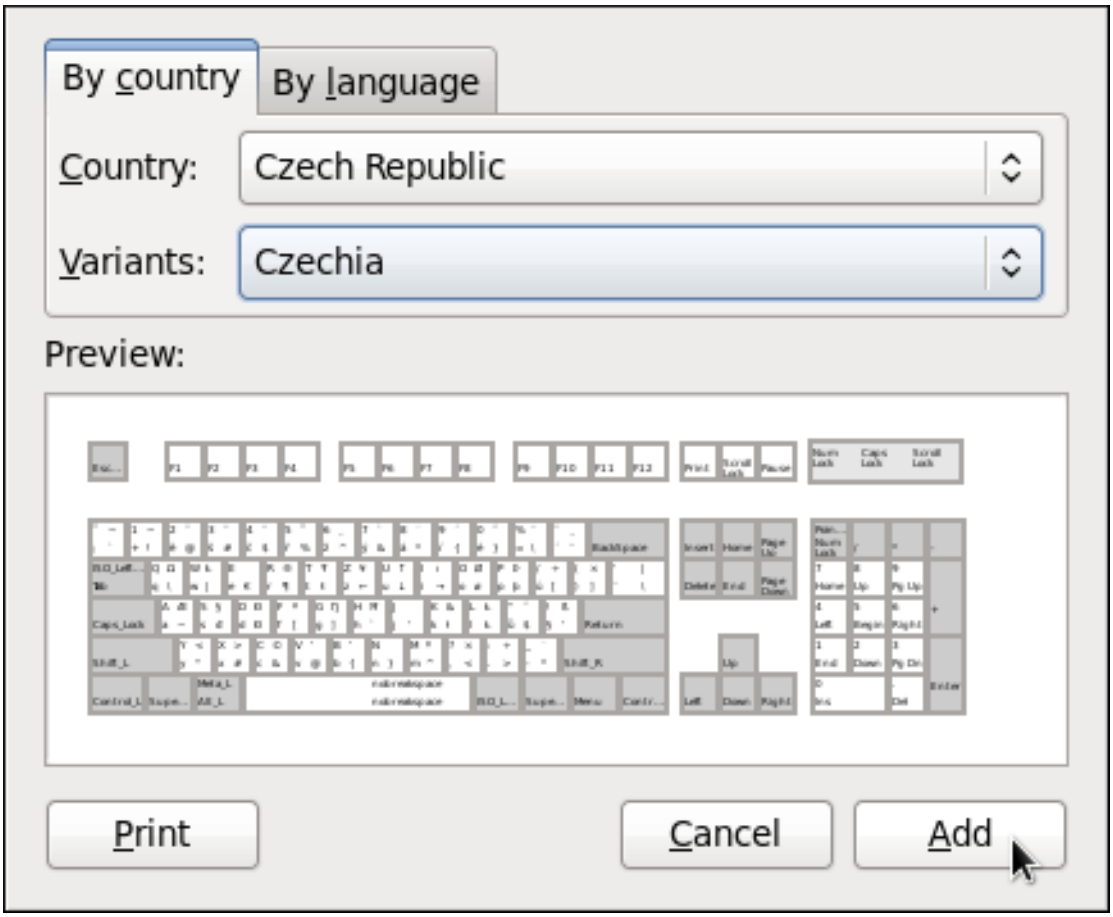
図1.1 キーボードのレイアウト設定



[D]

使用できるレイアウトの一覧が表示されます。新規レイアウトを追加するには、一覧の下の **追加** ボタンをクリックします。追加したいレイアウトを選択する画面が表示されます。

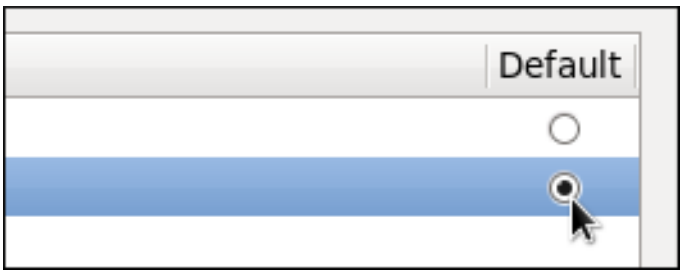
図1.2 レイアウトの選択



[D]

現時点では、キーボードのレイアウトを選択する方法は 2 つあります。関連する国から見つけるか（**By country (国別)** タブ、言語から選択するか（**By language (言語別)** タブ) のどちらかです。いずれの場合も、最初に **国** または **言語** のプルダウンメニューから希望する国か言語を選択します。次に、**Variants (系列)** メニューから系列を指定します。選択したら、レイアウトのプレビューは直ちに変更されます。選択を決定するには、**追加** をクリックします。

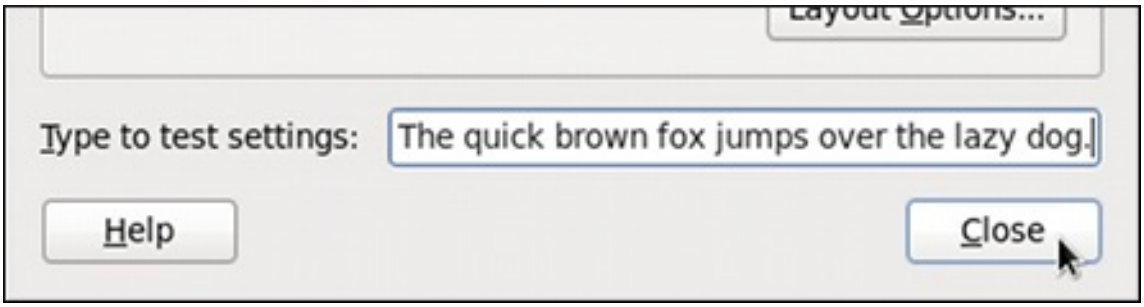
図1.3 デフォルトレイアウトの選択



[D]

選択したレイアウトが一覧に表示されるはずです。それをデフォルト設定にするには、名前の横にあるラジオボタンを選択して下さい。変更はすぐに有効になります。ウィンドウの最下部にはテキストを入力できるフィールドがあるため、そこで安心して設定を試すことができます。選択したレイアウトでよければ、**閉じる** をクリックしてウィンドウを閉じます。

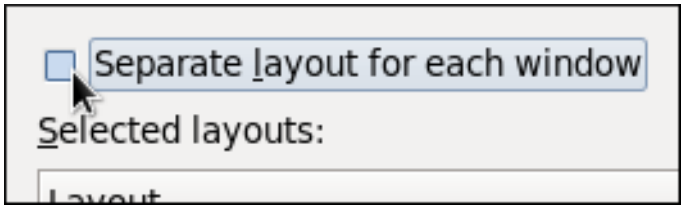
図1.4 レイアウトのテスト



[D]

ウィンドウごとに個別のレイアウトを無効化

デフォルトでは、キーボードのレイアウトを変更するとアクティブなウィンドウのみで変更が反映されるようになっています。つまり、レイアウトを変更した後に別のウィンドウに切り替えると、この別のウィンドウでは前のレイアウトが適用されるため、混乱する場合があります。この機能をオフにするには、**ウィンドウ毎にグループ化する** チェックボックスからチェックマークを外して下さい。



[D]

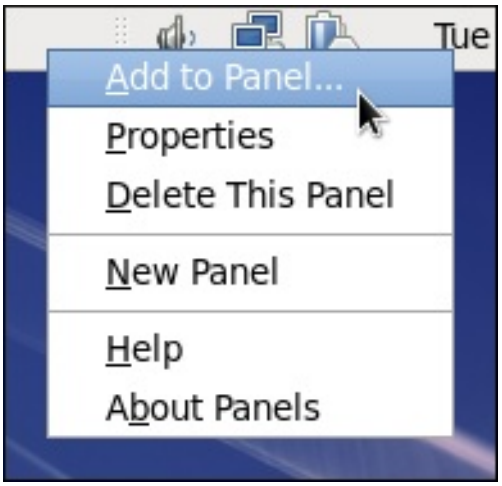
ただこれには、図1.3「デフォルトレイアウトの選択」 で示されたようにラジオボタンを選択することで、デフォルトのレイアウトを選択できなくなるという欠点があります。選択したレイアウトをデフォルトにするには、一覧の冒頭へドラッグするだけで可能です。

[D]

1.2. キーボードレイアウト表示器の追加

現在使用しているキーボードレイアウトを表示する場合や、1つのマウスクリックで異なるレイアウトを切り替える場合は、**キーボード** インダクレーターアプレットをパネルに追加します。そのためには、メインパネルの空いているスペースで右クリックし、プルダウンメニューから **Add to Panel... (パネルへ追加...)** オプションを選択します。

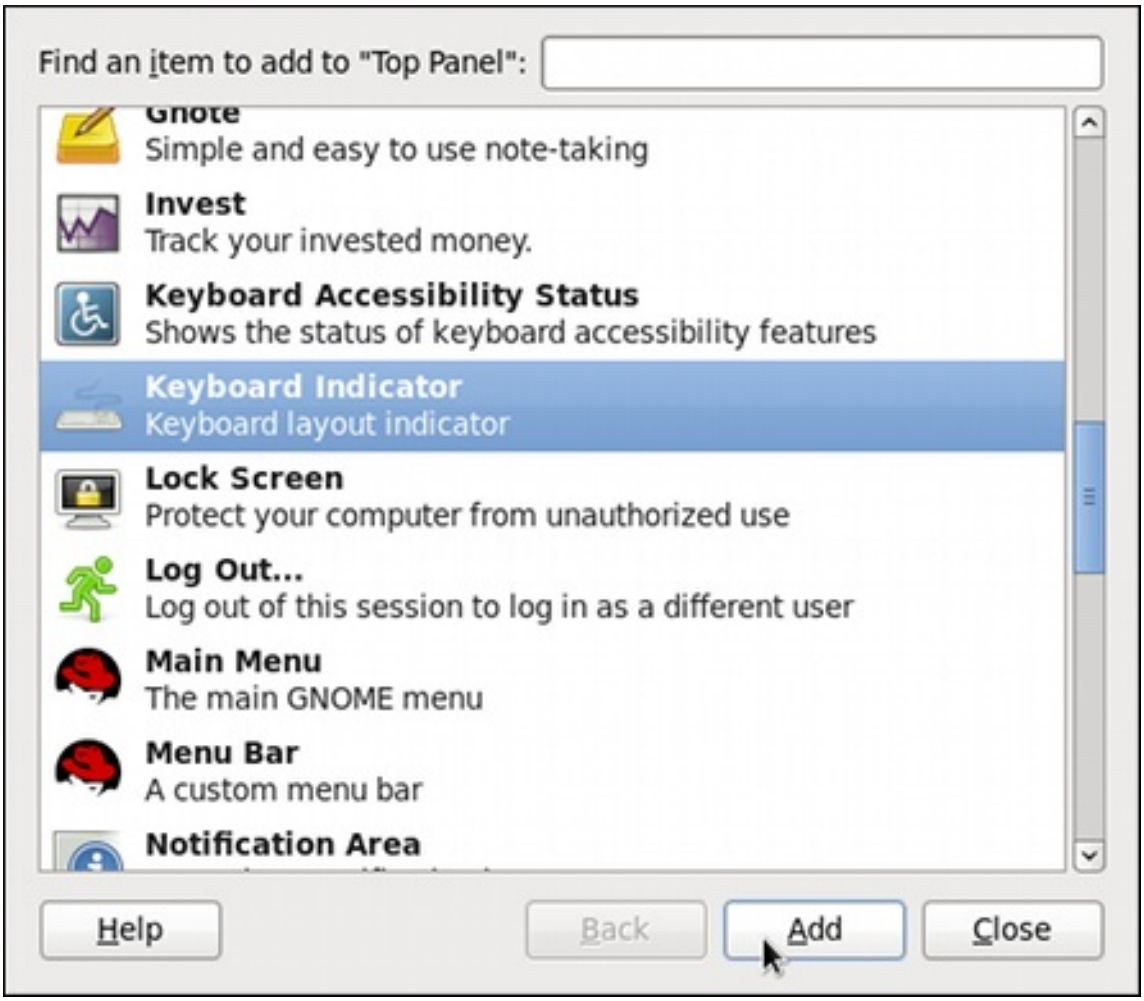
図1.5 新規アプレットの追加



[D]

利用可能なアプレットの一覧が表示されます。一覧をスクロールし (またはウィンドウの最上部にある検索フィールドに 「「キーボード」」 と入力し始める)、**キーボード表示器** を選択して、**追加** ボタンをクリックします。

図1.6 キーボードインダクレーターの選択



[D]

アプレットは直ちに表示され、現在のレイアウトと関連する短縮された国名が表示されます。実際の系列名を表示するには、アプレットのアイコン上にポインターを移動します。

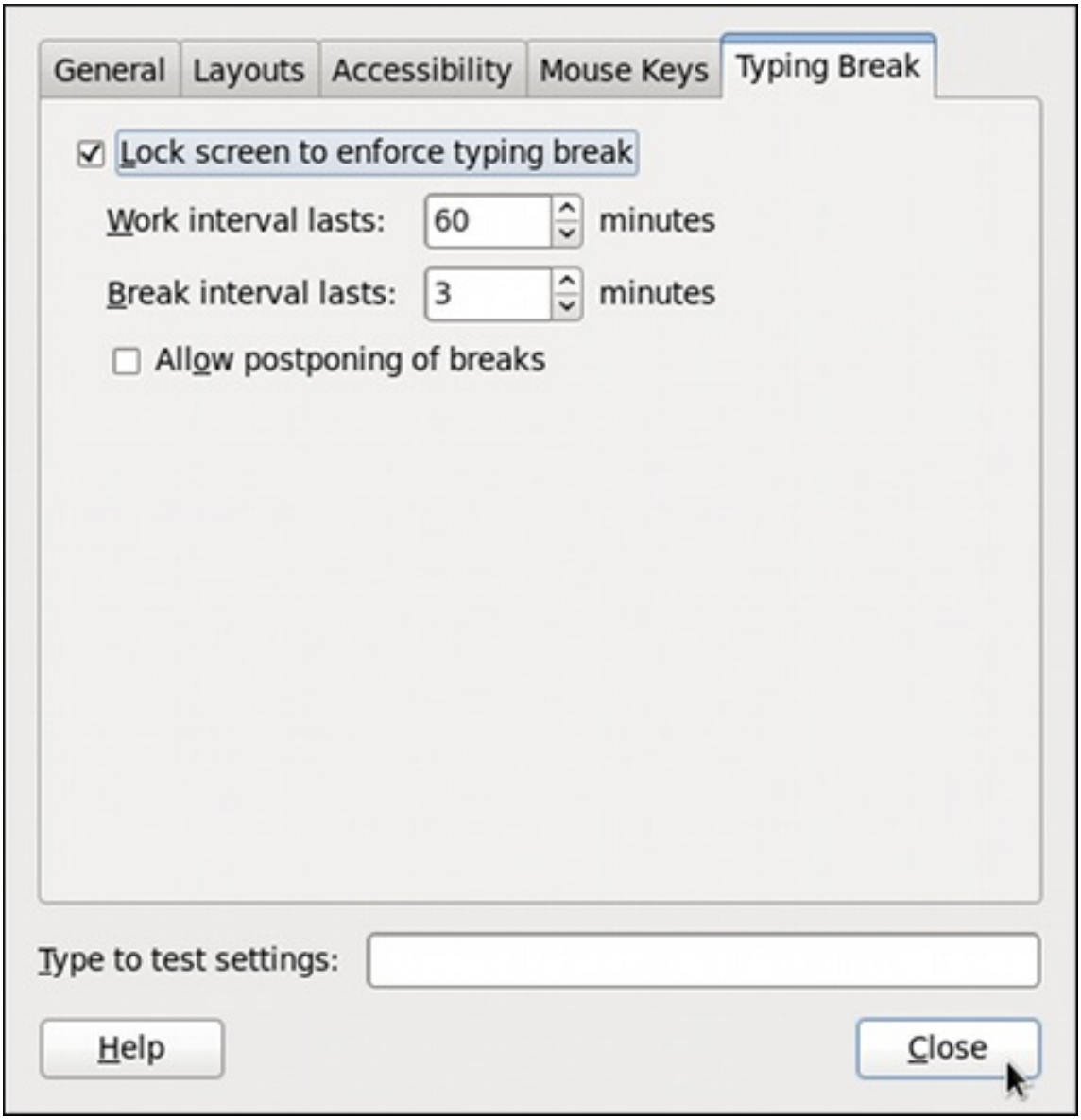
図1.7 Keyboard Indicator アプレット

[D]

1.3. 一休みの設定

長時間タイプし続けることは疲れるだけでなく、手根管症候群など深刻な健康問題のリスクを高める恐れもあります。これを防ぐ方法として、強制的に入力を休憩するようにシステムを設定することができます。設定方法は、パネルから **システム → 設定 → キーボード**の順に選択し、**一休み** タブをクリックし、**一休みに入ったら強制的に画面をロックする** のチェックボックスにチェックマークを入れます。

図1.8 一休みの設定

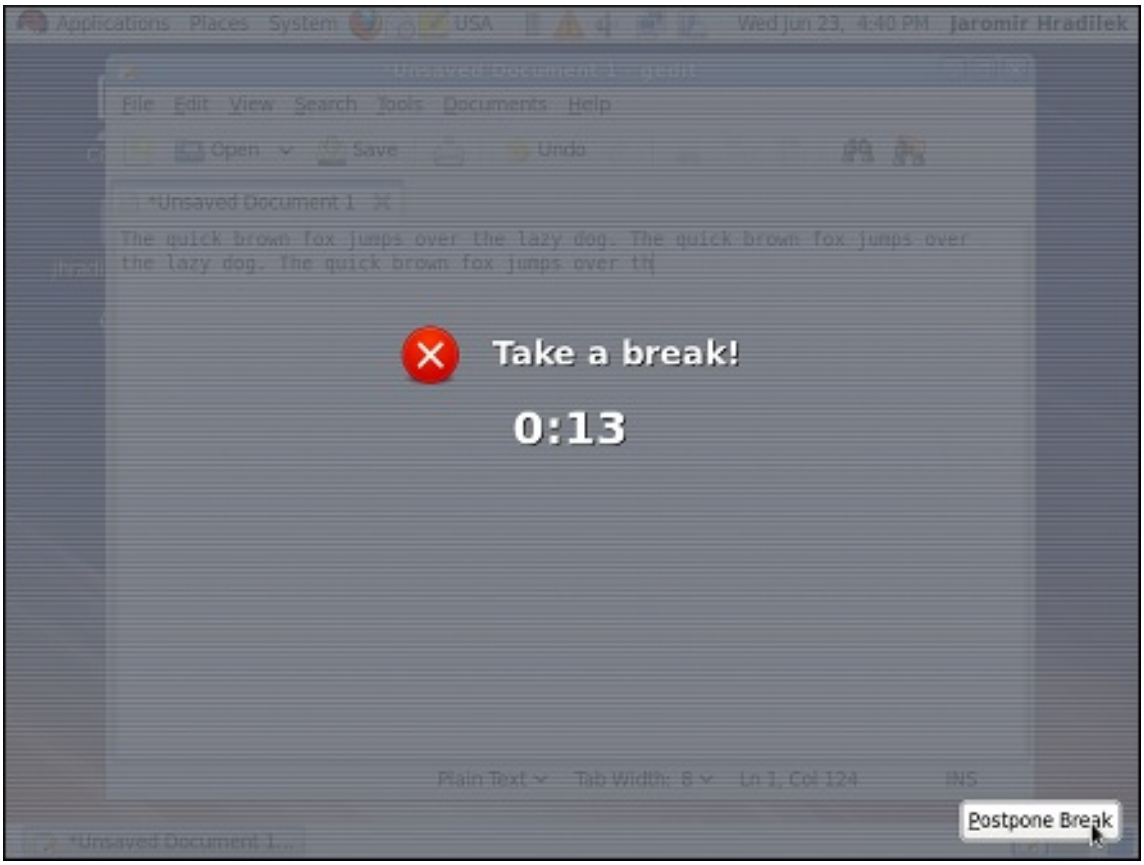


[D]

一休みの機能が強制実行されるまでの時間を調整するには、**一休みの警告を出すまでの時間** ラベルの横にある上向き/下向きボタンをクリックします。休憩時間の長さを変更するには、**一休みする時間** の

設定を同じように変更します。最後に、作業を終わらせたい場合など休憩を先送りしたい場合には、**一休みの延長を許可する** チェックボックスにチェックマークを入れます。変更は直ちに反映されます。

図1.9 休憩中



[D]

設定した制限時間に達すると、休憩するよう画面に通知が表示され、残りの休憩時間を表示した時計が現れます。 一休みの延長許可を有効にした場合は、**Postpone Break (中断を延長する)** ボタンが画面の右下隅に表示されます。

第2章 日付と時刻の設定

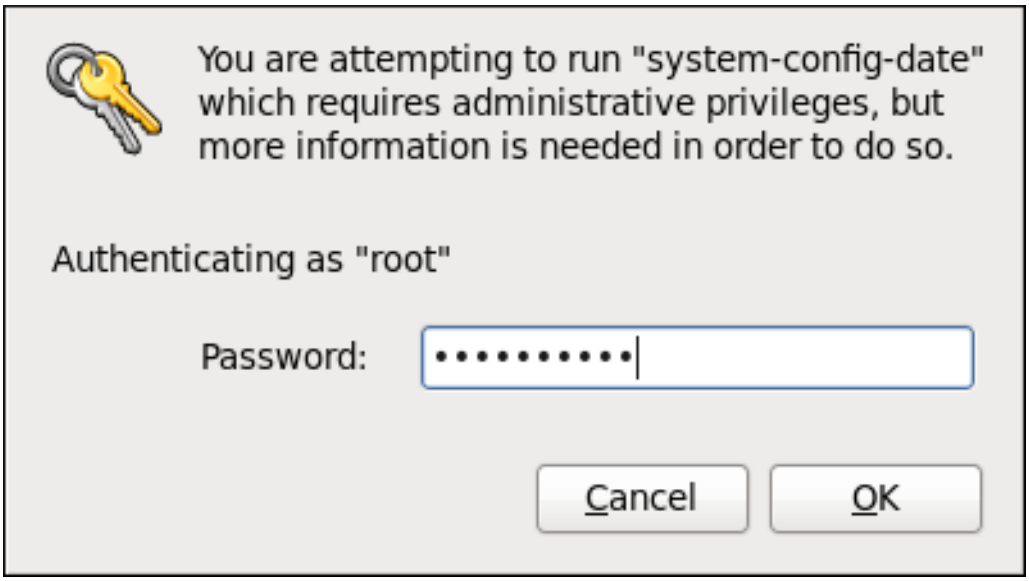
本章では、Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux でのシステムの日付と時刻（手動使用および Network Time Protocol(NTP)）の設定と、適切なタイムゾーンの設定について説明します。**Date/Time Properties** ツールを使用して日時を設定する方法と、コマンドラインで行う 2 つの方法があります。

2.1. 日付/時刻のプロパティのツール

Date/Time Properties ツールを使用すると、ユーザーはシステムの日付と時刻を変更し、システムが使用するタイムゾーンを設定し、システムクロックをタイムサーバーと同期するように Network Time Protocol デーモンを設定できます。このアプリケーションを使用するには、X Window System (このトピックの詳細については 付録C *X ウィンドウシステム* を参照) を実行している必要がある点に注意して下さい。

ツールを起動するには、パネルから **System → Administration → Date & Time** を選択するか、シェルプロンプトで **system-config-date** コマンド（ **xterm**、**GNOME ターミナル**など）を入力します。すでに認証されてない限りは、スーパーユーザーのパスワードを入力するよう求められます。

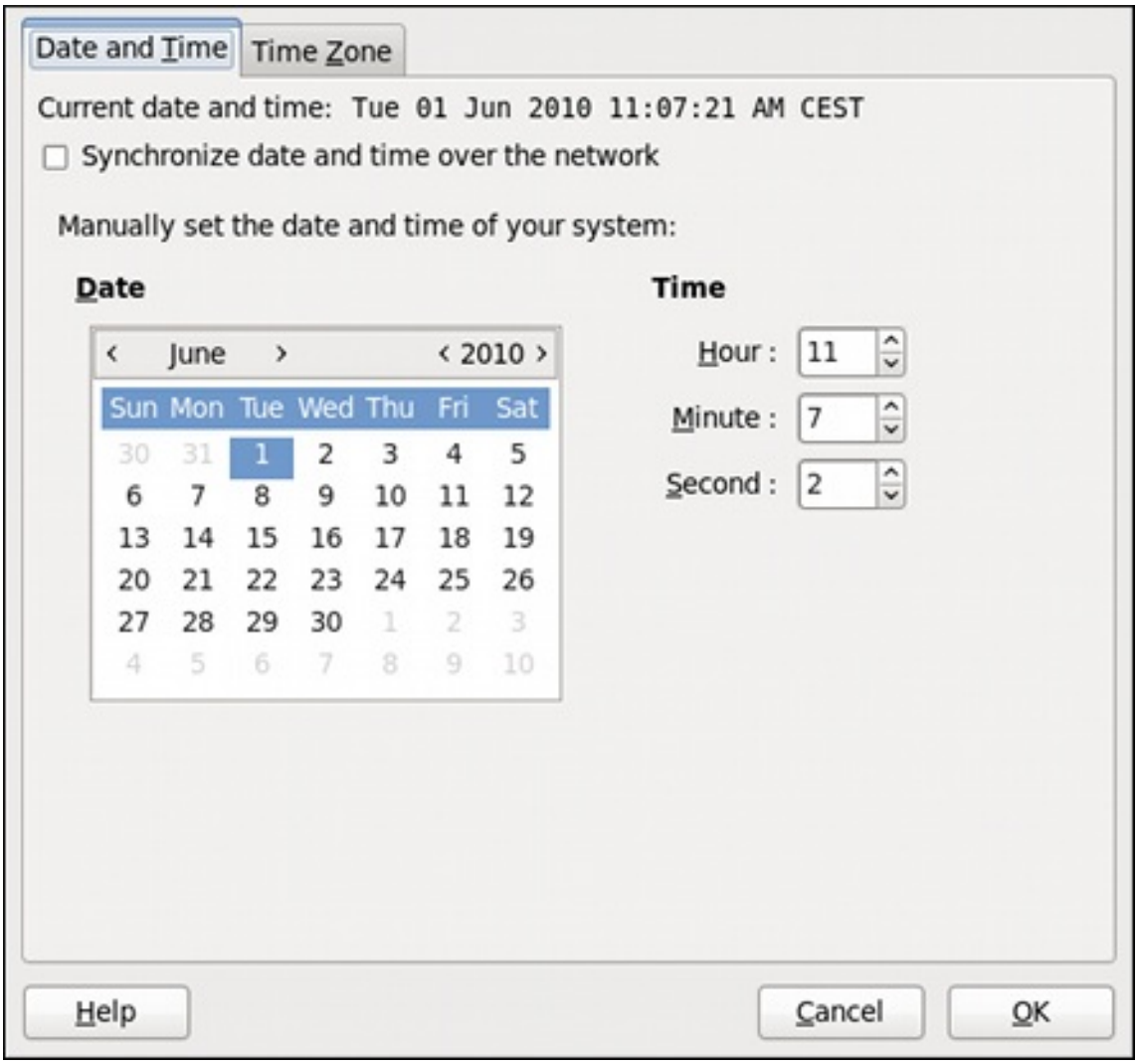
図2.1 認証クエリ



2.1.1. 日付と時刻のプロパティ

図2.2「日付と時刻のプロパティ」 のように、**Date/Time Properties** ツールは、2 つの別々のタブに分類されます。デフォルトでは、現在の日時設定があるタブが表示されます。

図2.2 日付と時刻のプロパティ



[D]

手動でシステムを設定するには、以下の手順に従って下さい:

1. **現在の日付を変更する。**月と年の左右にある矢印を使って、それぞれ変更します。次に、カレンダーの中で日付を選択します。

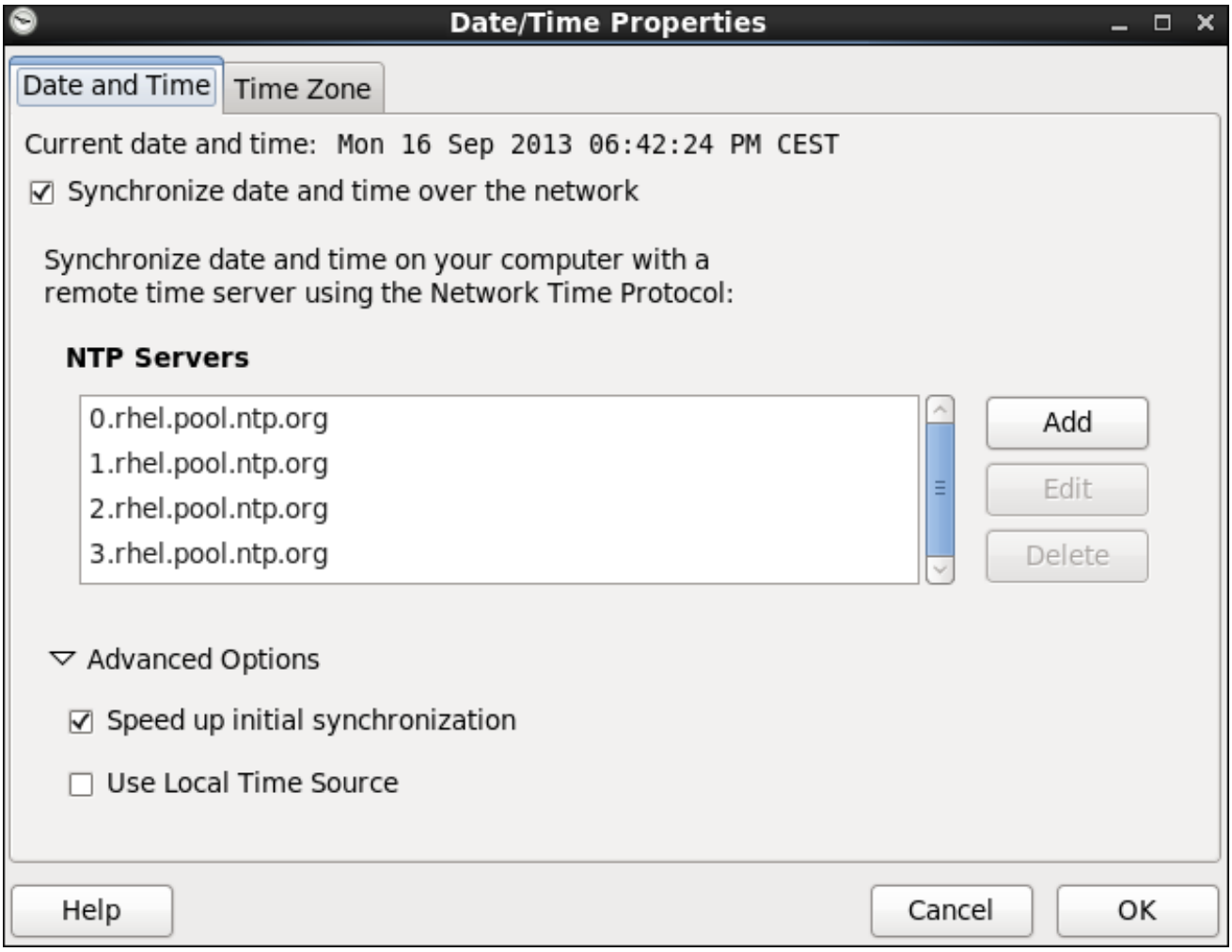
2. **現在の時刻を変更する。時、分、秒**の横にある上下の矢印ボタンを使用するか、数字を直接変更します。

OK ボタンをクリックして変更を適用し、アプリケーションを終了します。

2.1.2. ネットワーク時刻プロトコルのプロパティ

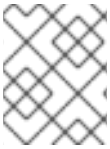
自動設定にしたい場合は、 **Synchronize date and time over the network (ネットワーク上で日付と時刻を同期化します)** というチェックボックスにチェックマークを入れて下さい。そうすると、利用可能な NTP サーバーの一覧が 図2.3「ネットワーク時刻プロトコルのプロパティ」 のように表示されます。

図2.3 ネットワーク時刻プロトコルのプロパティ



[D]

ここで、事前定義済みのサーバーから1つ選択し、**編集** ボタンをクリックして編集するか、**追加** をクリックして新規サーバー名を追加することも可能です。**Advanced Options (高度なオプション)** では、サービスを開始する前にシステムクロックを同期したいか、ローカルタイムソースを使用したいかどうかを選択することもできます。



注記

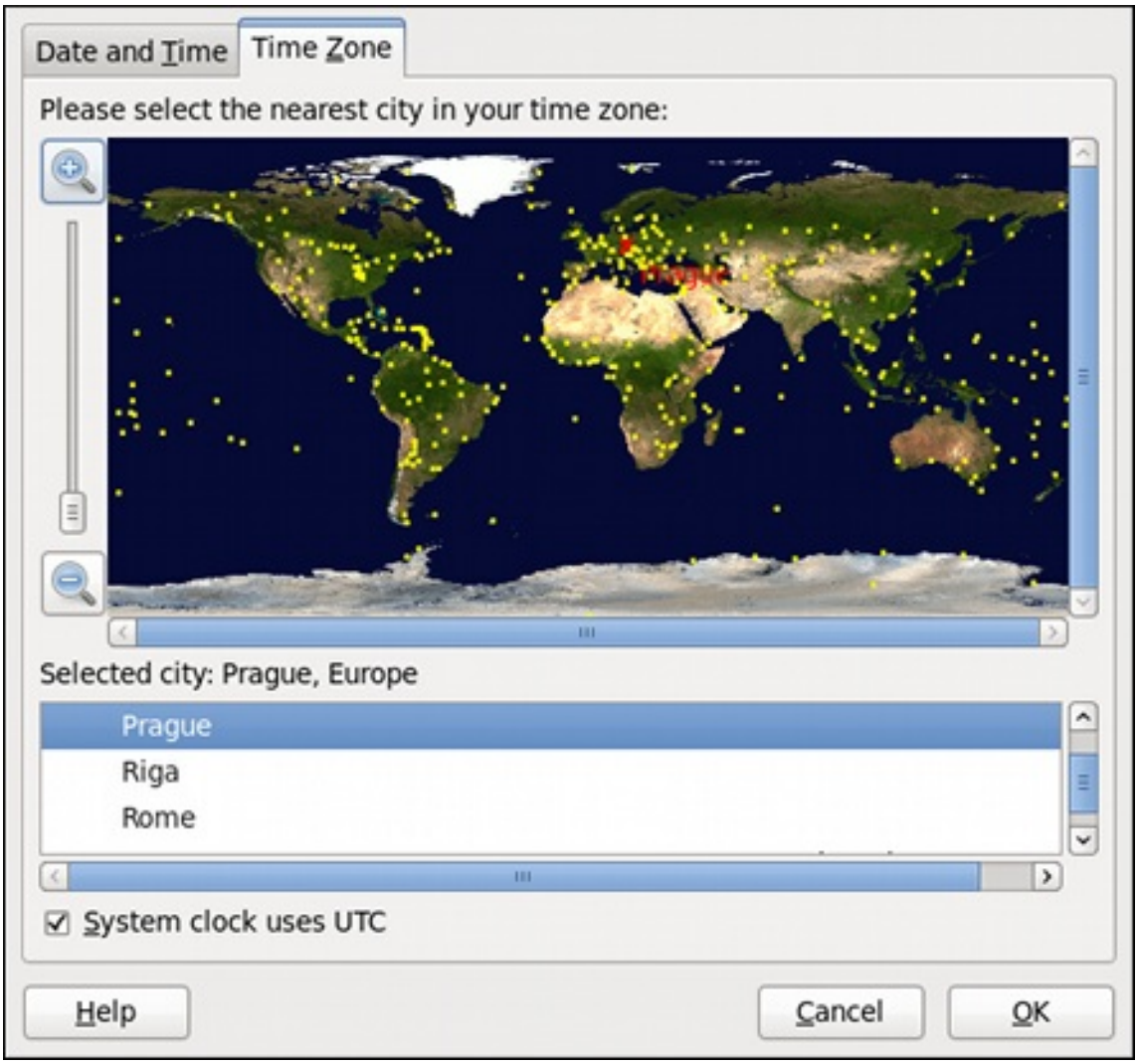
ウィンドウの最下部にある **OK** ボタンをクリックして変更を確定するまで、システムは NTP サーバーとの同期を開始しません。

OK ボタンをクリックして、日付と時刻の設定への変更を適用して、アプリケーションを終了します。

2.1.3. タイムゾーンのプロパティ

システムのタイムゾーンを設定するには、図2.4「タイムゾーンのプロパティ」 のように **Time Zone (タイムゾーン)** タブをクリックします。

図2.4 タイムゾーンのプロパティ



[D]

タイムゾーンを選択する一般的な方法としては、以下の 2 つがあります。

1. **インタラクティブマップを使用する。** マップの横にある “拡大”/“縮小” ボタンをクリックします。あるいは、マップ上でクリックして、選択した地域を拡大します。次に、希望するタイムゾーンに合う都市を選択します。赤い **X** が表示され、マップの下にある一覧内のタイムゾーンの選択肢が変更します。

2. **マップの下にある一覧を使用する。** 選択しやすいように特定の大陸内で都市と国がグループ化されています。科学界のニーズに合うように、地理的ではないタイムゾーンも追加されている点に注意して下さい。

ご使用のシステムクロックが UTC を使用するよう設定するには、**System clock uses UTC (システムクロックで UTC を使用)** オプションを選択します。UTC は *Universal Time, Coordinated (協定世界時)* の略で、GMT(*グリニッジ標準時*)とも呼ばれます。他のタイムゾーンは UTC 時間にプラス/マイナスすることで割り出されます。

OK をクリックし変更を適用して、プログラムを終了します。

2.2. コマンドラインからの設定

システムに **Date/Time Properties** ツールがインストールされていない場合や、X Window Server が実行されていない場合は、コマンドラインでシステムの日時を変更する必要があります。本項で説明する操作を行うには、スーパーユーザーとしてログインする必要がある点にご注意下さい:

```
~]$ su -
Password:
```

2.2.1. 日付と時刻の設定

date コマンドを使用すると、スーパーユーザーはシステムの日時を手動で設定できます。

1. **現在の日付を変更する。**ためには、シェルプロンプトで次の形式でコマンドを入力します。*YYYY* は 4 桁の年、*MM* は 2 桁の月、*DD* は 2 桁の日付に置き換えて下さい:

   ```
   ~]# date +%D -s YYYY-MM-DD
   ```

   例えば 2010 年 6 月 2 日と設定するには、以下のように入力します:

   ```
   ~]# date +%D -s 2010-06-02
   ```

2. **現在の時刻を変更する。**ためには、次のコマンドを使用します。*HH* は時間、*MM* は分、*SS*は秒を表し、すべて 2 桁で入力します:

   ```
   ~]# date +%T -s HH:MM:SS
   ```

   システムクロックが UTC を使用するように設定するには、次のオプションを追加します:

   ```
   ~]# date +%T -s HH:MM:SS -u
   ```

   例えば UTC を使用してシステムクロックを 11:26 PM に設定するには、以下のように入力します:

   ```
   ~]# date +%T -s 23:26:00 -u
   ```

現在の設定を確認するには、**date** を引数を付けずに入力します。

例2.1 システムの現在日時の表示

```
~]$ date
Wed Jun  2 11:58:48 CEST 2010
```

2.2.2. ネットワーク時刻プロトコルの設定

前述した手動設定とは対照的に、ネットワーク時刻プロトコル (NTP) でリモートサーバーとシステムクロックを同期することもできます。1 回限りの同期のみの場合は、**ntpdate** コマンドを使用します。

1. 最初に、選択した NTP サーバーがアクセス可能か確認します:

   ```
   ~]# ntpdate -q server_address
   ```

以下に例を示します。

```
~]# ntpdate -q 0.rhel.pool.ntp.org
```

2. satisfactory サーバーを見つけたら、**ntpdate** コマンドの後に 1 つ以上のサーバーアドレスを指定して実行します。

```
~]# ntpdate server_address...
```

たとえば、以下のようになります。

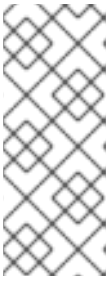
```
~]# ntpdate 0.rhel.pool.ntp.org 1.rhel.pool.ntp.org
```

エラーメッセージが表示されない限り、システムの時刻は設定されているはずです。現在確認するには、「日付と時刻の設定」 に記載されているように、追加の引数を指定せずに **date** を入力します。

3. ほとんどの場合、上記のステップで十分です。常に適切な時間を使用するためにシステムサービスが 1 つ以上必要な場合に限り、システムの起動時に **ntpdate** を実行できます。

```
~]# chkconfig ntpdate on
```

システムサービスとその設定の詳細については、*12章 サービスおよびデーモン* を参照して下さい。



注記

システムの起動時にタイムサーバーとの同期に失敗し続ける場合（**/var/log/boot.log** システムログに関連するエラーメッセージ）は、以下の行を **/etc/sysconfig/network** に追加します。

```
NETWORKWAIT=1
```

ただし、**ntpd** デーモンを、システムの起動時に時刻を自動的に同期するように設定すると便利な方法です。

1. **vi** や **nano** などのテキストエディターで NTP 設定ファイル **/etc/ntp.conf** を開くか、存在しない場合は新規のファイルを作成します。

```
~]# nano /etc/ntp.conf
```

2. 次に、パブリック NTP サーバーの一覧を追加/編集します。Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;LinuxRed Hat Enterprise Linux 6nbsp;6 を使用している場合、このファイルには以下の行がすでに含まれているはずですが、必要に応じてそれらを自由に変更または拡張してください。

```
server 0.rhel.pool.ntp.org iburst
server 1.rhel.pool.ntp.org iburst
server 2.rhel.pool.ntp.org iburst
server 3.rhel.pool.ntp.org iburst
```

各行の最後にある **iburst** ディレクティブは、初期同期を迅速化します。Red Hat Enterprise

Linux 6.5 では、デフォルトで追加されます。以前のマイナーリリースからアップグレードし、**/etc/ntp.conf** ファイルを変更した場合には、Red Hat Enterprise Linux 6.5 へのアップグレードで新しいファイル **/etc/ntp.conf.rpmnew** が作成され、既存の **/etc/ntp.conf** ファイルは変更されません。

3. サーバーの一覧での作業が完了したら、同じファイルで適切なパーミッションを設定し、ローカルホストのみに無制限アクセスを付与します:

```
restrict default kod nomodify notrap nopeer noquery
restrict -6 default kod nomodify notrap nopeer noquery
restrict 127.0.0.1
restrict -6 ::1
```

4. すべての変更を保存してエディターを終了し、NTP デーモンを再起動します。

```
~]# service ntpd restart
```

5. **ntpd** が起動時に起動されていることを確認します。

```
~]# chkconfig ntpd on
```

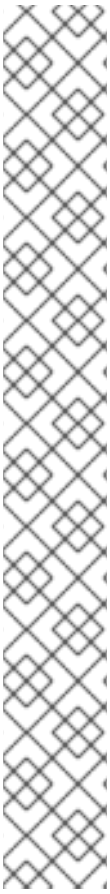
第3章 ユーザーとグループの管理

3.1. ユーザーとグループの概要

ユーザーおよびグループの制御は、Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux システム管理の中核となる要素です。システムのユーザーは、人間がいるか、*ユーザー ID(UID) と呼ばれる一意の数値 ID* で識別される特定のアプリケーションによって使用されるアカウントになります。グループ内のユーザーは、そのグループが所有するファイルの読み取り/書き込みパーミッション、実行パーミッション、実行パーミッション、読み取り/書き込み、実行パーミッション、または任意の組み合わせを指定できます。

Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux は、ファイルおよびディレクトリーの *アクセス制御リスト* (ACL)をサポートします。これにより、所有者外の特定ユーザーのパーミッションを設定できます。この機能の詳細は、『Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;Linux 『Red Hat Enterprise Linux 6nbsp;Linux Red Hat Enterprise Linux 6nbsp;6 ストレージ管理ガイド』 』の「 『アクセス制御リスト』 」の章を参照してください。

グループは、共通の目的でユーザーをまとめる組織単位です。これは、そのグループが所有するファイルのパーミッションの読み取り、パーミッションの書き込み、またはパーミッションの実行が可能です。UID と同様に、各グループはグループ ID(GID)に関連付けられています。



注記

Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux は、システムユーザーおよびグループ用に 500 未満のユーザー ID とグループ ID を確保します。デフォルトでは、**User Manager** にはシステムユーザーは表示されません。予約されているユーザー ID およびグループ ID は、setup パッケージに記載されています。このドキュメントを表示するには、以下のコマンドを実行します。

```
cat /usr/share/doc/setup-2.8.14/uidgid
```

予約済みの ID は将来的に増大するため、予約されていない ID を 5,000 から削除することが推奨されます。新規ユーザーに割り当てる ID を 5,000 から始まるようにするには、**/etc/login.defs** ファイルの **UID_MIN** ディレクティブおよび **GID_MIN** ディレクティブを変更します。

```
[file contents truncated]
UID_MIN                  5000
[file contents truncated]
GID_MIN                  5000
[file contents truncated]
```

新規ユーザーおよびグループの ID を 5,000 から始まるようにした場合でも、システムが予約する ID は 500 を超えることは推奨されません。これにより、500 の制限を保持するシステムとの競合を避けることができます。

各ユーザーは、正確に 1 つのプライマリーグループのメンバーで、ゼロまたは複数の補助グループになります。デフォルトでは、ファイルが作成される場合、ファイルの所有者は作成者であり、ファイルのグループは作成者のプライマリーグループです。ユーザーは、新しいグループによって所有された後に、**newgrp** コマンドを使用してプライマリーグループであるグループを一時的に変更できます。補助グループは、このグループが所有する特定のユーザーセットとそのメンバーに対し、特定のファイルセットへのアクセスを付与します。

ファイルには、所有者、グループ、その他に対して読み取り、書き込み、実行のパーミッションが別々に割り当てられます。ファイルの所有者は **root** でのみ変更でき、アクセスパーミッションは **root** ユーザーとファイル所有者の両方が変更できます。

デフォルトでは、ファイルまたはディレクトリーは、その作成者によってのみ変更できます。新規に作成するファイルまたはディレクトリーに適用される権限を判断する設定は **umask** と呼ばれ、すべてのユーザーの場合は **/etc/bashrc** ファイル、または各ユーザーの **~/.bashrc** で個別に設定できます。**~/.bashrc** の設定は、**/etc/bashrc** の設定を上書きします。また、**umask** コマンドは、シェルセッションの期間のデフォルトパーミッションを上書きします。

認証するには、ユーザーはパスワードを入力します。ハッシュ合計は、入力した文字列から生成され、ユーザーのパスワードのハッシュ合計と比較されます。ハッシュ合計が一致すると、ユーザーは正常に認証されます。

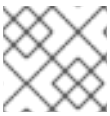
ユーザーパスワードのハッシュ合計は、**/etc/shadow** ファイルに保存されます。このファイルは **root** ユーザーのみが読み取り可能です。このファイルには、特定のアカウントのパスワードの変更およびポリシーに関する情報も保存されます。新規作成されたアカウントのデフォルト値は、**/etc/login.defs** ファイルおよび **/etc/default/useradd** ファイルに保存されます。『Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;LinuxRed Hat Enterprise Linux 6nbsp;6 セキュリティーガイド』 では、ユーザーとグループのセキュリティー関連情報が説明されています。

3.2. ユーザーマネージャーアプリケーションを使用したユーザーの管理

User Manager アプリケーションを使用すると、グラフィカルユーザーインターフェースでローカルのユーザーおよびグループを表示、変更、追加、削除できます。

User Manager アプリケーションを起動するには、以下を実行します。

- パネルから **システム → 管理 → ユーザーとグループ** の順にクリックします。
- または、シェルプロンプトで **system-config-users** を入力します。



注記

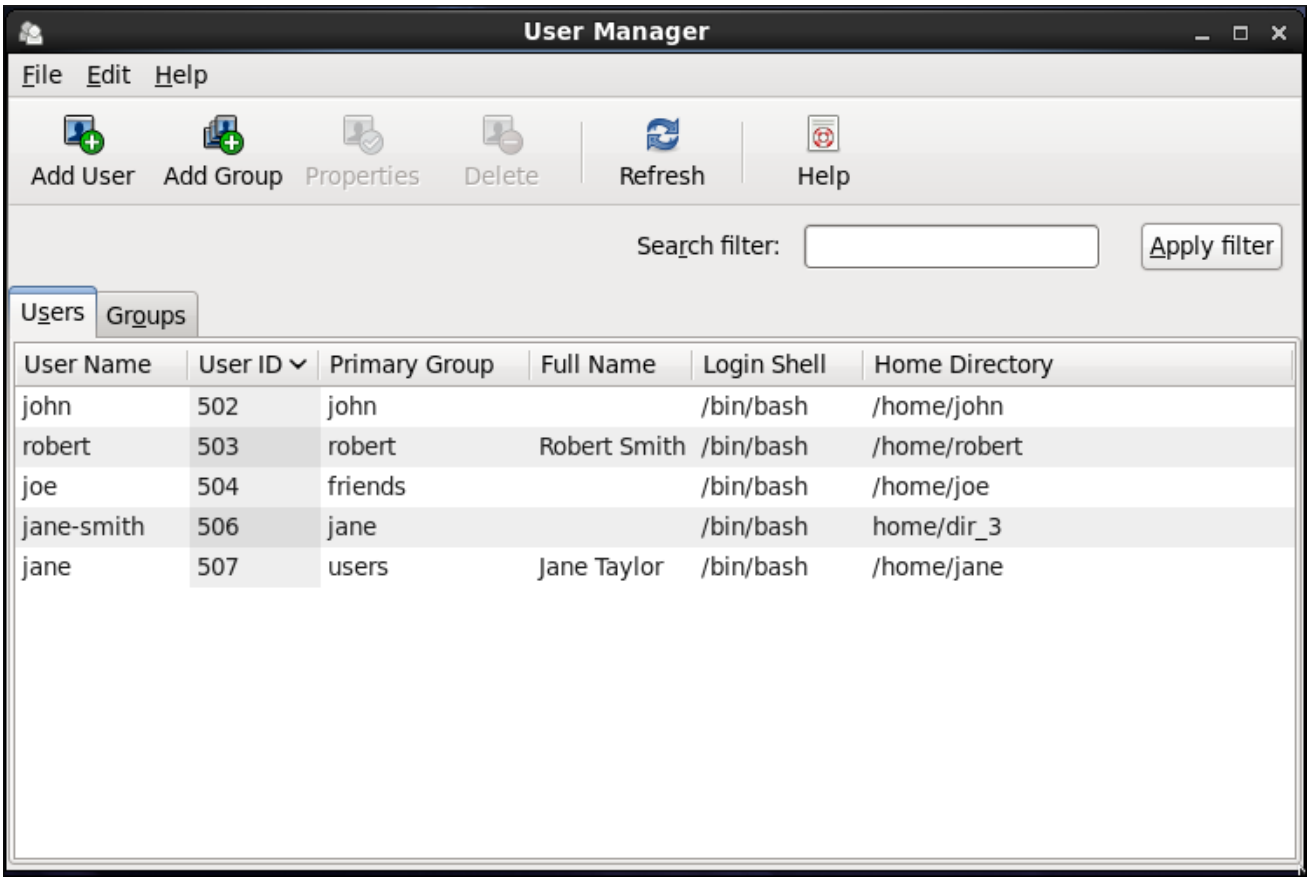
スーパーユーザー特権がない場合は、**root** で認証するよう求められます。

3.2.1. ユーザーの表示

ユーザーを表示する User Manager のメインウィンドウを表示するには、User Manager のツールバーから **Edit → Preferences** を選択します。システムユーザーを含む **すべて** のユーザーを表示するには、**Hide system users and groups** チェックボックスの選択を解除します。

ユーザー **タブ** では、ローカルユーザーの一覧と、ユーザー ID、プライマリーグループ、ホームディレクトリー、ログインシェル、およびフルネームに関する追加情報が表示されます。

図3.1 ユーザーの表示



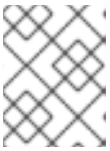
[D]

特定のユーザーやグループを検索したい場合は、**フィルターの探索** フィールドに検索したい名前の最初の数文字を入力します。そして、 **Enter** を押すか **フィルターの適用** ボタンをクリックします。また、いずれかのカラムのヘッダーをクリックするとそのカラムに従って項目を並び替えることもできます。

3.2.2. 新規ユーザーの追加

システムに追加する必要のある新規ユーザーがある場合は、以下の手順に従います。

1. **Add User** ボタンをクリックします。
2. 適切なフィールドにユーザー名とフル名を入力します。
3. Password フィールドおよび **Confirm Password** フィールドにユーザーのパスワードを入力します。パスワードは 6 文字以上でなくてはなりません。



注記

安全上の理由から、辞書用語に基づかない長いパスワードを選択し、文字、数字、および特殊文字の組み合わせを使用します。

4. **Login Shell** ドロップダウンリストからユーザーのログインシェルを選択するか、デフォルト値の **/bin/bash** を受け入れます。
5. **/ home/ *ユーザー 名*/ に新しいユーザーのホームディレクトリー** を作成したくない場合は、ホームディレクトリーの作成 チェックボックスの選択を解除します。

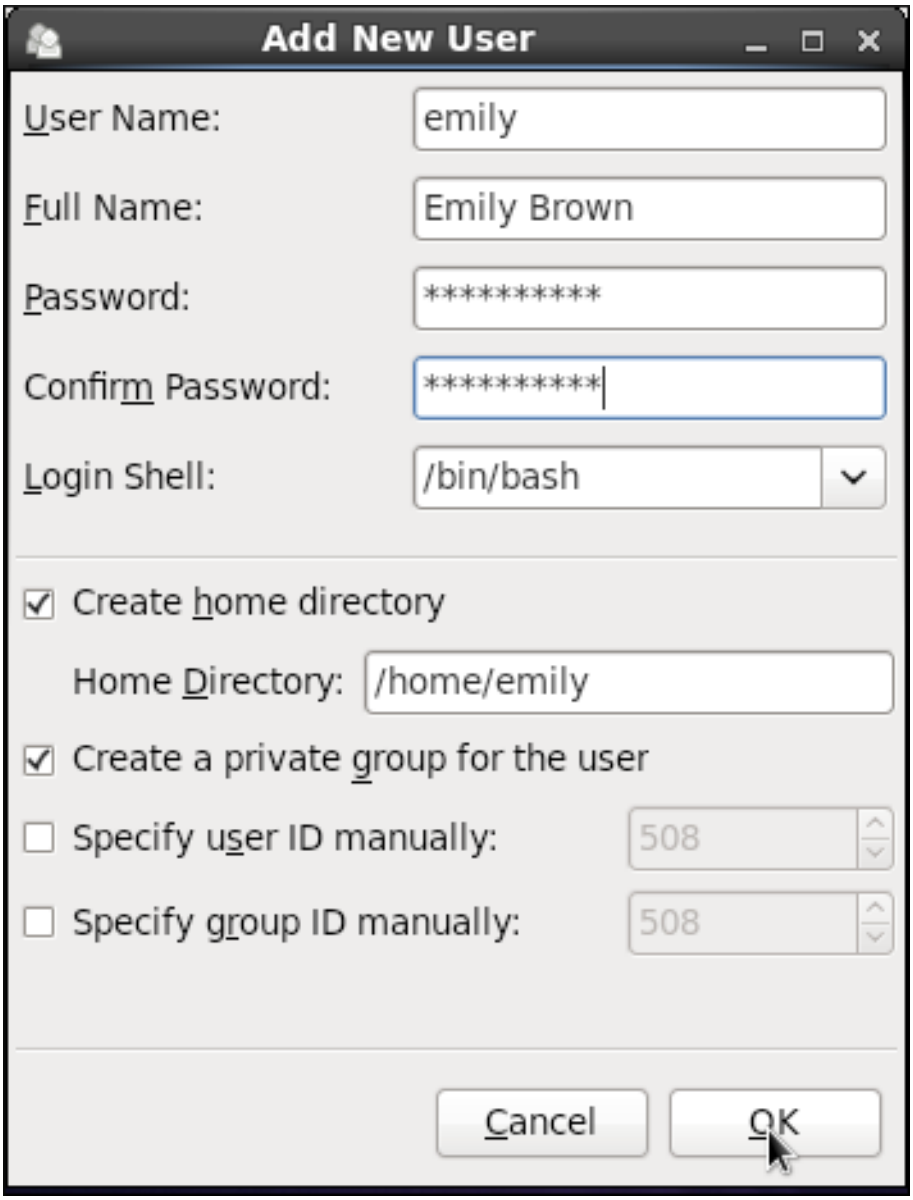
Home Directory テキストボックスに内容を編集して、このホームディレクトリーを変更することもできます。ホームディレクトリーが作成されると、デフォルトの設定ファイルが **/etc/skel/** ディレクトリーからコピーされることに注意してください。

6. **作成するユーザーと同じ名前の一意のグループが必要ない場合には、ユーザーのプライベートグループの作成** チェックボックスの選択を解除します。ユーザープライベートグループ(UPG)は、そのユーザーのみが属するユーザーアカウントに割り当てられたグループで、個別ユーザーのファイルパーミッションの管理に使用されます。

7. **Specify user ID manually** を選択して、ユーザーのユーザー ID を指定します。このオプションが選択されていない場合は、次に 500 を超える利用可能なユーザー ID が新規ユーザーに割り当てられます。

8. **OK** ボタンをクリックしてプロセスを完了します。

Add New User ダイアログボックスの設定内容を確認します。



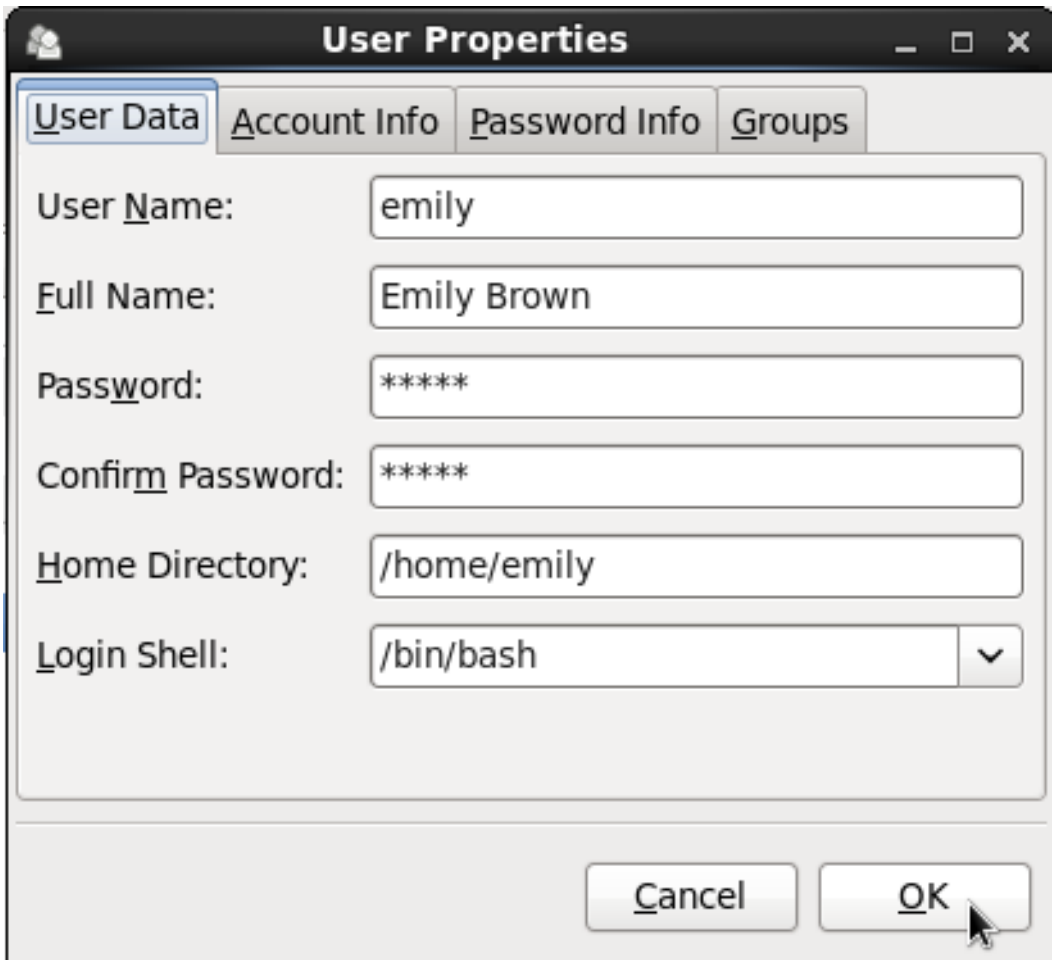
[D]

パスワードの有効期限などの高度なユーザープロパティーを設定するには、ユーザーを追加した後にユーザーのプロパティーを変更します。

3.2.3. ユーザープロパティーの変更

1. ユーザー名をクリックして、ユーザーリストからユーザーを選択します。
2. ツールバーから **Properties** をクリックするか、ドロップダウンメニューから **File** → **Properties** を選択します。

   図3.2 ユーザープロパティ



[D]

3. 設定に更新できるタブは 4 つあります。完了したら、**OK** ボタンをクリックして変更を保存します。

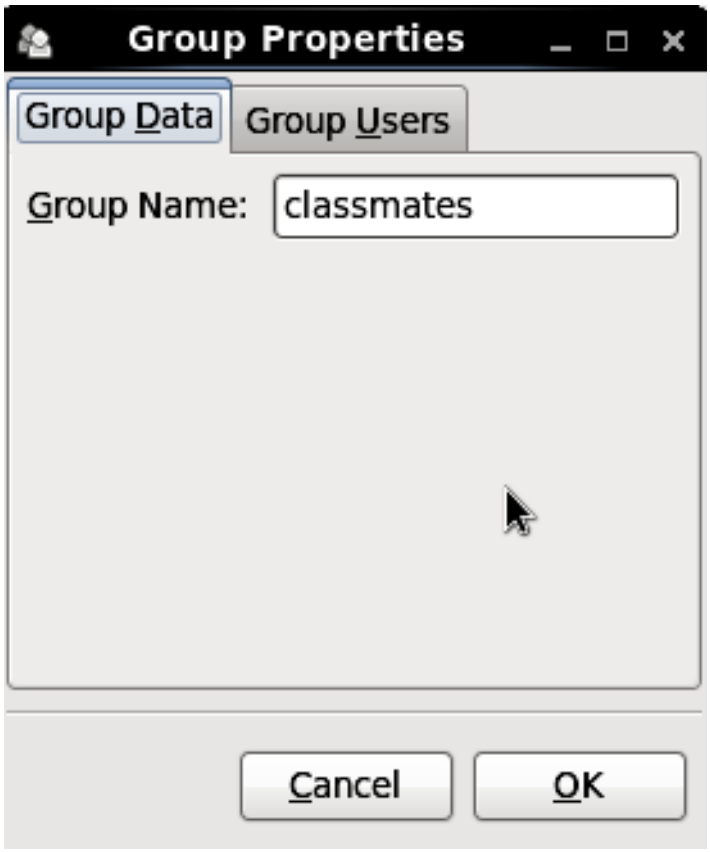
3.3. ユーザーマネージャーアプリケーションを使用したグループの管理

3.3.1. グループの表示

User Manager のメインウィンドウを表示してグループを表示するには、ツールバーから **Edit** → **Preferences** を選択します。**すべて** のグループを表示するには、**Hide system users and groups** チェックボックスの選択を解除します。

Groups タブには、以下の図に示すように、グループ ID とグループメンバーに関する情報を含むローカルグループの一覧が表示されます。

図3.3 グループの表示



[D]

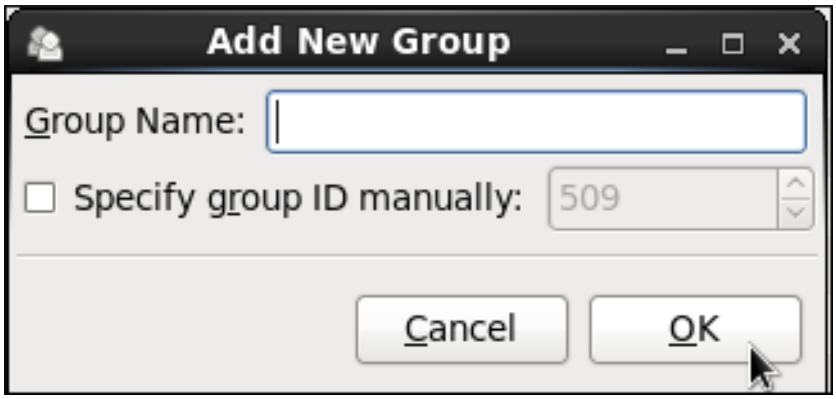
特定のグループを検索するには、**Search filter** フィールドに名前の最初の数文字を **入力** し、Enter を押すか、**Apply filter** ボタンをクリックします。また、いずれかのカラムのヘッダーをクリックするとそのカラムに従って項目を並び替えることもできます。

3.3.2. 新規グループの追加

システムに追加する必要のある新しいグループがある場合は、以下の手順に従います。

1. User Manager ツールバーから **Add Group** を選択します。

図3.4 新規グループ



[D]

2. 新規グループの名前を入力します。

3. 指定したグループ ID を手作業 **でチェックして、新しいグループのグループ ID(GID)を指定** します。

4. GID を選択します。Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux は、システムグループ用に 500 未満のグループ ID を確保することに注意してください。

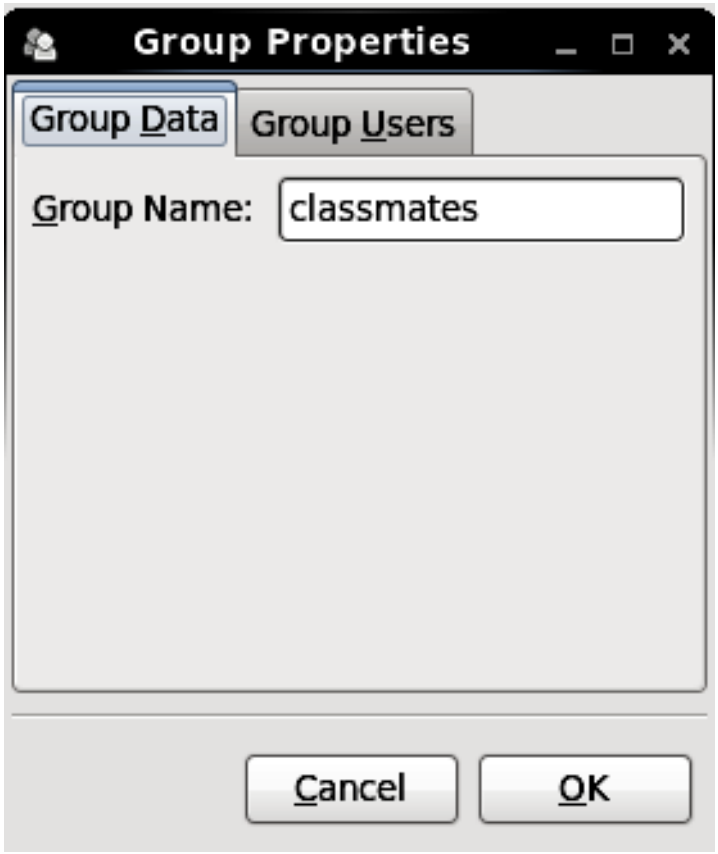
5. **OK** をクリックしてグループを作成します。新しいグループがグループ一覧に表示されます。

3.3.3. グループプロパティーの変更

1. 名前をクリックして、グループリストからグループを選択します。

2. ツールバーから **Properties** をクリックするか、ドロップダウンメニューから **File → Properties** を選択します。

図3.5 グループプロパティー



[D]

3. **Group Users** タブにはグループメンバーの一覧が表示されます。このタブを使用してグループに対してユーザーを追加または削除します。**OK** をクリックして変更を保存します。

3.4. コマンドラインツールを使用したユーザーの管理

コマンドラインでユーザーを管理する場合は、**useradd**、**usermod**、**userdel**、または **passwd** のコマンドが使用されます。影響を受けるファイルには、ユーザーアカウント情報を格納する **/etc/passwd** と、セキュアなユーザーアカウント情報を格納する **/etc/shadow** が含まれます。

3.4.1. ユーザーの作成

useradd ユーティリティーは、新しいユーザーを作成して、システムに追加します。以下の短い手順に従って、UID でデフォルトのユーザーアカウントを作成し、デフォルトユーザー設定を保存するホームディレクトリー(/**home**/***username***/)を作成し、デフォルトのシェルを /**bin**/**bash** に設定します。

1. *ユーザー名を任意の名前で置き換える* **root** として、シェルプロンプトで以下のコマンドを実行します。

```
useradd username
```

2. パスワードのロック解除を行い、アカウントのロックを解除します。プログラムが要求されたら、パスワードを 2 回入力します。

```
passwd
```

例3.1 デフォルト設定を使用したユーザーの作成

```
~]# useradd robert
~]# passwd robert
Changing password for user robert
New password:
Re-type new password:
passwd: all authentication tokens updated successfully.
```

useradd robert コマンドを実行すると、**robert** という名前のアカウントが作成されます。**cat /etc/passwd** を実行して /**etc**/**passwd** ファイルの内容を表示する場合は、表示された行から新規ユーザーの詳細を確認できます。

```
robert:x:502:502::/home/robert:/bin/bash
```

robert には 502 という UID が割り当てられました。これは、0 から 499 までのデフォルトの UID 値がシステムアカウント用に予約されるルールを反映しています。GID、**User Private Group** のグループ ID は UID に等しくなります。ホームディレクトリーは /**home**/**robert** に、ログインシェルを /**bin**/**bash** に設定します。X の文字 **は**、シャドウパスワードが使用され、ハッシュ化されたパスワードが /**etc**/**shadow** に保存されることを示唆します。

アカウントの作成時にユーザーの基本デフォルト設定を変更する場合は、**useradd** の動作を変更するコマンドラインオプションのリストから選択できます（オプションの一覧は、man ページの **useradd**(8) を参照してください）。コマンドの基本的な構文で分かるように、オプションを 1 つ以上追加できます。

```
useradd [option(s)] username
```

システム管理者は、**-c** オプションを使用して、作成時にユーザーのフルネームなどを指定できます。**-c** の後に文字列を付けて、ユーザーにコメントを追加します。

```
useradd -c "string" username
```

例3.2 ユーザー作成時のユーザーのフルネームの指定

```
~]# useradd -c "Robert Smith" robert
~]# cat /etc/passwd
robert:x:502:502:Robert Smith:/home/robert:/bin/bash
```

ユーザー名 **robert** でユーザーアカウントが作成されている。ログイン名およびフルネーム Robert Smith で作成されていること。

ユーザーアカウントにデフォルトの /**home**/ ***ユーザー名***/ ディレクトリーを作成したくない場合は、その代わりに別のユーザー名を設定します。以下のコマンドを実行します。

useradd -d *home_directory*

例3.3 デフォルト以外のホームディレクトリーでのユーザーの追加

~]# useradd -d /home/dir_1 robert

robert のホームディレクトリーはデフォルトの /**home**/**robert** ではなく、/**home**/**dir_1**/ **です**。

ユーザーのホームディレクトリーを作成したくない場合は、**useradd** に **-M** オプションを指定して実行できます。ただし、このようなユーザーが起動したばかりのシステムにログインし、ホームディレクトリーが存在しない場合は、ログインディレクトリーは root ディレクトリーになります。このようなユーザーが **su** コマンドを使用してシステムにログインすると、ログインディレクトリーは以前のユーザーの現在のディレクトリーになります。

useradd -M *username*

新規ユーザーの作成時にディレクトリーコンテンツを /**home** ディレクトリーにコピーする必要がある場合は、**-m** オプションと **-k** オプションの後にパスを指定して実行します。

例3.4 ホームディレクトリーへのコンテンツのコピー時のユーザーの作成

以下のコマンドは、/**dir_1** という名前のディレクトリーの内容を /**home**/**jane** にコピーします。これは、新しいユーザー **jane** のデフォルトのホームディレクトリーです。

~]# useradd -m -k /dir_1 jane

システム管理者は、一時的なアカウントを作成する必要がある場合があります。**useradd** コマンドを使用すると、特定の期間だけアカウントを作成し、特定日に無効にすることができます。特定のアカウントの削除にセキュリティー上のリスクがあるため、これは特に便利な設定です。このため、**-e** オプションは、YYYY-MM-DD 形式で指定された *expire_date* とともに使用されます。

注記

アカウントの有効期限とパスワードの有効期限を混同しないようにしてください。アカウントの有効期限は特定の日付で、アカウントが存在しなくなったため、アカウントにログインできなくなります。パスワードの有効期限、パスワード作成またはパスワード変更の最大有効期間と日付は、パスワードを使用してログインできない場合は日付になります（ただし、SSH キーを使用してログインする場合など）。

useradd -e *YYYY-MM-DD username*

例3.5 アカウントの有効期限の設定

```
~]# useradd -e 2015-11-05 emily
```

このアカウントは今後作成され、2015 年 11 月 5 日に自動的に無効になります。

ユーザーのログインシェルのデフォルトは /bin/bash ですが、-s オプションで bash、ksh、csh、tsh とは異なるシェルに変更できます。

```
useradd -s login_shell username
```

例3.6 デフォルト以外のシェルでユーザーの追加

```
~]# useradd -s /bin/ksh robert
```

このコマンドは、/bin/ksh シェルを持つ robert ユーザーを作成します。

-r オプションは、一部の root 権限ではなく、管理者が使用するアカウントとなるシステムアカウントを作成します。このようなアカウントは、/etc/login.defs で定義された *UID_MIN* の値よりも低い値を持ちます（通常は通常のユーザーの場合 500 以上）。

```
useradd -r username
```

3.4.2. 新規ユーザーのグループへの割り当て

useradd コマンドは、新しいユーザーがシステムに追加され、ユーザーの名前の後にグループに名前を付けるたびに、ユーザー プライベート グループ（*UPG*、ユーザーのみが属するユーザーアカウントに割り当てられたグループ）を作成します。たとえば、アカウントの robert が作成されると、robert という名前の UPG が同時に作成されます。そのメンバーは、ユーザー robert だけです。

何らかの理由でユーザープライベートグループ を作成し たくない場合は、useradd コマンドに以下のオプションを付けて実行します。

```
useradd -N username
```

UPG を自動的に作成したり、何も作成したりするのではなく、-g オプションおよび -G オプションでユーザーのグループメンバーシップを指定できます。-g オプションはプライマリーグループメンバーシップを指定しますが、-G はユーザーも含まれる補助グループを参照します。指定するグループ名がシステムにすでに存在している必要があります。

例3.7 グループへのユーザーの追加

```
~]# useradd -g "friends" -G "family,schoolmates" emily
```

useradd -g "friends" -G "family,schoolmates" emily コマンドは、ユーザーを作成しますが、's primary group は、-g オプションで指定したように friends に設定されます。Emily は、 補助グループ ファミリー と 授業におけるグループメンバーでもあります。

ユーザーがすでに存在し、特定の補助グループに追加する場合は、-G オプションを指定して usermod コマンドを使用し、コンマで区切られたグループの一覧（スペースなし）を使用します。

```
usermod -G group_1,group_2,group_3
```

3.4.3. ユーザーの認証の更新

基本的な useradd *username* コマンドを実行すると、パスワードは自動的に期限切れになりません（ /etc/shadow ファイルを参照）。

これを変更する場合は、passwd を使用して /etc/passwd ファイルを管理する標準ユーティリティーを使用します。passwd コマンドの構文は、以下のようになります。

```
passwd option(s) username
```

たとえば、指定したアカウントをロックできます。このロックは、暗号化された文字列を感嘆符(!)のプレフィックスを付けて無効な文字列にレンダリングすることで、暗号化されたパスワードを無効な文字列にレンダリングすることで実行されます。後でアカウントのロックを解除する理由を見つけた場合は、passwd にロックのためのリバース操作があります。これらの 2 つの操作を実行できるのは root のみです。

```
passwd -l username
passwd -u username
```

例3.8 ユーザーパスワードのロック解除

```
~]# passwd -l robert
Locking password for user robert.
passwd: Success
~]# passwd -u robert
passwd: Warning: unlocked password would be empty
passwd: Unsafe operation (use -f to force)
```

最初は、-l オプションが robert のアカウントパスワードを正常にロックします。ただし、passwd -u コマンドを実行すると、パスワードレスアカウントの作成を拒否するため、パスワードのロックを解除しません。

アカウントのパスワードの有効期限が切れるようにするには、-e オプションを指定して passwd を実行します。ユーザーは、次回のログイン時にパスワードの変更を強制します。

```
passwd -e username
```

パスワードの有効期間が重要であるため、パスワードの変更の最小時間を設定することは、ユーザーが実際にパスワードを変更するのに便利です。システム管理者は、最小（ -n オプション）および最大（ -x オプション）の有効期間を設定できます。パスワードの有効期限についてユーザーに通知するには、-w オプションを使用します。これらのオプションはすべて日数とともに指定する必要があり、root でのみ実行できます。

例3.9 ユーザーパスワードのデータの調整

```
~]# passwd -n 10 -x 60 -w 3 jane
```

上記のコマンドでは、最小パスワード有効期間を 10 日間、パスワードの最大有効期間を 60 日に設定し、自分のパスワードが 3 日後に期限切れになる前に警告を受信し始めます。

後で、パスワード設定を記憶できない場合は、-S オプションを使用して、特定のアカウントのパスワードステータスを把握するための短い情報を出力します。

```
~]# passwd -S jane
jane LK 2014-07-22 10 60 3 -1 (Password locked.)
```

また、useradd コマンドを使用して、パスワードの期限が切れるまでの日数を設定することもできます。これにより、アカウントが完全に無効になります。値が 0 の場合は、パスワードの期限が切れるとすぐにアカウントが無効になり、値が - 1 の場合はこの機能が無効になります。つまり、ユーザーはパスワードの期限が切れたときにパスワードを変更する必要があります。-f オプションは、アカウントが無効になるまで、パスワードの期限が切れてから日数を指定するために使用されます（ただし、システム管理者がブロック解除することもできます）。

```
useradd -f number-of-days username
```

passwd コマンドの詳細は、man ページの passwd(1)を参照してください。

3.4.4. ユーザー設定の変更

ユーザーがすでに存在し、現在のオプションのいずれかを指定する必要がある場合は、usermod コマンドを使用します。usermod を使用するロジックは、useradd と、その構文と同じです。

```
usermod option(s) username
```

ユーザーのユーザー名を変更する必要がある場合は、-l オプションを新しいユーザー名（または login）で使用します。

例3.10 ユーザーのログインの変更

```
~]# usermod -l "emily-smith" emily
```

-l オプションは、ユーザー名をログインから新しいログインに変更します(emily -smith)。それ以外は何も変更しません。特に、新しいユーザー名を反映するために手動で変更した場合を除き、ホームディレクトリー名(/home/emily)は変わりません。

同様に、ユーザーの UID またはユーザーのホームディレクトリーを変更できます。以下に例を示します。

注記

システムの指定された UID が所有するすべてのファイルを検索して、所有者を変更します。UID を参照するアクセス制御リスト(ACL)で同じことを行います。古い UID で実行中のプロセスがないことを確認することが推奨されます。

例3.11 ユーザーの UID およびホームディレクトリーの変更

```
~]# usermod -a -u 699 -d /home/dir_2 robert
```

-a -u オプションおよび -d オプションを指定すると、ユーザー robert の設定が変更されます。これで、ID は 501 ではなく 699 で、ホームディレクトリーは /home/robert ではなくなりましたが、/home/dir_2 はなくなりました。

usermod コマンドを使用すると、ユーザーのホームディレクトリーの内容を新しい場所に移動するか、パスワードをロックしてアカウントをロックすることもできます。

例3.12 ユーザーの変更

```
~]# usermod -m -d /home/jane -L jane
```

このサンプルコマンドでは、一緒に使用する **-m** オプションおよび **-d** オプションは、**jane** のホームディレクトリーのコンテンツを **/home/dir_3** ディレクトリーに移動します。**-L** オプションは、パスワードをロックして **jane** のアカウントへのアクセスをロックします。

usermod コマンドで使用するオプションの一覧は、**man** ページの **usermod (8)**か、コマンドラインで **usermod --help** を実行します。

3.4.5. ユーザーの削除

システムからユーザーアカウントを削除するには、**root** でコマンドラインで **userdel** コマンドを使用します。

```
userdel username
```

userdel と **-r** オプションを組み合わせると、ユーザーのホームディレクトリー内にあるファイルがホームディレクトリー自体と、ユーザーのメールスプールと共に削除されます。他のファイルシステムにあるファイルは、手動で検索して削除する必要があります。

```
userdel -r username
```

注記

-r オプションは比較的安全であるため、**-f** と比較すると、ユーザーがログインしている場合でもユーザーアカウントを強制的に削除することが推奨されます。

3.4.6. 機密ユーザー情報の表示

システムでユーザーとグループを管理する場合は、システム上の設定とアクティビティーを監視するための適切なツールが必要です。**Red Hat Enterprise Linux 6** は、**lslogins** コマンドラインユーティリティーを提供します。このユーティリティーは、ユーザーまたはグループの包括的な概要を提供します。これは、ユーザーまたはグループの設定だけでなく、システム上のアクティビティーも対象です。

lslogins の一般的な構文は以下のとおりです。

lslogins [*OPTIONS*]

OPTIONS **は、1 つ以上の利用可能なオプションとその関連パラメーターになります。利用可能なオプションとその使用の完全なリストは、lslogins(1)の man ページ、または lslogins --help コマンドの出力を参照してください。**

lslogins ユーティリティーは、選択したオプションに基づいて、さまざまな形式で汎用情報を提供します。以下の例では、最も基本的な組み合わせと、最も便利な組み合わせを紹介します。

オプションを指定せずに lslogins コマンドを実行すると、システム上のすべてのシステムおよびユーザーアカウントのデフォルト情報が表示されます。具体的には、UID、ユーザー名、および GECOS 情報、およびシステムへの最後にログインしたユーザーの情報、およびパスワードによるロックまたはログインが無効になっているかどうかなどです。

例3.13 システムにあるすべてのアカウントに関する基本情報の表示

```
~]# lslogins
  UID USER         PWD-LOCK PWD-DENY  LAST-LOGIN GECOS
    0 root                 0        0             root
    1 bin                  0        1             bin
    2 daemon               0        1             daemon
    3 adm                  0        1             adm
    4 lp                   0        1             lp
    5 sync                 0        1             sync
    6 shutdown             0        1 Jul21/16:20 shutdown
    7 halt                 0        1             halt
    8 mail                 0        1             mail
   10 uucp                 0        1             uucp
   11 operator             0        1             operator
   12 games                0        1             games
   13 gopher               0        1             gopher
   14 ftp                  0        1             FTP User
   29 rpcuser              0        1             RPC Service User
   32 rpc                  0        1             Rpcbind Daemon
   38 ntp                  0        1
   42 gdm                  0        1
   48 apache               0        1             Apache
   68 haldaemon            0        1             HAL daemon
   69 vcsa                 0        1             virtual console memory owner
   72 tcpdump              0        1
   74 sshd                 0        1             Privilege-separated SSH
   81 dbus                 0        1             System message bus
   89 postfix              0        1
   99 nobody               0        1             Nobody
  113 usbmuxd              0        1             usbmuxd user
  170 avahi-autoipd        0        1             Avahi IPv4LL Stack
  173 abrt                 0        1
  497 pulse                0        1             PulseAudio System Daemon
  498 saslauth             0        1             Saslauthd user
```

```
 499 rtkit              0       1           RealtimeKit
 500 jsmith             0       0    10:56:12 John Smith
 501 jdoe               0       0    12:13:53 John Doe
 502 esmith             0       0    12:59:05 Emily Smith
 503 jeyre              0       0    12:22:14 Jane Eyre
65534 nfsnobody         0       1           Anonymous NFS User
```

単一ユーザーに関する詳細情報を表示するには、lslogins *LOGIN* コマンドを実行します。ここで、*LOGIN* は UID またはユーザー名になります。以下の例は、John Doe のアカウントと、システム上のアクティビティーに関する詳細情報を表示します。

例3.14 1 つのアカウントに関する詳細情報の表示

```
~]# lslogins jdoe
Username:                       jdoe
UID:                            501
Gecos field:                    John Doe
Home directory:                 /home/jdoe
Shell:                          /bin/bash
No login:                       no
Password is locked:             no
Password no required:           no
Login by password disabled:     no
Primary group:                  jdoe
GID:                            501
Supplementary groups:           users
Supplementary group IDs:        100
Last login:                     12:13:53
Last terminal:                  pts/3
Last hostname:                  192.168.100.1
Hushed:                         no
Password expiration warn interval:  7
Password changed:               Aug01/02:00
Maximal change time:            99999
Password expiration:            Sep01/02:00
Selinux context:                unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023
```

--logins=*LOGIN* オプションを使用すると、UID またはユーザー名の一覧として指定されたアカウントのグループに関する情報を表示できます。--output=*COLUMNS* オプションの指定。*COLUMNS* は利用可能な出力パラメーターのリストであるため、lslogins コマンドの出力をカスタマイズできます。たとえば、以下のコマンドは、ユーザー root、jsmith、jdoe、および esmith のログインアクティビティーを表示します。

例3.15 ユーザーのグループに関する特定情報の表示

```
~]# lslogins --logins=0,500,jdoe,esmith \
> --output=UID,USER,LAST-LOGIN,LAST-TTY,FAILED-LOGIN,FAILED-TTY
UID USER   LAST-LOGIN LAST-TTY FAILED-LOGIN FAILED-TTY
```

```
  0 root
500 jsmith   10:56:12 pts/2
501 jdoe     12:13:53 pts/3
502 esmith   15:46:16 pts/3    15:46:09     ssh:notty
```

lslogins ユーティリティーは、システムとユーザーアカウントを区別します。クエリー内のシステムアカウントに対応するには、--system-accs オプションを使用します。ユーザーアカウントに対応するには、--user-accs を使用します。たとえば、以下のコマンドは、すべてのユーザーアカウントの補助グループとパスワードの有効期限に関する情報を表示します。

例3.16 すべてのユーザーアカウントの補助グループとパスワードの有効期限に関する情報の表示

```
~]# lslogins --user-accs --supp-groups --acc-expiration
  UID USER        GID GROUP     SUPP-GIDS SUPP-GROUPS PWD-WARN PWD-MIN PWD-
MAX PWD-CHANGE
PWD-EXPIR
    0 root          0 root                              7          99999 Jul21/02:00
  500 jsmith      500 jsmith    1000,100  staff,users       7          99999 Jul21/02:00
  501 jdoe        501 jdoe      100       users             7          99999 Aug01/02:00
Sep01/02:00
  502 esmith      502 esmith    100       users             7          99999 Aug01/02:00
  503 jeyre       503 jeyre     1000,100  staff,users       7          99999 Jul28/02:00
Sep01/02:00
65534 nfsnobody 65534 nfsnobody                                        Jul21/02:00
```

ユーザーのニーズに従って lslogins コマンドの出力をフォーマットする機能により、lslogins はスクリプトでの使用や自動処理に理想的なツールになります。たとえば、以下のコマンドは最後のログインの日時を表す単一の文字列を返します。この文字列は、さらなる処理のために別のユーティリティーへの入力として渡すことができます。

例3.17 見出しのない単一の情報の表示

```
~]# lslogins --logins=jsmith --output=LAST-LOGIN --time-format=iso | tail -1
2014-08-06T10:56:12+0200
```

3.5. コマンドラインツールを使用したグループの管理

グループは、異なるユーザー間で連携できるようにする便利なツールです。groupadd、groupmod、groupdel、gpasswd などのグループを操作するコマンドのセットがあります。影響を受けるファイルには、グループのアカウント情報を保存する /etc/group と、セキュアなグループのアカウント情報を格納する /etc/gshadow が含まれます。

3.5.1. グループの作成

デフォルト設定で新しいグループをシステムに追加するには、root で groupadd コマンドを実行します。

```
groupadd group_name
```

例3.18 デフォルト設定を使用したグループの作成

```
~]# groupadd friends
```

groupadd コマンドは、friends という新しいグループを作成します。グループの詳細は、/etc/group ファイルの新たに作成された行を参照してください。

```
classmates:x:30005:
```

自動的に、グループ friends は一意の GID（グループ ID）が 30005 にアタッチされ、どのユーザーも接続されません。必要に応じて、gpasswd *groupname* を実行してグループのパスワードを設定できます。

または、特定の設定でコマンドオプションを追加することもできます。

```
groupadd option(s) groupname
```

たとえば、グループの作成時にグループの ID(GID)の数値を指定する場合は、-g オプションを指定して groupadd コマンドを実行します。この値は固有で（ -o オプションを使用しない限り）一意でなければならず、値は負の値ではない必要があることに注意してください。

```
groupadd -g GID
```

例3.19 GID が指定されたグループの作成

以下のコマンドは schoolmates という名前のグループを作成し、そのグループに 60002 の GID を設定します。

```
~]# groupadd -g 60002 schoolmates
```

-g および *GID* がすでに存在する場合、groupadd は、既存の GID を持つ別のグループの作成を拒否します。回避策として、groupadd で -f オプションを使用してグループが作成されますが、別の *GID* を使用します。

```
groupadd -f GID
```

-r オプションを groupadd コマンドにアタッチして、システムグループを作成することもできます。システムグループは、実際には 999 の範囲内で 1 から 499 に割り当てられることを意味します。

```
groupadd -r group_name
```

groupadd の詳細は、man ページの groupadd (8)を参照してください。

3.5.2. グループへのユーザーのアタッチ

既存ユーザーを named グループに追加する場合は、gpasswd コマンドを使用できます。

```
gpasswd -a username which_group_to_edit
```

named グループからユーザーを削除するには、次のコマンドを実行します。

```
gpasswd -d username which_group_to_edit
```

グループメンバーの一覧を設定するには、--members オプションの後にユーザー名をコンマで区切って指定します。

```
gpasswd --members username_1,username_2 which_group_to_edit
```

3.5.3. グループ認証の更新

gpasswd コマンドは、/etc/group ファイルおよび /etc/gshadow ファイルを管理します。このコマンドは、グループ管理者が実行した場合に限り機能することに注意してください。

グループ管理者のユーザーグループ管理者は、ユーザーを追加および削除したり、グループパスワードを設定したり、削除したりすることができます。グループには、複数のグループ管理者を指定できます。root ユーザーは、gpasswd -A *users groupname* を使用してグループ管理者を追加できます。ここで、*ユーザー* は、グループ管理者にしたい既存ユーザーのコンマ区切りの一覧です（コンマの間にスペースは一切ありません）。

グループのパスワードを変更する場合は、関連するグループ名を指定して **gpasswd** コマンドを実行します。グループの新しいパスワードを入力するように求められます。

```
gpasswd groupname
```

例3.20 グループパスワードの変更

```
~]# gpasswd crowd
Changing password for group crowd
New password:
Re-enter new password:
```

グループ **crowd** のパスワードが変更されました。

-r オプションを使用して、**named** グループからパスワードを削除することもできます。

```
gpasswd -r schoolmates
```

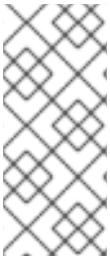
3.5.4. グループ設定の変更

グループが存在し、現在のオプションのいずれかを指定する必要がある場合は、**groupmod** コマンドを使用します。**groupmod** を使用するロジックは **groupadd** と、その構文と同じです。

```
groupmod option(s) groupname
```

指定したグループのグループ **ID** を変更するには、以下のように **groupmod** コマンドを使用します。

```
groupmod -g GID_NEW which_group_to_edit
```



注記

システムで指定の **GID** が所有するすべてのファイルを検索して、所有者を変更します。**GID** を参照するアクセス制御リスト(**ACL**)で同じことを行います。古い **GID** で実行しているため、実行中のプロセスを確認することが推奨されます。

グループの名前を変更するには、コマンドラインで次のコマンドを実行します。グループの名前は、**GROUP_NAME** から ***NEW_ GROUP_NAME*** 名に変更されます。

```
groupmod -n new_groupname groupname
```

例3.21 グループ名の変更

以下のコマンドは、グループ **schoolmates** の名前を **crowd** に変更します。

```
~]# groupmod -n crowd schoolmates
```

3.5.5. グループの削除

groupdel コマンドは、システムアカウントファイルを変更し、グループを表示するすべてのエントリーを削除します。このコマンドを実行すると、名前付きグループが存在する必要があります。

```
groupdel groupname
```

3.6. その他のリソース

ユーザーとグループの管理に関する詳細情報については、以下のリソースを参照して下さい。

3.6.1. インストールされているドキュメント

ユーザーおよびグループの管理に使用する各種ユーティリティーの詳細情報は、以下の **man** ページを参照してください。

- **chage(1) — パスワードエージングのポリシーとアカウントの有効期限を修正するコマンドです。**
- **gpasswd(1)- /etc/group ファイルを管理するコマンドです。**
- **groupadd(8) — グループを追加するコマンドです。**
- **grpck(8): /etc/group ファイルを検証するコマンドです。**

- groupdel(8) — グループを削除するコマンドです。
- groupmod(8) — グループのメンバーシップを修正するコマンドです。
- pwck(8): /etc/passwd ファイルおよび /etc/shadow ファイルを検証するコマンド。
- pwconv(8) — 通常のパスワードをシャドウパスワードに変換するツールです。
- pwunconv(8) — シャドウパスワードを通常のパスワードに変換するツールです。
- useradd(8) — ユーザーを追加するコマンドです。
- userdel(8) — ユーザーを削除するコマンドです。
- usermod(8) — ユーザーを修正するコマンドです。

関連する設定ファイルの詳細は、以下をご覧ください。

- group(5) — システムのグループ情報を含むファイルです。
- passwd(5) — システムのユーザー情報を含むファイルです。
- shadow(5) — システムのパスワードとアカウントの有効期限に関する情報を含むファイルです。ワードおよびアカウントの有効期限情報を含むファイル。
- login.defs(5): シャドウパスワードスイート設定を含むファイルです。
- useradd(8): /etc/default/useradd の場合は、man ページの「Changing the default values」セクションを参照してください。

第4章 権限の取得

システム管理者は (ユーザーも時には) 管理者アクセスでタスクを実行する必要があります。システムに root としてアクセスすることは危険を伴う可能性があり、システムおよびデータの著しい破損につながる場合もあります。本章では、su および sudo プログラムを使用して管理者権限を取得する方法を説明します。これらのプログラムを使うと、特定のユーザーが高レベルの制御およびシステムセキュリティーを維持しつつ、通常は root ユーザーしかできないタスクを実行することができます。

管理コントロール、潜在的な危険、および権限アクセスの不適切な使用を防ぐ方法についての詳細は、『Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;LinuxRed Hat Enterprise Linux 6nbsp;6 セキュリティーガイド』 を参照してください。

4.1. SU コマンド

ユーザーが su コマンドを実行すると、root パスワードが要求されます。認証されると root シェルプロンプトが表示されます。

su コマンドでログインすると、そのユーザーは root ユーザーと なり、システムへの絶対管理アクセスを持つことになります。[1].さらに、ユーザーが root になったら、パスワードを求められることなく、su コマンドを使用してシステム上の他のユーザーに変更を加えることができます。

このプログラムは非常に強力なので、組織内の管理者はこのコマンドにアクセスできるユーザーを制限してください。

最も簡単な方法は、*wheel* と呼ばれる特別な管理グループにユーザーを追加することです。これを実行するには、以下のコマンドを root で入力します。

```
~]# usermod -a -G wheel username
```

このコマンドで、*username* を、wheel グループに追加するユーザー名に置き換えます。

また、User Manager を使用して以下のようにグループメンバーシップを変更することもできます。この手順を実行するには、管理者権限が必要なことに注意してください。

1. パネル上の システム メニューをクリックして 管理 から ユーザーとグループ をクリックして User Manager を表示させます。または、シェルプロンプトで system-config-users コマンドを入力します。

2. ユーザー タブをクリックして、ユーザーリストの中から必要なユーザーを選択します。

3. ツールバーの 設定 をクリックして、ユーザー設定のダイアログボックスを表示させます (または ファイル メニューで 設定 を選択します)。

4. グループ タブをクリックし、wheel グループのチェックボックスにチェックマークを付けて OK をクリックします。

ユーザーマネージャーの詳細は、「ユーザーマネージャーアプリケーションを使用したユーザーの管理」 を参照 し てください。

wheel グループにユーザーを追加したら、この特定のユーザーにのみ su コマンドの使用を許可することが推奨されます。これを行うには、su:/etc/pam.d/su の PAM 設定ファイルを編集する必要があります。このファイルをテキストエディターで開き、以下の行からコメント (#) を削除します。

```
#auth          required        pam_wheel.so use_uid
```

この変更により、wheel の管理グループのメンバーのみが、su コマンドを使用して別のユーザーに切り換えることができるようになります。

注記

root ユーザーは、デフォルトで wheel グループに含まれます。

4.2. SUDO コマンド

ユーザーに管理アクセスを付与する別のアプローチとして sudo コマンドを利用できます。信頼されたユーザーが、管理コマンドの前に sudo を付けると、それらは 独自 のパスワードを要求します。ユーザーが認証され、コマンドが許可されると、管理コマンドは root 権限で実行されているかのように実行されます。

sudo コマンドの基本的なフォーマットは、以下のとおりです。

```
sudo <command>
```

上記の例では、& *lt;command&* gt; は、通常、mount などの root ユーザー用に予約されているコマンドに置き換えられます。

sudo コマンドでは、ハイレベルの柔軟性が可能になります。たとえば、/etc/sudoers 設定ファイルに一覧表示されているユーザーのみが sudo コマンドを使用することができ、root シェルではなく、その ユーザーの シェルでコマンドが実行されます。これは、Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;Linux 『Red Hat Enterprise Linux 6nbsp;6 セキュリティーガイド』 に示されるように、root シェルを完全に無効にできることを意味します。

sudo を使用した正常な認証のログはすべて /var/log/messages ファイルに記録され、このコマンドを実行したユーザー名で実行されたコマンドは /var/log/secure ファイルに記録されます。追加のロギングが必要な場合は、以下の行を /etc/pam.d/system-auth ファイルに追加して、pam_tty_audit モジュールを使用して、指定したユーザーの TTY 監査を有効にします。

```
session required pam_tty_audit.so disable=<pattern> enable=<pattern>
```

pattern は、オプションでグロブを使用し、ユーザーのコンマ区切りリストを表します。例えば、以下の設定は、root ユーザーのTTY 監査を有効にし、その他のユーザーについては無効にします。

```
session required pam_tty_audit.so disable=* enable=root
```

sudo コマンドのもう 1 つの利点は、管理者がそれぞれのニーズに応じて特定のコマンドへのアクセスを管理者が許可できることです。

管理者が sudo 設定ファイル /etc/sudoers を編集する場合は、visudo コマンドを使用する必要があります。

他のユーザーに完全な管理権限を付与するには、visudo と入力し、ユーザー権限の指定セクションに以下のような行を追加します。

```
juan ALL=(ALL) ALL
```

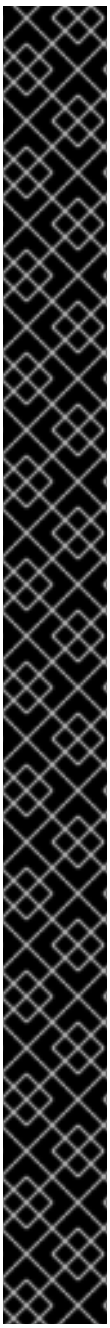
この例では、ユーザー juan は、任意のホストから sudo を使用し、任意のコマンドを実行できます。

以下の例では、sudo を設定する際に可能な粒度を示しています。

```
%users localhost=/sbin/shutdown -h now
```

この例では、コンソールから実行した限り、すべてのユーザーがコマンド /sbin/shutdown -h now を実行できることを示しています。

sudoers の man ページには、このファイルのオプションの詳細なリストが記載されています。



重要

sudo コマンドの使用時には、潜在的なリスクがいくつか存在することを覚えておく必要があります。上記のように visudo を使用して /etc/sudoers 設定ファイルを編集することで回避できます。/etc/sudoers ファイルをデフォルトの状態にしておくと、wheel グループのすべてのユーザーが無制限の root アクセスを許可します。

- デフォルトでは、sudo は sudoer のパスワードを 5 分間保存します。この間はコマンドを続けて使用しても、ユーザーはパスワードを要求されません。このため、ユーザーがログイン状態のままワークステーションを離れたりロックしない状態にしておくと、攻撃者に悪用されかねません。この動作は、以下の行を /etc/sudoers ファイルに追加することで変更できます。

```
Defaults    timestamp_timeout=<value>
```

<value> には、指定するタイムアウトの分数を入れます。< *value*> を 0 に設定すると、sudo は毎回パスワードを要求します。

- sudoer のアカウントが侵害されると、攻撃者は sudo を使用して管理権限のある新たなシェルを開くことができます。

```
sudo /bin/bash
```

この方法や同様の方法で root として新しいシェルを開くと、/etc/sudoers ファイルで指定されたタイムアウト時間を無視し、新たに開かれたセッションが閉じられるまで攻撃者に sudo のパスワード入力を要求することがありません。

4.3. その他のリソース

ユーザーに管理者権限を与えるプログラムは潜在的なセキュリティーリスクでありますが、セキュリティーの説明は本ガイドの対象外となります。このため、セキュリティーや権限のアクセスに関する情報に関しては、以下に挙げるリソースを参照してください。

インストールされているドキュメント

- su(1)- su の man ページには、このコマンドで利用可能なオプションの情報があります。
- sudo (8): sudo の man ページには、このコマンドの動作をカスタマイズするためのオプションの一覧と、このコマンドの詳細な説明が含まれています。
- pam(8) - man ページでは、 Linux 向け Pluggable Authentication Modules の使用方法が説明されています。

オンラインドキュメント

- Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;LinuxRed Hat Enterprise Linux 6nbsp;6『Security Guide』：セキュリティーガイドでは、セキュリティーリスクの詳細と権限を取得するプログラムに関連する技術を軽減しています。
- Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;LinuxRed Hat Enterprise Linux 6nbsp;6 Managing Single Sign-On and Smart Cards: 本書では、プラグ可能な認証モジュール(PAM)、その設定と使用について詳細に説明しています。

---

[1] このアクセスは、SELinux の制限が有効な場合は、この制限対象となります。

第5章 コンソールアクセス

通常（root 以外の）ユーザーがコンピューターにローカルでログインする場合は、以下の 2 種類の特別なパーミッションが付与されます。

1. そうでないと実行できない特定のプログラムを実行できます。

2. そうでない場合にはアクセスできない特定のファイルにアクセスできます。これらのファイルには通常、ディスクへのアクセスに使用される特殊なデバイスファイル、CD-ROM などが含まれます。

1 台のコンピューターには複数のコンソールがあり、複数のユーザーを同時にコンピューターにログインできるため、ユーザーの 1 つは基本的にファイルにアクセスするための競合を判断する必要があります。コンソールにログインする最初のユーザーはこれらのファイルを所有します。最初のユーザーがログアウトすると、ファイルを所有する次のユーザーが表示されます。

一方、コンソールにログインしているすべてのユーザーは、通常、root ユーザーに制限のあるタスクを実行するプログラムを実行できます。X を実行している場合は、グラフィカルユーザーインターフェースにメニュー項目としてこれらのアクションを追加できます。これらのコンソールアクセス可能なプログラムには、halt、poweroff、reboot が含まれます。

5.1. ROOT 以外のユーザーのコンソールプログラムアクセスの無効化

root 以外のユーザーは、/etc/security/console.apps/ ディレクトリー内のプログラムへのコンソールアクセスを拒否することができます。これらのプログラムを一覧表示するには、以下のコマンドを実行します。

```
~]$ ls /etc/security/console.apps
abrt-cli-root
config-util
eject
halt
poweroff
reboot
rhn_register
setup
subscription-manager
subscription-manager-gui
system-config-network
system-config-network-cmd
xserver
```

これらのプログラムごとに、プログラムの PAM(*Pluggable Authentication Module*)設定ファイルを使用してコンソールアクセス拒否を設定できます。PAM とその使用方法は、『Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;LinuxRed Hat Enterprise Linux 6nbsp;Linux Red Hat Enterprise Linux 6nbsp;6 『Managing Single Sign-On and Smart Cards』 』の「 Pluggable Authentication Modules 」の章を参照してください。

/etc/security/console.apps/ 内の各プログラムの PAM 設定ファイルは、/etc/pam.d/ ディレクトリーにあり、プログラムと同じ名前が付けられます。このファイルを使用して、ユーザーが root でない場合、プログラムへのアクセスを拒否するように PAM を設定できます。これを行うには、最初にコメント解除された行 auth requisite pam_deny.so の直後に auth sufficient pam_rootok.so 行を挿入します。

例5.1 再起動プログラムへのアクセスの無効化

/etc/security/console.apps/reboot への root 以外のコンソールアクセスを無効にするには、auth requisite pam_deny.so 行 を /etc/pam.d/reboot PAM 設定ファイルに挿入します。

```
#%PAM-1.0
auth       sufficient   pam_rootok.so
auth       requisite    pam_deny.so
auth       required     pam_console.so
#auth       include     system-auth
account    required     pam_permit.so
```

この設定では、reboot ユーティリティーへの root 以外のアクセスがすべて無効になります。

さらに、/etc/security/console.apps/ の複数のプログラムは、/etc/pam.d/config-util 設定ファイルから PAM 設定を部分的に派生します。これにより、/etc/pam.d/config-util を編集して、これらすべてのプログラムの設定を一度に変更できます。これらのプログラムをすべて検索するには、config-util ファイルを参照する PAM 設定ファイルを検索します。

```
~]# grep -l "config-util" /etc/pam.d/*
/etc/pam.d/abrt-cli-root
/etc/pam.d/rhn_register
/etc/pam.d/subscription-manager
/etc/pam.d/subscription-manager-gui
/etc/pam.d/system-config-network
/etc/pam.d/system-config-network-cmd
```

上記のようにコンソールプログラムアクセスを無効にすると、コンソールがセキュリティー保護されている環境では便利です。セキュリティー対策には、BIOS およびブートローダーのパスワード保護、Ctrl+Alt+Delete の操作で再起動の無効化、電源およびリセットスイッチの無効化などが含まれる

場合があります。このような場合、デフォルトでコンソールからアクセス可能なプログラム、電源オフ、 再 起動 など のプログラムへの通常のユーザーのアクセス権を制限することができます。

5.2. CTRL+ALT+DEL を使用した再起動の無効化

コンソールで Ctrl+Alt+Del を押す応答として発生するアクションは、/etc/init/control-alt-delete.conf ファイルに指定されます。デフォルトでは、-r オプションを指定した shutdown ユーティリティーが、システムのシャットダウンおよび再起動に使用されます。

このアクションを無効にするには、exec true コマンドを指定するオーバーライド設定ファイルを作成します。これを実行するには、root で以下のコマンドを実行します。

```
~]# echo "exec true" >> /etc/init/control-alt-delete.override
```

パート II. サブスクリプションおよびサポート

Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux システムのソフトウェアへの更新を受け取るには、*Red Hat コンテンツ配信ネットワーク* (CDN)にサブスクライブし、適切なリポジトリーを有効にする必要があります。ここでは、システムを Red Hat コンテンツ配信ネットワークにサブスクライブする方法を説明します。

Red Hat は カスタマーポータル からサポートを提供します。このサポートは、Red Hat Support Tool を使用してコマンドラインから直接アクセスできます。ここでは、そのコマンドラインツールの使用方法を説明します。

第6章 システム登録およびサブスクリプション管理

サブスクリプションサービスは、Red Hat ソフトウェアインベントリーを処理するメカニズムを提供し、yum または PackageKit パッケージマネージャーを使用して、追加のソフトウェアをインストールしたり、インストールされているプログラムを新しいバージョンに更新したりできます。Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;LinuxRed Hat Enterprise Linux 6nbsp;6 では、システムを登録し、サブスクリプションを割り当てる方法として、*Red Hat Subscription Management* を使用することが推奨されます。

注記

また、初回起動プロセス中にインストール後にシステムを登録し、サブスクリプションを割り当てることもできます。firstboot の詳細は、Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;LinuxRed Hat Enterprise Linux 6nbsp;6 の『インストールガイド』の「先起動」の章を参照してください。firstboot は、グラフィカルインストール後、またはデスクトップと X ウィンドウシステムがインストールされ、グラフィカルログインが有効になっているキックスタートのインストール後にのみ使用できることに注意してください。

6.1. システム登録およびサブスクリプションの割り当て

Red Hat Subscription Management を使用してシステムを登録し、1 つ以上のサブスクリプションを割り当てる手順を完了してください。subscription-manager コマンドはすべて root で実行することに注意してください。

1. 以下のコマンドを実行してシステムを登録します。ユーザー名とパスワードを入力するように求められます。ユーザー名とパスワードは、Red Hat カスタマーポータルのログイン認証情報と同じであることに注意してください。

```
subscription-manager register
```

2. 必要なサブスクリプションのプール ID を確認します。これを行うには、シェルプロンプトで以下のコマンドを入力し、システムで利用できるサブスクリプションの一覧を表示します。

```
subscription-manager list --available
```

このコマンドは、利用可能な各サブスクリプションの名前、固有 ID、有効期限、およびそのサブスクリプションに関連するその他の詳細情報を表示します。すべてのアーキテクチャーのサブスクリプションを一覧表示するには、--all オプションを追加します。プール ID は、Pool ID で始まる行に一覧表示されます。

3. 以下のコマンドを実行して、該当するサブスクリプションをシステムに割り当てます。

```
subscription-manager attach --pool=pool_id
```

pool_id を、直前のステップで確認したプール ID に置き換えます。

システムに割り当てているサブスクリプションの一覧を随時確認するには、以下を実行します。

```
subscription-manager list --consumed
```

注記

ファイアウォールまたはプロキシーを使用する場合は、yum および subscription-manager が正常に機能するように追加の設定が必要になる場合があります。ファイアウォールを使用し、プロキシーを使用する場合は、『Red Hat Enterprise Linux 6 サブスクリプション管理の管理』の 「コンテンツ配信ネットワーク(Firewall Access for Content Delivery)の設定 」 セクションを参照してください。

Red Hat Subscription Management を使用してシステムを登録し、サブスクリプションに関連付ける方法は、指定された ソリューションの記事 を参照してください。サブスクリプションに関する包括的な情報は、Red Hat Subscription Management のガイドを参照してください。

6.2. ソフトウェアリポジトリーの管理

Red Hat コンテンツ配信ネットワークにシステムをサブスクライブすると、/etc/yum.repos.d/ ディレクトリーにリポジトリーファイルが作成されます。これを確認するには、yum を使用して有効なリポジトリーの一覧を表示します。

```
yum repolist
```

Red Hat Subscription Management を使用すると、Red Hat が提供するソフトウェアリポジトリーを手動で有効または無効にすることもできます。利用可能なリポジトリーの一覧を表示するには、以下のコマンドを実行します。

```
subscription-manager repos --list
```

リポジトリー名は、使用している Red Hat Enterprise Linux のバージョンによって異なり、以下の

フォーマットに基づいています。

```
rhel-variant-rhscl-version-rpms
rhel-variant-rhscl-version-debug-rpms
rhel-variant-rhscl-version-source-rpms
```

variant は Red Hat Enterprise Linux システムのバリアント（サーバー または ワークステーション）で、*version* は Red Hat Enterprise Linux システムのバージョン（6 または 7）を示します。以下に例を示します。

```
rhel-server-rhscl-6-eus-rpms
rhel-server-rhscl-6-eus-source-rpms
rhel-server-rhscl-6-eus-debug-rpms
```

リポジトリーを有効にするには、以下のコマンドを入力します。

```
subscription-manager repos --enable repository
```

repository を、有効にするリポジトリーの名前に置き換えます。

同様に、リポジトリーを無効にするには以下のコマンドを使用します。

```
subscription-manager repos --disable repository
```

「Yum と Yum リポジトリーの設定」 では、yum を使用したソフトウェアリポジトリー管理の詳細情報を説明します。

6.3. サブスクリプションの削除

特定のサブスクリプションを削除するには、以下の手順を行います。

1. すでに割り当てられているサブスクリプションの情報を一覧表示し、削除する必要があるサブスクリプションのシリアル番号を確認します。

   ```
   subscription-manager list --consumed
   ```

   シリアル番号は、serial に記載されている番号です。たとえば、以下の例では 744993814251016831 になります。

```
SKU:              ES0113909
Contract:         01234567
Account:          1234567
Serial:           744993814251016831
Pool ID:          8a85f9894bba16dc014bccdd905a5e23
Active:           False
Quantity Used:    1
Service Level:    SELF-SUPPORT
Service Type:     L1-L3
Status Details:
Subscription Type: Standard
Starts:           02/27/2015
Ends:             02/27/2016
System Type:      Virtual
```

2. 以下のコマンドを実行して、選択したサブスクリプションを削除します。

```
subscription-manager remove --serial=serial_number
```

serial_number を、直前の手順で確認したシリアル番号に置き換えます。

システムに割り当てられているすべてのサブスクリプションを削除するには、以下のコマンドを実行します。

```
subscription-manager remove --all
```

6.4. その他のリソース

Red Hat Subscription Management を使用してシステムを登録し、サブスクリプションに関連付ける方法は、以下の資料を参照してください。

インストールされているドキュメント

- subscription-manager(8): Red Hat Subscription Management の man ページでは、サポートされるオプションおよびコマンドの完全なリストが提供されます。

関連書籍

- Red Hat Subscription Management の一連のガイド： これらのガイドには、Red Hat Subscription Management の使用方法の詳細情報が記載されています。

- インストールガイド: 初回起動プロセス中に登録する方法の詳細については、「初回起動時」の章を参照してください。https://access.redhat.com/documentation/ja-JP/Red_Hat_Enterprise_Linux/6/html/Installation_Guide/ch-firstboot.html

オンラインリソース

- Red Hat Access Labs: Red Hat Access Labs には 「Registration Assistant」 が含まれています。

その他の参考資料

- 4章*権限の取得* では、su および sudo コマンドを使用して管理者権限を取得する方法を説明しています。

- 8章*Yum* では、yum パッケージマネージャーを使用したソフトウェアのインストールおよび更新に関する情報を提供します。

- 9章*PackageKit* PackageKit パッケージマネージャーを使用してソフトウェアをインストールおよび更新する方法を提供します。

第7章 RED HAT SUPPORT TOOL を使用したサポートへのアクセス

redhat-support-tool パッケージの Red Hat Support Tool は、インタラクティブシェルおよび単一実行プログラムの両方として機能します。SSH または任意のターミナルで実行できます。また、コマンドラインから Red Hat ナレッジベースを検索したり、コマンドラインでソリューションを直接コピーしたり、サポートケースを作成または更新したり、分析のために Red Hat にファイルを送信したりできます。

7.1. RED HAT SUPPORT TOOL のインストール

Red Hat Support Tool は、Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux にデフォルトでインストールされます。必要な場合は、確実にインストールするために root で以下のコマンドを入力します。

```
~]# yum install redhat-support-tool
```

7.2. コマンドラインを使用した RED HAT SUPPORT TOOL の登録

コマンドラインを使用して Red Hat Support Tool をカスタマーポータルに登録するには、以下の手順を実行します。

1.

```
~]# redhat-support-tool config user username
```

username は、Red Hat カスタマーポータルアカウントのユーザー名に置き換えます。

2.

```
~]# redhat-support-tool config password
Please enter the password for username:
```

7.3. インタラクティブシェルモードでの RED HAT SUPPORT TOOL の使用

インタラクティブモードでツールを起動するには、以下のコマンドを入力します。

```
~]$ redhat-support-tool
Welcome to the Red Hat Support Tool.
Command (? for help):
```

ツールは、非特権ユーザーまたは root として実行できます。

? 文字を入力するとコマンドの一覧を表示できます。プログラムまたはメニューの選択は、q または e の文字を入力して終了できます。ナレッジベースまたはサポートケースを初めて検索する場合は、Red Hat カスタマーポータルのユーザー名とパスワードを入力するよう求められます。また、インタラクティブモードで Red Hat カスタマーポータルアカウントのユーザー名とパスワードを設定し、オプションで設定ファイルに保存することもできます。

7.4. RED HAT SUPPORT TOOL の設定

インタラクティブモードの場合は、コマンド config --help を入力して設定オプションの一覧を表示できます。

```
~]# redhat-support-tool
Welcome to the Red Hat Support Tool.
Command (? for help): config --help

Usage: config [options] config.option <new option value>

Use the 'config' command to set or get configuration file values.
Options:
  -h, --help    show this help message and exit
  -g, --global  Save configuration option in /etc/redhat-support-tool.conf.
  -u, --unset   Unset configuration option.

The configuration file options which can be set are:
 user      : The Red Hat Customer Portal user.
 password  : The Red Hat Customer Portal password.
 debug     : CRITICAL, ERROR, WARNING, INFO, or DEBUG
 url       : The support services URL.  Default=https://api.access.redhat.com
 proxy_url : A proxy server URL.
 proxy_user: A proxy server user.
 proxy_password: A password for the proxy server user.
 ssl_ca    : Path to certificate authorities to trust during communication.
 kern_debug_dir: Path to the directory where kernel debug symbols should be downloaded
and cached. Default=/var/lib/redhat-support-tool/debugkernels

Examples:
- config user
- config user my-rhn-username
- config --unset user
```

手順7.1. インタラクティブモードでの Red Hat Support Tool の登録

手順7.1 インタラクティブモードでの Red Hat Support Tool の登録

インタラクティブモードを使用して Red Hat Support Tool をカスタマーポータルに登録するには、以下のコマンドを実行します。

1. 以下のコマンドを入力してツールを起動します。

```
~]# redhat-support-tool
```

2. Red Hat カスタマーポータルのユーザー名を入力します。

```
Command (? for help): config user username
```

ユーザー名をグローバル設定ファイルに保存するには、-g オプションを追加します。

3. Red Hat カスタマーポータルのパスワードを入力します。

```
Command (? for help): config password
Please enter the password for username:
```

7.4.1. 設定ファイルへの設定の保存

Red Hat Support Tool は、（他の方法が設定されていない場合に） ~/.redhat-support-tool/redhat-support-tool.conf 設定ファイルを使用して現在のユーザーのホームディレクトリーに値とオプションをローカルで保存します。必要に応じて、パスワードをこのファイルに保存することが推奨されます。ツールが起動すると、グローバル設定ファイル /etc/redhat-support-tool.conf とローカル設定ファイルから値が読み取られます。ローカルに保存された値とオプションは、グローバルに保存された設定よりも優先されます。

警告

パスワードは base64 でエンコードされ簡単にデコードできるため、グローバルな /etc/redhat-support-tool.conf 設定ファイルにパスワードを保存することは推奨され ませ ん。また、ファイルは誰でも読み取り可能です。

値またはオプションをグローバル設定ファイルに保存するには、以下のように -g, --global オプションを追加します。

Command (? for help): config *setting* -g *value*

注記

-g, --global オプションを使用して設定をグローバルで保存できるようにするには、Red Hat Support Tool を root で実行する必要があります。これは、通常のユーザーには /etc/redhat-support-tool.conf への書き込みに必要なパーミッションがないためです。

値またはオプションをローカル設定ファイルから削除するには、以下のように -u, --unset オプションを追加します。

Command (? for help): config *setting* -u *value*

これにより、ツールからパラメーターが削除および設定解除され、(利用可能な場合は) グローバル設定ファイルにある同等の設定が使用されます。

注記

非特権ユーザーとして実行している場合は、 -u, --unset オプションでグローバル設定ファイルに保存される値は削除できませんが、-g, --global オプションと -u, --unset オプションを同時に使用して、ツールの現在の実行中のインスタンスから、設定を解除することができます。root として実行している場合は、-g, --global と -u, --unset オプションを同時に使用してグローバル設定ファイルから値とオプションを削除できます。

7.5. インタラクティブモードでのサポートケースの作成および更新

手順7.2 インタラクティブモードでの新しいサポートケースの作成

インタラクティブモードで新しいサポートケースを作成するには、以下の手順を実行します。

1. **以下のコマンドを入力してツールを起動します。**

```
~]# redhat-support-tool
```

2. **opencase コマンドを入力します。**

```
Command (? for help): opencase
```

3. **画面に表示されたプロンプトに従って製品とバージョンを選択します。**

4. **ケースの要約を入力します。**

5. **ケースの説明を入力し、完了したら空の行で Ctrl+D を押します。**

6. **ケースの重大度を選択します。**

7. **オプションで、サポートケースを作成する前に、この問題のソリューションが存在するかどうかを確認することを選択します。**

8. **サポートケースを作成することを確定します。**

```
Support case 0123456789 has successfully been opened
```

9. オプションで、SOS レポートを添付することを選択します。

10. オプションで、ファイルを添付することを選択します。

手順7.3 インタラクティブモードでの既存のサポートケースの表示および更新

インタラクティブモードで既存のサポートケースを表示および更新するには、以下の手順を実行します。

1. 以下のコマンドを入力してツールを起動します。

```
~]# redhat-support-tool
```

2. getcase コマンドを入力します。

```
Command (? for help): getcase case-number
```

case-number は、表示および更新するケースの番号です。

3. 画面に表示されたプロンプトに従ってケースを表示し、コメントを変更または追加して、添付ファイルを取得または追加します。

手順7.4 インタラクティブモードでの既存のサポートケースの変更

インタラクティブモードで既存のサポートケースの属性を変更するには、以下の手順を実行します。

1. 以下のコマンドを入力してツールを起動します。

```
~]# redhat-support-tool
```

2.
modifycase コマンドを入力します。

```
Command (? for help): modifycase case-number
```

case-number は、表示および更新するケースの番号です。

3.
変更の選択リストが表示されます。

```
Type the number of the attribute to modify or 'e' to return to the previous menu.
 1 Modify Type
 2 Modify Severity
 3 Modify Status
 4 Modify Alternative-ID
 5 Modify Product
 6 Modify Version
End of options.
```

画面に表示されたプロンプトに従って 1 つまたは複数のオプションを変更します。

4.
たとえば、ステータスを変更する場合は、3 と入力します。

```
Selection: 3
 1   Waiting on Customer
 2   Waiting on Red Hat
 3   Closed
Please select a status (or 'q' to exit):
```

7.6. コマンドラインでのサポートケースの表示

コマンドラインでケースの内容を表示すると、コマンドラインからソリューションを素早く簡単に適用できます。

コマンドラインで既存のサポートケースを表示するには、以下のようにコマンドを入力します。

```
~]# redhat-support-tool getcase case-number
```

case-number は、ダウンロードするケースの番号です。

7.7. その他のリソース

Red Hat ナレッジベースの記事 『Red Hat Support Tool』 には、追加情報、例、および動画チュートリアルが含まれます。

パート III. ソフトウェアのインストールおよび管理

Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux システムのソフトウェアはすべて RPM パッケージに分割され、インストール、アップグレード、または削除が可能です。ここでは、製品のサブスクリプションとエンタイトルメントに重点を置いており、Yum と、グラフィカルパッケージ管理ツールの PackageKit スイートを使用して、Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux でパッケージを管理する方法を説明します。

第8章 YUM

yum は Red Hatnbsp;Hat パッケージマネージャーで、利用可能なパッケージに関する情報のクエリー、リポジトリーからのパッケージのフェッチ、パッケージのインストールおよびアンインストール、さらに利用可能な最新バージョンへのシステム全体の更新が可能です。Yum は、更新/インストール/削除を実行しているパッケージで依存関係の自動解決を行います。そのため、利用可能なすべての依存パッケージを自動的に決定/フェッチ/インストールすることができます。

yum は、新しいリポジトリー、追加リポジトリー、または *パッケージソース* で設定でき、その機能を強化し、拡張するプラグインを多数提供します。yum は、RPM が実行できる同じタスクを多数実行できます。さらに、多くのコマンドラインオプションも似ています。Yum を使用することで、1 つのマシンまたはマシンのグループ上でのパッケージ管理を簡単かつシンプルに行うことができます。

以下のセクションでは、『Red Hat Enterprise Linux 6 インストールガイド』で説明されているように、インストール時にシステムが Red Hat Subscription Management に登録されていることを前提としています。システムがサブスクライブされていない場合は、*6章システム登録およびサブスクリプション管理* を参照してください。

GPG 署名パッケージによるセキュアなパッケージ管理

yum は、GPG（Gnu Privacy Guard（別名 GnuPG））の署名検証をすべてのパッケージリポジトリー（パッケージソースなど）または個々のリポジトリーで有効にすることで、セキュアなパッケージ管理を提供します。署名の検証を有効にすると、Yum はリポジトリーの正しいキーで GPG 署名されていないパッケージのインストールを拒否します。つまり、お使いのシステムにダウンロードしてインストールする RPM パッケージが Red Hatnbsp;Hat などの信頼できるソースからいることを信頼でき、転送中に変更されなかったことを意味します。Yum での署名確認を有効にする方法は 「Yum と Yum リポジトリーの設定」 または 「パッケージの署名の確認」 で一般的な GPG 署名 RPM パッケージを使用および検証する方法についての詳細は、を参照してください。

Yum を使用すると、他のマシンへダウンロードし、インストールするための RPM パッケージのリポジトリーを簡単に設定することができます。

Yum の学習は、システム管理タスクを実行する最も高速な方法であり、PackageKit グラフィカルパッケージ管理ツールが提供する以上の機能を提供します。PackageKit の使用方法は、*9章PackageKit* を参照してください。

YUM および SUPERUSER の特権

yum を使用して、システムにパッケージをインストール、更新、または削除するにはスーパーユーザー権限が必要です。本章のすべての例では、su または sudo コマンドを使用することでスーパーユーザー権限をすでに持っていると仮定しています。

8.1. パッケージの確認と更新

8.1.1. 更新の確認

使用しているシステムに利用可能な更新があるインストール済みのパッケージを確認するには、以下のコマンドを実行します。

```
yum check-update
```

以下に例を示します。

```
~]# yum check-update
Loaded plugins: product-id, refresh-packagekit, subscription-manager
Updating Red Hat repositories.
INFO:rhsm-app.repolib:repos updated: 0
PackageKit.x86_64                0.5.8-2.el6              rhel
PackageKit-glib.x86_64           0.5.8-2.el6              rhel
PackageKit-yum.x86_64            0.5.8-2.el6              rhel
PackageKit-yum-plugin.x86_64     0.5.8-2.el6              rhel
glibc.x86_64                     2.11.90-20.el6           rhel
glibc-common.x86_64              2.10.90-22               rhel
kernel.x86_64                    2.6.31-14.el6            rhel
kernel-firmware.noarch           2.6.31-14.el6            rhel
rpm.x86_64                       4.7.1-5.el6              rhel
rpm-libs.x86_64                  4.7.1-5.el6              rhel
rpm-python.x86_64                4.7.1-5.el6              rhel
udev.x86_64                      147-2.15.el6             rhel
yum.noarch                       3.2.24-4.el6             rhel
```

上記の出力に表示されているパッケージには利用可能な更新があります。一覧の最初のパッケージは、グラフィカルパッケージマネージャーである PackageKit です。出力例の行は、以下のようになります。

- **PackageKit - パッケージの名前**
- **x86_64 - パッケージがビルドされた CPU アーキテクチャー**

- 0.5.8 - インストールする更新パッケージのバージョン

- rhel - 更新パッケージが置かれているリポジトリー

この出力では、yum を使用してカーネル（ kernel パッケージ）、Yum および RPM（ yum パッケージ）、およびその依存関係（ rpm、kernel-firmware、および rpm-libs パッケージ）をすべて更新できることも示しています。rpm-python

8.1.2. パッケージの更新

一度に更新するパッケージ数を 1 つ、複数、または全てのパッケージから選択できます。更新するパッケージの依存関係（またはパッケージ）に利用可能な更新がある場合は、更新も更新されます。

単一パッケージの更新

1 つのパッケージを更新するには、root で以下のコマンドを実行します。

```
yum update package_name
```

たとえば、udev パッケージを更新するには、以下を入力します。

```
~]# yum update udev
Loaded plugins: product-id, refresh-packagekit, subscription-manager
Updating Red Hat repositories.
INFO:rhsm-app.repolib:repos updated: 0
Setting up Update Process
Resolving Dependencies
--> Running transaction check
---> Package udev.x86_64 0:147-2.15.el6 set to be updated
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
 Package       Arch          Version              Repository     Size
================================================================================
Updating:
 udev          x86_64        147-2.15.el6         rhel          337 k

Transaction Summary
================================================================================
Install       0 Package(s)
Upgrade       1 Package(s)
```

```
Total download size: 337 k
Is this ok [y/N]:
```

この出力で重要となる項目がいくつかあります。

1. Loaded plugins: product-id, refresh-packagekit, subscription-manager - yum は常に、どの Yum プラグインがインストールされ、有効になっているかを通知します。Yum プラグインに関する一般的な情報は、「yum のプラグイン」 を参照してください。特定のプラグインの説明は、「プラグインの説明」 を参照してください。

2. udev.x86_64 - 新しい udev パッケージをダウンロードしてインストールできます。

3. yum は更新情報を表示し、更新を実行するかどうかを尋ねられます。yum はデフォルトで対話的に実行されます。yum コマンドが実行する予定のトランザクションがすでに分かっている場合は、-y オプションを使用して、yum が質問する質問（この場合は非対話的に実行）に対して、自動的に yes と回答できます。ただし、発生する可能性のある問題を簡単にトラブルシューティングできるように、yum がシステムに加える変更を必ず確認する必要があります。

   トランザクションが到着する場合は、「トランザクション履歴の活用」 で説明されているように yum history コマンドを使用して Yum のトランザクション履歴を表示できます。

YUM を使用したカーネルの更新とインストール

yum は、rpm -i kernel コマンドを使用する際に RPM が新しいカーネルをインストールするのと同様に、新しいカーネルを常にインストールします。*したがって、yum を使用する際に、カーネルパッケージの インストール と アップグレード の違いを考慮する必要はありません。 yum update コマンドまたは yum install コマンドを使用しているかどうかに関わらず、適切な操作を行います。*

一方、RPM を使用する場合は、rpm -u kernel （現在のカーネルを 置き換える ）の代わりに rpm -i kernel コマンド（新しいカーネルをインストール）を使用することが重要です。RPM を使用したカーネルのインストール/アップグレードの詳細は、「インストールおよび設定ガイド」 を参照してください。

すべてのパッケージとそれらの依存関係の更新

パッケージとその依存関係をすべて更新するには、引数なしで yum update を入力します。

```
yum update
```

セキュリティー関連パッケージの更新

どのパッケージでセキュリティー更新が利用可能かを検出し、それらのパッケージを迅速かつ簡単に更新することが重要になります。yum は、この目的のためにプラグインを提供します。security プラグインは、yum コマンドを、高度なセキュリティー中心のコマンド、サブコマンド、およびオプションのセットで拡張します。具体的な情報については、「プラグインの説明」 を参照してください。

パッケージの自動更新

パッケージに定期的な自動更新を設定することもできます。このため、Red Hat Enterprise Linux 6 では yum-cron パッケージを使用します。これは cron デーモンの Yum インターフェースを提供し、パッケージリポジトリーからメタデータをダウンロードします。yum-cron サービスを有効にすると、ユーザーは毎日自動的に Yum の更新を cron ジョブとしてスケジュールできます。

注記

yum-cron パッケージは、Optional サブスクリプションチャンネルで提供されます。Red Hat 追加チャンネルの詳細は、「Optional および Supplementary リポジトリーの追加」 を参照してください。

yum-cron をインストールするには、以下のコマンドを実行します。

```
~]# yum install yum-cron
```

デフォルトでは、yum-cron サービスは無効になっており、手動でアクティブ化して起動する必要があります。

```
~]# chkconfig yum-cron on
```

```
~]# service yum-cron start
```

サービスのステータスを確認するには、以下のコマンドを実行します。

```
~]# service yum-cron status
```

yum-cron パッケージに含まれるスクリプトは、更新のエクステントと頻度を変更し、通知をメールに送信するように設定できます。yum-cron をカスタマイズするには、/etc/sysconfig/yum-cron ファイルを編集します。

yum-cron の詳細は、/etc/sysconfig/yum-cron 内のコメントと yum-cron(8)man ページを参照してください。

8.1.3. 設定ファイルの変更の保存

Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux システムの使用時に、パッケージによりインストールされた設定ファイルに変更を加えます。Yum がシステムの変更の実行に使用する RPM は、整合性を確保するメカニズムを提供します。パッケージアップグレード全体で設定ファイルへの変更を管理する方法は、「インストールおよび設定ガイド」 を参照してください。

8.1.4. ISO と Yum を使用してシステムをオフラインでアップグレード

インターネットまたは Red Hat Network から切断されたシステムの場合は、yum update コマンドと Red Hat Enterprise Linux インストール ISO イメージを使用すると、システムを最新のマイナーバージョンに簡単かつ迅速にアップグレードできます。以下の手順はアップグレードプロセスを示しています。

1. ISO イメージをマウントするターゲットディレクトリーを作成します。このディレクトリーは、マウント時に自動的に作成されません。root で以下のコマンドを実行します。

```
mkdir mount_dir
```

mount_dir をマウントディレクトリーへのパスに置き換えます。通常、ユーザーは /media ディレクトリーのサブディレクトリーとして作成します。

2. 以前に作成したターゲットディレクトリーに Red Hat Enterprise Linux 6 インストール ISO イメージをマウントします。root で以下のコマンドを実行します。

```
mount -o loop iso_name mount_dir
```

iso_name は ISO イメージへのパスに、mount_dir はターゲットディレクトリーへのパスに置き換えます。ブロックデバイスとしてファイルをマウントするには、-o loop オプションが必要です。

3. media.repo ファイルをマウントディレクトリーから /etc/yum.repos.d/ ディレクトリーにコピーします。正常に機能するために、このディレクトリーの設定ファイルの拡張子は .repo である必要があります。

```
cp mount_dir/media.repo /etc/yum.repos.d/new.repo
```

これにより、yum リポジトリーの設定ファイルが作成されます。new.repo をファイル名に置き換えます（例： rhel6.repo ）。

4.

Red Hat Enterprise Linux インストール ISO を参照するように新しい設定ファイルを編集します。/etc/yum.repos.d/new.repoファイルに以下の行を追加します。

```
baseurl=file:///mount_dir
```

mount_dir は、マウントポイントへのパスに置き換えます。

5.

前の手順で作成された /etc/yum.repos.d/new.repo を含む、yum リポジトリーをすべて更新します。root で以下のコマンドを実行します。

```
yum update
```

これにより、システムはマウントされた ISO イメージで提供されたバージョンにアップグレードされます。

6.

アップグレードに成功したら、ISO イメージをアンマウントできます。root で以下のコマンドを実行します。

```
umount mount_dir
```

mount_dir はマウントディレクトリーへのパスです。また、最初の手順で作成されたマウントディレクトリーを削除することもできます。root で以下のコマンドを実行します。

```
rmdir mount_dir
```

7.

以前に作成された設定ファイルを別のインストールまたは更新に使用しない場合は、その設定ファイルを削除できます。root で以下のコマンドを実行します。

```
rm /etc/yum.repos.d/new.repo
```

例8.1 Red Hat Enterprise Linux 6.3 から 6.4 へのアップグレード

インターネットにアクセスせずにシステムをアップグレードする必要があるとします。これを行うには、新しいバージョンのシステムとともに ISO イメージを使用します（例： RHEL6.4-Server-20130130.0-x86_64-DVD1.iso ）。マウント用に作成されたターゲットディレクトリーは

/media/rhel6/ です。root で ISO イメージがあるディレクトリーに移動し、以下のコマンドを入力します。

```
~]# mount -o loop RHEL6.4-Server-20130130.0-x86_64-DVD1.iso /media/rhel6/
```

次に、マウントディレクトリーから media.repo ファイルをコピーして、イメージ用の yum リポジトリーをセットアップします。

```
~]# cp /media/rhel6/media.repo /etc/yum.repos.d/rhel6.repo
```

yum にマウントポイントをリポジトリーとして認識させるには、前の手順でコピーした /etc/yum.repos.d/rhel6.repo に以下の行を追加します。

```
baseurl=file:///media/rhel6/
```

この時点で、yum リポジトリーを更新すると、RHEL6.4-Server-20130130.0-x86_64-DVD1.iso により提供されたバージョンにシステムがアップグレードされます。root で以下を実行します。

```
~]# yum update
```

システムが正常にアップグレードされたら、イメージをアンマウントし、ターゲットディレクトリーと設定ファイルを削除できます。

```
~]# umount /media/rhel6/
```

```
~]# rmdir /media/rhel6/
```

```
~]# rm /etc/yum.repos.d/rhel6.repo
```

8.2. パッケージおよびパッケージグループ

8.2.1. パッケージの検索

以下のコマンドを使用して、すべての RPM パッケージ名、説明、サマリーを検索できます。

```
yum search term…
```

term を、検索するパッケージ名に置き換えます。

例8.2 特定の文字列に一致するパッケージの検索

「vim」、「gvim」、または 「emacs」 に一致するパッケージの一覧を表示するには、以下を入力します。

```
~]$ yum search vim gvim emacs
Loaded plugins: langpacks, product-id, search-disabled-repos, subscription-manager
============================== N/S matched: vim ==============================
vim-X11.x86_64 : The VIM version of the vi editor for the X Window System
vim-common.x86_64 : The common files needed by any version of the VIM editor
[output truncated]

============================== N/S matched: emacs ==============================
emacs.x86_64 : GNU Emacs text editor
emacs-auctex.noarch : Enhanced TeX modes for Emacs
[output truncated]

  Name and summary matches mostly, use "search all" for everything.
Warning: No matches found for: gvim
```

yum search コマンドは、パッケージ名は分からないものの、関連用語を知っている場合にパッケージを検索する際に役立ちます。デフォルトでは、yum search はパッケージ名とサマリーが一致したものを返すため、検索には時間がかかりません。yum search all コマンドを使用して、より詳細な検索を行いますが、検索は遅くなります。

8.2.2. パッケージの一覧表示

yum list および関連コマンドは、パッケージ、パッケージグループ、リポジトリーに関する情報を提供します。

*Yum の list コマンドはすべて 1 つ以上の glob 表現 を引数として追加することで、結果をフィルタリングできます。glob 表現は、1 つ以上のワイルドカード文字 * （任意の文字を複数回一致するように拡張）と ？（任意の文字に拡張）を含む通常の文字列です。*

GLOB 表現を使用した結果のフィルタリング

yum コマンドに glob 表現を引数として渡す場合には、glob 表現をエスケープするように注意してください。エスケープしないと、bash シェルはこの表現を パス名の展開 と解釈し、glob に一致する現在のディレクトリー内の全ファイルを yum に渡す可能性があります。確実に glob 表現を yum に渡すには、以下のいずれかを行います。

- ワイルドカード文字の前にバックスラッシュ記号を入力して、ワイルドカード文字をエスケープする
- glob 表現全体を二重引用符または単一引用符でくくる

これらの両方の方法の使用例は、例8.3「glob 表現を使用した ABRT アドオンおよびプラグインの一覧表示」 および 例8.5「エスケープされたワイルドカード文字を含む 1 つの glob 式を使用した利用可能なパッケージの一覧表示」 を参照してください。

yum list glob_expression

すべての glob 表現に一致するインストール済み および 利用可能なパッケージに関する情報を一覧表示します。

例8.3 glob 表現を使用した ABRT アドオンおよびプラグインの一覧表示

さまざまな ABRT アドオンとプラグインを含むパッケージは 「abrt-addon-」 または 「abrt-plugin-」 で始まります。これらのパッケージを一覧表示するには、シェルプロンプトで以下を入力します。

```
~]# yum list abrt-addon\* abrt-plugin\*
Loaded plugins: product-id, refresh-packagekit, subscription-manager
Updating Red Hat repositories.
INFO:rhsm-app.repolib:repos updated: 0
Installed Packages
abrt-addon-ccpp.x86_64                1.0.7-5.el6          @rhel
abrt-addon-kerneloops.x86_64          1.0.7-5.el6          @rhel
abrt-addon-python.x86_64              1.0.7-5.el6          @rhel
abrt-plugin-bugzilla.x86_64           1.0.7-5.el6          @rhel
abrt-plugin-logger.x86_64             1.0.7-5.el6          @rhel
abrt-plugin-sosreport.x86_64          1.0.7-5.el6          @rhel
abrt-plugin-ticketuploader.x86_64     1.0.7-5.el6          @rhel
```

yum list all

インストール済み および 利用可能なパッケージの一覧を表示します。

yum list installed

システムにインストールされているパッケージの一覧を表示します。出力の右端の列には、パッケージを取得したリポジトリーが表示されます。

例8.4 二重引用符表現を使用したインストール済みパッケージの一覧表示

「krb」 で始まり、1 文字とハイフンが続くインストール済みパッケージを一覧表示するには、以下を入力します。

```
~]# yum list installed "krb?-*"
Loaded plugins: product-id, refresh-packagekit, subscription-manager
Updating Red Hat repositories.
INFO:rhsm-app.repolib:repos updated: 0
Installed Packages
krb5-libs.x86_64                 1.8.1-3.el6               @rhel
krb5-workstation.x86_64          1.8.1-3.el6               @rhel
```

yum list available

有効なすべてのリポジトリーで利用可能なパッケージの一覧を表示します。

例8.5 エスケープされたワイルドカード文字を含む 1 つの glob 式を使用した利用可能なパッケージの一覧表示

「gstreamer」 と 「プラグイン」 を含む名前で利用可能なパッケージを一覧表示するには、以下のコマンドを実行します。

```
~]# yum list available gstreamer\*plugin\*
Loaded plugins: product-id, refresh-packagekit, subscription-manager
Updating Red Hat repositories.
INFO:rhsm-app.repolib:repos updated: 0
Available Packages
gstreamer-plugins-bad-free.i686       0.10.17-4.el6          rhel
gstreamer-plugins-base.i686           0.10.26-1.el6          rhel
gstreamer-plugins-base-devel.i686     0.10.26-1.el6          rhel
gstreamer-plugins-base-devel.x86_64   0.10.26-1.el6          rhel
gstreamer-plugins-good.i686           0.10.18-1.el6          rhel
```

yum grouplist

パッケージグループの一覧を表示します。

yum repolist

有効 なリポジトリーごとにリポジトリー ID、名前、提供するパッケージ数を一覧表示します。

8.2.3. パッケージ情報の表示

1 つ以上のパッケージに関する情報（ここでは glob 表現も有効）を表示するには、以下のコマンドを使用します。

```
yum info package_name
```

たとえば、abrt パッケージに関する情報を表示するには、以下を入力します。

```
~]# yum info abrt
Loaded plugins: product-id, refresh-packagekit, subscription-manager
Updating Red Hat repositories.
INFO:rhsm-app.repolib:repos updated: 0
Installed Packages
Name       : abrt
Arch       : x86_64
Version    : 1.0.7
Release    : 5.el6
Size       : 578 k
Repo       : installed
From repo  : rhel
Summary    : Automatic bug detection and reporting tool
URL        : https://fedorahosted.org/abrt/
License    : GPLv2+
Description: abrt is a tool to help users to detect defects in applications
           : and to create a bug report with all informations needed by
           : maintainer to fix it. It uses plugin system to extend its
           : functionality.
```

yum info package_name コマンドは rpm -q --info package_name コマンドに似ていますが、追加情報として、RPM パッケージが見つかる Yum リポジトリーの ID です（出力の From repo: 行を参照）。

Yum データベースに対して、次のコマンドを使用してパッケージに関する代替情報や有用な情報をクエリーすることもできます。

```
yumdb info package_name
```

このコマンドは、パッケージのチェックサム（および SHA-256 などのチェックサムを算出するために使用されるアルゴリズム）、パッケージのインストール開始に使用されたコマンドラインのコマンド（存在する場合）、パッケージがシステムにインストールされた理由（ user はユーザーがインストールしたことを、dep は依存関係として取り入れたことを意味します）などのパッケージに関する追加情報を提供します。たとえば、yum パッケージに関する追加情報を表示するには、以下を入力します。

```
~]# yumdb info yum
Loaded plugins: product-id, refresh-packagekit, subscription-manager
yum-3.2.27-4.el6.noarch
    checksum_data = 23d337ed51a9757bbfbdceb82c4eaca9808ff1009b51e9626d540f44fe95f771
    checksum_type = sha256
    from_repo = rhel
    from_repo_revision = 1298613159
    from_repo_timestamp = 1298614288
    installed_by = 4294967295
    reason = user
    releasever = 6.1
```

yumdb コマンドの詳細は、man ページの yumdb(8)を参照してください。

パッケージに含まれるファイルの一覧表示

repoquery は、rpm クエリーと同様に yum リポジトリーから情報をクエリーするプログラムです。パッケージグループと個々のパッケージの両方をクエリーできます。特定のパッケージに含まれるファイルの一覧を表示するには、以下を入力します。

```
repoquery --list package_name
```

package_name を、検査するパッケージ名に置き換えます。repoquery コマンドの詳細は、man ページの repoquery を参照してください。

特定のファイルを提供するパッケージを確認するには、で説明している yum provides コマンドを使用できます。 ファイルを所有するパッケージの検索

8.2.4. パッケージのインストール

yum を使用すると、1 つのパッケージと複数のパッケージ、および選択したパッケージグループの両方をインストールできます。

個別パッケージのインストール

1 つのパッケージと、そのパッケージの依存関係でインストールされていないものをすべてインストールするには、以下の形式のコマンドを入力します。

```
yum install package_name
```

複数のパッケージを同時にインストールするには、その名前を引数として追加します。

```
yum install package_name package_name
```

AMD64 マシンや Intel 64 マシンなどの multilib システムにパッケージをインストールする場合は、パッケージ名に .arch を追加して、（有効なリポジトリーで利用可能な限り）パッケージのアーキテクチャーを指定できます。たとえば、i686 の sqlite パッケージをインストールするには、以下を入力します。

```
~]# yum install sqlite.i686
```

glob 表現を使用すると、名前が似ている複数のパッケージを迅速にインストールできます。

```
~]# yum install perl-Crypt-\*
```

パッケージ名と glob 表現に加えて、yum install にはファイル名も追加できます。インストールするバイナリー名が分かっていて、パッケージ名が分からない場合は、yum install にパス名を付けて実行します。

```
~]# yum install /usr/sbin/named
```

yum はパッケージ一覧で検索を行い、/usr/sbin/named を提供するパッケージを探します。パッケージが存在すると、yum により、そのパッケージをインストールするかどうかを尋ねられます。

ファイルを所有するパッケージの検索

named バイナリーを含むパッケージをインストールする場合は、どの bin ディレクトリーまたは sbin ディレクトリーがインストールされているかわからない場合は、glob 表現を付けて yum provides コマンドを実行します。

```
~]# yum provides "*bin/named"
Loaded plugins: product-id, refresh-packagekit, subscription-manager
Updating Red Hat repositories.
INFO:rhsm-app.repolib:repos updated: 0
32:bind-9.7.0-4.P1.el6.x86_64 : The Berkeley Internet Name Domain (BIND)
                              : DNS (Domain Name System) server
Repo        : rhel
Matched from:
Filename    : /usr/sbin/named
```

yum は「/file_name"」を提供 します。これは、file_name を含むパッケージを検索するのに一般的かつ便利な方法です。*

パッケージグループのインストール

パッケージグループはパッケージと似ています。それ自体は役に立ちませんが、共通の目的に対応する依存パッケージのグループを 1 つプルします。パッケージグループには、名前と groupid があります。yum grouplist -v コマンドはすべてのパッケージグループの名前を表示し、各パッケージグループの横に groupid を括弧で表示します。groupid は、以下の例のように、括弧の最後のペアで使用される用語です（以下の例のように kde-desktop ）。

```
~]# yum -v grouplist kde\*
Loading "product-id" plugin
Loading "refresh-packagekit" plugin
Loading "subscription-manager" plugin
Updating Red Hat repositories.
INFO:rhsm-app.repolib:repos updated: 0
Config time: 0.123
Yum Version: 3.2.29
Setting up Group Process
Looking for repo options for [rhel]
rpmdb time: 0.001
group time: 1.291
Available Groups:
   KDE Desktop (kde-desktop)
Done
```

パッケージグループをインストールするには、（groupid 部分なしで）完全なグループ名を groupinstall に渡します。

```
yum groupinstall group_name
```

groupid でインストールすることもできます。

```
yum groupinstall groupid
```

groupid（または引用付き名前）に @記号を追加しても（ groupinstallと同じように yum に指示）、install コマンドに groupid（または引用付き名前）を渡すこともできます。

```
yum install @group
```

たとえば、以下は代替ですが、KATK Desktop グループをインストールする方法と同じです。

```
~]# yum groupinstall "KDE Desktop"
~]# yum groupinstall kde-desktop
~]# yum install @kde-desktop
```

8.2.5. パッケージの削除

パッケージのインストールと同様に、Yum では個別のパッケージとパッケージグループの両方をアンインストール（ RPM と Yum の用語で削除）できます。

個々のパッケージの削除

特定のパッケージと、そのパッケージの依存関係パッケージをすべてアンインストールするには、root で以下のコマンドを実行します。

```
yum remove package_name
```

複数のパッケージをインストールする場合と同様、コマンドに複数のパッケージ名を追加すると、一度に複数のパッケージを削除できます。たとえば、totem、rhythmbox、および sound-juicer を削除するには、シェルプロンプトで以下を入力します。

```
~]# yum remove totem rhythmbox sound-juicer
```

インストール と同様に、remove では、以下の引数を使用できます。

- パッケージ名

- *glob 表現*
- *ファイル一覧*
- *パッケージが提供する機能*

他のパッケージがそれに依存している場合にパッケージを削除

Yum では、パッケージを削除して、その依存パッケージを残すことはできません。このタイプの操作は RPM によってのみ実行でき、推奨されません。システムが機能しなくなるか、アプリケーションに誤作動したり、クラッシュしたりする可能性があります。詳細は、RPM の章の「アンインストール」を参照してください。

パッケージグループの削除

install 構文で構文 congruent を使用すると、パッケージグループを削除できます。

```
yum groupremove group
```

```
yum remove @group
```

以下は代替方法になりますが、KK Desktop グループを削除する方法と同等の方法になります。

```
~]# yum groupremove "KDE Desktop"
~]# yum groupremove kde-desktop
~]# yum remove @kde-desktop
```

インテリジェントなパッケージグループの削除

パッケージグループを削除するように yum に指示すると、そのパッケージが他のパッケージグループのメンバーである場合でも、そのグループ内の全パッケージが削除されます。ただし、group remove_leaf_only=1 ディレクティブを /etc/ yum.conf 設定ファイルの [main] セクションに追加すると、他のパッケージやグループで必要のないパッケージのみ を削除するように yum に指示することができます。このディレクティブの詳細は、「[main] オプションの設定」を参照してください。

8.3. トランザクション履歴の活用

yum history コマンドを使用すると、Yum トランザクションのタイムライン、トランザクションの発生日時、影響を受けたパッケージ数、トランザクション成功の有無、RPM データベースがトランザクション間で変更されたかどうかなどに関する情報を確認できます。さらに、このコマンドを使用すると、特定のトランザクションを元に戻す、またはやり直すことが可能です。

8.3.1. トランザクションの一覧表示

最近発生した 20 件のトランザクションを一覧表示するには、root で引数なしで **yum history** を実行するか、シェルプロンプトで以下を入力します。

```
yum history list
```

すべてのトランザクションを表示するには、**all** のキーワードを追加します。

```
yum history list all
```

特定の範囲内のトランザクションのみを表示したい場合は、以下の形式でコマンドを使用します。

```
yum history list start_id..end_id
```

特定のパッケージに関するトランザクションのみを一覧表示することもできます。そのためには、パッケージ名か **glob** 表現を付けてコマンドを実行します。

```
yum history list glob_expression
```

たとえば、最初の 5 つのトランザクションのリストは以下のようになります。

```
~]# yum history list 1..5
Loaded plugins: product-id, refresh-packagekit, subscription-manager
ID     | Login user               | Date and time    | Action(s)      | Altered
-------------------------------------------------------------------------------
     5 | Jaromir ... <jhradilek>  | 2011-07-29 15:33 | Install        |    1
     4 | Jaromir ... <jhradilek>  | 2011-07-21 15:10 | Install        |    1
     3 | Jaromir ... <jhradilek>  | 2011-07-16 15:27 | I, U           |   73
     2 | System <unset>           | 2011-07-16 15:19 | Update         |    1
     1 | System <unset>           | 2011-07-16 14:38 | Install        | 1106
history list
```

yum history list コマンドのすべての形式で、以下のコラムで構成される各行を含む表形式出力を生成します。

- *ID - 特定のトランザクションを識別する整数値です。*
- *Login user - トランザクションが開始したログインセッションのユーザー名。この情報は、通常 Full Name <username> 形式で表示されます。ユーザーが実行しなかったトランザクション（システムの自動更新など）については、代わりに System <unset> が使用されます。*
- *Date and time - トランザクションが発生した日時。*
- *Action(s) - 表8.1「Action フィールドの値」 の説明通りに、トランザクション中に実行されたアクションのリスト。*
- *Altered - トランザクションによって影響を受けるパッケージの数。場合によっては 表8.2「Altered フィールドの値」 で説明されているように、追加情報が続きます。*

表8.1 Action フィールドの値

Action	省略形	詳細
Downgrade	**D**	1つ以上のパッケージが旧バージョンにダウングレードされました。
Erase	**E**	1つ以上のパッケージが削除されました。
Install	**I**	1つ以上の新しいパッケージがインストールされました。
Obsoleting	**O**	1つ以上のパッケージが廃止として記録されました。
Reinstall	**-R**	1つ以上のパッケージが再インストールされました。
Update	**U**	1つ以上のパッケージが新しいバージョンに更新されました。

表8.2 Altered フィールドの値

記号	詳細
<	トランザクションが終了する前に、**rpmdb** データベースが Yum 以外で変更されました。

記号	詳細
>	トランザクションが終了した後に、**rpmdb** データベースが Yum 以外で変更されました。
*****	トランザクションは失敗して終了しました。
#	トランザクションは正常に終了しましたが、**yum** はゼロ以外の終了コードを返しました。
E	トランザクションは正常に終了しましたが、エラーまたは警告が表示されました。
%P	トランザクションは正常に終了しましたが、**rpmdb** データベースに問題がすでに存在します。
s	トランザクションは正常に終了しましたが、**--skip-broken** コマンドラインオプションが指定され、特定のパッケージが省略されました。

Yum を使用すると、過去に発生したすべてのトランザクションのサマリーを表示できます。これを行うには、root で以下の形式のコマンドを実行します。

```
yum history summary
```

特定の範囲内でのトランザクションのみを表示するには、以下を入力します。

```
yum history summary start_id..end_id
```

yum history list コマンドと同様に、パッケージ名または glob 表現を指定することで、特定のパッケージに関するトランザクションのサマリーを表示できます。

```
yum history summary glob_expression
```

たとえば、上記で表示されるトランザクション履歴の概要は以下のようになります。

```
~]# yum history summary 1..5
Loaded plugins: product-id, refresh-packagekit, subscription-manager
Login user               | Time               | Action(s)        | Altered
-------------------------------------------------------------------------------
Jaromir ... <jhradilek>  | Last day           | Install          |        1
Jaromir ... <jhradilek>  | Last week          | Install          |        1
Jaromir ... <jhradilek>  | Last 2 weeks       | I, U             |       73
System <unset>           | Last 2 weeks       | I, U             |     1107
history summary
```

yum history summary コマンドのすべての形式で、**yum history list** の出力に似た、簡略化された表形式出力を生成します。

上記のように、**yum history list** および **yum history summary** の両方がトランザクション向けに設計されていますが、特定のパッケージに関連するトランザクションのみを表示できますが、パッケージバージョンなどの重要な詳細は表示されません。パッケージに関連するトランザクションを一覧表示するには、**root** で以下のコマンドを実行します。

```
yum history package-list glob_expression
```

たとえば、**subscription-manager** および関連するパッケージの履歴を追跡するには、シェルプロンプトで以下を入力します。

```
~]# yum history package-list subscription-manager\*
Loaded plugins: product-id, refresh-packagekit, subscription-manager
ID     | Action(s)      | Package
-------------------------------------------------------------------------------
     3 | Updated        | subscription-manager-0.95.11-1.el6.x86_64
     3 | Update         |                  0.95.17-1.el6_1.x86_64
     3 | Updated        | subscription-manager-firstboot-0.95.11-1.el6.x86_64
     3 | Update         |                  0.95.17-1.el6_1.x86_64
     3 | Updated        | subscription-manager-gnome-0.95.11-1.el6.x86_64
     3 | Update         |                  0.95.17-1.el6_1.x86_64
     1 | Install        | subscription-manager-0.95.11-1.el6.x86_64
     1 | Install        | subscription-manager-firstboot-0.95.11-1.el6.x86_64
     1 | Install        | subscription-manager-gnome-0.95.11-1.el6.x86_64
history package-list
```

この例では、初期システムのインストール時に、**subscription-manager**、**subscription-manager-firstboot**、および **subscription-manager-gnome** の **3** つのパッケージがインストールされています。**3** つ目のトランザクションでは、これらのパッケージはすべてバージョン **0.95.11** からバージョン **0.95.17** に更新されました。

8.3.2. トランザクションの検証

単一のトランザクションのサマリーを表示するには、**root** で以下の形式で **yum history summary** コマンドを使用します。

```
yum history summary id
```

特定のトランザクションを詳しく調べる場合は、**root** で以下のコマンドを実行します。

yum history info id

id の引数は任意であり、省略すると、yum は自動的に最後のトランザクションを使用します。複数のトランザクションを指定する場合は、範囲を指定することもできます。

yum history info start_id..end_id

以下は、2 つのトランザクションに関する出力のサンプルです。それぞれ新しいパッケージを 1 つインストールしています。

```
~]# yum history info 4..5
Loaded plugins: product-id, refresh-packagekit, subscription-manager
Transaction ID : 4..5
Begin time     : Thu Jul 21 15:10:46 2011
Begin rpmdb    : 1107:0c67c32219c199f92ed8da7572b4c6df64eacd3a
End time       :          15:33:15 2011 (22 minutes)
End rpmdb      : 1109:1171025bd9b6b5f8db30d063598f590f1c1f3242
User           : Jaromir Hradilek <jhradilek>
Return-Code    : Success
Command Line   : install screen
Command Line   : install yum-plugin-security
Transaction performed with:
   Installed     rpm-4.8.0-16.el6.x86_64
   Installed     yum-3.2.29-17.el6.noarch
   Installed     yum-metadata-parser-1.1.2-16.el6.x86_64
Packages Altered:
   Install screen-4.0.3-16.el6.x86_64
   Install yum-plugin-security-1.1.30-17.el6.noarch
history info
```

また、トランザクション時に使用された設定オプション、特定のパッケージをインストールしたリポジトリー、その理由などの追加情報も閲覧できます。特定のトランザクションに関して入手可能な追加情報を表示する場合は、root で次のコマンドを実行します。

yum history addon-info id

yum history info と同様に、id を指定しないと、yum は自動的に最新のトランザクションを使用します。最新のトランザクションを確認する別の方法として、last キーワードを使用することもできます。

yum history addon-info last

たとえば、前述の例の最初のトランザクションでは、yum history addon-info コマンドは以下の出力を提供します。

```
~]# yum history addon-info 4
Loaded plugins: product-id, refresh-packagekit, subscription-manager
Transaction ID: 4
Available additional history information:
  config-main
  config-repos
  saved_tx

history addon-info
```

この例では、3 種類の情報が表示されます。

- config-main - トランザクション中に使用された yum のグローバルオプション。グローバルオプションの変更方法は 「[main] オプションの設定」 を参照してください。
- config-repos - 個々の Yum リポジトリー用のオプション。個々のリポジトリー用のオプションを変更する方法は 「[repository] オプションの設定」 を参照してください。
- saved_tx - 別のマシンでトランザクションを繰り返すために yum load-transaction コマンドで使用できるデータ（以下を参照）。

選択した種類の追加情報を表示するには、root で以下のコマンドを実行します。

```
yum history addon-info id information
```

8.3.3. トランザクションを元に戻す/繰り返す

トランザクション履歴の確認以外に、yum history コマンドは、選択したトランザクションを元に戻す、または繰り返す方法を提供します。トランザクションを元に戻すには、root で次のコマンドを実行します。

```
yum history undo id
```

特定のトランザクションを繰り返すには、root で次のコマンドを実行します。

```
yum history redo id
```

どちらのコマンドでも last キーワードを使用して、最新のトランザクションを元に戻す、または繰り返すことができます。

yum history undo コマンドおよび yum history redo コマンドはいずれも、トランザクション中に実行されたステップのみを元に戻すか、繰り返すことに注意してください。トランザクションで新しいパッケージがインストールされた場合に、yum history undo コマンドを実行すると、そのパッケージがアンインストールされ、トランザクションでパッケージがアンインストールされた場合は、このコマンドにより再度インストールされます。またこのコマンドは、(古いパッケージが引き続き利用可能な場合に) 更新済みパッケージをすべて以前のバージョンにダウングレードする試みも行います。

複数の同一システムを管理する場合、Yum はいずれかのシステムでトランザクションを実行し、トランザクションの詳細をファイルに格納し、テスト期間後に残りのシステムで同じトランザクションを繰り返すこともできます。トランザクションの詳細をファイルに保存するには、root で次のコマンドを実行します。

```
yum -q history addon-info id saved_tx > file_name
```

このファイルを目的のシステムにコピーしたら、root で以下のコマンドを使用してトランザクションを繰り返すことができます。

```
yum load-transaction file_name
```

ただし、ファイルに保存されている rpmdb バージョンは、ターゲットシステムのバージョンと同じである必要があります。yum version nogroups コマンドを使用すると、rpmdb バージョンを確認できます。

8.3.4. トランザクションの完了

電源損失やシステムクラッシュなどの予期しない状況では、yum トランザクションを完了できない場合があります。このようなイベントがトランザクションの途中で発生する場合は、root で以下のコマンドを使用して、後で再開できます。

```
yum-complete-transaction
```

yum-complete-transaction ツールは、システムで不完全な yum トランザクションまたは中断した yum トランザクションを検索し、それらを完了しようとします。デフォルトでは、これらのトランザクションは /var/lib/yum/transaction-all ファイルおよび /var/lib/yum/transaction-done ファイルに一覧表示されます。未完了のトランザクションが多い場合、yum-complete-transaction は最新のトランザクションを最初に完了しようとします。

中止したトランザクションを再開せずにトランザクションジャーナルファイルを消去するには、--cleanup-only オプションを使用します。

```
yum-complete-transaction --cleanup-only
```

8.3.5. 新しいトランザクション履歴の開始

Yum は単一の SQLite データベースファイルにトランザクション履歴を保存します。新しいトランザクションの履歴を開始するには、root で以下のコマンドを実行します。

```
yum history new
```

これにより、/var/lib/yum/history/ ディレクトリーに新しい空のデータベースファイルが作成されます。古いトランザクション履歴は保存されますが、新しいデータベースファイルがディレクトリーにある限りアクセスすることはできません。

8.4. YUM と YUM リポジトリーの設定

yum および関連ユーティリティー用の設定ファイルは /etc/yum.conf にあります。このファイルには、必須の [main] セクションが 1 つあり、ここで全体に影響を与える Yum オプションを設定できます。また、[repository] セクションを 1 つ以上追加して、リポジトリー固有のオプションを設定することもできます。ただし、/etc/yum.repos.d/ ディレクトリーにある、新規または既存の .repo ファイルに個々のリポジトリーを定義することが推奨されます。/etc/yum.conf ファイルの個別の [repository] セクションで定義した値は、[main] セクションに設定した値をオーバーライドします。

このセクションでは以下の方法を紹介します。

- /etc/yum.conf 設定ファイルの [main] セクションを編集して、yum のグローバルオプションを設定します。
- /etc/yum.conf ファイルおよび .repo ファイルの [repository] セクションを編集して、個々のリポジトリーにオプション を設定します。
- 動的バージョンとアーキテクチャーの値が正しく処理されるように、/etc/yum.conf 内の Yum 変数と /etc/yum.repos.d/ ディレクトリーのファイルを使用します。
- コマンドラインで Yum リポジトリーを追加、有効化、および無効化します。
- 独自のカスタム Yum リポジトリーを設定します。

8.4.1. [main] オプションの設定

/etc/yum.conf 設定ファイルには [main] セクションが 1 つだけ含まれます。本セクションにあるキー値ペアの中には、yum の動作に影響を与えるものもあれば、Yum がリポジトリーを処理する方法に影響を与えるものもあります。/etc/yum.conf の [main] セクションの下に、オプションを多数追加できます。

/etc/yum.conf 設定ファイルの例を以下に示します。

```
[main]
cachedir=/var/cache/yum/$basearch/$releasever
keepcache=0
debuglevel=2
logfile=/var/log/yum.log
exactarch=1
obsoletes=1
gpgcheck=1
plugins=1
installonly_limit=3

[comments abridged]

# PUT YOUR REPOS HERE OR IN separate files named file.repo
# in /etc/yum.repos.d
```

以下は、[main] セクションで最も一般的に使用されるオプションです。

assumeyes=value

ここで、value は以下のいずれかになります。

0: yum は、実行する重要な動作の確認を求めるプロンプトを出します。これはデフォルトです。

1: yum は、重要な動作の確認を行いません。assumeyes=1 を 設定すると、yum はコマンドラインオプション -y が行うのと同じ方法で動作します。

cachedir=directory

directory は、Yum がキャッシュとデータベースファイルを保存するディレクトリーへの絶対パスです。デフォルトでは、Yum のキャッシュディレクトリーは /var/cache/yum/$basearch/$releasever です。

「yum 変数の使用」 および $basearch Yum 変数の説明は、$releasever を参照してください。

debuglevel=value

value は、1 から 10 までの整数になります。debuglevel 値を高く設定すると、yum はより詳細なデバッグ出力を表示します。debuglevel=0 はデバッグ出力を無効にしますが、debuglevel=2 がデフォルトです。

exactarch=value

ここで、value は以下のいずれかになります。

0: パッケージの更新時には正しいアーキテクチャーを考慮に入れて実行しません。

1: パッケージの更新時に正しいアーキテクチャーを考慮します。この設定では、yum は、システムにインストールされている i386 パッケージを更新する i686 パッケージをインストールしません。これはデフォルトです。

exclude=package_name [more_package_names]

このオプションを使用すると、インストール/更新時にキーワードでパッケージを除外できます。除外する複数のパッケージの一覧を表示するには、スペースで区切ったパッケージの一覧を引用符で囲みます。ワイルドカードを使用したシェルグロブ（ や ?など）を使用できます。*

gpgcheck=value

ここで、value は以下のいずれかになります。

0: インストールされるローカルパッケージなど、全リポジトリー内のパッケージでの GPG 署名確認を無効にします。

1 - インストールされるローカルパッケージなど、全リポジトリー内の全パッケージで GPG 署名確認を有効にします。gpgcheck=1 がデフォルトであるため、すべてのパッケージ署名が確認されます。

このオプションが /etc/yum.conf ファイルの [main] セクションで設定されている場合は、全リポジトリーに対して GPG チェックルールが設定されます。ただし、個々のリポジトリーに gpgcheck=値 を設定することもできます。つまり、あるリポジトリーで GPG チェックを有効にしつつ、別のリポジトリーで無効にすることができます。対応する .repo ファイルの個々のリポジトリーに gpgcheck=値 を設定すると、/etc/yum.conf に存在する場合にはデフォルトが上書きされます。

GPG 署名の確認に関する詳細は、「パッケージの署名の確認」 を参照してください。

groupremove_leaf_only=value

ここで、value は以下のいずれかになります。

0: パッケージグループを削除するときに、yum は各パッケージの依存関係を確認し ない でください。この設定では、yum は、他のパッケージまたはグループで必要なパッケージが必要であるかどうかに関わらず、パッケージグループ内の全パッケージを削除します。groupremove_leaf_only=0 がデフォルトです。

1: yum はパッケージグループを削除するときに各パッケージの依存関係を確認し、他のパッケージやグループを必要としないパッケージのみを削除します。

パッケージの削除に関する詳細は、インテリジェントなパッケージグループの削除 を参照してください。

installonlypkgs=space separated list of packages

ここでは、yum がインストール できるものの、更新 を行わないパッケージの一覧をスペースで区切って提供できます。デフォルトでインストールのみに設定されているパッケージの一覧は、yum.conf(5)man ページを参照してください。

installonlypkgs ディレクティブを /etc/yum.conf に追加する場合は、yum.conf(5)の installonlypkgs セクションに一覧表示されているものを含め、インストールのみのパッケージを すべて 一覧表示する必要があります。特に、（デフォルトで）カーネルパッケージを常に installonlypkgs に一覧表示し、installonly_limit を常に 2 よりも大きな値に設定し、デフォルトが起動に失敗した場合にバックアップカーネルを常に利用可能にできるようにする必要があります。

installonly_limit=value

value は、installonlypkgs ディレクティブに記載されている単一パッケージに同時にインストールできるバージョンの最大数を示す整数です。

installonlypkgs ディレクティブのデフォルトには複数のカーネルパッケージが含まれるため、installonly_limit の値を変更すると、インストール済みの単一のカーネルパッケージのバージョンの最大数にも影響することに注意してください。/etc/yum.conf に一覧表示されるデフォルト値は installonly_limit=3 です。この値は特に 2 未満で減らすことは推奨されません。

keepcache=value

ここで、value は以下のいずれかになります。

0: インストールの成功後は、ヘッダーとパッケージのキャッシュを保持しません。これはデフォルトです。

1: インストールの成功後も、キャッシュを保持します。

logfile=file_name

file_name は、yum がログ出力を書き込むファイルへの絶対パスです。デフォルトでは、yum は /var/log/yum.log にログを記録します。

multilib_policy=value

ここで、value は以下のいずれかになります。

best: このシステムに最適なアーキテクチャーをインストールします。たとえば、AMD64 システムに multilib_policy=best を設定すると、yum は全パッケージの 64 ビットバージョンをインストールします。

all: 常に全パッケージ用の可能なあらゆるアーキテクチャーをインストールします。たとえば、AMD64 システムで multilib_policy を all に設定すると、yum は i686 および AMD64 のパッケージが利用可能であれば、両方をインストールします。

obsoletes=value

ここで、value は以下のいずれかになります。

0: 更新の実行時に yum の廃止処理ロジックを無効にします。

1: 更新の実行時に yum の廃止処理ロジックを有効にします。あるパッケージが仕様ファイルで別のパッケージを 廃止 することを宣言すると、廃止されたパッケージのインストール時に、廃止後のパッケージが former パッケージに置き換えられます。たとえば、パッケージ名が変更された場合などに廃止が宣言されます。デフォルトは obsoletes=1 です。

plugins=value

ここで、value は以下のいずれかになります。

0: すべての Yum プラグインをグローバルに無効にします。

プラグインをすべて無効にすることは推奨されません。

一部のプラグインは重要な Yum サービスを提供するため、すべてのプラグインを無効にすることは推奨されません。特に、rhnplugin は RHN Classic に対応しており、product-id プラグインおよび subscription-manager プラグインは、証明書ベースの コンテンツ配信ネットワーク (CDN)のサポートを提供します。グローバルにプラグインを無効にすることは便利なオプションとして提供されますが、通常は Yum の潜在的な問題を診断する場合にのみ推奨されます。

1: すべての Yum プラグインをグローバルに有効にします。plugins=1 で は、そのプラグインの設定ファイルに enabled=0 を設定して、特定の Yum プラグインを無効にすることができます。

Yum のさまざまなプラグインの詳細は、「yum のプラグイン」 を参照してください。プラグインの制御に関する詳細は 「yum プラグインを有効、設定、および無効にする方法」 を参照してください。

reposdir=directory

directory は、.repo ファイルがあるディレクトリーへの絶対パスです。すべての .repo ファイルにはリポジトリー情報が含まれています（ /etc/yum.confの [repository] セクションと同様）。yum は、/etc/yum.conf ファイルの .repo ファイルおよび [repository] セクションからすべてのリポジトリー情報を収集し、トランザクションに使用するリポジトリーのマスター一覧を作成

します。reposdir が設定されていない場合、yum はデフォルトのディレクトリー /etc/yum.repos.d/ を使用します。

retries=value

value は、整数 0 以上です。この値は、エラーを返す前に yum がファイルの取得を試行する回数を設定します。これを 0 に設定すると、yum はその試行を何度も続けます。デフォルト値は 10 です。

利用可能な [main] オプションの完全なリストは、yum.conf(5)man ページの [main] OPTIONS セクションを参照してください。

8.4.2. [repository] オプションの設定

[repository] セクションは、個別の Yum リポジトリーの定義を可能にする my_personal_repo（スペースは使用不可）などの一意のリポジトリー ID です。競合を回避するために、カスタムリポジトリーには、Red Hat リポジトリーで使用されている名前を使用しないでください。

以下は、[repository] セクションの形式が最小限の例です。

```
[repository]
name=repository_name
baseurl=repository_url
```

すべての [repository] セクションには、以下のディレクティブを含める必要があります。

name=repository_name

repository_name は、人間が判読できる文字列で、リポジトリーを記述します。

baseurl=repository_url

repository_url は、リポジトリーの repodata ディレクトリーが置かれているディレクトリーの URL に置き換えます。

- リポジトリーが HTTP にある場合は、http://path/to/repoを使用します。

- リポジトリーが FTP にある場合は、ftp://path/to/repoを使用します。

- リポジトリーがマシンのローカルにある場合は、file:///path/to/local/repoを使用します。

- 特定のオンラインリポジトリーでベーシック HTTP 認証が必要な場合は、username: password @link として URL の前に付けてユーザー名とパスワードを指定できます。たとえば、http://www.example.com/repo/ にあるリポジトリーでユーザー名とパスワードのパスワードが必要な場合、baseurl リンクは http://user:password@www.example.com/repo/ として指定できます。「」 「」

通常この URL は以下のような HTTP リンクになります。

```
baseurl=http://path/to/repo/releases/$releasever/server/$basearch/os/
```

Yum は常に URL の $releasever、$arch、および $basearch 変数を展開することに注意してください。Yum 変数の詳細は、「yum 変数の使用」 を参照してください。

もう 1 つの便利な [repository] ディレクティブを以下に示します。

enabled=value

ここで、value は以下のいずれかになります。

0: 更新およびインストールの実行時には、パッケージソースとしてこのリポジトリーを含めません。これはリポジトリーを迅速に有効または無効にする簡単な方法です。更新またはインストールには無効にしているリポジトリーから、単一パッケージが欲しい場合に便利です。

1: パッケージソースとしてこのリポジトリーを含めます。

リポジトリーのオンとオフは、--enablerepo=repo_nameオプションまたは --disablerepo=repo_name オプションを yum に渡すか、PackageKit ユーティリティーのソフトウェアの 追加/削除 ウィンドウから実行できます。

他にも多くの [repository] オプションが存在します。完全な一覧は、yum.conf(5)man ページの [repository] OPTIONS セクションを参照してください。

例8.6 /etc/yum.repos.d/redhat.repo ファイルのサンプル

以下は、/etc/yum.repos.d/redhat.repo ファイルの例です。

```
#
# Red Hat Repositories
# Managed by (rhsm) subscription-manager
#

[red-hat-enterprise-linux-scalable-file-system-for-rhel-6-entitlement-rpms]
name = Red Hat Enterprise Linux Scalable File System (for RHEL 6 Entitlement) (RPMs)
baseurl = https://cdn.redhat.com/content/dist/rhel/entitlement-
6/releases/$releasever/$basearch/scalablefilesystem/os
enabled = 1
gpgcheck = 1
gpgkey = file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release
sslverify = 1
sslcacert = /etc/rhsm/ca/redhat-uep.pem
sslclientkey = /etc/pki/entitlement/key.pem
sslclientcert = /etc/pki/entitlement/11300387955690106.pem

[red-hat-enterprise-linux-scalable-file-system-for-rhel-6-entitlement-source-rpms]
name = Red Hat Enterprise Linux Scalable File System (for RHEL 6 Entitlement) (Source
RPMs)
baseurl = https://cdn.redhat.com/content/dist/rhel/entitlement-
6/releases/$releasever/$basearch/scalablefilesystem/source/SRPMS
enabled = 0
gpgcheck = 1
gpgkey = file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release
sslverify = 1
sslcacert = /etc/rhsm/ca/redhat-uep.pem
sslclientkey = /etc/pki/entitlement/key.pem
sslclientcert = /etc/pki/entitlement/11300387955690106.pem

[red-hat-enterprise-linux-scalable-file-system-for-rhel-6-entitlement-debug-rpms]
name = Red Hat Enterprise Linux Scalable File System (for RHEL 6 Entitlement) (Debug
RPMs)
baseurl = https://cdn.redhat.com/content/dist/rhel/entitlement-
6/releases/$releasever/$basearch/scalablefilesystem/debug
enabled = 0
gpgcheck = 1
gpgkey = file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release
sslverify = 1
sslcacert = /etc/rhsm/ca/redhat-uep.pem
sslclientkey = /etc/pki/entitlement/key.pem
sslclientcert = /etc/pki/entitlement/11300387955690106.pem
```

8.4.3. yum 変数の使用

yum コマンドおよびすべての Yum 設定ファイル（ /etc/ yum.conf および /etc/yum.repos.d/ ディレクトリー内のすべての .repo ファイル）で、以下の組み込み変数を使用および参照できます。

$releasever

この変数を使用して、Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux のリリースバージョンを参照できます。yum は、/etc/yum.conf 設定ファイルの distroverpkg=値 行から $releasever の値を取得します。/etc/yum.conf にそのような行がない場合、yum は redhat-release-server パッケージからバージョン番号を取得することで正しい値を推測します。$releasever の値は、通常、メジャーリリース番号と Red Hat Enterprise Linuxnbsp;Hat Enterprise Red Hat Enterprise Linuxnbsp;Linux のバリアント（例： 6Client、6Server ）で構成されます。

$arch

この変数を使用して、Python の os.uname() 関数を呼び出す際に返り値としてシステムの CPU アーキテクチャーを参照できます。$arch の有効な値には、i686 および x86_64 が含まれます。

$basearch

$basearch を使用して、システムのベースアーキテクチャーを参照できます。たとえば、i686 マシンには i386 のベースアーキテクチャーがあり、AMD64 および Intel 64 のマシンには x86_ 64 のベースアーキテクチャーがあります。

$YUM0-9

これら 10 個の変数は、それぞれ同じ名前を持つシェル環境変数の値に置換されます。これらの変数の 1 つが（たとえば /etc/yum.conf で）参照され、同じ名前のシェル環境変数が存在しない場合、設定ファイルの変数は置換されません。

カスタム変数の定義、既存の変数値の上書きを行うには、/etc/yum/vars/ ディレクトリーに変数と同じ名前を持つファイルを作成して（「$」 記号なし）、必要な値を 1 行目に追加します。

たとえば多くの場合、リポジトリーの詳細にはオペレーティングシステムの名前が含まれます。$osname と呼ばれる新しい変数を定義するには、「Red Hat Enterprise Linuxnbsp;Hat

Enterprise Linuxnbsp;Linux」 の新規ファイルを 1 行目に作成し、/etc/yum/vars/osname として保存します。

```
~]# echo "Red Hat Enterprise Linux" > /etc/yum/vars/osname
```

「Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;LinuxRed Hat Enterprise Linux 6nbsp;6」 の代わりに、.repo ファイルで以下を使用できるようになりました。

```
name=$osname $releasever
```

8.4.4. 現在の設定の表示

yum グローバルオプションの現在の値（ /etc/yum.conf ファイルの [main] セクションで指定されているオプション）を表示するには、コマンドラインオプションなしで yum-config-manager を実行します。

```
yum-config-manager
```

別の設定セクションの内容を一覧表示するには、以下の形式でコマンドを実行します。

```
yum-config-manager section
```

glob 表現を使用して、適合する全セクションの設定を表示することもできます。

```
yum-config-manager glob_expression
```

たとえば、設定オプションとその対応する値を一覧表示するには、シェルプロンプトで以下を入力します。

```
~]$ yum-config-manager main \*
Loaded plugins: product-id, refresh-packagekit, subscription-manager
================================== main ==================================
[main]
alwaysprompt = True
assumeyes = False
bandwith = 0
bugtracker_url = https://bugzilla.redhat.com/enter_bug.cgi?
product=Red%20Hat%20Enterprise%20Linux%206&component=yum
cache = 0
[output truncated]
```

8.4.5. yum リポジトリーの追加、有効化、および無効化

「[repository] オプションの設定」 Yum リポジトリーの定義に使用できる各種のオプションを説明しました。本セクションでは、yum-config-manager コマンドを使用してリポジトリーを追加、有効化、および無効化する方法を説明します。

/ETC/YUM.REPOS.D/REDHAT.REPO ファイル

システムが証明書ベースの Red Hat Network に登録されると、Red Hat サブスクリプションマネージャーツールを使用して、/etc/yum.repos.d/redhat.repo ファイル内のリポジトリーを管理します。Red Hat Network にシステムを登録する方法や、Red Hat Subscription Manager ツールを使用してサブスクリプションを管理する方法は、6章システム登録およびサブスクリプション管理 を参照してください。

yum リポジトリーの追加

新しいリポジトリーを定義するには、[repository] セクションを /etc/yum.conf ファイル、または /etc/yum.repos.d/ ディレクトリーの .repo ファイルに追加します。このディレクトリーにある .repo ファイル拡張子が付いたすべてのファイルは、yum が読み取ります。リポジトリーは、/etc/yum.conf ではなく、ここに定義することが推奨されます。

信頼できないソフトウェアソースを使用する場合は注意してください。

ソフトウェアパッケージを、Red Hat Network 以外の未検証または信頼できないソフトウェアソースから取得してインストールする場合には、セキュリティー上のリスクが伴います。セキュリティー、安定性、互換性、保全性の問題が発生する可能性があります。

yum リポジトリーは、一般的に .repo ファイルを提供します。このようなリポジトリーをシステムに追加して有効にするには、root で以下のコマンドを実行します。

```
yum-config-manager --add-repo repository_url
```

repository_url は、.repo ファイルへのリンクになります。たとえば、http://www.example.com/example.repo にあるリポジトリーを追加するには、シェルプロンプトで以下を入力します。

```
~]# yum-config-manager --add-repo http://www.example.com/example.repo
Loaded plugins: product-id, refresh-packagekit, subscription-manager
adding repo from: http://www.example.com/example.repo
```

```
grabbing file http://www.example.com/example.repo to /etc/yum.repos.d/example.repo
example.repo                                     |  413 B     00:00
repo saved to /etc/yum.repos.d/example.repo
```

yum リポジトリーの有効化

特定のリポジトリーを有効にするには、root で次のコマンドを実行します。

```
yum-config-manager --enable repository
```

repository は一意のリポジトリー ID です（yum repolist all を使用して、利用可能なリポジトリー ID を一覧表示します）。別の方法では、glob 表現を使用すると、一致するすべてのリポジトリーを有効にできます。

```
yum-config-manager --enable glob_expression
```

たとえば、[example] セクション、[example-debuginfo] セクション、および [example-source]セクションで定義されたリポジトリーを有効にするには、以下を入力します。

```
~]# yum-config-manager --enable example\*
Loaded plugins: product-id, refresh-packagekit, subscription-manager
================================ repo: example ================================
[example]
bandwidth = 0
base_persistdir = /var/lib/yum/repos/x86_64/6Server
baseurl = http://www.example.com/repo/6Server/x86_64/
cache = 0
cachedir = /var/cache/yum/x86_64/6Server/example
[output truncated]
```

成功すると、yum-config-manager --enable コマンドは現在のリポジトリー設定を表示します。

yum リポジトリーの無効化

Yum リポジトリーを無効にするには、root で以下のコマンドを実行します。

```
yum-config-manager --disable repository
```

repository は一意のリポジトリー ID です（yum repolist all を使用して、利用可能なリポジトリー ID を一覧表示します）。yum-config-manager --enable と同様に、glob 表現を使用して、一致するすべてのリポジトリーを同時に無効にできます。

```
yum-config-manager --disable glob_expression
```

成功すると、yum-config-manager --disable コマンドは現在の設定を表示します。

8.4.6. yum リポジトリーの作成

Yum リポジトリーを設定するには、以下の手順に従います。

1. createrepo パッケージをインストールします。これを行うには、root で次のコマンドを実行します。

```
yum install createrepo
```

2. リポジトリー内に存在させるパッケージをすべて 1 つのディレクトリー（/mnt/local_repo/ など）にコピーします。

3. このディレクトリーに移動し、以下のコマンドを実行します。

```
createrepo --database /mnt/local_repo
```

これにより、yum リポジトリーに必要なメタデータと、yum 操作を迅速化する sqlite データベースが作成されます。

RED HAT ENTERPRISE LINUXNBSP;HAT ENTERPRISE LINUXNBSP;LINUX 5 での CREATEREPO コマンドの使用

Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux 5 と比較すると、Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;LinuxRed Hat Enterprise Linux 6nbsp;6 の RPM パッケージは、XZ の損失なしのデータ圧縮形式で圧縮され、SHA-256 などの新しいハッシュアルゴリズムで署名できます。したがって、Red Hat Enterprise Linuxnbsp;Hat Enterprise Red Hat Enterprise Linux 6nbsp;Linux 5 で createrepo コマンドを使用して Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;Linux Red Hat Enterprise Linux 6nbsp;Linux Red Hat Enterprise Linux 6nbsp;6 のパッケージメタデータを作成することは推奨されません。

8.4.7. yum キャッシュの使用

デフォルトでは、操作が正常に実行されると、yum はダウンロードしたデータファイルを削除します。これにより、yum が使用するストレージ容量が最小限に抑えられます。ただし、yum がダウンロードしたパッケージファイルがキャッシュディレクトリーに残るように、キャッシュを有効にできま

す。キャッシュされたデータを使用して、ネットワーク接続なしで特定の操作を実行できます。また、キャッシュに保存されているパッケージをコピーして別の場所で再利用することもできます。

yum は、/var/cache/yum/$basearch/$releasever/ ディレクトリーに一時ファイルを保存します。$basearch および $releasever は、システムのベースアーキテクチャーと Red Hat Enterprise Linux のリリースバージョンを参照する Yum 変数です。設定した各リポジトリーには 1 つのサブディレクトリーがあります。たとえば、/var/cache/yum/$basearch/$releasever/development/packages/ ディレクトリーには、開発リポジトリーからダウンロードしたパッケージを保持します。yum version コマンドの出力で、$basearch 変数および $releasever 変数の値を確認できます。

デフォルトのキャッシュの場所を変更するには、/etc/yum.conf 設定ファイルの [main] セクションの cachedir オプションを変更します。yum の設定に関する詳細は、「Yum と Yum リポジトリーの設定」 を参照してください。

キャッシュの有効化

インストールに成功するとパッケージのキャッシュを保持するには、/etc/yum.conf の [main] セクションに以下のテキストを追加します。

```
keepcache = 1
```

キャッシュを有効にすると、yum 操作はすべて、設定したリポジトリーからパッケージデータをダウンロードできます。

現在有効な yum リポジトリーのメタデータをダウンロードして利用できるようにするには、以下を入力します。

```
yum makecache
```

これは、キャッシュがすべてのメタデータで完全に最新の状態である場合に便利です。メタデータの有効期限を設定するには、/etc/yum.conf で metadata-expire 設定を使用します。

キャッシュ専用モードでの yum の使用

ネットワーク接続なしで yum コマンドを実行するには、コマンドラインオプション -C または --cacheonly を追加します。このオプションを使用すると、yum はネットワークリポジトリーを確認せずに続行し、キャッシュされたファイルのみを使用します。このモードでは、yum は以前の操作によってダウンロードおよびキャッシュされているパッケージのみをインストールできます。

たとえば、「gstreamer」 が含まれる名前で現在キャッシュされたデータを使用するパッケージを一覧表示するには、以下のコマンドを入力します。

```
yum -C list gstreamer*
```

yum キャッシュの消去

多くの場合、/var/cache/yum/ ディレクトリーに累積したエントリーを削除すると便利です。キャッシュからパッケージを削除すると、システムにインストールされているソフトウェアのコピーには影響を及ぼしません。現在有効なリポジトリーのエントリーをキャッシュから削除するには、root で以下を入力します。

```
yum clean all
```

削除するキャッシュデータのタイプに応じて、yum を clean モードで起動する方法は複数あります。利用可能な設定オプションの一覧は、表8.3「利用可能な yum clean オプション」 を参照してください。

表8.3 利用可能な yum clean オプション

オプション	説明
expire-cache	各リポジトリーのメタデータおよびミラーリストのダウンロードの時間レコードを取り除きます。これにより、yum は次回の使用時に各リポジトリーのキャッシュを再無効にします。
packages	キャッシュされたパッケージをシステムから削除
ヘッダー	以前のバージョンの yum が依存関係解決に使用したヘッダーファイルをすべて削除します。
metadata	パッケージのリモートの可用性を決定するために yum が使用するファイルをすべて削除します。これらのメタデータは、次回 yum の実行時にダウンロードされます。
dbcache	メタデータへの迅速なアクセスに使用する sqlite キャッシュを削除します。このオプションを使用すると、yum が、次回実行時に sqlite メタデータをダウンロードするように強制します。これは、.xml データのみが含まれるリポジトリーには適用されません。この場合、sqlite データは削除されますが、後続のダウンロードはありません。
rpmdb	ローカルの rpmdb からキャッシュされたデータを削除します。

オプション	説明
plugins	キャッシュされたデータを取り除くために有効なプラグインは強制的に実行されます。
all	上記をすべて削除します。

expire-cache オプションは、上記の一覧から最も推奨されます。多くの場合、クリーン な 状態にするには不十分です。

8.4.8. Optional および Supplementary リポジトリーの追加

オプションおよび Supplementary サブスクリプションチャンネルは、オープンソースのライセンス付きソフトウェア（Optional チャンネル）および商用ライセンス付きソフトウェア（Supplementary チャンネル）に対応する Red Hat Enterprise Linux 用の追加ソフトウェアパッケージを提供します。

Optional および Supplementary チャンネルをサブスクライブする前に、「対象範囲の詳細」を参照してください。これらのチャンネルからパッケージをインストールする場合は、Red Hat カスタマーポータルの記事「How to access Optional and Supplementary channel, and -devel packages using Red Hat Subscription Manager(RHSM)?」に記載されている手順に従います。

8.5. YUM のプラグイン

yum は、その操作を拡張し、強化するプラグインを提供します。特定のプラグインが、デフォルトでインストールされています。yum コマンドを呼び出すたびに、読み込まれ、アクティブになっているプラグインがあれば、yum がそれを通知します。以下に例を示します。

```
~]# yum info yum
Loaded plugins: product-id, refresh-packagekit, subscription-manager
[output truncated]
```

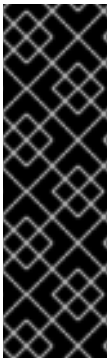
Loaded plugins に続くプラグイン名は、--disableplugins=plugin_nameオプションに指定できる名前 です。

8.5.1. yum プラグインを有効、設定、および無効にする方法

Yum プラグインを有効にするには、/etc/yum.conf の [main] セクションに plugins= で始まる行を追加し、その値が 1 であることを確認します。

```
plugins=1
```

すべてのプラグインを無効にするには、この行を plugins=0 に変更します。



プラグインをすべて無効にすることは推奨されません。

一部のプラグインは重要な Yum サービスを提供するため、すべてのプラグインを無効にすることは推奨されません。特に、rhnplugin は RHN Classic に対応しており、product-id プラグインおよび subscription-manager プラグインは、証明書ベースの コンテンツ配信ネットワーク (CDN)のサポートを提供します。グローバルにプラグインを無効にすることは便利なオプションとして提供されますが、通常は Yum の潜在的な問題を診断する場合にのみ推奨されます。

インストールされたすべてのプラグインには、/etc/yum/pluginconf.d/ ディレクトリーに独自の設定ファイルがあります。このファイルに、プラグイン固有のオプションを設定できます。たとえば、以下のように refresh-packagekit プラグインの refresh-packagekit.conf 設定ファイルがあるとします。

```
[main]
enabled=1
```

プラグイン設定ファイルには、[main] セクション（Yum の /etc/yum.conf ファイルと同様）が常に含まれます（または存在しない場合は配置可能）、yum コマンドの実行時にプラグインを有効にするかどうかを制御する enabled= オプションを使用できます。

/etc/yum.conf に enabled=0 を設定してすべてのプラグインを無効にすると、個々の設定ファイルで有効かどうかに関わらず、すべてのプラグインが無効になります。

1 つの yum コマンドで Yum プラグインをすべて無効にする場合は、--noplugins オプションを使用します。

1 つの yum コマンドで yum プラグインを無効にする場合は、--disableplugin=plugin_name オプションをコマンドに追加します。たとえば、システムの更新中に presto プラグインを無効にするには、以下を入力します。

```
~]# yum update --disableplugin=presto
```

--disableplugin= オプションに指定したプラグイン名は、yum コマンドの出力の Loaded プラグイン 行の後に一覧表示される名前と同じです。名前をコンマで区切ることにより、複数のプラグインを

無効にすることができます。さらに、glob 表現を使用して、複数のプラグイン名に一致したり、長いプラグイン名を短くすることができます。

```
~]# yum update --disableplugin=presto,refresh-pack*
```

8.5.2. 追加の Yum プラグインのインストール

yum プラグインは通常、yum-plugin-plugin_name package-naming 規則に従いますが、常に presto プラグインを提供するパッケージの名前は yum-presto です。Yum プラグインは、他のパッケージをインストールするのと同じ方法でインストールできます。たとえば、セキュリティー プラグインをインストールするには、シェルプロンプトで以下を入力します。

```
~]# yum install yum-plugin-security
```

8.5.3. プラグインの説明

以下では、便利な yum プラグインの説明と使用方法を紹介しています。プラグインは名前で表示されており、括弧内はパッケージ名になります。

search-disabled-repos (subscription-manager)

search-disabled-repos プラグインを使用すると、依存関係の解決に役立つ無効なリポジトリーを一時的または永続的に有効にすることができます。このプラグインが有効な場合は、依存関係の解決に失敗して Yum がパッケージのインストールに失敗したときに、無効なリポジトリーを一時的に有効し、再試行することが提示されます。インストールが成功した場合、Yum は使用されているリポジトリーを永久的に有効にすることも提示します。プラグインは subscription-manager で管理されるリポジトリーとのみ機能し、カスタムリポジトリーとは機能しないことに注意してください。

重要

yum が --assumeyes または -y オプションで実行されるか、/etc/yum.conf で assumeyes ディレクティブが有効になっている場合、プラグインは、確認を求めるプロンプトなしに、一時的に、かつ永続的に無効なリポジトリーを有効にします。この結果、有効にしたくないリポジトリーが有効になるといった問題が発生することがあります。

search-disabled-repos プラグインを設定するには、/etc/yum/pluginconf.d/search-disabled-repos.conf にある設定ファイルを編集します。[main] セクションで使用できるディレクティブのリストについては、以下の表を参照してください。

表8.4 サポートされている search-disabled-repos.conf ディレクティブ

ディレクティブ	詳細
enabled=value	プラグインを有効または無効にできます。値 は 1 （有効）または 0 （無効）のいずれかにする必要があります。プラグインはデフォルトで有効です。
notify_only=value	プラグインの動作を通知のみに制限できます。値 は 1 （Yum の動作の変更なしで通知のみ）または 0 （Yum の動作の変更）のいずれかにする必要があります。デフォルトでは、プラグインはユーザーへの通知のみを行います。
ignored_repos=repositories	プラグインで有効でないリポジトリーを指定できます。

kabi (kabi-yum-plugins)

kabi プラグインは、ドライバー更新パッケージが公式の Red Hatnbsp;Hat kernel Application Binary Interface (kABI)に準拠するかどうかを確認します。このプラグインが有効な状態で、ユーザーがホワイトリストにないカーネルシンボルを使用するパッケージのインストールを試行する場合は、警告メッセージがシステムログに書き込まれます。さらには、プラグインを enforcing モードで実行するように設定すると、そうしたパッケージがインストールされないようにできます。

kabi プラグインを設定するには、/etc/yum/pluginconf.d/kabi.conf にある設定ファイルを編集します。[main] セクションに使用できるディレクティブの一覧は、表8.5「サポートされる kabi.conf ディレクティブ」 を参照してください。

表8.5 サポートされる kabi.conf ディレクティブ

ディレクティブ	詳細
enabled=value	プラグインを有効または無効にできます。値 は 1 （有効）または 0 （無効）のいずれかにする必要があります。インストール時には、プラグインはデフォルトで有効です。
whitelists=directory	サポートされるカーネルシンボルを持つファイルがある ディレクトリー を指定できます。デフォルトでは、kabi プラグインは kernel-abi-whitelists パッケージ（ /lib/modules/kabi/ ディレクトリー）が提供するファイルを使用します。
enforce=value	enforcing モードを有効または無効にできます。値 は 1 （有効）または 0 （無効）のいずれかにする必要があります。デフォルトでは、このオプションはコメントアウトされ、kabi プラグインは警告メッセージのみを表示します。

presto (yum-presto)

presto プラグインは、更新中に delta RPM パッケージをダウンロードする Yum へのサポートを、事前メタデータが有効になっているリポジトリーから追加します。デルタ RPM には、RPM パッケージを要求するクライアントにインストールされているパッケージのバージョンと、リポジトリーの更新バージョンの違いのみが含まれます。

デルタ RPM のダウンロードは、更新されたパッケージ全体をダウンロードするよりもはるかに高速であり、更新を大幅に高速化できます。デルタ RPM をダウンロードしたら、現在インストールされているパッケージに違いを適用して完全に更新されたパッケージを作成するために再構築する必要があります。このプロセスは、インストールマシンで CPU 時間を使用します。したがって、デルタ RPM の使用は、ネットワーク接続に依存する time-to-download と、CPU をバインドしている time-to-rebuild 間の危険性があります。presto プラグインの使用は、ネットワーク接続が遅い高速マシンやシステムに推奨されますが、非常に高速の接続では、通常の RPM パッケージをダウンロードすることで、非常に高速のマシン(presto)を無効にすることで多くの利点があります。

product-id (subscription-manager)

product-id プラグインは、コンテンツ配信ネットワークからインストールされた製品の製品アイデンティティー証明書を管理します。product-id プラグインはデフォルトでインストールされています。

refresh-packagekit (PackageKit-yum-plugin)

refresh-packagekit プラグインは、yum が実行されるたびに PackageKit のメタデータを更新します。refresh-packagekit プラグインはデフォルトでインストールされます。

rhnplugin (yum-rhn-plugin)

rhnplugin は、RHN Classic への接続に対応します。これにより、RHN Classic に登録されたシステムは、このシステムからパッケージを更新し、インストールできます。RHN Classic は、以前の Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux システム（Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux 4.x、Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;）に対してのみ提供されます。 Linux 5.x および Satellite 5.x は、それらを Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux 6 に移行するためのものです。rhnplugin はデフォルトでインストールされます。

プラグインの詳細は、man ページの rhnplugin(8)を参照してください。

セキュリティー (yum-plugin-security)

セキュリティー更新に関する情報を検出し、これらを適用することは、すべてのシステム管理者にとって重要です。このため、Yum は セキュリティー プラグインを提供します。これは、高度なセキュリティー関連コマンド、サブコマンド、およびオプションのセットで yum を拡張します。

セキュリティー関連の更新は、以下のように確認できます。

```
~]# yum check-update --security
Loaded plugins: product-id, refresh-packagekit, security, subscription-manager
Updating Red Hat repositories.
INFO:rhsm-app.repolib:repos updated: 0
Limiting package lists to security relevant ones
Needed 3 of 7 packages, for security
elinks.x86_64                0.12-0.13.el6              rhel
kernel.x86_64                2.6.30.8-64.el6            rhel
kernel-headers.x86_64        2.6.30.8-64.el6            rhel
```

その後、yum update --security または yum update-minimal --security のいずれかを使用して、セキュリティーアドバイザリーの影響を受けるパッケージを更新できます。これらのコマンドはいずれも、セキュリティーアドバイザリーが発行されたシステムのパッケージをすべて更新します。yum update-minimal --security は、セキュリティーアドバイザリーの一部としてリリースされた最新パッケージに更新されます。一方、yum update --security は、セキュリティーアドバイザリーの影響を受けるすべてのパッケージを、利用可能なパッケージの最新バージョンに 更新します。

言い換えると、以下のようになります。

- *kernel-2.6.30.8-16 パッケージがシステムにインストールされている。*
- *kernel-2.6.30.8-32 パッケージがセキュリティー更新としてリリースされました。*
- *次に、 kernel-2.6.30.8-64 がバグ修正の更新としてリリースされました。*

...then yum update-minimal --security により kernel-2.6.30.8-32 に更新され、yum update --security により kernel-2.6.30.8-64 が更新されます。システム管理者は、可能な限りパッケージを更新することで、update-minimal を使用してリスクを低減します。

yum に追加されたセキュリティープラグインの拡張に関する詳細は、yum- security (8)man ページを参照してください。

subscription-manager (subscription-manager)

subscription-manager プラグインは、Red Hat Network への接続をサポートします。これにより、Red Hat Network に登録したシステムが、証明書ベースのコンテンツ配信ネットワークからパッケージを更新およびインストールできます。subscription-manager プラグインはデフォルトでインストールされています。

製品のサブスクリプションとエンタイトルメントを管理する方法は、6章システム登録およびサブスクリプション管理 を参照してください。

yum-downloadonly (yum-plugin-downloadonly)

yum-downloadonly プラグインは、--downloadonly コマンドラインオプションを提供します。このオプションを使用すると、パッケージをインストールせずに Red Hat Network または設定済みの Yum リポジトリーからパッケージをダウンロードできます。

パッケージをインストールするには、「追加の Yum プラグインのインストール」 に記載の手順に従います。インストール後に、/etc/yum/pluginconf.d/downloadonly.conf ファイルの内容を確認し、プラグインが有効であることを確認します。

```
~]$ cat /etc/yum/pluginconf.d/downloadonly.conf
[main]
enabled=1
```

以下の例では、yum install --downloadonly コマンドを実行して、インストールせずに最新バージョンの httpd パッケージをダウンロードします。

```
~]# yum install httpd --downloadonly
Loaded plugins: downloadonly, product-id, refresh-packagekit, rhnplugin,
              : subscription-manager
Updating Red Hat repositories.
Setting up Install Process
Resolving Dependencies
--> Running transaction check
---> Package httpd.x86_64 0:2.2.15-9.el6_1.2 will be updated
---> Package httpd.x86_64 0:2.2.15-15.el6_2.1 will be an update
--> Processing Dependency: httpd-tools = 2.2.15-15.el6_2.1 for package: httpd-2.2.15-
15.el6_2.1.x86_64
--> Running transaction check
---> Package httpd-tools.x86_64 0:2.2.15-9.el6_1.2 will be updated
---> Package httpd-tools.x86_64 0:2.2.15-15.el6_2.1 will be an update
--> Finished Dependency Resolution
```

```
Dependencies Resolved

================================================================================
 Package       Arch      Version                Repository             Size
================================================================================
Updating:
 httpd         x86_64    2.2.15-15.el6_2.1      rhel-x86_64-server-6    812 k
Updating for dependencies:
 httpd-tools   x86_64    2.2.15-15.el6_2.1      rhel-x86_64-server-6     70 k

Transaction Summary
================================================================================
Upgrade       2 Package(s)

Total download size: 882 k
Is this ok [y/N]: y
Downloading Packages:
(1/2): httpd-2.2.15-15.el6_2.1.x86_64.rpm               | 812 kB     00:00
(2/2): httpd-tools-2.2.15-15.el6_2.1.x86_64.rpm         |  70 kB     00:00
--------------------------------------------------------------------------------
Total                                   301 kB/s | 882 kB     00:02


exiting because --downloadonly specified
```

デフォルトでは、Red Hat Enterprise Linux のバリアントおよびアーキテクチャーに応じて、--downloadonly オプションを使用してダウンロードしたパッケージは、/var/cache/yum ディレクトリーのサブディレクトリーのいずれかに保存されます。

パッケージを保存する代替ディレクトリーを指定する場合は、--downloaddir オプションを --downloadonly とともに渡します。

```
~]# yum install --downloadonly --downloaddir=/path/to/directory httpd
```

注記

yum-downloadonly プラグインの代わりに、パッケージをインストールせずにパッケージをダウンロードする代わりに、yum-utils パッケージが提供する yumdownloader ユーティリティーを使用できます。

8.6. その他のリソース

Red Hat Enterprise Linux でソフトウェアパッケージを管理する方法は、以下の資料を参照してください。

インストールされているドキュメント

- *yum(8): yum コマンドラインユーティリティーの man ページには、サポートされるオプションおよびコマンドの完全なリストを提供します。*
- *yumdb(8): yumdb コマンドラインユーティリティーの man ページでは、このツールを使用してクエリーを行い、必要な場合は yum データベースを変更する方法が説明されています。*
- *yum.conf (5): yum.conf の man ページでは、利用可能な yum 設定オプションが説明されています。*
- *yum-utils (1): yum-utils の man ページでは、yum 設定の管理、リポジトリーの操作、yum データベースでの作業を行う追加ユーティリティーの一覧表示と簡単な説明が提供されます。*

オンラインリソース

- *yum Guides: プロジェクトのホームページの 『Yum Guides』 ページには、追加のドキュメントへのリンクがあります。*
- *Red Hat Access Labs: Red Hat Access Labs に 「Yum Repository Configuration Helper」 が含まれています。*

その他の参考資料

- *4章権限の取得 では、su および sudo コマンドを使用して管理者権限を取得する方法を説明しています。*
- *付録B RPM では、Red Hat Enterprise Linux で使用されるパッケージ化システムである RPM パッケージマネージャー(RPM)を説明します。*

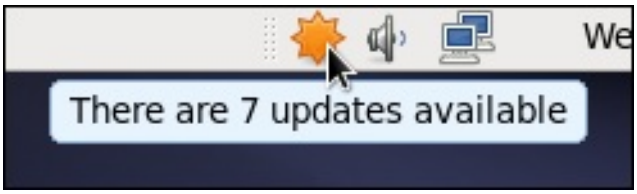
第9章 PACKAGEKIT

Red Hat は、お使いのシステムと互換性のあるパッケージを表示、管理、更新、インストール、およびアンインストールする PackageKit を提供します。PackageKit は、GNOME パネルメニューから開くことができる複数のグラフィカルインターフェース、または PackageKit が更新が利用可能であることを通知エリアから開くことができます。PackageKit のアーキテクチャーおよび利用可能なフロントエンドの 詳細は、「PackageKit Architecture」 を参照してください。

9.1. ソフトウェア更新によるパッケージの更新

システムにインストールできる更新が利用できる場合には、Kugo は通知エリアに星のバーストアイコンを表示します。

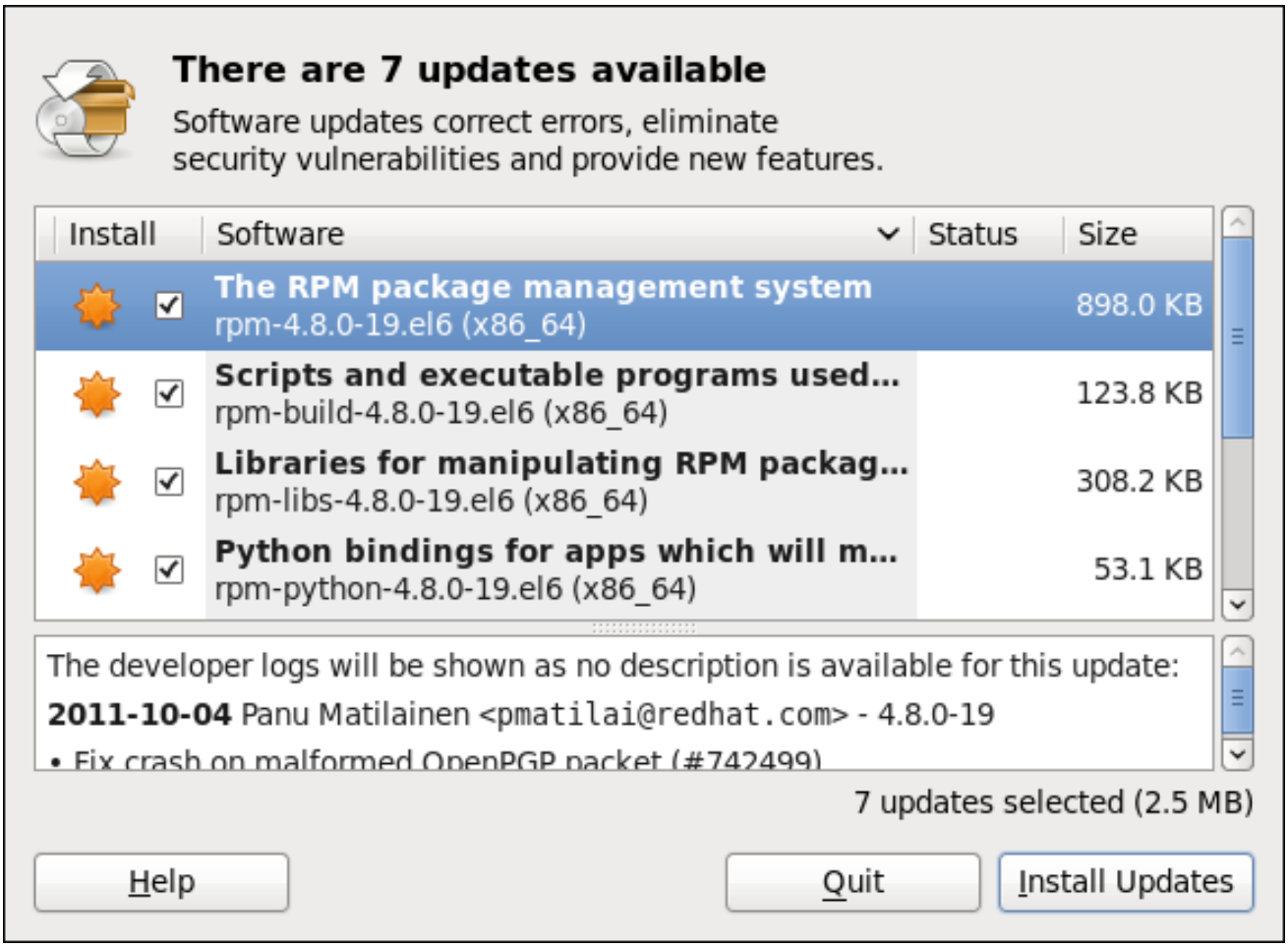
図9.1 通知エリアの PackageKit's アイコン



[D]

通知アイコンをクリックすると、Software Update ウィンドウが開きます。また、GNOME パネルから System → Administration → Software Update をクリックするか、シェルプロンプトで gpk-update-viewer コマンドを実行すると、ソフトウェアの更新を開くことができます。ソフトウェア 更新 ウィンドウでは、利用可能な更新はすべて、更新中のパッケージ名（ .rpm 接尾辞を引いたもの（CPU アーキテクチャーを含む）、パッケージの短い説明）とともに表示されます。通常、更新が提供する変更の短い説明です。インストールしたくない更新は、更新に対応するチェックボックスの選択を解除して、ここで選択解除できます。

図9.2 ソフトウェア更新による更新のインストール



[D]

ソフトウェア 更新 画面に表示される更新は、更新が利用可能なシステムで現在インストールされているパッケージのみを表します。パッケージの依存関係（システムの既存パッケージであるか、新しいパッケージであるかにかかわらず、更新の インストールをクリックするまで表示されません。

PackageKit は、システムに変更を加えるように要求するたびに、PolicyKit ツールキットが提供する粒度の細かいユーザー認証機能を使用します。PackageKit にパッケージを更新、インストール、または削除するように指示すると、システムへの変更が加えられる前にスーパーユーザーパスワードを入力するように求められます。

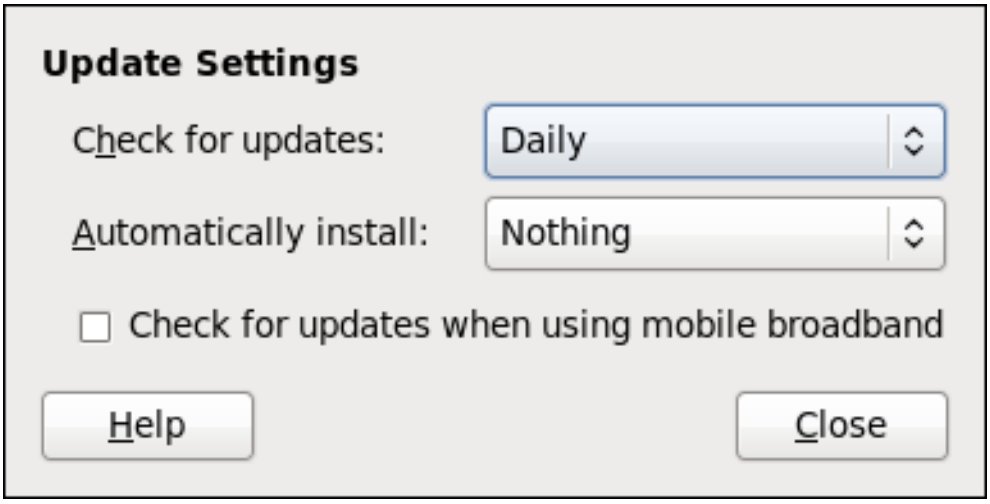
PackageKit により、kernel パッケージが更新されると、インストール後に求められ、システムを再起動して、新たにインストールしたカーネルを起動するかどうかを尋ねられます。

更新間隔の設定

PackageKit の Notification Area アイコンを右クリックし、Preferences をクリックすると、Software Update Preferences ウィンドウが表示されます。ここで、PackageKit がパッケージの更新をチェックする間隔や、すべての更新を自動的にインストールするか、セキュリティー更新のみを自動的にインストールするかを定義できます。モバイルブロードバンドボックスを使用する場合 は、ダ

ウンロードデータの量に対して課金するワイヤレス接続を使用する際に、不要な帯域幅の使用を回避するために、チェックを外します。

図9.3 PackageKit の更新チェック間隔の設定

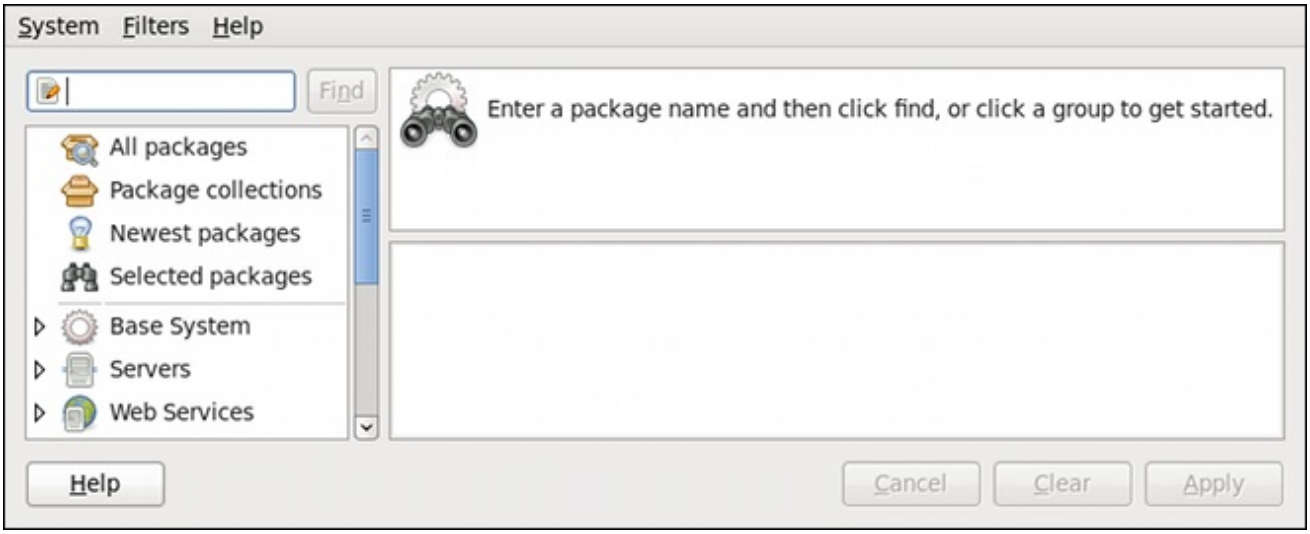


[D]

9.2. ソフトウェアの追加/削除

新しいパッケージを見つけてインストールするには、GNOME パネルで System → Administration → Add/Remove Software をクリックするか、シェルプロンプトで gpk-application コマンドを実行します。

図9.4 PackageKit's Add/Remove Software ウィンドウ



[D]

9.2.1. ソフトウェアソースの更新（Yum リポジトリー）

PackageKit は Yum リポジトリーをソフトウェアソースとして参照します。有効なソフトウェアソースからすべてのパッケージを取得します。ソフトウェアの 追加/削除を開き、System → Software をクリックすると、設定され たすべての Yum リポジトリーおよびフィルターが適用されていない（以

下を参照）Yum リポジトリーの一覧を表示できます。Software Sources ダイアログでは、/etc/yum.conf 設定ファイルのすべての [repository] セクション、および /etc/yum.repos.d/ ディレクトリーにあるすべての リポジトリー.repo ファイルで作成されるリポジトリー名が表示されます。name=<My Repository Name>

有効 コラムでチェックされるエントリーは、対応するリポジトリーを使用して更新およびインストール要求すべてを満たすパッケージを特定するために使用されます（依存関係解決を含む）。一覧表示された Yum リポジトリーのいずれかを有効または無効にするには、チェックボックスを選択または選択解除できます。これを実行すると、PolicyKit によりスーパーユーザー認証が要求されます。

Enabled 列は、[repository] セクションの enabled=<1 or 0> フィールドに対応します。チェックボックスをクリックすると、PackageKit は enabled= <1 または 0 > 行を正しい [repository] セクションに挿入します。存在しない場合は値を変更します。つまり、Software Sources ウィンドウでリポジトリーを有効または無効にすると、ウィンドウを閉じるか、またはシステムを再起動した後に変更が維持されます。

PackageKit から Yum リポジトリーを追加または削除できないことに注意してください。

ソース RPM、テスト、および DEBUGINFO リポジトリーの表示

Software Sources ウィンドウの下部にあるボックスを選択すると、PackageKit がソース RPM、テスト、および debuginfo リポジトリーも表示します。デフォルトでは、このボックスのチェックは解除されます。

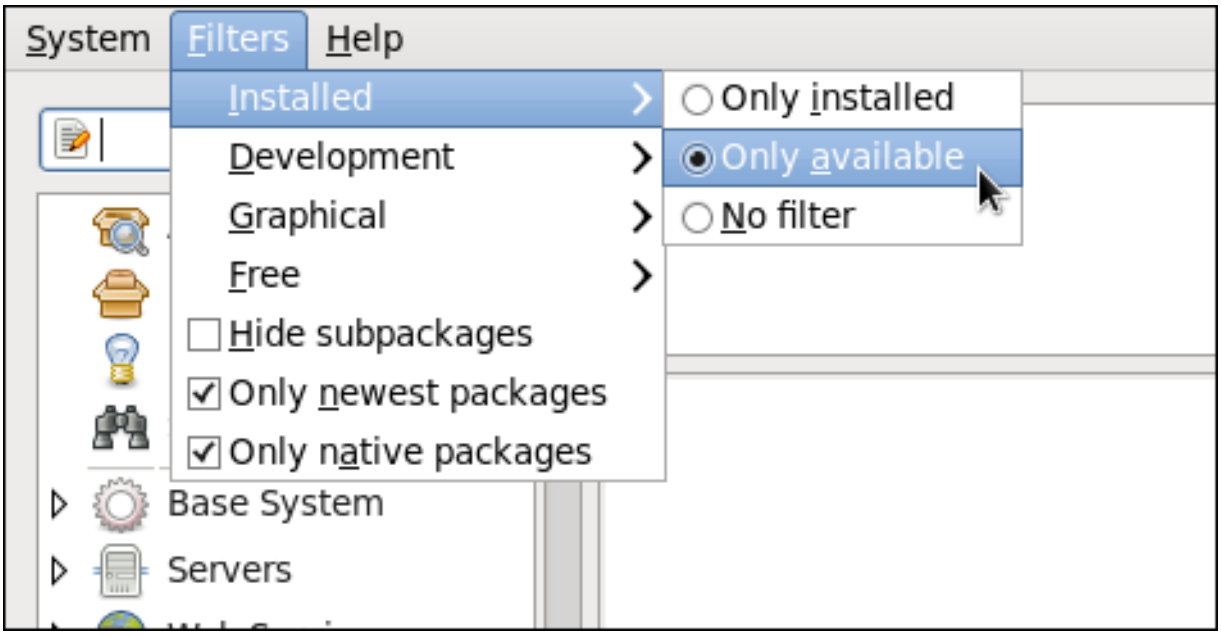
利用可能な Yum リポジトリーに変更を加えたら、システム → 更新 パッケージ一覧 をクリックして、パッケージリストが最新であることを確認します。

9.2.2. フィルターを使用したパッケージの検索

ソフトウェアソースが更新されると、PackageKit が Find クエリーの結果をより迅速に取得できるようにフィルターを適用すると便利です。これは、多くのパッケージ検索を実行する場合に特に便利です。Filters ドロップダウンメニューのフィルターの 4 つを使用して、単一の基準に一致させたり、一致させたりして結果を分割したりします。デフォルトでは、PackageKit が開始すると、これらのフィルターはすべて適用されない（フィルターなし）されますが、そのフィルターは変更するか、PackageKit を閉じるまで、そのフィルターが設定されます。

通常、システムにインストールされていない 利用可能なパッケージを検索するため、Filters → Installed をクリックし、利用可能なラジオボタンのみ を選択します。

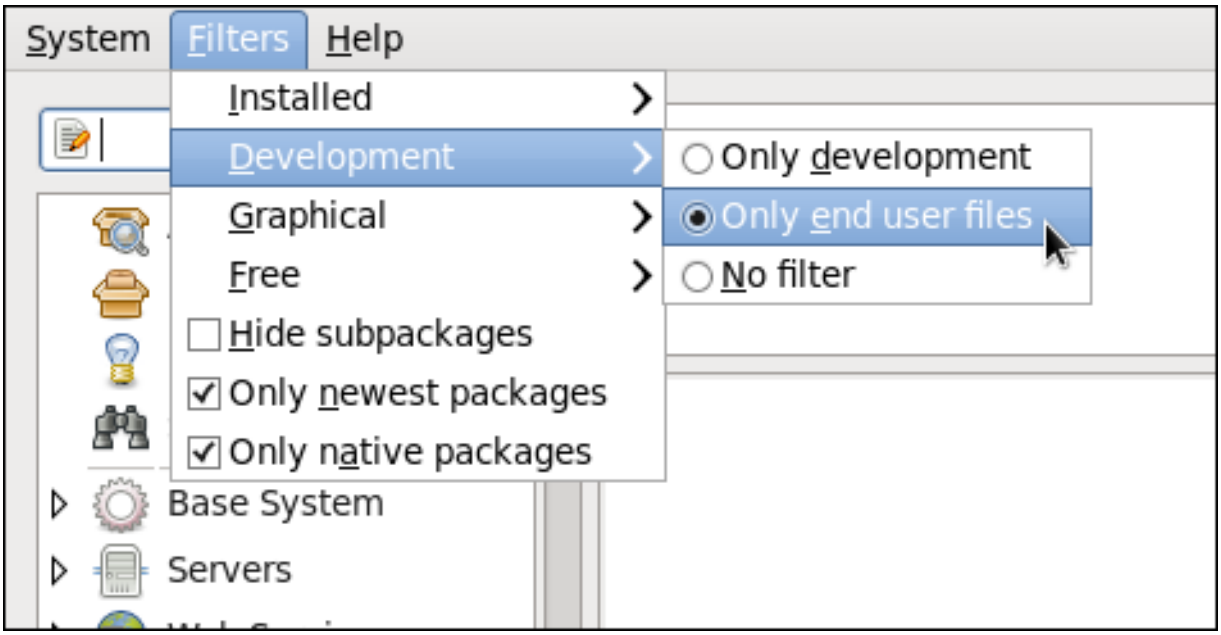
図9.5 インストール済みのパッケージのフィルタリング



[D]

また、C ヘッダーファイルなどの開発ファイルが必要でない限り、Filterd Development+クリックして、Only end user files ラジオボタンを選択します。このフィルターは、興味のない <package_name> -devel パッケージをすべて除外します。

図9.6 Find results 一覧から開発パッケージのフィルタリング



[D]

サブメニューの残りの 2 つのフィルターは以下のとおりです。

graphical

GUI インターフェースを提供するアプリケーション（グラフィカルのみ）か、そうでないアプリケーションに検索を絞り込みます。このフィルターは、特定の機能を実行する GUI アプリケーションを確認する場合に便利です。

Free

空きソフトウェアとして考慮されるパッケージを検索します。承認されたライセンスの詳細は、「Fedora Licensing List」を参照してください。

残りのフィルターを有効にするには、そのフィルターの横にあるチェックボックスを選択します。

サブパッケージの非表示

Hide サブ パッケージ チェックボックスを確認すると、通常、必要な他のパッケージの依存関係のみであるパッケージにフィルターが除外されます。たとえば、Hide サブ パッケージを確認し、<package> を検索すると、以下の関連パッケージが 検索 結果から除外されます（存在する場合）。

- *& lt;package> -devel*
- *& lt;package> -libs*
- *<package>-libs-devel*
- *<package>-debuginfo*

最新のパッケージのみ

最新の パッケージのみ を確認すると、結果のリストから同じパッケージの全古いバージョンがフィルタリングされます。通常、これは必要なものです。多くの場合、このフィルターは唯一の利用可能なフィルターと組み合わせて、利用可能な 新バージョン（インストールされていない）パッケージの最新バージョンを検索することに注意してください。

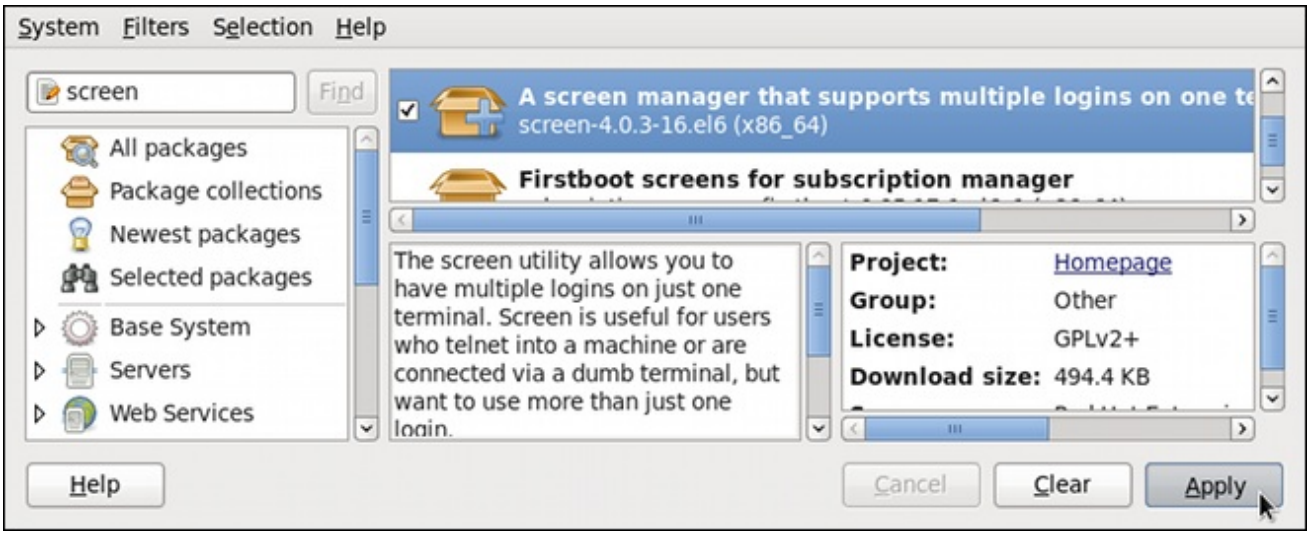
ネイティブパッケージのみ

multilib システムで 唯一のネイティブパッケージ ボックスを確認すると、互換性モード で実行されるアーキテクチャー用にコンパイルしたパッケージの結果の一覧表示が PackageKit が省略されます。たとえば、AMD64 CPU を備えた 64 ビットシステムでこのフィルターを有効にすると、AMD64 マシン上で実行できても、32 ビットの x86 CPU アーキテクチャー用に構築されたパッケージがすべて、結果の一覧に表示されなくなります。アーキテクチャーに依存しないパッケージ（crontabs-1 . 10-32.1.el6.noarch.rpm などの noarchパッケージ）は、ネイティブパッケージのみをチェックして除外されません。このフィルターは、x86 マシンなど、multilib 以外のシステムでは影響を受けません。

9.2.3. パッケージ（および依存関係）のインストールおよび削除

2 つのフィルターが選択さ れて おり、エンドユーザーファイル のみを選択し、画面 ウィンドウマネージャーでコマンドラインを検索し、パッケージを強調表示します。これで、プロジェクトのホームページへのクリック可能なリンク、パッケージのライセンス、アプリケーションが開くことのできる GNOME メニューロケーションへのポインター、およびダウンロードとインストール時に関連するパッケージのサイズなど、プロジェクトのホームページに関する非常に便利な情報にアクセスできます。

図9.7 PackageKit's Add/Remove Software ウィンドウを使用したパッケージの表示およびインストール



[D]

パッケージまたはグループの横にあるチェックボックスを選択すると、そのアイテムがシステムにインストールされていることになります。チェックボックスにチェックを入れるとインストールのマーク が付けられ、適用 ボタンをクリックするとのみ発生します。これにより、実際のインストールトランザクションを実行する前に、複数のパッケージまたはパッケージグループを検索して選択できます。さらに、チェックボックスの選択を解除して、インストールしたパッケージを削除することもできます。また、適用 が押される際に保留中のインストールと共に削除が行われます。インストールまたは削除する追加パッケージを追加する依存関係の解決は、Apply を押した後 に実行します。PackageKit は、インストールまたは削除する追加パッケージを一覧表示するウィンドウを表示し、確認を求めるプロンプトを表示します。

screen を選択し、Apply ボタンをクリックします。その後スーパーユーザーパスワードの入力が求

められます。入力すると PackageKit により 画面 がインストールされます。インストールが完了すると、PackageKit が新たにインストールしたアプリケーションの一覧を表示し、すぐに実行を選択できる場合があります。または、パッケージを検索して ソフトウェアの追加ウィンドウで選択すると、GNOME メニューのアプリケーションショートカットが置かれている 場所 が表示されます。これは、実行したい場合に便利です。

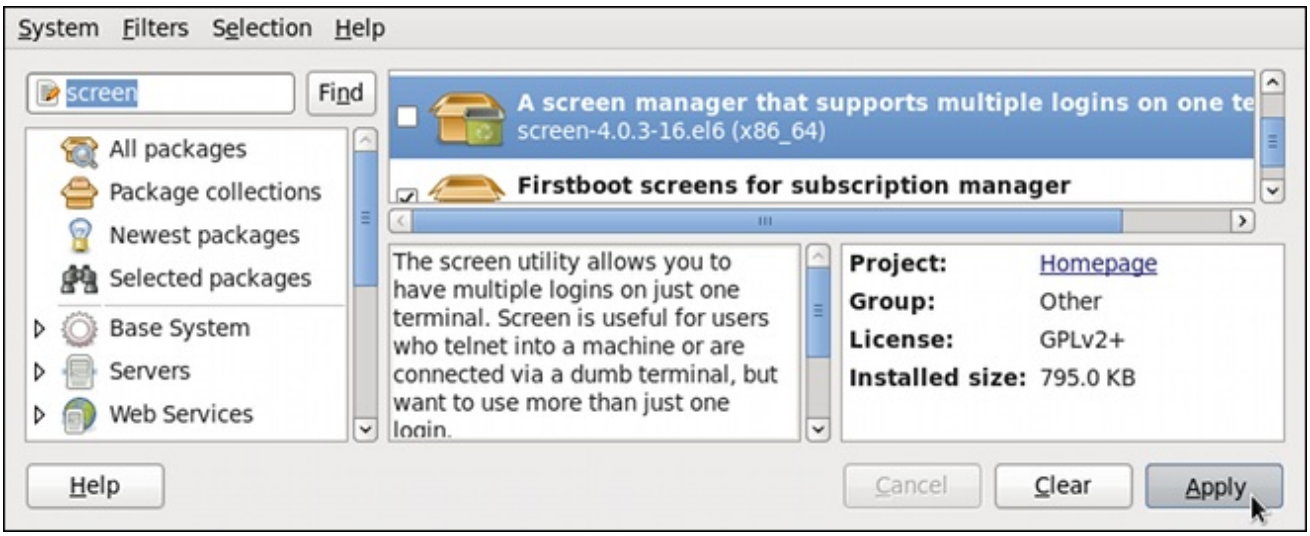
インストールしたら、シェルプロンプトに screen を入力して、1 つのターミナルで複数のログインを可能にする 画面 マネージャーを実行できます。

screen は非常に便利なユーティリティーですが、これは必要ありませんが、アンインストールしたいと判断します。最近 利用可能な フィルターのみを変更して Filters → Installed にインストールしただけ に変更する必要があります。再度 画面 を検索して選択解除します。プログラムが独自の依存関係をインストールしませんでした。これがあった場合、これらのパッケージも自動的に削除されます。他のパッケージの依存関係もシステムにインストールされている限り、そのプログラムは自動的に削除されます。

他のパッケージがそれに依存している場合にパッケージを削除

PackageKit は、パッケージのインストールおよび削除時に依存関係を自動的に解決しますが、そのパッケージに依存するパッケージを削除せずにパッケージを削除することはできません。このタイプの操作は RPM によってのみ実行でき、推奨されません。システムが機能しなくなるか、アプリケーションが誤作動したり、クラッシュしたりする可能性があります。

図9.8 PackageKit のソフトウェアの追加/削除ウィンドウでパッケージの削除

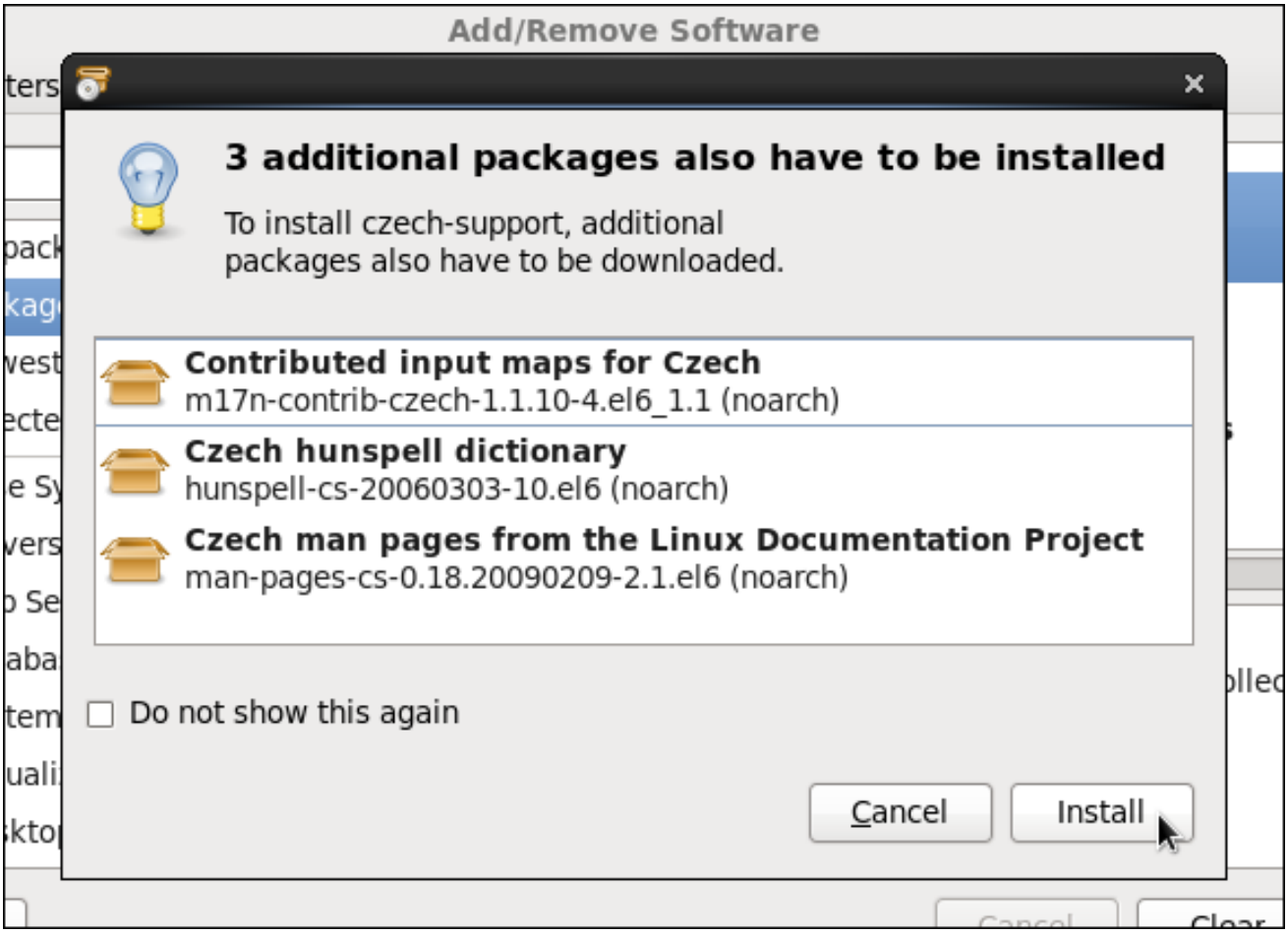


[D]

9.2.4. パッケージグループのインストールおよび削除

PackageKit には、パッケージ コレクション を呼び出す Yum パッケージグループをインストールする機能もあります。Software Updates ウィンドウにあるカテゴリーの左上にある Package collections をクリックすると、スクロールして、インストールするパッケージグループを検索できます。この場合、Czech 言語サポート（Czech サポート グループ）をインストールします。このボックスをチェックして apply をクリックすると、パッケージグループの依存関係を満たすために、インストールする必要のある 追加 パッケージの数が表示されます。

図9.9 Czech サポートパッケージグループのインストール



[D]

同様に、インストールされているパッケージグループをアンインストールするには、Package collections を選択し、適切なチェックボックスの選択を解除して適用します。

9.2.5. トランザクションログの表示

PackageKit は、実行するトランザクションのログを維持します。ログを表示するには、Add/Remove Software ウィンドウから System → Software log をクリックするか、シェルプロンプトで gpk-log コマンドを実行します。

Software Log Viewer には、以下の情報が表示されます。

- *date：トランザクションが実行された日付。*
- *アクション - トランザクション中に実行されたアクション（ 更新パッケージや インストール されたパッケージ など）。*
- *詳細: 更新され た、Installed、また は Removed などのトランザクションタイプと、影響を受けるパッケージの一覧が続きます。*
- *username: アクションを実行するユーザーの名前。*
- *アプリケーション： Update System などのアクションの実行に使用したフロントエンドアプリケーション。*

上部のテキスト入力フィールドにパッケージの名前を入力すると、そのパッケージに影響するトランザクションのリストをフィルタリングします。

図9.10 Software Log Viewer を使用したパッケージ管理トランザクションのログの表示

[D]

9.3. PACKAGEKIT ARCHITECTURE

Red Hat は、お使いのシステムと互換性のあるパッケージおよびパッケージグループを表示、更新、インストール、およびアンインストールするアプリケーションの PackageKit スイートを提供します。アーキテクチャーの観点的には、Packagekit d デーモンバックエンドと通信する複数のグラフィ

カルフロントエンドで構成されます。これは、Yum を使用してパッケージのインストールや削除などの実際のトランザクションを実行するパッケージマネージャー固有のバックエンドと通信します。

表9.1「PackageKit GUI ウィンドウ、メニューの場所、およびシェルプロンプトコマンド」 GUI ウィンドウの名前、GNOME デスクトップからウィンドウを開始する方法、または ソフトウェアの追加/削除 ウィンドウからウィンドウを開始する方法と、そのウィンドウを開くコマンドラインアプリケーションの名前を表示します。

表9.1 PackageKit GUI ウィンドウ、メニューの場所、およびシェルプロンプトコマンド

ウィンドウタイトル	関数	開く方法	shell コマンド
ソフトウェアの追加/削除	パッケージ情報をインストール、削除、または表示します。	GNOME パネルの使用： System → Administration → Add/Remove Software	gpk-application
ソフトウェアの更新	パッケージの更新の実行	GNOME パネルの使用： System → Administration → Software Update	gpk-update-viewer
ソフトウェアソース	Yum リポジトリーの有効化および無効化	ソフトウェアの追加/削除:システム → ソフトウェアソース	gpk-repo

ウィンドウタイトル	関数	開く方法	shell コマンド
Software Log Viewer	トランザクションログの表示	ソフトウェアの追加/削除：システムログの追加/ 削除	gpk-log
ソフトウェア更新の設定	PackageKit 設定の設定		gpk-prefs
（通知エリアアラート）	更新が利用可能になるとのアラート	GNOME パネルから - System → Preferences → Startup Applications、Start up Programs タブ	gpk-update-icon

packagekitd デーモンは、ユーザーセッションの外部で実行され、さまざまなグラフィカルフロントエンドと通信します。packagekitd デーモン[2] Yum の Python API を使用してクエリーを実行し、システムに変更を加えている別のバックエンドで DBus システムメッセージバスを介して通信します。Red Hat Enterprise Linux や Fedora 以外の Linux システムでは、packagekitd は、そのシステムにネイティブパッケージマネージャーを使用できる他のバックエンドと通信できます。このモジュラーアーキテクチャーは、グラフィカルインターフェースと多くの異なるパッケージマネージャーが動作するために必要な抽象化を提供し、基本的に同じタイプのパッケージ管理タスクを実行します。PackageKit フロントエンドの使用方法を理解すると、Yum 以外のネイティブパッケージマネージャーを使用する場合でも、多くの異なる Linux ディストリビューションで同じグラフィカルインターフェースを使用できます。

さらに、PackageKit は、GUI ウィンドウの 1 つをクラッシュさせる、またはユーザーの X Window セッションもクラッシュさせることで、ユーザーセッション外で実行される packagekitd デーモンによって監視されているパッケージ管理タスクに影響を与えないという信頼性を提供します。

本章で説明するフロントエンドのグラフィカルアプリケーションはすべて、PackageKit およびその依存関係ではなく、gnome-packagekit パッケージで提供されます。

最後に、PackageKit には pkcon と呼ばれるコンソールベースのフロントエンドが同梱されます。

9.4. その他のリソース

PackageKit の詳細は、以下に記載のドキュメントを参照してください。

インストールされているドキュメント

- gpk-application(1): gpk-application コマンドに関する情報が含まれる man ページです。
- gpk-backend-status(1): gpk-backend-status コマンドに関する情報が含まれる man ページです。
- gpk-install-local-file(1): gpk-install-local-file コマンドに関する情報が含まれる man ページです。
- gpk-install-mime-type(1): gpk-install-mime-type コマンドについての情報が含まれる man ページです。
- gpk-install-package-name(1): qpk-install-package-name コマンドに関する情報が含まれる man ページです。
- gpk-install-package-name(1): gpk-install-package-name コマンドに関する情報が含まれる man ページです。
- gpk-prefs(1): gpk-prefs コマンドについての情報が含まれる man ページです。
- gpk-repo(1): gpk-repo コマンドに関する情報が含まれる man ページです。
- gpk-update-icon(1): gpk-update-icon コマンドについての情報が含まれる man ページです。
- gpk-update-viewer(1): gpk-update-viewer コマンドについての情報が含まれる man ページです。

- *pkcon(1) および pkmon(1): PackageKit コンソールクライアントに関する情報が含まれる man ページ。*

- *pkgenpack(1): PackageKit Pack Generator に関する情報が含まれる man ページです。*

オンラインドキュメント

- *PackageKit ホームページ - PackageKit ホームページで、PackageKit ソフトウェアスイートに関する詳細情報が記載されます。*

- *PackageKit FAQ - PackageKit ソフトウェアスイートに関する Frequently Asked Questions に関する情報一覧。*

その他の参考資料

- *8章Yum Yum（Red Hat パッケージマネージャー）について説明しています。*

---

[2]
システムデーモンは通常、ユーザーまたは他のプログラムにサービスを提供する長時間実行されるプロセスであり、多くの場合は、特別な初期化スクリプト（init スクリプトに短縮）により起動時に起動されます。デーモンは サービス コマンドに応答し、chkconfig on コマンドまたは chkconfig off コマンドを使用して永続的にオンまたはオフにすることができます。通常、これらは packagekit 「d」デーモンなど、d に追加される名前で認識できます。システムサービスに関する情報は、12章サービスおよびデーモン を参照してください。

パート IV. ネットワーク

ここでは、Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux でネットワークを設定する方法を説明します。

第10章 NETWORKMANAGER

NetworkManager は、動的ネットワーク制御および構成システムで、利用可能なときにネットワークデバイスと接続が起動してアクティブな状態を維持しようとします。NetworkManager は、ネットワークステータス情報を提供する GNOME Notification Area アプレットであるコアデーモンと、接続およびインターフェースの作成、編集、削除が可能なグラフィカル設定ツールで構成されます。NetworkManager は、Ethernet、ワイヤレス、モバイルブロードバンド(cellular 3G)、DSL および PPPoE （イーサネット経由のPoint-to-Point over Ethernet）などの接続の種類を設定できます。また、NetworkManager は、ネットワークエイリアス、静的ルート、DNS 情報、VPN 接続、および多くの接続固有のパラメーターの設定を可能にします。最後に、NetworkManager は、D-Bus を介して豊富な API を提供します。これにより、アプリケーションはネットワーク設定と状態をクエリーし、制御できます。

以前のバージョンの Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux には、コマンドラインの呼び出し後に system-config-network と呼ばれるネットワーク 管理ツール が含まれていました。Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;LinuxRed Hat Enterprise Linux 6nbsp;6 では、 NetworkManager は former Network Administration Tool を置き換え、ユーザー固有やモバイルブロードバンド設定などの強化された機能を提供します。インターフェース設定ファイルを編集して、Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;LinuxRed Hat Enterprise Linux 6nbsp;6 でネットワークを設定することもできます。詳細は、11章Network Interfaces を参照してください。

NetworkManager は、お使いの Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux バージョンにデフォルトでインストールできます。インストールされていることを確認するには、root で以下のコマンドを実行します。

```
~]# yum install NetworkManager
```

10.1. NETWORKMANAGER デーモン

NetworkManager デーモンは root 権限で実行され、通常起動時に起動するように設定されています。NetworkManager デーモンが実行しているかどうかを確認するには、root で次のコマンドを実行します。

```
~]# service NetworkManager status
   NetworkManager (pid  1527) is running...
```

service コマンドは、NetworkManager サービスが実行していない場合は NetworkManager is stopped を報告します。現行セッションで開始するには、以下を実行します。

```
~]# service NetworkManager start
```

chkconfig コマンドを実行して、システムが起動するたびに NetworkManager が起動していることを確認します。

```
~]# chkconfig NetworkManager on
```

サービスおよびランレベルの開始、停止、および管理の詳細は、12章サービスおよびデーモン を参照してください。

10.2. NETWORKMANAGER との対話

ユーザーは、NetworkManager システムサービスを直接対話しません。代わりに、NetworkManager の Notification Area アプレットを介してネットワーク設定タスクを実行できます。アプレットには、現在使用している接続タイプの視覚的なインジケーターとして機能する複数の状態があります。現在の接続状態に関するツールチップ情報のアプレットアイコンにポインターを合わせます。

図10.1 NetworkManager アプレットの状態

[D]

GNOME パネルに NetworkManager アプレットが表示されておらず、NetworkManager パッケージがシステムにインストールされていることを仮定すると、以下のコマンドを通常のユーザーとして（rootではなく）実行してアプレットを起動できます。

```
~]$ nm-applet &
```

このコマンドを実行すると、アプレットが Notification Area に表示されます。System → Preferences → Startup Applications+Startup Applications Preferences ウィンドウを開くことで、アプレットがログインするたびに実行されることを確認できます。次に、Startup Programs タブを選択し、NetworkManager の横にあるチェックボックスを選択します。

10.2.1. ネットワークへの接続

アプレットアイコンをクリックすると、以下が表示されます。

- 現在接続している分類されたネットワークの一覧（Wire red や Wirelessなど）

- *NetworkManager が検出した 利用可能なネットワーク の一覧*
- *設定済みの仮想プライベートネットワーク(VPN)への接続オプション、および*
- *非表示ネットワークまたは新しいワイヤレスネットワークに接続するためのオプション。*

ネットワークに接続している場合は、その名前は、Wired や Wireless などのネットワークタイプの下に太字の型エリアで表示されます。ワイヤレスアクセスポイントなど、多くのネットワークが利用可能な場合には、その他のネットワーク 拡張可能なメニューエントリーが表示されます。

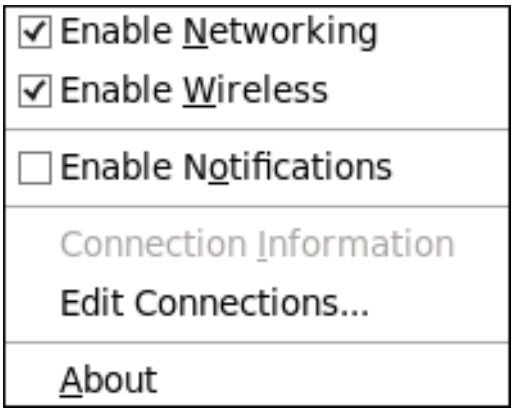
図10.2 NetworkManager アプレットの左側のメニューで、利用可能かつ接続されたネットワークをすべて表示します。

[D]

10.2.2. 既存接続の設定および編集

次に、NetworkManager アプレットを右クリックしてコンテキストメニューを開きます。コンテキストメニューは、NetworkManager と対話して接続を設定するための主要なエントリーポイントです。

図10.3 NetworkManager アプレットのコンテキストメニュー



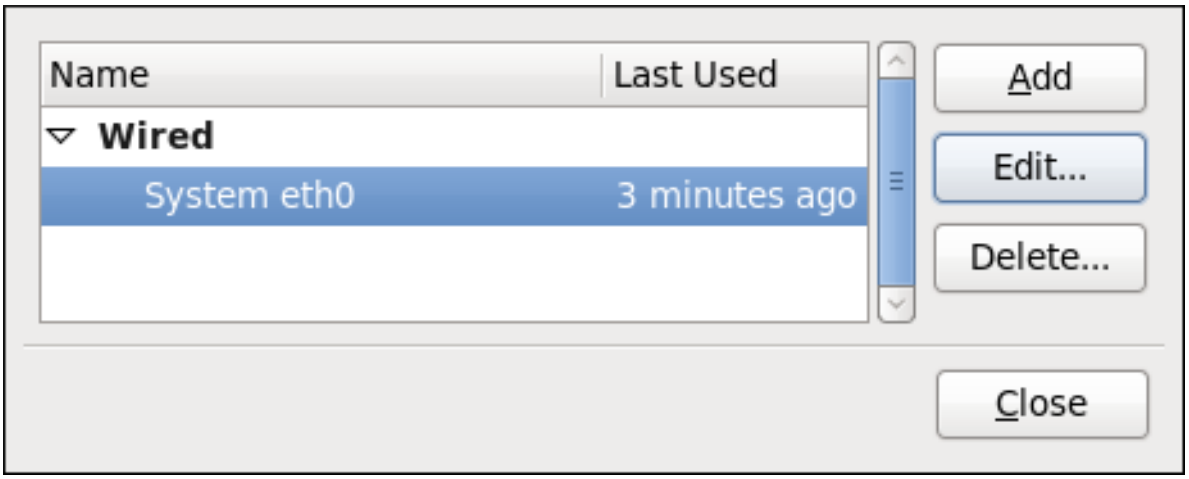
[D]

Enable Networking ボックスにチェックマークが付いていることを確認します。システムがワイヤレスカードを検出した場合は、Enable Wireless メニューオプションも表示されます。Enable Wireless チェックボックスもチェックします。通知の 有効化 ボックスにチェックを入れると、NetworkManager によりネットワーク接続のステータスの変更が通知 されます。Connection Information エントリーをクリックすると、接続タイプおよびインターフェース、IP アドレス、ルーティングの詳細などを一覧表示する参考 Connection Information ウィンドウが表示されます。

最後に、Edit Connections をクリックすると、ネットワーク設定タスクのほとんどを実行できる Network Connections ウィンドウが開きます。このウィンドウは、通常のユーザーとして実行して開くこともできます。

```
~]$ nm-connection-editor &
```

図10.4 ネットワーク接続ウィンドウを使用したネットワークの設定



[D]

左に矢印記号があり、クリックしてエントリーを非表示にし、必要に応じてエントリーを表示します。新しい接続を作成するには、Add ボタンをクリックして選択一覧を表示し、接続タイプを選択して

Create ボタンをクリックします。既存の接続を編集するには、一覧からインターフェース名を選択し、編集 ボタンをクリックします。

そして、以下のいずれかの設定をします。

- *有線イーサネット接続は、「Wired（イーサネット）接続の確立」 に進みます。*
- *ワイヤレス接続、または 「ワイヤレス接続の確立」 に進みます。*
- *モバイルブロードバンド接続。「モバイルブロードバンド接続の確立」 に進みます。*
- *VPN 接続で、「VPN 接続の確立」 に進みます。*

10.2.3. ネットワークの自動接続

追加や設定を行うすべての接続で、ネットワークが利用可能な時に NetworkManager が自動的に接続を試行するかどうか選択することができます。

手順10.1 検出時にネットワークに自動的に接続する NetworkManager の設定

1. *Notification Area の NetworkManager アプレットアイコンを右クリックし、Edit Connections をクリックします。ネットワーク接続 ウィンドウが表示されます。*
2. *必要に応じて矢印ヘッドをクリックして、接続の一覧を表示します。*
3. *設定する特定のコネクションを選択し、Edit をクリックします。*
4. *Connect を自動的 にチェックすると、NetworkManager が接続が利用可能であることを NetworkManager が検出されるたびに、NetworkManager が接続に自動接続できるようにします。NetworkManager が自動的に接続しない場合は、チェックボックスの選択を解除します。ボックスのチェックを外す場合は、NetworkManager アプレットの左側のメニューでその接続を手動で選択して、接続できるようにする必要があります。*

10.2.4. ユーザーおよびシステム接続

NetworkManager 接続は常に、ユーザー接続または システム接続 のいずれかです。管理者が設定したシステム固有のポリシーによっては、システム接続の作成および変更には root 権限が必要になる場合があります。NetworkManager のデフォルトポリシーを使用すると、ユーザーはユーザー接続を作成および変更できますが、システム接続を追加、変更、または削除するには root 権限が必要になります。

ユーザー接続は、作成するユーザーに固有のため、この接続と呼ばれます。システムの接続とは対照的に、設定が /etc/sysconfig/network-scripts/ ディレクトリー（主に ifcfg- <network_type > インターフェース設定ファイル）に保存され、ユーザー接続設定は GConf 設定データベースと GNOME キーリングに保存され、作成したユーザーのログインセッションにのみ利用できます。したがって、デスクトップセッションからログアウトすると、ユーザー固有の接続が利用できなくなります。

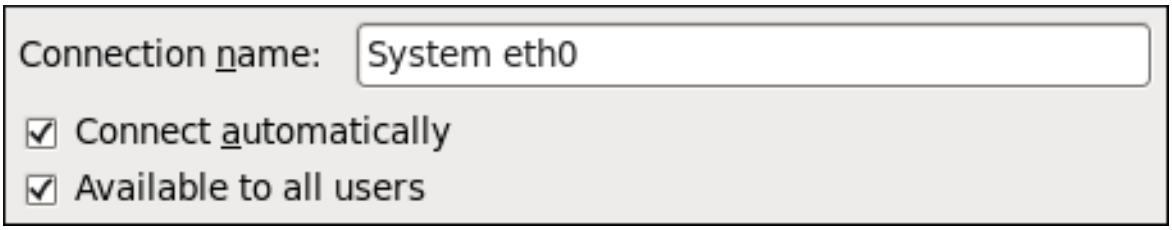
ユーザー固有の VPN 接続を作成してセキュリティーを強化する

NetworkManager は GConf および GNOME キーリングアプリケーションを使用してユーザー接続設定を保存します。これらの設定はデスクトップセッションに固有のため、ユーザー接続として個人の VPN 接続を設定することを強く推奨します。これを行うと、システム上の他のroot 以外のユーザーは、これらの接続を表示したり、アクセスしたりできません。

一方、システム接続は起動時に利用可能になり、デスクトップセッションに最初にログインせずにシステム上の他のユーザーが使用できます。

NetworkManager は、システム接続に迅速かつ便利に変換できます。また、その逆も同様です。ユーザー接続をシステム接続に変換すると、NetworkManager は /etc/sysconfig/network-scripts/ ディレクトリーに関連するインターフェース設定ファイルを作成し、ユーザーのセッションから GConf 設定を削除します。逆に、システムをユーザー固有の接続に変換すると、NetworkManager はシステム全体の設定ファイルを削除し、対応する GConf/GNOME キーリング設定を作成します。

図10.5 すべてのユーザーが使用できる チェックボックスは、接続がユーザー固有か、またはシステム全体であるかを制御します。



[D]

手順10.2 System-Wide または Vice-Versa の代わりに接続をユーザー固有のものに変更

ROOT 権限が必要になる場合があります。

システムのポリシーによっては、接続がユーザー固有か、またはシステム全体であるかを変更するために、システムで root 権限が必要になる場合があります。

1. Notification Area の NetworkManager アプレットアイコンを右クリックし、Edit Connections をクリックします。ネットワーク接続 ウィンドウが表示されます。

2. 必要に応じて、矢印ヘッド（左側上）を選択して非表示にし、利用可能なネットワーク接続の種類を表示します。

3. 設定する特定のコネクションを選択し、Edit をクリックします。

4. Available to all users チェックボックスをチェックして、NetworkManager にシステム全体の接続させるように要求します。システムポリシーによっては、PolicyKit アプリケーションによって root パスワードの入力が求められます。その場合は、root パスワードを入力して変更の最終処理を行います。

   逆に、Available to all users チェックボックスの選択を解除して、接続ユーザー固有のものにします。

10.3. 接続の確立

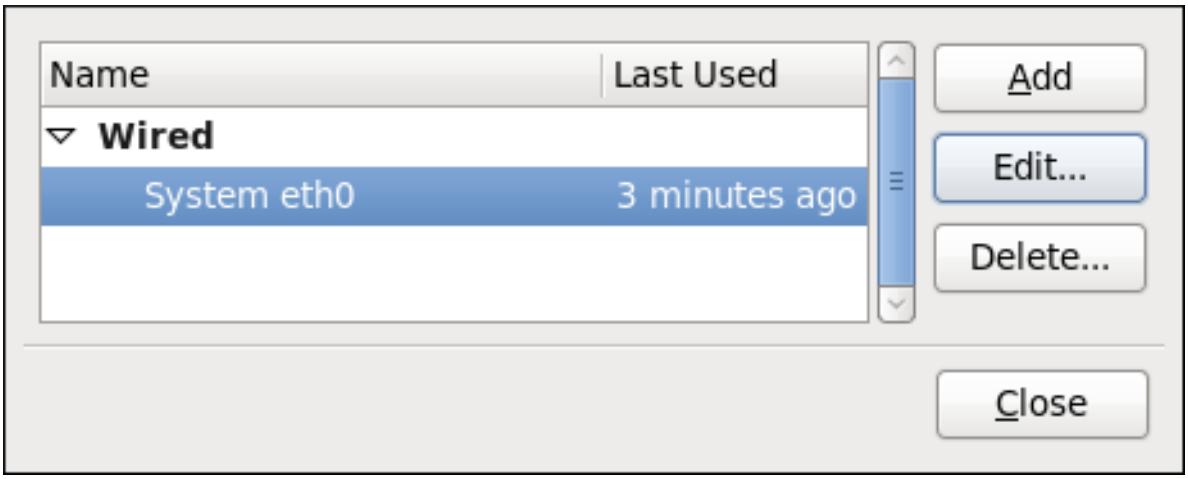
10.3.1. Wired（イーサネット）接続の確立

有線ネットワーク接続を確立するには、NetworkManager アプレットを右クリックしてコンテキストメニューを開き、Enable Networking ボックスをチェックして Edit Connections をクリックします。これにより、Network Connections ウィンドウが開きます。このウィンドウは、通常のユーザーとして実行して開くこともできます。

```
~]$ nm-connection-editor &
```

矢印ヘッドをクリックして、必要に応じて接続の一覧を表示し、非表示にすることができます。

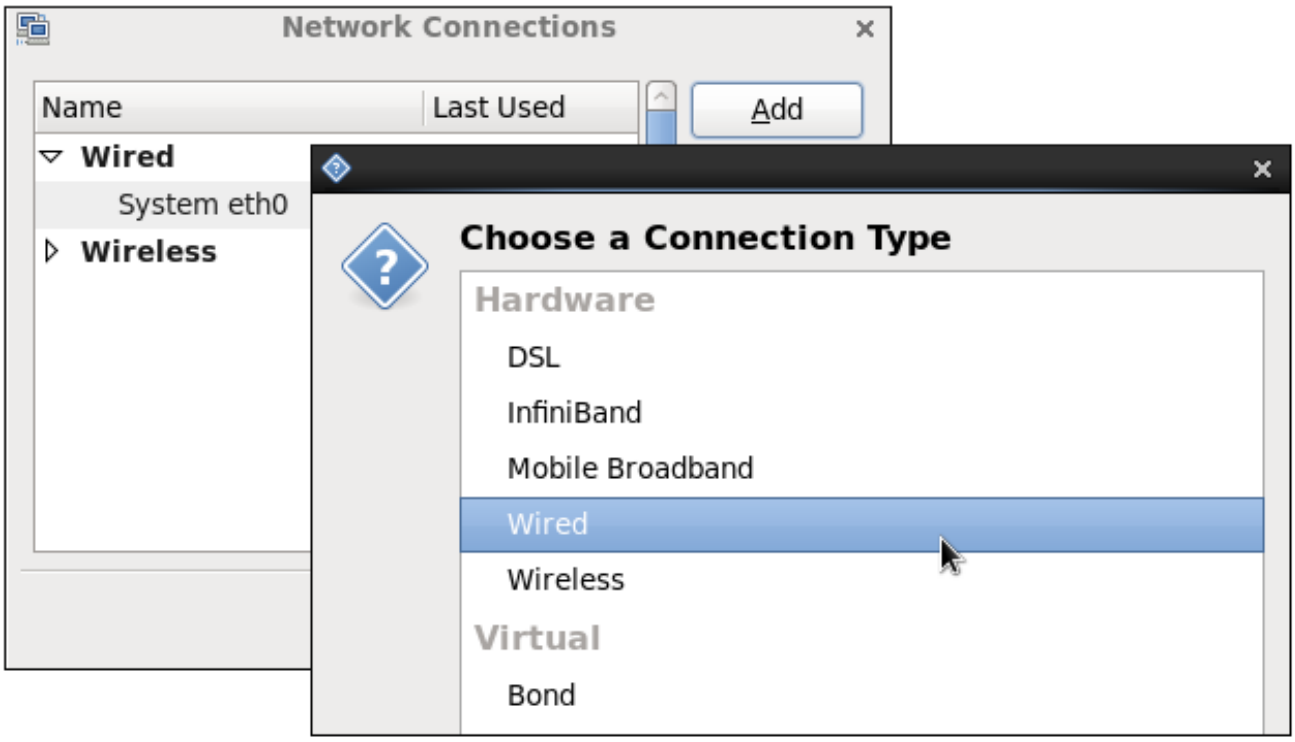
図10.6 新規作成したシステム eth0 接続を示す Network Connections ウィンドウ



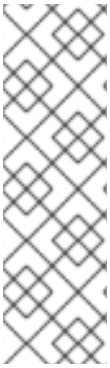
[D]

システム起動スクリプトは、デフォルトですべてのシステムで System eth0 という名前の単一の有線接続を作成して設定します。System eth0 を編集できますが、カスタム設定に新しい有線接続を作成することが推奨されます。新しい有線接続を作成するには、Add ボタンをクリックして、表示される一覧から Wired エントリーを選択してから Create ボタンをクリックします。

図10.7 「Choose a Connection Type」リストから新しい接続タイプの選択



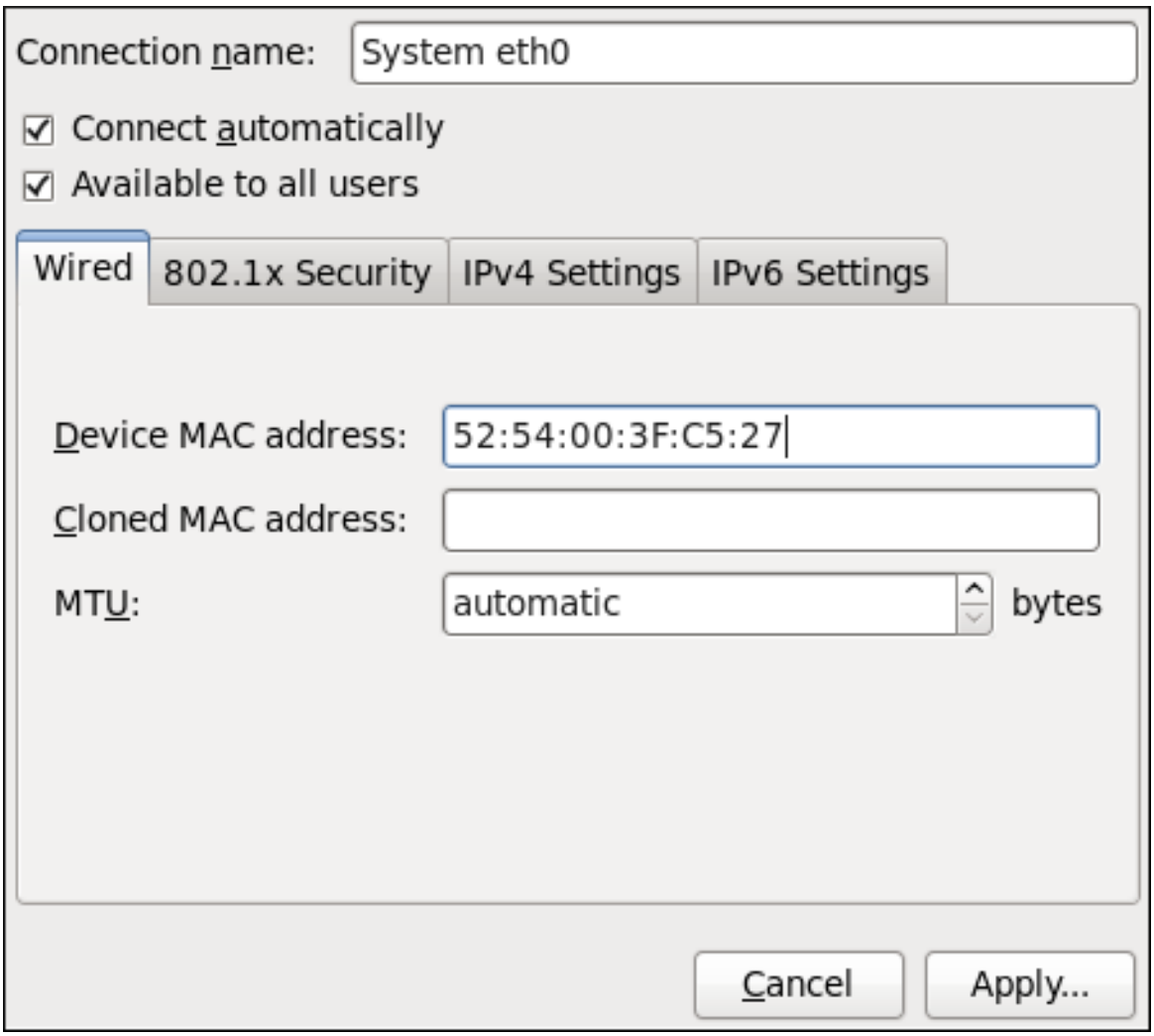
[D]



接続を追加および編集するダイアログは同じです。

Add ボタンをクリックして新しい接続を追加すると、接続タイプの一覧が表示されます。選択を作成して Create ボタンをクリックすると、NetworkManager により、その接続用の新しい設定ファイルが作成され、既存の接続の編集に使用するのと同じダイアログが表示されます。このダイアログには違いがありません。実際、接続は常に編集されます。相違点は、接続がすでに存在しているか、Create をクリックして NetworkManager が 作成 した場合にのみ、相違点になります。

図10.8 新規作成した Wired 接続システム eth0 の編集



[D]

接続名、自動接続の動作、可用性の設定

編集ダイアログの 3 つの設定は、すべての接続タイプに共通します。

- *接続名: ネットワーク接続の名前を入力します。この名前は、Network Connections ウィンドウの Wired セクションでこの接続を一覧表示するために使用されます。*

Connect automatically - NetworkManager が利用可能なときにこの接続に自動接続する場合は、このボックスにチェックを付けます。詳細は、「ネットワークの自動接続」 を参照してください。

- *すべてのユーザーが使用 - このボックスにチェックを入れて、システム上のすべてのユーザーが利用可能な接続を作成します。この設定を変更するには、root 権限が必要になる場合があります。詳しくは、「ユーザーおよびシステム接続」 を参照してください。*

Wired タブの設定

最後の 3 つの設定可能な設定は Wired タブ自体にあります。1 つ目は、MAC(Media Access Control)アドレスを指定でき、2 つ目はクローンされた MAC アドレスを指定でき、3 つ目は MTU(Maximum Transmission Unit)値を指定できます。通常、MAC アドレス フィールドを空白のままにし、MTU は automatic に設定できます。有線接続を 2 つまたは特定の NIC に関連付けるか、または高度なネットワークを実行していない限り、これらのデフォルト設定で十分です。このような場合は、以下の説明を参照してください。

MAC Address

ネットワークインターフェースカード(NIC)などのネットワークハードウェアには、システムを識別する一意の MAC アドレス（Media Access Control とも呼ばれる）があります。ip addr コマンドを実行すると、各インターフェースに関連付けられた MAC アドレスが表示されます。たとえば、以下の ip addr 出力では、eth0 インターフェースの MAC アドレス(52:54:00:26:9e:f1)は即座に link/ether キーワードに従います。

```
~]# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state
UNKNOWN qlen 1000
    link/ether 52:54:00:26:9e:f1 brd ff:ff:ff:ff:ff:ff
    inet 192.168.122.251/24 brd 192.168.122.255 scope global eth0
    inet6 fe80::5054:ff:fe26:9ef1/64 scope link
       valid_lft forever preferred_lft forever
```

1 つのシステムに 1 つ以上の NIC をインストールできます。したがって、MAC アドレス フィールドを使用すると、特定の NIC を特定の接続（または接続）に関連付けることができます。説明したように、ip addr コマンドを使用して MAC アドレスを判別し、その値を MAC アドレス の text-entry フィールドにコピーして貼り付けます。

このような状況で使用するクローンされた MAC アドレスフィールドは、ネットワークサービスが特定の MAC アドレスに制限されているため、その MAC アドレスをエミュレートする必要があります。

MTU

MTU(Maximum Transmission Unit)の値は、接続が送信に使用する最大パケットのサイズ（バイト単位）を表します。IPv6 に IPv4 または変数名 1500 以上を使用する場合、この値はデフォルトで 1280 に設定されるため、通常は指定したり変更したりする必要はありません。

新規 (または修正した) 接続を保存して他の設定を行う

有線接続の編集が終了したら、適用 ボタンをクリックすると、NetworkManager により、カスタマイズされた設定がすぐに保存されます。正しい設定がある場合は、NetworkManager Notification Area アプレットから新しい接続またはカスタマイズされた接続を選択して接続できます。新しい接続または変更された接続の使用に関する詳細は、「ネットワークへの接続」 を参照してください。

既存の接続をさらに設定をするには、ネットワーク接続 ウィンドウ内でその接続を選択し、編集 をクリックして 編集 ダイアログに戻ります。

そして、以下のいずれかの設定をします。

- *ポートベースのネットワークアクセス制御(PNAC)をクリックし、802.1X Security タブをクリックして 「802.1X セキュリティーの設定」 に進みます。*
- *接続の IPv4 設定、IPv4 設定 タブをクリックして、「IPv4 設定の構成」 に進んでください。*
- *接続の IPv6 設定、IPv6 設定 タブをクリックして、「IPv6 セッティングの設定」 に進みます。*

10.3.2. ワイヤレス接続の確立

本セクションでは、NetworkManager を使用して、アクセスポイントへのワイヤレス（Wi-Fi または 802.11a/b/g/n）接続を設定する方法を説明します。

モバイルブロードバンド（3G など）接続を設定するには、「モバイルブロードバンド接続の確立」 を参照してください。

利用可能なアクセスポイントへの迅速な接続

利用可能なアクセスポイントに接続する最も簡単な方法は、NetworkManager アプレットで、利用可能な ネットワーク一覧のアクセスポイントの Service Set Identifier (SSID)を特定することです。ア

クセスポイントが保護されると、認証を求めるダイアログが表示されます。

図10.9 ワイヤレスアクセスポイントへの認証

[D]

NetworkManager は、アクセスポイントが使用するセキュリティーの出塁を自動検出しようとします。複数の可能性がある場合、NetworkManager はセキュリティータイプを推測し、ワイヤレスセキュリティー ドロップダウンメニューで表示します。複数の選択肢があるかどうかを確認するには、ワイヤレスセキュリティー ドロップダウンメニューをクリックし、アクセスポイントが使用するセキュリティーのタイプを選択します。不明な場合は、各タイプに順次接続を試みます。最後に、パスワードフィールドにキーまたはパスフレーズを 入力 します。40 ビットの WEP キーまたは 128 ビットの WPA キーは、必要な長さがないと無効になります。選択したセキュリティーの種類に必要な分だけ鍵を入力するまで、Connect ボタンは無効になります。ワイヤレスセキュリティーの詳細は、「ワイヤレスセキュリティーの設定」 を参照してください。

同じアクセスポイント上でのリング防止

WPA および WPA2（Personal と Enterprise）の場合、Auto、WPA、および WPA2 間で選択するオプションが追加されました。このオプションは、WPA と WPA2 の両方を提供するアクセスポイントでの使用を目的としています。2 つのプロトコル間でローミングされないようにする場合は、いずれかのプロトコルを選択します。同じアクセスポイントの WPA と WPA2 をローミングすると、サービスが失われる可能性があります。

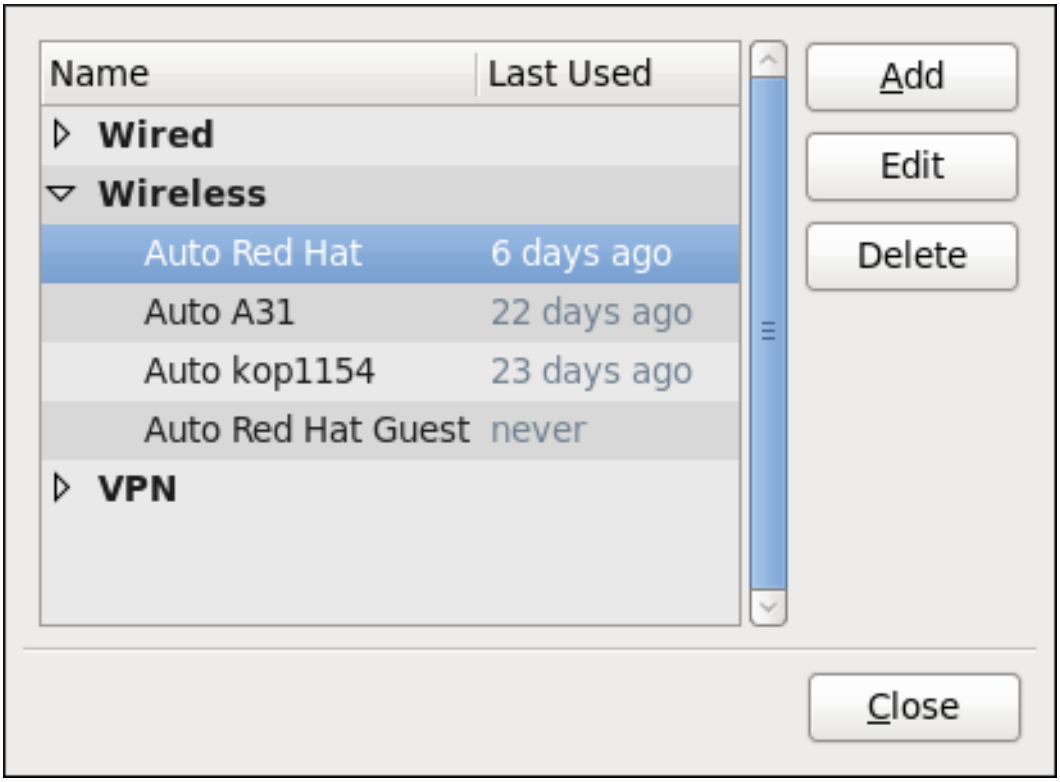
NetworkManager がアクセスポイントに正常に接続すると、アプレットアイコンがワイヤレス接続のシグナル強度のグラフィックインジケーターに変更されます。

図10.10 ワイヤレス接続シグナルの強度が 75% を示すアプレットアイコン

[D]

また、自動作成されたアクセスポイント接続の設定を、あたかも自分で追加したように編集することもできます。Network Connections ウィンドウの Wireless タブには、接続を試みた接続がすべて表示されます。NetworkManager の名前は Auto <SSID> です。SSID はアクセスポイント のサービスセット識別子 です。

図10.11 以前に接続されていたアクセスポイントの例



[D]

非表示のネットワークへの接続

すべてのアクセスポイントには、識別用の SSID(Service Set Identifier)があります。ただし、アクセスポイントはその SSID をブロードキャストしないように設定されていることがあります。この場合は非表示となり、NetworkManager の利用可能なネットワーク一覧には表示されなくなります。ただし、その SSID と認証方法と秘密情報が分かっていれば、SSID を非表示としているワイヤレスアクセスポイントに接続することは可能です。

非表示のワイヤレスネットワークに接続するには、NetworkManager のアプレットアイコンの左側にある Connect to Hidden Wireless Network を選択するとダイアログが表示されます。非表示のネットワークに接続している場合は、Connection ドロップダウンを使用して選択し、Connect をクリックします。作成していない場合は、Connection ドロップダウンを New のままにし、非表示ネットワー

クの SSID を入力し、ワイヤレスセキュリティー メソッドを選択して正しい認証シークレットを入力して Connect をクリックします。

ワイヤレスセキュリティーの設定に関する詳細は、「ワイヤレスセキュリティーの設定」 を参照してください。

接続の編集または新規作成

ネットワーク 接続を試みた、または過去に接続した既存の接続を編集するには、ネットワーク接続の ワイヤレス タブを開き、名前で接続を選択します（ Auto に続くキーワードはアクセスポイントの SSID を参照）、Edit をクリックします。

Network Connections ウィンドウを開き、Add ボタンをクリックして ワイヤレス を選択し、Create ボタンをクリックします。

1. Notification Area の NetworkManager アプレットアイコンを右クリックし、Edit Connections をクリックします。ネットワーク接続 ウィンドウが表示されます。
2. Add ボタンをクリックします。
3. 一覧から Wireless エントリーを選択します。
4. 作成 ボタンをクリックします。

図10.12 新規作成されたワイヤレス接続 1 の編集

[D]

接続名、自動接続の動作、可用性の設定

編集ダイアログの 3 つの設定は、すべての接続タイプに共通します。

- *接続名: ネットワーク接続の名前を入力します。この名前は、Network Connections ウィンドウの Wireless セクションでこの接続を一覧表示するために使用されます。デフォルトでは、ワイヤレス接続の名前はワイヤレスアクセスポイントの SSID と同じです。接続機能に影響を与えることなくワイヤレス接続の名前を変更できますが、SSID 名を保持することが推奨されます。*

- *Connect automatically - NetworkManager が利用可能なときにこの接続に自動接続する場合は、このボックスにチェックを付けます。詳細は、「ネットワークの自動接続」 を参照し*

てください。

- すべてのユーザーが使用 - このボックスにチェックを入れて、システム上のすべてのユーザーが利用可能な接続を作成します。この設定を変更するには、root 権限が必要になる場合があります。詳しくは、「ユーザーおよびシステム接続」 を参照してください。

ワイヤレスタブの設定

SSID

すべてのアクセスポイントには、サービスセット識別子 で特定できます。ただし、アクセスポイントはその SSID をブロードキャストしないように設定されていることがあります。この場合は非表示となり、NetworkManager の利用可能なネットワーク一覧には表示されなくなります。SSID（および認証シークレット）を知っている限り、SSID を隠しているワイヤレスアクセスポイントに接続できます。

非表示のワイヤレスネットワークへの接続方法は、「非表示のネットワークへの接続」 を参照してください。

モード

infrastructure: 専用のワイヤレスアクセスポイントに接続する場合、またはルーターやスイッチなどのネットワークデバイスに構築されたインフラストラクチャーの場合は、Mode を Infrastructure に設定します。

アドホック： 2 つ以上のモバイルデバイス間で直接通信するようにピアツーピアネットワークを作成する場合は、Mode を Ad-hoc に設定します。802.11 標準で Independent Basic Service Set (IBSS)と呼ばれる アドホック モードを使用する場合は、参加するすべてのワイヤレスデバイスに同じ SSID を設定し、それらすべてが同じチャンネルで通信されるようにする必要があります。

BSSID

Basic Service Set Identifier(BSSID)は、インフラストラクチャー モードで接続する特定のワイヤレスアクセスポイントの MAC アドレスです。このフィールドはデフォルトで空白になっており、BSSID を指定せずに SSID でワイヤレスアクセスポイントに接続できます。BSSID を指定している場合は、システムによる特定のアクセスポイントのみへの関連付けが強制的に実行されます。

アドホックネットワークが作成されると、BSSID は、アドホックネットワークの作成時に mac80211 サブシステムによって無作為に生成されます。これは NetworkManager では表示されません。

MAC アドレス

イーサネットネットワークインターフェースカード(NIC)と同様に、ワイヤレスアダプターには、システムを識別する一意の MAC アドレス（Media Access Control、または ハードウェアアドレス）があります。ip addr コマンドを実行すると、各インターフェースに関連付けられた MAC アドレスが表示されます。たとえば、以下の ip addr 出力では、wlan0 インターフェースの MAC アドレス（ 00:1c:bf:02:f8:70は link/ether キーワードの直後）に従います。

```
~]# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state
UNKNOWN qlen 1000
    link/ether 52:54:00:26:9e:f1 brd ff:ff:ff:ff:ff:ff
    inet 192.168.122.251/24 brd 192.168.122.255 scope global eth0
    inet6 fe80::5054:ff:fe26:9ef1/64 scope link
       valid_lft forever preferred_lft forever
3: wlan0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP qlen 1000
    link/ether 00:1c:bf:02:f8:70 brd ff:ff:ff:ff:ff:ff
    inet 10.200.130.67/24 brd 10.200.130.255 scope global wlan0
    inet6 fe80::21c:bfff:fe02:f870/64 scope link
       valid_lft forever preferred_lft forever
```

単一システムには、1 つまたは複数のワイヤレスネットワークアダプターを接続することができます。そのため、MAC アドレス フィールドで、特定のワイヤレスアダプターと特定の接続 (単一または複数) の関連付けを可能にしています。説明したように、ip addr コマンドを使用して MAC アドレスを判別し、その値を MAC アドレス の text-entry フィールドにコピーして貼り付けます。

MTU

MTU(Maximum Transmission Unit)の値は、接続が送信に使用する最大パケットのサイズ（バイト単位）を表します。 ゼロ以外の数値を設定すると、指定されたサイズのパケットまたは小さいパケットのみが送信されます。より大きなパケットは複数のイーサネットフレームに分割されます。この設定は、自動 のままにすることが推奨されます。

新規 (または修正した) 接続を保存して他の設定を行う

ワイヤレス接続の編集が終了したら、適用 ボタンをクリックすると、NetworkManager により、カスタマイズされた設定がすぐに保存されます。正しい設定がある場合は、NetworkManager Notification Area アプレットからこれを選択して、修正した接続に正常に接続できます。ネットワークの選択および接続の詳細は、「ネットワークへの接続」 を参照してください。

既存の接続をさらに設定をするには、ネットワーク接続 ウィンドウ内でその接続を選択し、編集 をクリックして 編集 ダイアログに戻ります。

そして、以下のいずれかの設定をします。

- ワイヤレス接続のセキュリティー認証。ワイヤレスセキュリティー タブを クリックして、「ワイヤレスセキュリティーの設定」 に進みます。

- 接続の IPv4 設定、IPv4 設定 タブをクリックして、「IPv4 設定の構成」 に進んでください。

- 接続の IPv6 設定、IPv6 設定 タブをクリックして、「IPv6 セッティングの設定」 に進みます。

10.3.3. モバイルブロードバンド接続の確立

NetworkManager のモバイルブロードバンド接続機能を使用すると、以下の 2G と 3G のサービスに接続することができます。

- 2g - GPRS （一般パケット通信サービス）または EDGE （GSM Evolution 用のデータレートの強化）

- 3g - UMTS (Universal Mobile Telecommunications System)または HSPA (High Speed Packet Access)

接続を作成するには、使用中のシステムがすでに発見して認識しているモバイルブロードバンドのデバイス (モデム) をコンピューターが備えている必要があります。そのようなデバイスはコンピューターに内蔵されている場合 (多くのノートブックやネットブック) と、外付けまたは内蔵のハードウェアとして提供されている場合があります。たとえば、PC カードや USB モデム、ドングル、モデムとして機能する携帯電話などです。

手順10.3 新しいモバイルブロードバンド接続の追加

Network Connections ウィンドウを開き、Add をクリックして、一覧から Mobile Broadband を選択して、モバイルブロードバンド接続を設定できます。

1. Notification Area の NetworkManager アプレットアイコンを右クリックし、Edit Connections をクリックします。ネットワーク接続 ウィンドウが表示されます。

2. *Add ボタンをクリックして選択リストを開きます。Mobile Broadband を選択し、Create を クリックします。Set up a Mobile Broadband Connection Assistant が表示されます。*

3. *このモバイルブロードバンドデバイス用の接続を作成 の下で、その接続で使用する 2G または 3G に対応したデバイスを選択します。ドロップダウンメニューが非アクティブの場合、これはシステムがモバイルブロードバンドが可能なデバイスを検出できなかったことを示します。この状況では、キャンセル をクリックして、モバイルブロードバンドの機能を持ったデバイスが接続されており、それがコンピューターに認識されていることを確認してください。Forward ボタンをクリックします。*

4. *一覧からサービスプロバイダーが置かれている国を選択し、forward ボタンをクリックします。*

5. *一覧からプロバイダーを選択するか、手動で入力します。Forward ボタンをクリックします。*

6. *ドロップダウンメニューから支払い計画を選択し、Access Point Name (APN)が正しいことを確認します。Forward ボタンをクリックします。*

7. *設定を確認して、適用 ボタンをクリックします。*

8. の手順に従って、モバイルブロードバンド固有の設定を編集します。これには、の下の Mobile Broadband タブ の説明を参照し、モバイルブロードバンド固有の設定を編集します。

手順10.4 既存のモバイルブロードバンド接続を編集する

以下の手順に従って、既存のモバイルブロードバンド接続を編集します。

1. *Notification Area の NetworkManager アプレットアイコンを右クリックし、Edit Connections をクリックします。ネットワーク接続 ウィンドウが表示されます。*

2. *編集する接続を選択して、編集 ボタンをクリックします。*

3. *Mobile Broadband タブを選択します。*

4. *接続名、自動接続の動作、および可用性のセッティングを設定します。*

   編集ダイアログの 3 つの設定は、すべての接続タイプに共通します。

   - *接続名: ネットワーク接続の名前を入力します。この名前は、Network Connections ウィンドウの Mobile Broadband セクションでこの接続を一覧表示するために使用されます。*
   - *Connect automatically - NetworkManager が利用可能なときにこの接続に自動接続する場合は、このボックスにチェックを付けます。詳細は、「ネットワークの自動接続」を参照してください。*
   - *すべてのユーザーが使用 - このボックスにチェックを入れて、システム上のすべてのユーザーが利用可能な接続を作成します。この設定を変更するには、root 権限が必要になる場合があります。詳しくは、「ユーザーおよびシステム接続」を参照してください。*

5. の手順に従って、モバイルブロードバンド固有の設定を編集します。これには、の下の Mobile Broadband タブ の説明を参照し、モバイルブロードバンド固有の設定を編集します。

新規 (または修正した) 接続を保存して他の設定を行う

モバイルブロードバンド接続の編集が終了したら、適用 ボタンをクリックすると、NetworkManager により、カスタマイズされた設定がすぐに保存されます。正しい設定がある場合は、NetworkManager Notification Area アプレットから新しい接続またはカスタマイズされた接続を選択して接続できます。新しい接続または変更された接続の使用に関する詳細は、「ネットワークへの接続」を参照してください。

既存の接続をさらに設定をするには、ネットワーク接続 ウィンドウ内でその接続を選択し、編集 をクリックして 編集 ダイアログに戻ります。

そして、以下のいずれかの設定をします。

- *接続のポイントツーポイント設定をクリックして、PPP Settings タブをクリックし、「PPP (ポイントツーポイント) セッティングの設定」に進みます。*
- *接続の IPv4 設定、IPv4 設定 タブをクリックして、「IPv4 設定の構成」に進んでください。*

- 接続の IPv6 設定、IPv6 設定 タブをクリックして、「IPv6 セッティングの設定」 に進みます。

モバイルブロードバンドタブの設定

アシスタントを使用した新しいモバイルブロードバンド接続がすでに追加されている場合（指示については 手順10.3「新しいモバイルブロードバンド接続の追加」 を参照）、ホームページが利用できない場合に Mobile Broadband タブを編集してロミングを無効にするか、ネットワーク ID を割り当てるか、NetworkManager が接続を使用する際に特定のテクノロジー（3G または 2G など）を選択するように指示できます。

数値

GSM ベースのモバイルブロードバンドネットワークでの PPP 接続を確立するためにダイアルする番号です。このフィールドは、ブロードバンドデバイスの初期インストールの際に自動設定されている場合があります。通常、このフィールドは空白で残し、代わりに APN を記入します。

Username

ネットワークでの認証に使用するユーザー名を記入します。一部のプロバイダーは、ユーザー名を提供しないことや、ネットワーク接続の時点でユーザー名を受け付けたりすることがあります。

Password

ネットワークで認証に使用するパスワードを記入します。一部のプロバイダーはパスワードを提供しなかったり、またはすべてのパスワードを受け付けたりします。

APN

GSM ベースのネットワークとの接続を確立するために使用する Access Point Name (APN) を記入します。これは以下の項目を決定するので、正しい APN を記入することが重要になります。

- ネットワーク使用率についてユーザーが請求する方法、またはその両方
- ユーザーがインターネット、イントラネット、サブネットワークにアクセスできるかどうか。

ネットワーク ID

ネットワーク ID を記入すると、NetworkManager は強制的にデバイスが特定のネットワークのみに登録されるようにします。これにより、ローミングを直接に制御できない時に接続がローミングしないようにします。

Type

Any: デフォルト値の Any では、モデムが最速のネットワークを選択します。

3G (UMTS/HSPA): 接続が 3G ネットワーク技術のみを使用するように強制します。

2G (GPRS/EDGE): 接続が 2G ネットワーク技術のみを使用するように強制します。

Prefer 3G (UMTS/HSPA): 最初に HSPA または UMTS などの 3G 技術を使用した接続を試み、失敗した後にのみ GPRS または EDGE にフォールバックします。

Prefer 2G (GPRS/EDGE): 最初に GPRS または EDGE などの 2G 技術を使用した接続を試み、失敗した後にのみ HSPA または UMTS にフォールバックします。

ホームネットワークが使用できない場合にローミングを許可

ホームネットワークからローミングへの移行ではなく、NetworkManager が接続を終了するようにするには、このボックスからチェックを外します。これにより、ローミング料金を回避できます。ボックスにチェックが入っていると、NetworkManager はホームネットワークからローミングに、またはその逆に切り替えることで接続を維持しようとします。

PIN

デバイスの SIM (Subscriber Identity Module (購読者識別モジュール)) が PIN (Personal Identification Number (個人識別番号)) でロックされている場合は、その PIN を入力して NetworkManager がデバイスのロックを解除できるようにします。どんな目的でもデバイスの使用に PIN を必要とする場合は、NetworkManager は SIM をロック解除する必要があります。

10.3.4. VPN 接続の確立

暗号化された仮想プライベートネットワーク(VPN)を確立することで、ローカルエリアネットワーク(LAN)と、別のリモート LAN との間で安全に通信できます。VPN 接続を正常に確立した後、VPN ルーターまたはゲートウェイは送信したパケットに対して以下のアクションを実行します。

1. *ルーティングおよび認証目的で 認証ヘッダー を追加します。*

2. *パケットデータを暗号化します。*

3. *データを暗号化および処理手順を構成する Encapsulating Security Payload(ESP)で囲みます。*

受信側の VPN ルーターはヘッダー情報を開いてデータを暗号化解読し、それを目的地 (ネットワーク上のワークステーションまたは他のノード) に送信します。ネットワーク対ネットワークの接続を使用すると、ローカルネットワーク上の受信側ノードはすでに暗号化解読されいてすぐに処理ができる状態のパケットを受信します。したがって、ネットワーク間 VPN 接続の暗号化/復号化プロセスは、クライアントに対して透過的になります。

VPN は認証と暗号化で複数のレイヤーを使用するため、複数のリモートノードを統合してひとつのイントラネットとして作動させる上で安全かつ効果的な手段となります。

手順10.5 新しい VPN 接続の追加

1. *Network Connections ウィンドウを開き、Add ボタンをクリックして、新しい接続リストの VPN セクションから VPN の種類を選択すると、新しい VPN 接続を設定できます。*

2. *Notification Area の NetworkManager アプレットアイコンを右クリックし、Edit Connections をクリックします。ネットワーク接続 ウィンドウが表示されます。*

3. *Add ボタンをクリックします。*

4. *接続の種類の選択 が表示されます。*

5. 

VPN プラグインが必要です。

設定する VPN タイプに適した NetworkManager VPN プラグインがインストールされている必要があります（Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;LinuxRed Hat Enterprise Linux 6nbsp;6 に新しいパッケージをインストールする方法は 「パッケージのインストール」 を参照してください）。

適切なプラグインがインストールされてい ない 場合は、Choose a Connection Type リストの VPN セクションは表示されません。

6. *Choose a Connection Type リストから、接続しているゲートウェイの VPN プロトコルを選択します。一覧の選択に使用できる VPN プロトコルは、インストールされている NetworkManager VPN プラグインに対応します。たとえば、NetworkManager-openswan の場合、libreswan 用の NetworkManager VPN プラグインがインストールされていると、IPsec ベースの VPN は Choose a Connection Type 一覧から選択可能です。*

注記

Red Hat Enterprise Linux 6.8 では、openswan は libreswan で廃止されました。NetworkManager-openswan openswan と libreswan の両方をサポートするように変更されました。

正しいものを選択したら、Create ボタンを押します。

7. *Editing VPN Connection 1 ウインドウが表示されます。このウィンドウでは、ステップ 6 で選択した VPN 接続のタイプにカスタマイズされた設定が表示されます。*

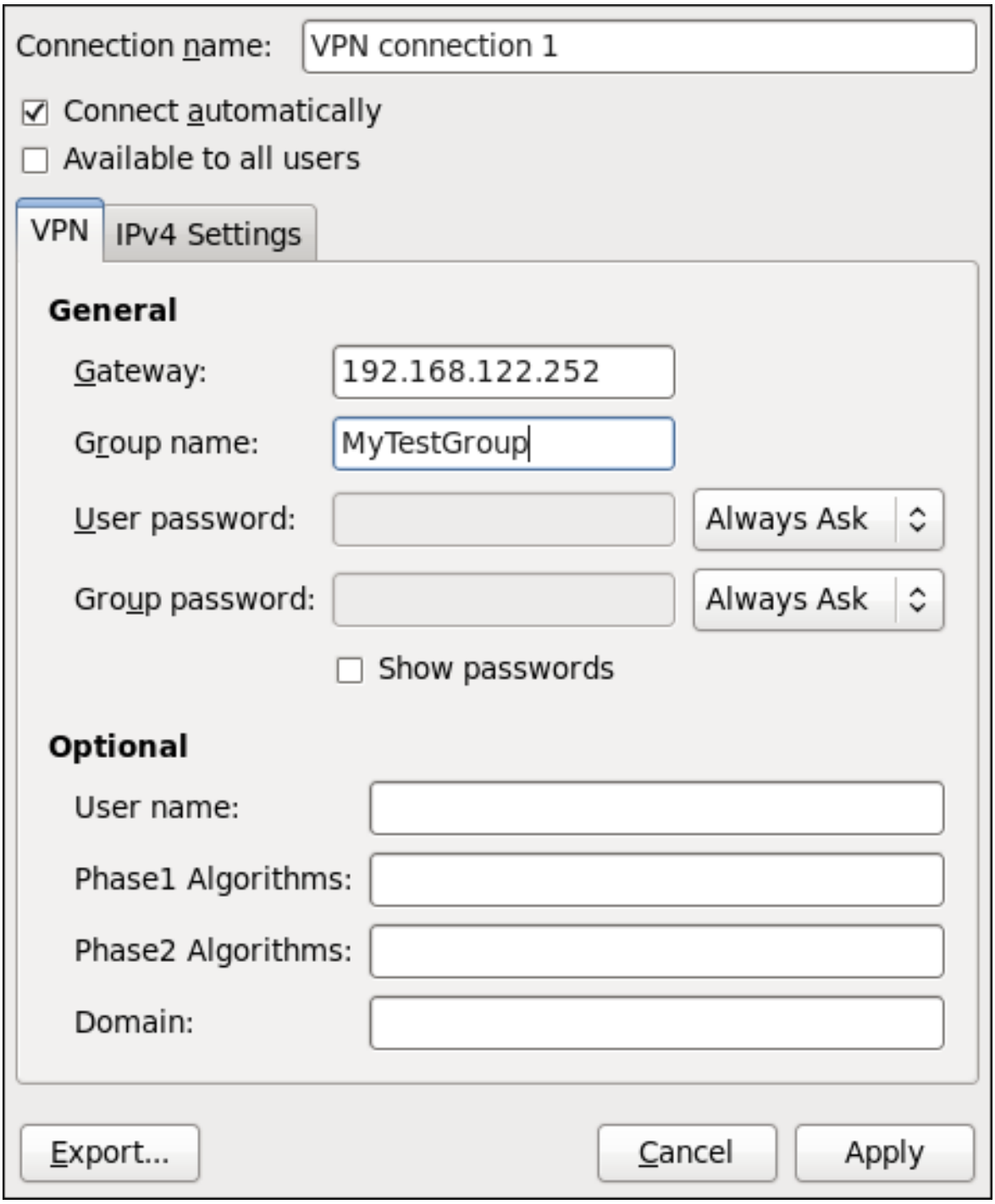
手順10.6 既存の VPN 接続を編集する

Network Connections ウィンドウを開き、一覧から接続の名前を選択して、既存の VPN 接続を設定できます。次に、編集 ボタンをクリックします。

1. *Notification Area の NetworkManager アプレットアイコンを右クリックし、Edit Connections をクリックします。ネットワーク接続 ウィンドウが表示されます。*

2. *編集する接続を選択して、編集 ボタンをクリックします。*

図10.13 新規作成された IPsec VPN 接続 1 の編集



[D]

接続名、自動接続の動作、可用性の設定

編集ダイアログの 3 つの設定は、すべての接続タイプに共通します。

- *接続名: ネットワーク接続の名前を入力します。この名前は、Network Connections ウィンドウの VPN セクションでこの接続を一覧表示するために使用されます。*

- *Connect automatically - NetworkManager が利用可能なときにこの接続に自動接続する*

場合は、このボックスにチェックを付けます。詳細は、「ネットワークの自動接続」 を参照してください。

- すべてのユーザーが使用 - このボックスにチェックを入れて、システム上のすべてのユーザーが利用可能な接続を作成します。この設定を変更するには、root 権限が必要になる場合があります。詳しくは、「ユーザーおよびシステム接続」 を参照してください。

VPN タブの設定

ゲートウェイ

リモート VPN ゲートウェイの名前または IP アドレス。

グループ名

リモートゲートウェイで設定された VPN グループ名です。

ユーザーパスワード

必要に応じて、VPN での認証に使用するパスワードを入力します。

グループパスワード

必要に応じて、VPN での認証に使用するパスワードを入力します。

ユーザー名

必要に応じて、VPN での認証に使用するユーザー名を入力します。

フェーズ1 アルゴリズム

必要な場合は、暗号化チャンネルの認証および設定で使用するアルゴリズムを入力します。

フェーズ2 アルゴリズム

必要な場合は、IPsec ネゴシエーションに使用するアルゴリズムを入力します。

ドメイン

必要な場合は、ドメイン名を入力します。

NAT トラバーサル

Cisco UDP（デフォルト）: UDP 経由の IPsec

NAT-T - ESP のカプセル化と IKE 拡張機能は、NAT トラバーサルを処理するために使用されます。

disabled: 特別な NAT 計測値は必要ありません。

Disable Dead Peer Detection - プローブのリモートゲートウェイまたはエンドポイントへの送信を無効にします。

新規 (または修正した) 接続を保存して他の設定を行う

新しい VPN 接続の編集が終了したら、適用 ボタンをクリックすると、NetworkManager により、カスタマイズされた設定がすぐに保存されます。正しい設定がある場合は、NetworkManager Notification Area アプレットから新しい接続またはカスタマイズされた接続を選択して接続できます。新しい接続または変更された接続の使用に関する詳細は、「ネットワークへの接続」 を参照してください。

既存の接続をさらに設定をするには、ネットワーク接続 ウィンドウ内でその接続を選択し、編集 をクリックして 編集 ダイアログに戻ります。

そして、以下のいずれかの設定をします。

- *接続の IPv4 設定、IPv4 設定 タブをクリックして、「IPv4 設定の構成」 に進みます。*

10.3.5. DSL 接続の確立

このセクションでは、個人ユーザーや SOHO インストールでよくある DSL モデムルーターの外部の組み合わせではなく、ホスト内に DSL カードが組み込まれているインストールについて説明します。

手順10.7 新しい DSL 接続の追加

Network Connections ウィンドウを開き、Add ボタンをクリックして、新しい接続リストの Hardware セクションから DSL を選択して、新しい DSL 接続を設定できます。

1. *Notification Area の NetworkManager アプレットアイコンを右クリックし、Edit Connections をクリックします。ネットワーク接続 ウィンドウが表示されます。*

2. *Add ボタンをクリックします。*

3. *接続の種類の選択 が表示されます。*

4. *DSL を選択し、Create ボタンを押します。*

5. *DSL 接続 1 の編集 ウィンドウが表示されます。*

手順10.8 既存の DSL 接続を編集する

Network Connections ウィンドウを開き、一覧から接続の名前を選択して、既存の DSL 接続を設定できます。次に、編集 ボタンをクリックします。

1. *Notification Area の NetworkManager アプレットアイコンを右クリックし、Edit Connections をクリックします。ネットワーク接続 ウィンドウが表示されます。*

2. *編集する接続を選択して、編集 ボタンをクリックします。*

接続名、自動接続の動作、可用性の設定

編集ダイアログの 3 つの設定は、すべての接続タイプに共通します。

- *接続名: ネットワーク接続の名前を入力します。この名前は、Network Connections ウィンドウの DSL セクションでこの接続を一覧表示するために使用されます。*

- *Connect automatically - NetworkManager が利用可能なときにこの接続に自動接続する場合は、このボックスにチェックを付けます。詳細は、「ネットワークの自動接続」 を参照してください。*

- *すべてのユーザーが使用 - このボックスにチェックを入れて、システム上のすべてのユー*

ザーが利用可能な接続を作成します。この設定を変更するには、root 権限が必要になる場合があります。詳しくは、「ユーザーおよびシステム接続」 を参照してください。

DSL タブの設定

Username

サービスプロバイダー認証で使用するユーザー名を入力します。

サービス

指示がない限り、空白のままにします。

パスワード

サービスプロバイダーから提供されたパスワードを入力します。

新規 (または修正した) 接続を保存して他の設定を行う

DSL 接続の編集が完了したら、適用 ボタンをクリックすると、NetworkManager により、カスタマイズされた設定がすぐに保存されます。正しい設定がある場合は、NetworkManager Notification Area アプレットから新しい接続またはカスタマイズされた接続を選択して接続できます。新しい接続または変更された接続の使用に関する詳細は、「ネットワークへの接続」 を参照してください。

既存の接続をさらに設定をするには、ネットワーク接続 ウィンドウ内でその接続を選択し、編集 をクリックして 編集 ダイアログに戻ります。

そして、以下のいずれかの設定をします。

- MAC アドレスと MTU の設定で Wired タブをクリックして、「Wired タブの設定」 に進みます。
- 接続のポイントツーポイント設定をクリックして、PPP Settings タブをクリックし、「PPP (ポイントツーポイント) セッティングの設定」 に進みます。
- 接続の IPv4 設定、IPv4 設定 タブをクリックして、「IPv4 設定の構成」 に進みます。

10.3.6. ボンド接続の確立

NetworkManager を使用して、2 つ以上の Wired 接続または Infiniband 接続からボンドを作成できます。最初にボンディングする接続を作成する必要はありません。それは、ボンディングを構成するためのプロセスで設定できます。設定プロセスを完了するには、利用可能なインターフェースの MAC アドレスが必要です。

注記

ボンディングに対する NetworkManager のサポートは、NMI _BOND_VLAN_ENABLED ディレクティブで有効にしてから、NetworkManager を再起動する必要があります。NM_CONTROLLED および NM_BOND_VLAN_ENABLED ディレクティブの説明は、「イーサネットインターフェース」 を参照してください。コマンドラインから NetworkManager などのサービスを再起動する方法は、「サービスの再開」 を参照してください。また、グラフィカルツールの場合は、「Service 設定ユーティリティーの使用」 を参照してください。

手順10.9 新しいボンド接続の追加

Bond 接続を設定するには、Network Connections ウィンドウを開き、Add をクリックして、一覧から Bond を選択します。

1. *Notification Area の NetworkManager アプレットアイコンを右クリックし、Edit Connections をクリックします。ネットワーク接続 ウィンドウが表示されます。*

2. *Add ボタンをクリックして選択リストを開きます。Bond を選択し、Create をクリックします。ボンド接続 1 の編集 ウィンドウが表示されます。*

3. *Bond タブで 追加 をクリックし、このボンド接続で使用するインターフェースのタイプを選択します。作成 ボタンをクリックします。スレーブタイプを選択するダイアログが表示されるのは、最初のスレーブを作成する時のみです。その後は、すべてのスレーブに同じタイプが自動的に使われます。*

4. *bond0 スレーブ 1 の編集 ウィンドウが表示されます。ボンディングする最初のインターフェースの MAC アドレスを入力します。最初のスレーブの MAC アドレスがボンドインターフェース用の MAC アドレスとして使用されます。必要な場合は、ボンドの MAC アドレスとして使用するクローンした MAC アドレスを入力します。Apply ボタンをクリックします。*

5. *Authenticate ウインドウが表示されます。続行するには root パスワードを入力します。認証 ボタンをクリックします。*

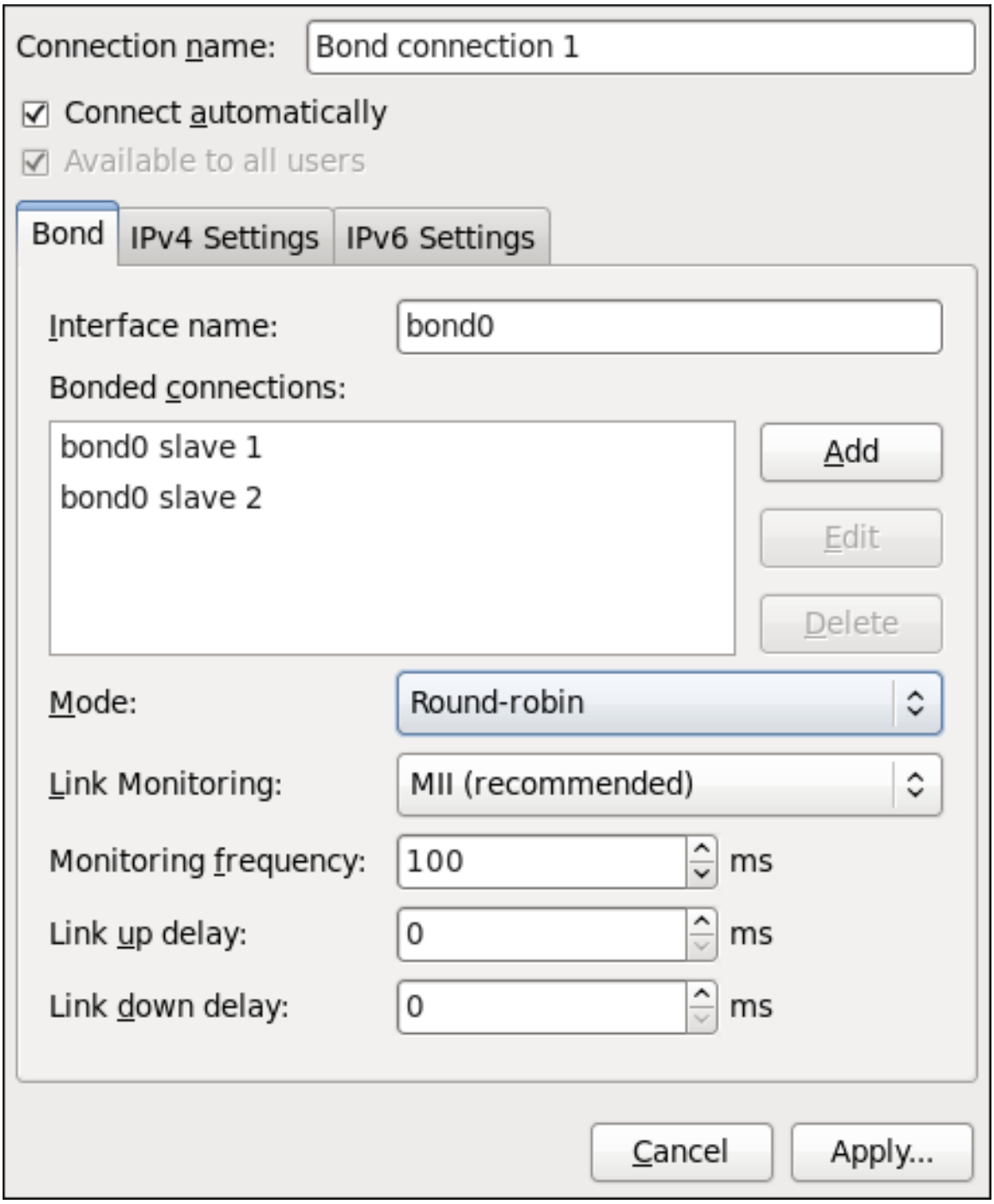
6.

ボンディングスレーブの名前が Bonded Connections ウィンドウ に表示されます。追加 ボタンをクリックしてさらにスレーブ接続を追加します。

7. *設定を確認して、適用 ボタンをクリックします。*

8. *以下の 「Bond タブの設定」 に移動して、ボンディング固有の設定を編集します。*

図10.14 新規作成したボンディング接続 1 の編集



[D]

手順10.10 既存のボンド接続を編集する

既存のボンド接続を編集するには以下のステップに従います。

1. *Notification Area の NetworkManager アプレットアイコンを右クリックし、Edit Connections をクリックします。ネットワーク接続 ウィンドウが表示されます。*

2. *編集する接続を選択して、編集 ボタンをクリックします。*

3. *Bond タブを選択します。*

4. *接続名、自動接続の動作、および可用性のセッティングを設定します。*

   編集ダイアログの 3 つの設定は、すべての接続タイプに共通します。

   - *接続名: ネットワーク接続の名前を入力します。この名前は、Network Connections ウィンドウの Bond セクションでこの接続を一覧表示するために使用されます。*

   - *Connect automatically - このチェックボックスを選択すると、NetworkManager が利用可能なときにこの接続に自動接続します。詳細は、「ネットワークの自動接続」 を参照してください。*

   - *Available to all users - このボックスを選択すると、システムですべてのユーザーが利用できる接続が作成されます。この設定を変更するには、root 権限が必要になる場合があります。詳しくは、「ユーザーおよびシステム接続」 を参照してください。*

5. *以下の 「Bond タブの設定」 に移動して、ボンディング固有の設定を編集します。*

新規 (または修正した) 接続を保存して他の設定を行う

ボンディング接続の編集が完了したら、適用 ボタンをクリックして、カスタマイズされた設定を保存します。正しい設定がある場合は、NetworkManager Notification Area アプレットから新しい接続またはカスタマイズされた接続を選択して接続できます。新しい接続または変更された接続の使用に関する詳細は、「ネットワークへの接続」 を参照してください。

既存の接続をさらに設定をするには、ネットワーク接続 ウィンドウ内でその接続を選択し、編集 をクリックして 編集 ダイアログに戻ります。

そして、以下のいずれかの設定をします。

- 接続の IPv4 設定、IPv4 設定 タブをクリックして、「IPv4 設定の構成」 に進んでください。

- 接続の IPv6 設定、IPv6 設定 タブをクリックして、「IPv6 セッティングの設定」 に進みます。

Bond タブの設定

新しいボンディング接続がすでに追加されている場合（手順は 手順10.9「新しいボンド接続の追加」 を参照）、Bond タブで負荷共有モードと、スレーブ接続の失敗を検出するために使用するリンク監視の種別を設定できます。

モード

ボンドを構成するスレーブ接続でのトラフィック共有に使われるモード。デフォルトは、ラウンドロビン です。802.3ad などの他の負荷分散モードは、ドロップダウンリストから選択することができます。

リンク監視

ネットワークトラフィックを伝送するスレーブの能力を監視する方法。

以下の負荷分散モードが、モード のドロップダウンリストから選択できます。

ラウンドロビン

耐障害性とロードバランシングにラウンドロビンポリシーを設定します。利用可能な最初のインターフェースからそれぞれのボンディングされたスレーブインターフェースで送受信が順次行われます。このモードは、仮想マシンのブリッジの背後では追加のスイッチ設定がないと機能しない可能性があります。

アクティブバックアップ

耐障害性のためアクティブなバックアップポリシーを設定します。ボンディングインターフェースの中で最初に利用可能になったものから送受信が行われます。別のボンディングされたスレーブインターフェースは、アクティブなボンディングされたスレーブインターフェースが失敗し

た場合にのみ使用されます。これは、InfiniBand デバイスのボンドで利用可能な唯一のモードです。

XOR

XOR (排他的理論和) を設定します。送受信は選択されたハッシュポリシーに基づいて行われます。デフォルトでは、ハッシュはソースの XOR とスレーブインターフェース数による剰余で宛先 MAC アドレスを掛けて導き出します。このモードでは、宛先が特定のピアになっているトラフィックは常に同一インターフェースで送信されます。宛先は MAC アドレスで決められるので、この方法は同一リンクまたはローカルネットワーク上にあるピアが宛先のトラフィックに最適なものです。トラフィックが単一ルーターを通過する必要がある場合は、このトラフィックバランスのモードは最適ではなくなります。

ブロードキャスト

耐障害性にブロードキャストポリシーを設定します。すべての送信は、すべてのスレーブインターフェースで行われます。このモードは、仮想マシンのブリッジの背後では追加のスイッチ設定がないと機能しない可能性があります。

802.3ad

IEEE 802.3ad 動的リンクアグリゲーションのポリシーを設定します。同一の速度とデュプレックス設定を共有するアグリゲーショングループを作成します。アクティブなアグリゲーターのすべてのスレーブで送受信を行います。802.3ad に対応するネットワークスイッチが必要です。

適応送信のロードバランシング

耐障害性とロードバランシングのための適応型送信ロードバランシング (TLB) ポリシーを設定します。発信トラフィックは、各スレーブインターフェースの現在の負荷に従って分散されます。受信トラフィックは、現在のスレーブにより受信されます。受信しているスレーブが失敗すると、別のスレーブが失敗したスレーブの MAC アドレスを引き継ぎます。このモードは、カーネルボンディングモジュールが認識しているローカルアドレスにのみ、適したものになります。このため、仮想マシンのブリッジの背後では使用できません。

適応ロードバランス

耐障害性とロードバランシングに適応型ロードバランシング (ALB) ポリシーを設定します。IPv4 トラフィック用の送受信ロードバランシングが含まれます。ARP ネゴシエーションにより、受信ロードバランシングが可能です。このモードは、カーネルボンディングモジュールが認識しているローカルアドレスにのみ、適したものになります。このため、仮想マシンのブリッジの背後では使用できません。

以下のリンク監視のタイプは、リンク監視 ドロップダウンリストから選択できます。ボンディングされたインターフェースでどのチャンネルボンディングのモジュールパラメーターが最適な動作をする

かテストするとよいでしょう。

MII (Media Independent Interface)

インターフェースのキャリア波の状態を監視します。実行方法は、ドライバーへのクエリー、MII レジスターへの直接クエリー、ethtool を使ったデバイスへのクエリーがあります。利用可能な 3 つのオプションは以下のとおりです。

監視周期

ドライバーもしくはMII レジスターへのクエリーの間隔時間 (ミリ秒単位)

接続遅延

up とレポートされたリンクの使用を試みるまでの待機時間 (ミリ秒単位)。リンクが「up」とレポートされてからすぐに余計な ARP リクエストが失われた場合に、この遅延は使用できます。これが発生するのは、たとえばスイッチ初期化などの間です。

接断遅延

これまでアクティブだったリンクが「down」とレポートされた際に、別のリンクに変更するまでの待ち時間 (ミリ秒単位)。アタッチされたスイッチがバックアップモードに変更するまで比較的長い時間がかかる場合に、この遅延は使用できます。

ARP

アドレス解決プロトコル (ARP) は、1 つ以上のピアにプローブしてリンク層接続の動作具合を判断するために使用されます。これは、送信開始時間および最終受信時間を提供しているデバイスドライバーに依存しています。

以下の 2 つのオプションがあります。

監視周期

ARP リクエストを送信する間隔時間 (ミリ秒単位)。

ARP ターゲット

ARP リクエスト送信先の IP アドレスのコンマ区切り。

10.3.7. VLAN 接続の確立

NetworkManager を使用して、既存のインターフェースを使用して VLAN を作成できます。現在、作成時には、イーサネットデバイスの VLAN のみを作成できます。

手順10.11 新しい VLAN 接続の追加

VLAN 接続を設定するには、Network Connections ウィンドウを開き、Add をクリックして、一覧から VLAN を選択します。

1. Notification Area の NetworkManager アプレットアイコンを右クリックし、Edit Connections をクリックします。ネットワーク接続 ウィンドウが表示されます。

2. Add ボタンをクリックして選択リストを開きます。VLAN を選択し、Create をクリックします。VLAN Connection 1の 編集 ウィンドウが表示されます。

3. VLAN タブで、VLAN 接続に使用する親インターフェースをドロップダウンリストから選びます。

4. VLAN ID を入力します。

5. VLAN インターフェース名を入力します。これは、作成される VLAN インターフェースの名前です。たとえば、「eth0.1」または「vlan2"」などです。通常、これは親インターフェース名に「.」および VLAN ID を加えたか、または VLAN ID を加えた「vlan」のいずれかです。

6. 設定を確認して、適用 ボタンをクリックします。

7. の VLANタブの説明を参照して、VLAN 固有の設定を 編集します。

手順10.12 既存の VLAN 接続を編集する

既存の VLAN 接続を編集するには以下の手順に従います。

1. Notification Area の NetworkManager アプレットアイコンを右クリックし、Edit Connections をクリックします。ネットワーク接続 ウィンドウが表示されます。

2. 編集する接続を選択して、編集 ボタンをクリックします。

3. VLAN タブを選択します。

4. 接続名、自動接続の動作、および可用性のセッティングを設定します。

   編集ダイアログの 3 つの設定は、すべての接続タイプに共通します。

   - 接続名: ネットワーク接続の名前を入力します。この名前は、ネットワーク 接続 ウィンドウの VLAN セクションでこの接続を一覧表示するために使用されます。
   - Connect automatically - NetworkManager が利用可能なときにこの接続に自動接続する場合は、このボックスにチェックを付けます。詳細は、「ネットワークの自動接続」を参照してください。
   - すべてのユーザーが使用 - このボックスにチェックを入れて、システム上のすべてのユーザーが利用可能な接続を作成します。この設定を変更するには、root 権限が必要になる場合があります。詳しくは、「ユーザーおよびシステム接続」 を参照してください。

5. の VLAN タブの説明を参照して、VLAN 固有の設定を 編集します。

新規 (または修正した) 接続を保存して他の設定を行う

VLAN 接続の編集が終了したら、適用 ボタンをクリックすると、NetworkManager により、カスタマイズされた設定がすぐに保存されます。正しい設定がある場合は、NetworkManager Notification Area アプレットから新しい接続またはカスタマイズされた接続を選択して接続できます。新しい接続または変更された接続の使用に関する詳細は、「ネットワークへの接続」 を参照してください。

既存の接続をさらに設定をするには、ネットワーク接続 ウィンドウ内でその接続を選択し、編集 をクリックして 編集 ダイアログに戻ります。

そして、以下のいずれかの設定をします。

- 接続の IPv4 設定、IPv4 設定 タブをクリックして、「IPv4 設定の構成」 に進みます。

VLAN タブの設定

新しい VLAN 接続がすでに追加されている場合（手順は 手順10.11「新しい VLAN 接続の追加」 を参照）、VLAN タブを編集して親インターフェースと VLAN ID を設定できます。

親インターフェース

ドロップダウンリストから以前に設定したインターフェースを選択できます。

VLAN ID

VLAN ネットワークのトラフィックのタグ付けに使用する ID 番号。

VLAN インターフェース名

作成される VLAN インターフェースの名前。たとえば、「eth0.1」または「vlan2"」などです。

クローンした MAC アドレス

VLAN インターフェースの特定に使用する別の MAC アドレスをオプションで設定します。このアドレスを使って、この VLAN 上で送信されたパケットのソース MAC アドレスを変更することができます。

MTU

VLAN 接続で送信されるパケットに使用する最大転送単位 (MTU) のサイズをオプションで設定します。

10.3.8. IPoIB(IP-over-InfiniBand)接続の確立

NetworkManager を使用して InfiniBand 接続を作成できます。

手順10.13 新しい InfiniBand 接続の追加

Network Connections ウィンドウを開き、Add をクリックして、一覧から InfiniBand 接続を選択すると、InfiniBand 接続を設定できます。

1. *Notification Area の NetworkManager アプレットアイコンを右クリックし、Edit Connections をクリックします。ネットワーク接続 ウィンドウが表示されます。*

2. *Add ボタンをクリックして選択リストを開きます。InfiniBand を選択し、Create をクリックします。Editing InfiniBand Connection 1 ウインドウが表示されます。*

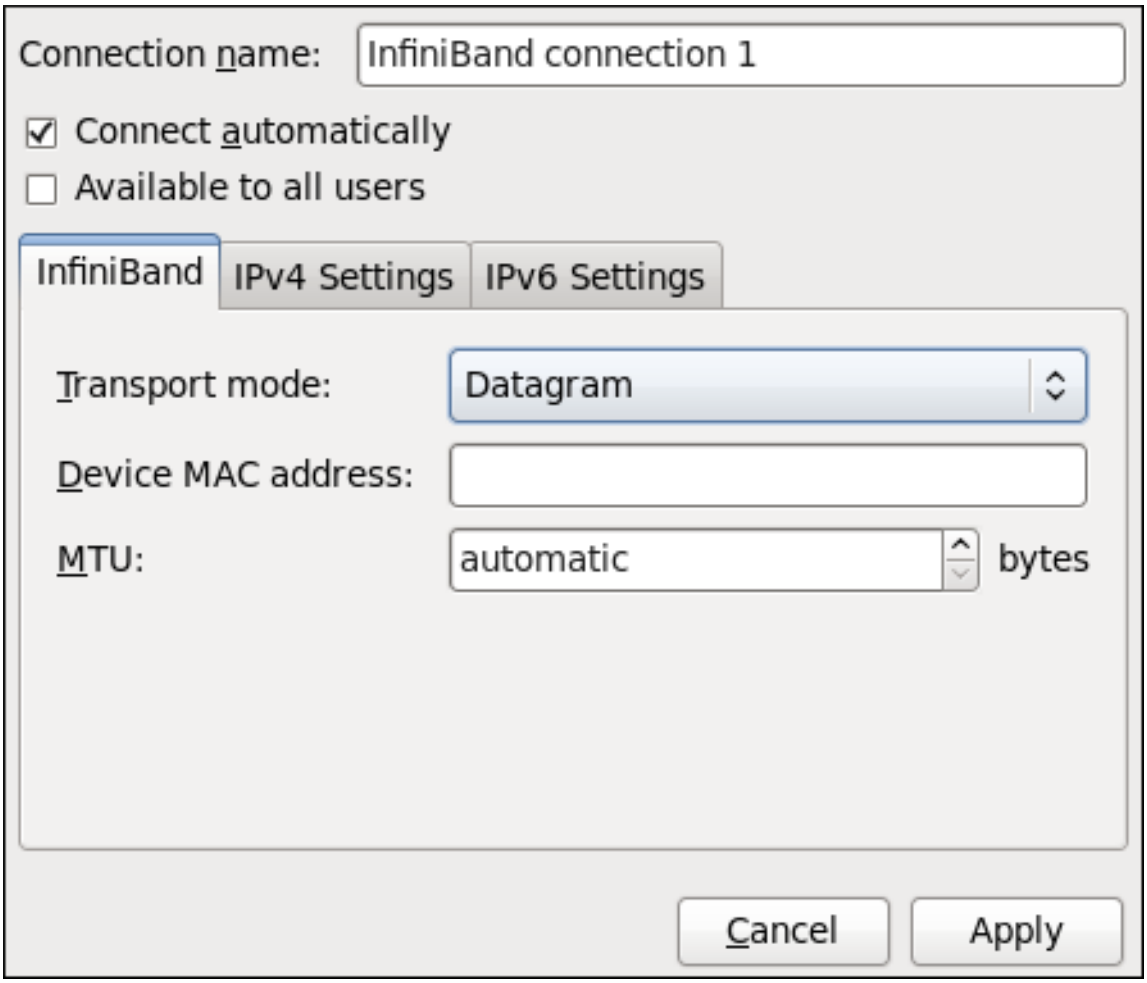
3. *InfiniBand タブで、InfiniBand 接続に使用するトランスポートモードをドロップダウンリストから選びます。*

4. *InfiniBand MAC アドレスを入力します。*

5. *設定を確認して、適用 ボタンをクリックします。*

6. の手順に従って、InfiniBand タブの設定を参照して、InfiniBand 固有の設定を *編集します。*

図10.15 新規作成した InfiniBand 接続 1 の編集



[D]

手順10.14 既存の InfiniBand 接続を編集する

既存の InfiniBand 接続を編集するには以下の手順に従います。

1. *Notification Area の NetworkManager アプレットアイコンを右クリックし、Edit Connections をクリックします。ネットワーク接続 ウィンドウが表示されます。*
2. *編集する接続を選択して、編集 ボタンをクリックします。*
3. *InfiniBand タブを選択します。*
4. *接続名、自動接続の動作、および可用性のセッティングを設定します。*

編集ダイアログの 3 つの設定は、すべての接続タイプに共通します。

- 接続名: ネットワーク接続の名前を入力します。この名前は、Network Connections ウィンドウの InfiniBand セクションでこの接続を一覧表示するために使用されます。

- Connect automatically - NetworkManager が利用可能なときにこの接続に自動接続する場合は、このボックスにチェックを付けます。詳細は、「ネットワークの自動接続」を参照してください。

- すべてのユーザーが使用 - このボックスにチェックを入れて、システム上のすべてのユーザーが利用可能な接続を作成します。この設定を変更するには、root 権限が必要になる場合があります。詳しくは、「ユーザーおよびシステム接続」 を参照してください。

5. の手順に従って、InfiniBand タブの設定を参照して、InfiniBand 固有の設定を 編集します。

新規 (または修正した) 接続を保存して他の設定を行う

InfiniBand 接続の編集が終了したら、適用 ボタンをクリックすると、NetworkManager により、カスタマイズされた設定がすぐに保存されます。正しい設定がある場合は、NetworkManager Notification Area アプレットから新しい接続またはカスタマイズされた接続を選択して接続できます。新しい接続または変更された接続の使用に関する詳細は、「ネットワークへの接続」 を参照してください。

既存の接続をさらに設定をするには、ネットワーク接続 ウィンドウ内でその接続を選択し、編集 をクリックして 編集 ダイアログに戻ります。

そして、以下のいずれかの設定をします。

- 接続の IPv4 設定、IPv4 設定 タブをクリックして、「IPv4 設定の構成」 に進んでください。

- 接続の IPv6 設定、IPv6 設定 タブをクリックして、「IPv6 セッティングの設定」 に進みます。

InfiniBand タブの設定

新しい InfiniBand 接続がすでに追加されている場合（手順は 手順10.13「新しい InfiniBand 接続の

追加」 を参照）、InfiniBand タブを編集して親インターフェースと InfiniBand ID を設定できます。

トランスポートモード

ドロップダウンリストから、Datagram または Connected モードを選択できます。他の IPoIB ネットワークで使用しているモードと同じものを選びます。

Device MAC アドレス

InfiniBand ネットワークのトラフィックで使用される InfiniBand 対応デバイスの MAC アドレスです。InfiniBand ハードウェアがインストールされていれば、このハードウェアのアドレスフィールドは事前に記入されます。

MTU

InfiniBand 接続で送信されるパケットに使用する最大転送単位 (MTU) のサイズをオプションで設定します。

10.3.9. 接続設定の構成

10.3.9.1. 802.1X セキュリティーの設定

802.1X セキュリティーは、ポートベースのネットワークアクセス制御(PNAC)用の IEEE 標準の名前です。簡単に言うと、802.1X セキュリティーは、物理ネットワークから 論理ネットワーク を定義する方法です。論理ネットワークに参加するクライアントはすべて、正しい 802.1X 認証方法を使用して、ルーターなどのサーバーで認証を行う必要があります。

802.1X セキュリティーは、ほとんどの場合、ワイヤレスネットワーク (WLAN) のセキュリティー保護に関連付けられていますが、ネットワーク (LAN) に物理的にアクセスする侵入者が侵入するのを防ぐためにも使用できます。以前は、DHCP サーバーは、権限のないユーザーに IP アドレスをリースしないように設定されていましたが、このプラクティスは実用的で安全でないため、推奨されなくなりました。代わりに、802.1X セキュリティーを使用して、ポートベースの認証を通じて、論理的に安全なネットワークを確保します。

802.1X は、WLAN と LAN のアクセス制御のためのフレームワークを提供して、EAP (Extensible Authentication Protocol) タイプの 1 つを運搬するエンベロープとして機能します。EAP タイプは、ネットワーク上で WLAN セキュリティーを達成する方法を定義するプロトコルです。

Network Connections ウィンドウ(「既存接続の設定および編集」)を開き、適用可能な手順に従って、有線またはワイヤレス接続タイプに 802.1X セキュリティーを設定できます。

手順10.15 有線接続の場合...

1. *Add をクリックして、802.1X セキュリティーを設定する新しいネットワーク接続を選択してから Create をクリックするか、既存の接続を選択して Edit をクリックします。*

2. *次に 802.1X Security タブを選択し、この接続に 802.1X セキュリティーを使用 して設定を有効にします。*

3. *次に進みます 「TLS(Transport Layer Security)の設定」*

手順10.16 ワイヤレス接続の場合...

1. *Add をクリックして、802.1X セキュリティーを設定する新しいネットワーク接続を選択してから Create をクリックするか、既存の接続を選択して Edit をクリックします。*

2. *ワイヤレスセキュリティー タブを 選択します。*

3. *次に Security ドロップダウンメニューをクリックし、LEAP、Dynamic WEP(802.1X)、または W PA & WPA2 Enterprise のセキュリティーメソッドのいずれかを選択します。*

4. *セキュリティードロップダウンでの選択に対応する EAP タイプの説明は、「TLS(Transport Layer Security)の設定」 を参照 し てください。*

10.3.9.1.1. TLS(Transport Layer Security)の設定

Transport Layer Security では、クライアントとサーバーは TLS プロトコルを使用して相互に認証します。サーバーはデジタル証明書を維持していることを示し、クライアントはクライアント側の証明書を使用して自身の ID を証明することで、キー情報が交換されます。認証が完了すると、TLS トンネルの使用は終了します。その代わりにクライアントとサーバーは交換したキーで、AES、TKIP、WEP のいずれかを使用してデータを暗号化します。

認証を希望する全クライアントに証明書が配布される必要があるということは、EAP-TLS 認証のメソッドが非常に強力であることを意味しますが、セットアップはより複雑になります。TLS セキュリティーを使用すると、証明書を管理する公開鍵インフラストラクチャー (PKI) のオーバーヘッドが必要になります。TLS セキュリティーを使用する利点は、パスワードが危険にさらされても (W)LAN へのアクセスが許可されないことです。侵入者は、認証するクライアントのプライベートキーにもアクセスを必要とします。

NetworkManager は、対応する TLS のバージョンを決定しません。NetworkManager は、ユーザーが入力するパラメーターを集め、手順を処理するデーモンである wpa_supplicant にこれらを渡します。このデーモンは、OpenSSL を使用して TLS トンネルを確立します。OpenSSL 自体は、SSL/TLS プロトコルバージョンを処理します。両端が対応する一番高いバージョンが使用されます。

アイデンティティー

ユーザー名やログイン名などの EAP 認証方法の ID 文字列。

ユーザー証明書

クリックしてユーザーの証明書を参照し、選択します。

CA 証明書

クリックしてブラウズし、認証局 (CA) の証明書を選択します。

秘密鍵

クリックして、ユーザーの秘密鍵ファイルを参照し、選択します。鍵はパスワードで保護される必要があることに注意してください。

秘密鍵のパスワード

ユーザーの秘密鍵に対応するユーザーパスワードを入力します。

10.3.9.1.2. Tunneled TLS の設定

Anonymous identity

この値は、非暗号化 ID として使用されます。

CA 証明書

クリックしてブラウズし、認証局 (CA) の証明書を選択します。

Inner authentication

PAP - パスワード認証プロトコル

MSCHAP - チャレンジ ハンドシェイク認証プロトコル

MSCHAPv2 - Microsoft チャレンジハンドシェイク認証プロトコルバージョン 2

CHAP - チャレンジ ハンドシェイク認証プロトコル

Username

認証プロセスで使用するユーザー名を入力します。

Password

認証プロセスで使用するパスワードを入力します。

10.3.9.1.3. Protected EAP (PEAP) の設定

Anonymous Identity

この値は、非暗号化 ID として使用されます。

CA 証明書

クリックしてブラウズし、認証局 (CA) の証明書を選択します。

PEAP version

使用する、保護された EAP のバージョン。Automatic、0、1 のいずれか。

Inner authentication

MSCHAPv2 - Microsoft チャレンジハンドシェイク認証プロトコルバージョン 2

MD5 - メッセージダイジェスト 5、暗号化ハッシュ関数。

GTC: Generic Token Card

Username

認証プロセスで使用するユーザー名を入力します。

Password

認証プロセスで使用するパスワードを入力します。

10.3.9.2. ワイヤレスセキュリティーの設定

セキュリティー

なし - Wi-Fi 接続を暗号化しません。

WEP 40/128-bit キー - IEEE 802.11 標準からの Wired Equivalent Privacy (WEP)。共有キー (PSK) を 1 つ使用します。

WEP 128-bit パスフレーズ - パスフレーズの MD5 ハッシュを使用して WEP キーを引き出します。

LEAP - Cisco Systems の Lightweight Extensible Authentication Protocol。

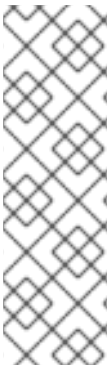
動的 WEP (802.1X) - WEP キーは動的に変更します。

WPA & WPA2 Personal - IEEE 802.11i 標準からの Wi-Fi Protected Access (WPA)。WEP の代替。802.11i-2004 規格の Wi-Fi Protected Access II (WPA2)。個人モードは、事前共有キー (WPA-PSK) を使用します。

WPA & WPA2 Enterprise - IEEE 802.1X ネットワークアクセス制御を提供するために RADIUS 認証サーバーとともに使用する WPA。

パスワード

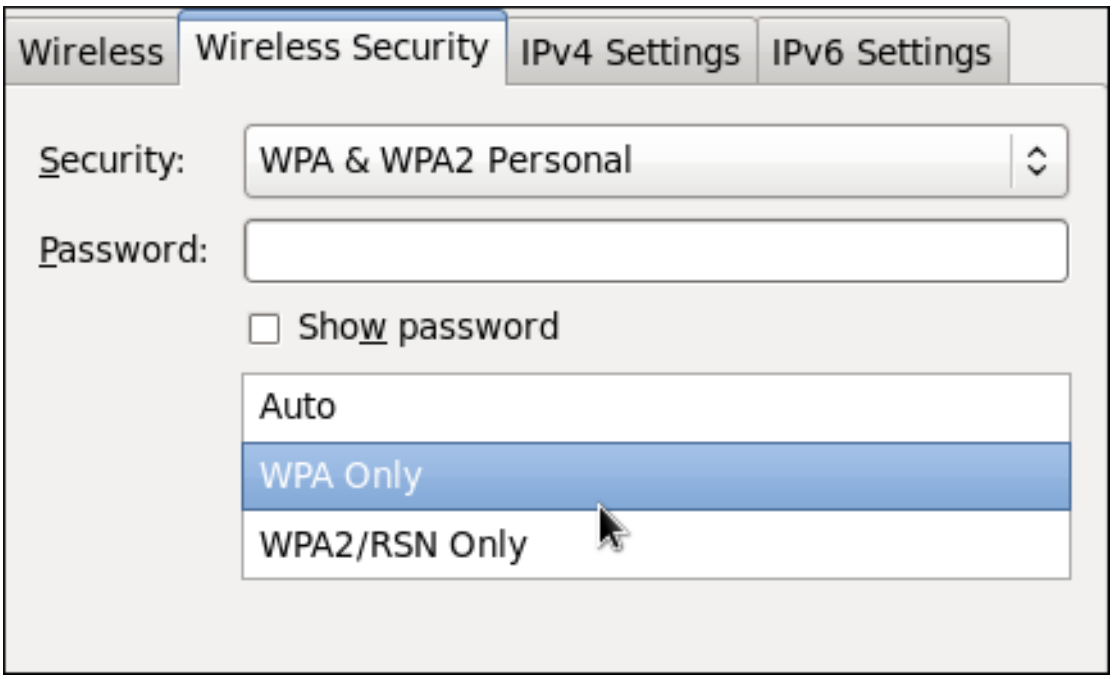
認証プロセスで使用するパスワードを入力します。



同じアクセスポイント上でのリング防止

WPA および WPA2（Personal と Enterprise）の場合、Auto、WPA、および WPA2 間で選択するオプションが追加されました。このオプションは、WPA と WPA2 の両方を提供するアクセスポイントでの使用を目的としています。2 つのプロトコル間でローミングされないようにする場合は、いずれかのプロトコルを選択します。同じアクセスポイントの WPA と WPA2 をローミングすると、サービスが失われる可能性があります。

図10.16 ワイヤレスセキュリティータブの編集および WPA プロトコルの選択



[D]

10.3.9.3. PPP (ポイントツーポイント) セッティングの設定

メソッドの設定

MPPE (ポイントツーポイント暗号化) を使用

Microsoft Point-To-Point Encryption Protocol(RFC 3078)

BSD データ圧縮を許可する

PP BSD 圧縮プロトコル(RFC 1977)

Deflate データ圧縮を許可する

PP デフレートプロトコル(RFC 1979)

TCP ヘッダー圧縮を使用

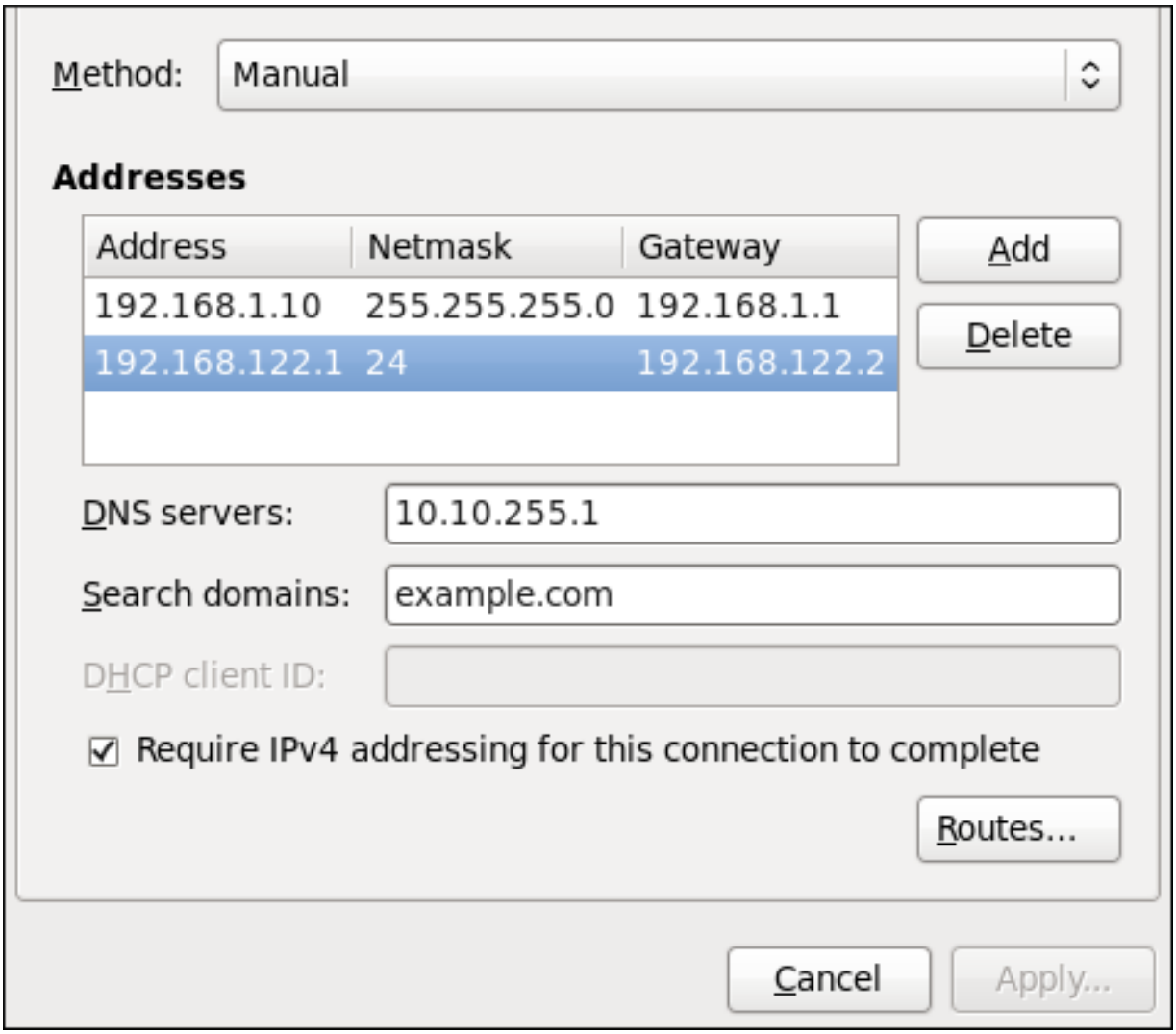
低速度シリアル番号(RFC 1144)の TCP/IP ヘッダーの圧縮。

PPP echo のパケットを送信

ループバックテスト用の LCP Echo-Request および Echo-Reply Code(RFC 1661)

10.3.9.4. IPv4 設定の構成

図10.17 IPv4 設定タブの編集



[D]

IPv4 設定 タブ では、インターネットに接続する方法を設定し、必要に応じて IP アドレス、ルート、および DNS 情報を入力します。IPv4 設定 タブは、有線、無線、モバイルブロードバンド、VPN、または DSL の接続タイプのいずれかを作成して変更する場合に利用できます。

DHCP を使用して DHCP サーバーから動的 IP アドレスを取得する場合は、方法 を Automatic(DHCP) に設定します。

メソッドの設定

接続の種類別で利用可能な IPv4 方式

設定する接続のタイプに応じて Method ドロップダウンメニューをクリックすると、以下の IPv4 接続方法のいずれかを選択できます。すべてのメソッドは、関連付けられている接続タイプまたはタイプに従ってここに一覧表示されます。

方法

Automatic(DHCP) - 接続しているネットワークが DHCP サーバーを使用して IP アドレスを割り当てる場合は、このオプションを選択します。DHCP クライアント ID フィールドの記入は必要ありません。

Automatic(DHCP)アドレスのみ - 接続しているネットワークが DHCP サーバーを使用して IP アドレスを割り当てる場合に、DNS サーバーを手動で割り当てる場合は、このオプションを選択します。

Link-Local Only - 接続しているネットワークに DHCP サーバーがなく、IP アドレスを手動で割り当てない場合は、このオプションを選択します。RFC 3927 に従ってランダムなアドレスが選択されます。

他のコンピューターへ共有 - 設定中のインターフェースがインターネットまたは WAN 接続の共有用である場合は、このオプションを選択します。

有線、ワイヤレス、DSL 接続の方式

Manual - 接続しているネットワークに DHCP サーバーがなく、IP アドレスを手動で割り当てる場合は、このオプションを選択します。

モバイルブロードバンド接続の方式

Automatic(PPP): 接続しているネットワークが DHCP サーバーを使用して IP アドレスを割り当てる場合は、このオプションを選択します。

Automatic(PPP)アドレスのみ - DHCP サーバーを使用して IP アドレスを割り当てるが、DNS サーバーを手動で割り当てる場合は、このオプションを選択します。

VPN 接続の方式

Automatic(VPN): 接続しているネットワークが DHCP サーバーを使用して IP アドレスを割り当てる場合は、このオプションを選択します。

Automatic(VPN)アドレスのみ - 接続しているネットワークが DHCP サーバーを使用して IP アドレスを割り当てる場合に、DNS サーバーを手動で割り当てる場合は、このオプションを選択します。

DSL 接続の方式

Automatic(PPPoE) - 接続しているネットワークが DHCP サーバーを使用して IP アドレスを割り当てる場合は、このオプションを選択します。

Automatic(PPPoE)アドレスのみ - DHCP サーバーを使用して IP アドレスを割り当てるが、DNS サーバーを手動で割り当てる場合は、このオプションを選択します。

PPPoE 固有の設定手順

複数の NIC がインストールされており、PPPoE が 1 つの NIC でのみ実行され、別の NIC しか実行しない場合は、正しい PPPoE 操作の場合には、特定のイーサネットデバイスの PPPoE への接続をロックする必要があります。特定の NIC への接続をロックするには、以下の いずれか を行います。

- *その接続の nm-connection-editor に MAC アドレスを入力します。必要に応じて、システムの起動時 にユーザーのログインを必要とせずに、自動的に Connect と Available をすべて のユーザーが接続して接続を確立します。*

- *以下のように、/etc/NetworkManager/system-connections/ でその接続に適切なファイルの [802-3-ethernet] セクションに hardware-address を設定します。*

  ```
  [802-3-ethernet]
  mac-address=00:11:22:33:44:55
  ```

  /etc/NetworkManager/system-connections/ にファイルが存在 「するのは、すべてのユーザーが使用でき」 ることを意味します。システムの起動後にユーザーログインを必要とせずに接続を起動する [connection] セクションに autoconnect=true が表示されていることを確認します。

ネットワーク接続に静的ルートを設定する方法は、「ルートの作成」 を参照してください。

10.3.9.5. IPv6 セッティングの設定

方法

ignore: IPv6 設定を無効にする場合は、このオプションを選択します。

Automatic - 接続しているネットワークが DHCP サーバーを使用して IP アドレスを割り当てる場合は、このオプションを選択します。

Automatic, addresses only - DHCP サーバーを使用して IP アドレスを割り当て、DNS サーバーを手動で割り当てる場合は、このオプションを選択します。

Manual - 接続しているネットワークに DHCP サーバーがなく、IP アドレスを手動で割り当てる場合は、このオプションを選択します。

Link-Local Only - 接続しているネットワークに DHCP サーバーがなく、IP アドレスを手動で割り当てない場合は、このオプションを選択します。RFC 4862 に従ってランダムなアドレスが選択されます。

他のコンピューターへ共有 - 設定中のインターフェースがインターネットまたは WAN 接続の共有用である場合は、このオプションを選択します。

アドレス

DNS サーバー: DNS サーバーのコンマ区切りの一覧を入力します。

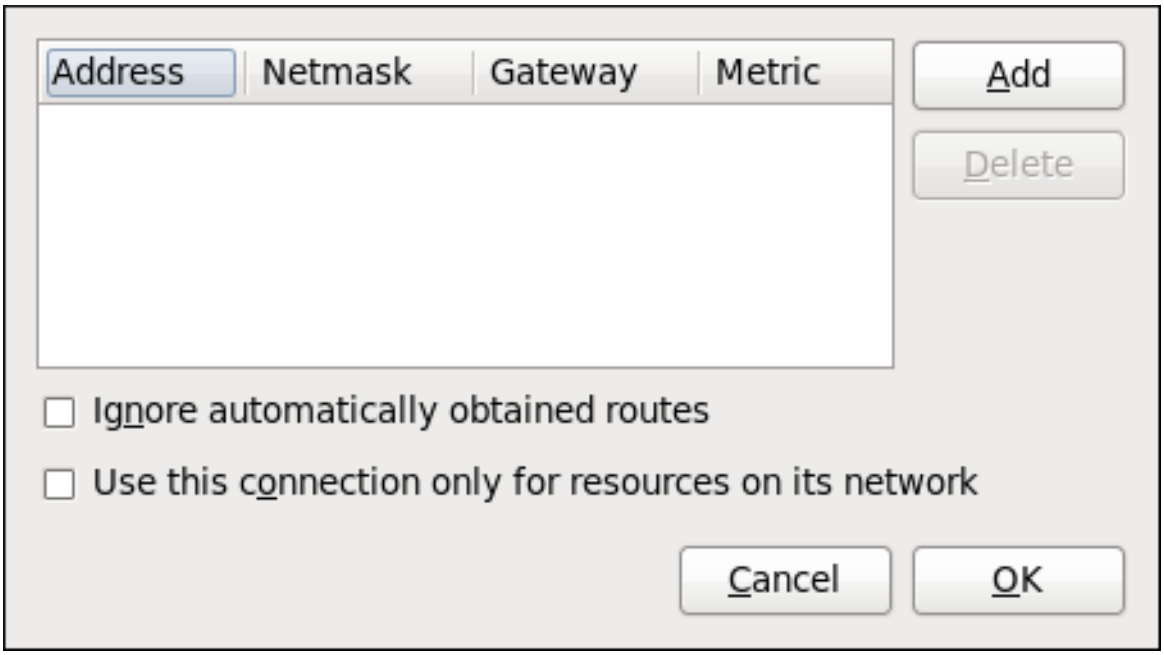
ドメインを検索: コンマで区切られたドメインコントローラーのリストを入力します。

ネットワーク接続に静的ルートを設定する方法は、「ルートの作成」 を参照してください。

10.3.9.6. ルートの作成

ホストのルーティングテーブルには、ネットワークに直接接続しているルートが自動的に追加されます。ルートは、ネットワークインターフェースの 「起動」 時にこれらを確認することで確認できます。このセクションでは、VPN や leased 行などの中間ネットワークや接続をトラバースすることで到達できるネットワークまたはホストへの静的ルートを入力します。

図10.18 静的ルートの設定



[D]

アドレス

アドレス： ネットワーク、サブネット、またはホストの IP アドレス

netmask : 入力した IP アドレスのネットマスクまたはプレフィックス長。

ゲートウェイ - ネットワーク、サブネット、またはホストにつながるゲートウェイの IP アドレス。

メトリック: このルートに付与する優先値となるネットワークコスト。数値が低い方が優先されます。

自動的に取得したルートを無視します。

このチェックボックスを選択すると、この接続に手動で入力したルートのみを使用します。

Use this connection only for resources on its network (この接続はネットワーク上のリソースのためだけに使用)

このチェックボックスを選択すると、この接続はデフォルトルートになりません。典型的な例としては、接続がヘッドオフィスの VPN またはリース線で、インターネットバインドされたトラフィックが接続を通過させないようにする場合などです。このオプションを選択すると、この接続

で自動的に学習したルートを使用することが明確なトラフィックか、手動で入力したトラフィックのみがこの接続を経由します。

第11章 NETWORK INTERFACES

Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux では、すべてのネットワーク通信は、設定されているソフトウェア インターフェース と、システムに接続されている 物理ネットワークデバイス の間で行われます。

ネットワークインターフェースの設定ファイルは /etc/sysconfig/network-scripts/ ディレクトリーにあります。これらのネットワークインターフェースをアクティベートおよび非アクティブにするために使用するスクリプトもここにあります。インターフェースファイルの数と種類はシステムによって異なりますが、このディレクトリーに存在するファイルにはカテゴリーが 3 つあります。

1. インターフェース設定ファイル
2. インターフェース制御スクリプト
3. ネットワーク関数ファイル

各カテゴリーのファイルは一緒に機能し、さまざまなネットワークデバイスを有効にします。

本章では、これらのファイル間の関係と、その使用方法について説明します。

11.1. ネットワーク設定ファイル

インターフェース設定ファイルに移動する前に、まずネットワーク設定で使用される主な設定ファイルを項目にします。ネットワークスタックのセットアップでこれらのファイルの役割を理解することは、Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux システムをカスタマイズする際に役立ちます。

主要なネットワーク設定ファイルは以下のとおりです。

/etc/hosts

このファイルの主な目的は、他の方法で解決できないホスト名を解決することです。DNS サーバーを使用しない小ネットワーク上のホスト名を解決することもできます。コンピューターが置かれるネットワークのタイプにかかわらず、このファイルには localhost.localdomain としてループバックデバイスの IP アドレス(127.0.0.1)を指定する行が含まれている必要があります。詳細は、Hosts (5)man ページ を参照してください。

/etc/resolv.conf

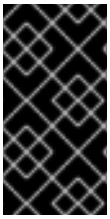
このファイルは、DNS サーバーの IP アドレスと検索ドメインを指定します。特に設定しない限り、ネットワークの初期化スクリプトにこのファイルが設定されます。このファイルの詳細は、resolv.conf(5) man ページを参照してください。

/etc/sysconfig/network

このファイルは、すべてのネットワークインターフェースのルーティングおよびホスト情報を指定します。これは、インターフェース固有ではなく、グローバルな効果を持つディレクティブを含めるために使用されます。このファイルおよび許可されるディレクティブの詳細は、「/etc/sysconfig/network」 を参照してください。

/etc/sysconfig/network-scripts/ifcfg-interface-name

ネットワークインターフェースごとに、対応するインターフェース設定スクリプトがあります。これらの各ファイルは、特定のネットワークインターフェースに固有の情報を提供します。このタイプのファイルおよび受け入れ可能なディレクティブの詳細については、「インターフェース設定ファイル」 を参照してください。



ネットワークインターフェース名

ネットワークインターフェース名は、ハードウェアタイプごとに異なる可能性があります。詳細は、付録A ネットワークデバイス命名における一貫性 を参照してください。

/ETC/SYSCONFIG/NETWORKING/ ディレクトリー

/etc/sysconfig/networking/ ディレクトリーは、現在非推奨になった Network Administration Tool (system-config-network)によって使用されます。そのコンテンツは手動で編集し ない でください。設定を削除するリスクがあるため、ネットワーク設定に 1 つだけ使用することが強く推奨されます。グラフィカル設定ツールを使用したネットワークインターフェースの設定に関する詳細は、10 章NetworkManager を参照してください。

11.1.1. ホスト名の設定

静的ホスト名を永続的に変更するには、/etc/sysconfig/network ファイルの HOSTNAME ディレクティブを変更します。以下に例を示します。

```
HOSTNAME=penguin.example.com
```

Red Hat では、静的ホスト名は、host.example.com などの DNS のマシンに使用される 完全修飾ドメイン名 (FQDN)と一致することを推奨しています。また、静的なホスト名は、小文字の 7 ビット ASCII のみで構成されており、スペースやドットはなく、DNS ドメイン名ラベルで許可されている形式に制限することが推奨されます。ただし、これは厳密な要件ではありません。従来の仕様ではアンダースコアは禁止されているので、この使用も推奨されません。変更は、ネットワークサービスまたはシステムが再起動されている場合にのみ有効になります。

ホストの FQDN は、/etc/sysconfig/network の設定、または /etc/hosts ファイルにより DNS リゾルバーで提供できることに注意してください。/etc/nsswitch.conf における hosts: files dns のデフォルト設定により、リゾルバーの前に設定ファイルが確認されます。/etc/host.conf ファイルの multi on のデフォルト設定は、最初にではなく、/etc/hosts ファイル内のすべての有効な値が返されることを意味します。

システムの起動時に DNS が実行されていない場合など、/etc/sysconfig/network の HOSTNAME ディレクティブの代わりに、/etc/hosts ファイルでホストテーブルを使用する必要がある場合があります。

/etc/hosts ファイルを使用してホスト名を変更するには、次の形式で行を追加します。

```
192.168.1.2 penguin.example.com penguin
```

11.2. インターフェース設定ファイル

インターフェース設定ファイルは、個々のネットワークデバイスのソフトウェアインターフェースを制御します。これは、システムの起動時に、このファイルを使用して、どのインターフェースを起動するかと、どのように設定するかを決定します。このファイルは、通常 ifcfg-name です。name は、設定ファイルが制御するデバイスの名前を指します。

11.2.1. イーサネットインターフェース

最も一般的なインターフェースファイルの 1 つが /etc/sysconfig/network-scripts/ifcfg-eth0 で、システム内の最初のイーサネット ネットワークインターフェースカード または NIC を制御します。複数の NIC を持つシステムには、複数の ifcfg-ethXファイル（ X は特定のインターフェースに対応する一意の番号）があります。各デバイスには独自の設定ファイルがあるため、管理者は各インターフェースがどのように個別に機能するかを制御できます。

以下は、固定 IP アドレスを使用してシステム用の ifcfg-eth0 ファイルのサンプルです。

```
DEVICE=eth0
BOOTPROTO=none
ONBOOT=yes
NETMASK=255.255.255.0
IPADDR=10.0.1.27
USERCTL=no
```

インターフェース設定ファイルに必要な値は、他の値に基づいて変更できます。たとえば、DHCP を使用するインターフェースの ifcfg-eth0 ファイルは、 DHCP サーバーが IP 情報を提供するためとは異なります。

```
DEVICE=eth0
BOOTPROTO=dhcp
ONBOOT=yes
```

NetworkManager は、さまざまなネットワークインターフェースの設定ファイルを簡単に変更できるグラフィカル設定ツールです（このツールの使用方法は 10章NetworkManager を参照してください）。

ただし、特定のネットワークインターフェースの設定ファイルを手動で編集することもできます。

以下は、イーサネットインターフェース設定ファイルの設定可能なパラメーターの一覧です。

BONDING_OPTS=parameters

ボンディングデバイスの設定パラメーターを設定し、/etc/sysconfig/network-scripts/ifcfg-bondN で使用されます（「チャンネルボンディングインターフェース」を参照）。このパラメーターは、ボンディングモジュールのディレクティブで説明されているように、/sys/class/net/bonding_device /bonding のボンディングデバイスに使用されるパラメーターと、ボンディングドライバーのモジュールパラメーターと同じです。

この設定方法を使うと、複数のボンディングデバイスに異なる設定をすることが可能になります。Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;LinuxRed Hat Enterprise Linux 6nbsp;6 では、ifcfg-name ファイルの BONDING_OPTS ディレクティブの後に、すべてのインターフェース固有のボンディングオプションを配置します。詳細は、ボンディングモジュールパラメーターを指定する場所 を参照してください。

BOOTPROTO=protocol

protocol は以下のいずれかになります。

- *none: ブートタイムプロトコルは使用できません。*
- *BOOTP - BOOTP プロトコルを使用する必要があります。*
- *DHCP - DHCP プロトコルを使用する必要があります。*

BROADCAST=address

ここでの address はブロードキャストアドレスになります。この値は ipcalc で自動的に計算されるため、このディレクティブは非推奨になりました。

DEVICE=name

name は、物理デバイスの名前です（論理名である動的に割り当てられている PPP デバイスを除く）。

DHCP_HOSTNAME=name

name は、DHCP サーバーに送信される短いホスト名です。このオプションは、DHCP サーバーが IP アドレスを受け取る前にクライアントがホスト名を指定する必要がある場合にのみ使用します。

DHCPV6C=answer

ここでの answer は以下のいずれかになります。

- *はい: DHCP を使用して、このインターフェースの IPv6 アドレスを取得します。*
- *No: DHCP を使用してこのインターフェースの IPv6 アドレスを取得しないでください。これはデフォルト値になります。*

デフォルトでは、IPv6 リンクローカルアドレスは引き続き割り当てられます。リンクローカルアドレスは、『RFC 4862』 に従ってインターフェースの MAC アドレスに基づいています。

DHCPV6C_OPTIONS=answer

ここでの answer は以下のいずれかになります。

- *-p - IPv6 プレフィックス委譲を有効にします。*
- *-s: DHCP を使用して、このインターフェースのアドレスではなくステートレス設定のみを取得します。*
- *-n - -T オプションまたは -P オプションを使用してから、通常の操作を復元します。*
- *- t: DHCP を使用して、このインターフェースの一時的な IPv6 アドレスを取得します。*
- *-d - 使用する DHCP Unique Identifier (DUID)のタイプを選択する際にデフォルトを上書きします。*

デフォルトでは、DHCP クライアント(dhclient)は、ステートレスモード（アドレスを要求しない -S オプションを使用）で実行している場合は、リンク層アドレス(DUID-LL)に基づいて DHCP Unique Identifier (DUID-LLT)を作成します。-D オプションは、LL または LLT のいずれかの値で、このデフォルトを上書きします。

DNS{1,2}=address

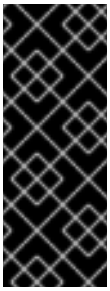
ここでの address は、PEERDNS ディレクティブが no に設定されていない限り、/etc/resolv.conf に配置されるネームサーバーアドレスです。

ETHTOOL_OPTS=options

ここでの options は、ethtool で対応しているデバイス固有のオプションになります。たとえば、100Mb を強制する場合は、duplex 全体を実行します。

```
ETHTOOL_OPTS="autoneg off speed 100 duplex full"
```

カスタムの initscript の代わりに ETHTOOL_OPTS を使用してインターフェース速度およびデュプレックスを設定します。ネットワーク init スクリプト外でカスタムの init スクリプトを実行すると、ブート後のネットワークサービスの再起動時に予期しない結果が得られます。



速度またはデュプレックスの設定を変更する前に 「AUTONEG OFF」 を設定します。

speed または duplex 設定を変更するには、autoneg off オプションを指定して自動ネゴシエーションを無効にする必要があります。オプションエントリーは順序に依存するため、このオプションを最初に指定する必要があります。

ethtool その他のオプションは、 「ethtool」 を参照してください。

HOTPLUG=answer

ここでの answer は以下のいずれかになります。

- はい: このデバイスはホットプラグされるとアクティベートする必要があります（これはデフォルトのオプションです）。
- No: ホットプラグ さ れると、このデバイスはアクティベートされ ません。

ボンディングカーネルモジュールが読み込まれると、チャネルボンディングインターフェースがアクティブにならないように、Live PLUG=no オプションを使用できます。

チャンネルボンディングインターフェースについての詳しい情報は、 「チャンネルボンディングインターフェース」 を参照してください。

HWADDR=MAC-address

MAC-address は、AA:BB:CC:DD:EE:FF 形式のイーサネットデバイスのハードウェアアドレスです。このディレクティブは、各 NIC のモジュールに設定された負荷順序に関係なく、インターフェースに適切なデバイス名が割り当てられるように、複数の NIC が含まれるマシンで使用する必要があります。このディレクティブは、MACADDR と併用し ない でください。

注記

- *永続的なデバイス名は、/etc/udev/rules.d/70-persistent-net.rules により処理されるようになりました。*
- *HWADDR は、System z ネットワークデバイスと併用しないでください。*
- *Red Hat Enterprise Linux 6nbsp;Linux Red Hat Enterprise Linux 6nbsp;Linux『Red Hat Enterprise Linux 6nbsp;6 Installation Guide』のセクション 25.3.3、「Mapping subchannels and network device names」を参照してください。*

IPADDRn=address

ここで、address は IPv4 アドレスで、n は 0 から始まる連続する正の整数になることが予想されます（例： IPADDR0）。これは、インターフェースで複数の IP アドレスを持つ設定に使用されます。これは、設定されたアドレスが 1 つしかない場合は省略できます。

IPV6ADDR=address

ここでの address は、インターフェースの最初の静的（プライマリー） IPv6 アドレスになります。

形式は Address/Prefix-length です。プレフィックスの長さが指定されていない場合、/64 が想定されます。この設定は、有効にする IPV6INIT に依存することに注意してください。

IPV6ADDR_SECONDARIES=address

ここでの address は、スペースで区切られた、追加の IPv6 アドレスになります。

形式は Address/Prefix-length です。プレフィックスの長さが指定されていない場合、/64 が想定されます。この設定は、有効にする IPV6INIT に依存することに注意してください。

IPV6INIT=answer

ここでの answer は以下のいずれかになります。

- *はい: IPv6 アドレス設定用にこのインターフェースを初期化します。*
- *No: IPv6 アドレス設定用にこのインターフェースを初期化しません。これはデフォルト値になります。*

この設定は、IPv6 アドレスの IPv6 静的および DHCP の割り当てに必要です。『RFC 4862』に従って、IPv6 ステートレスアドレス自動設定 (SLAAC)には影響を与えません。

IPv6 を無効にする方法は、「/etc/sysconfig/network」を参照してください。

IPV6_AUTOCONF=answer

ここでの answer は以下のいずれかになります。

- *はい: このインターフェースの IPv6 自動conf 設定を有効にします。*
- *no: このインターフェースの IPv6 自動conf 設定を無効にします。*

有効にすると、radvd デーモンを実行するルーターから Neighbor Discovery (ND)を使用して IPv6 アドレスを要求します。

IPV6_AUTOCONF のデフォルト値は、以下のように IPV6FORWARDING に依存することに注意してください。

- *IPV6FORWARDING=yes である場合、IPV6_AUTOCONF はデフォルトで no に設定されます。*
- *IPV6FORWARDING=no である場合、IPV6_AUTOCONF はデフォルトで yes に設定され、IPV6_ROUTER には影響がありません。*

IPV6_MTU=value

value は、このインターフェースの任意の専用 MTU です。

IPV6_PRIVACY=rfc3041

rfc3041 では、『IPv6 の Stateless Address Autoconfiguration の RFC 3041 Privacy Extensions をサポートするようにこのインターフェースを設定』します。この設定は、有効になっている IPV6INIT オプションに依存することに注意してください。

デフォルトは『RFC 3041』サポートを無効にします。ステートレス自動設定は、変更した EUI-64 メソッドを使用して、利用可能な場合に MAC アドレスに基づいてアドレスを取得します。アドレスは接頭辞に追加されますが、アドレスは通常 MAC アドレスから派生するので、接頭辞が変更されてもグローバルで一意となります。リンクローカルアドレスの場合、『RFC 2462 IPv6 Stateless Address Autoconfiguration』に従ってプレフィックスが fe 801-1 になります。

LINKDELAY=time

ここでの time は、デバイスを設定する前にリンクネゴシエーションを待つ秒数です。デフォルトは 5 秒です。たとえば、STTP が原因とするリンクネゴシエーション の 遅延は、この値を増やすことで解決できます。

MACADDR=MAC-address

MAC-address は、AA:BB:CC:DD:EE:FF 形式のイーサネットデバイスのハードウェアアドレスです。

このディレクティブは、MAC アドレスをインターフェースに割り当てるのに使用されます。物理 NIC に割り当てられたアドレスを上書きします。このディレクティブは HWADDR ディレクティブと併用し ない でください。

MASTER=bond-interface

bond-interface は、イーサネットインターフェースがリンクされるチャネルボンディングインターフェースです。

このディレクティブは、SLAVE ディレクティブと併用されます。

チャンネルボンディングインターフェースについての詳しい情報は、「チャンネルボンディングインターフェース」 を参照してください。

NETMASKn=mask

mask はネットマスクの値で、n は 0 から始まる連続する正の整数になります（例：NETMASK0）。これは、インターフェースで複数の IP アドレスを持つ設定に使用されます。これは、設定されたアドレスが 1 つしかない場合は省略できます。

NETWORK=address

ここでの address はネットワークアドレスになります。この値は ipcalc で自動的に計算されるため、このディレクティブは非推奨になりました。

NM_CONTROLLED=answer

ここでの answer は以下のいずれかになります。

- はい - NetworkManager はこのデバイスの設定を許可しています。これはデフォルトの動作であり、省略可能です。
- no - NetworkManager はこのデバイスの設定を許可していません。

注記

NM_CONTROLLED ディレクティブは、Red Hat Enterprise Linux 6.3 の時点で、/etc/sysconfig/network の NM_BOND_VLAN_ENABLED ディレクティブに依存します。そのディレクティブが存在し、yes、y、または true のいずれかである場合にのみ、NetworkManager はボンディングおよび VLAN インターフェースを検出し、管理します。

ONBOOT=answer

ここでの answer は以下のいずれかになります。

- *はい - このデバイスは、システムの起動時にアクティブにする必要があります。*
- *No - このデバイスは、システムの起動時にアクティブにしないでください。*

PEERDNS=answer

ここでの answer は以下のいずれかになります。

- *はい - DNS ディレクティブが設定されている場合、または PPP で Microsoft の『RFC 1877』 IPCP 拡張を使用している場合は /etc/resolv.conf を変更します。いずれの場合も、デフォルトは yes です。*
- *No: /etc/resolv.conf は変更しないでください。*

SLAVE=answer

ここでの answer は以下のいずれかになります。

- *はい: このデバイスは、MASTER ディレクティブで指定されたチャンネルボンディングインターフェースによって制御されます。*
- *no: このデバイスは、MASTER ディレクティブで指定されたチャンネルのボンディングインターフェースで制御され ません。*

このディレクティブは、MASTER ディレクティブと併用されます。

チャンネルボンディングインターフェースについての詳しい情報は、「チャンネルボンディングインターフェース」 を参照してください。

SRCADDR=address

ここでの address は、送信パケットの指定されたソース IP アドレスになります。

USERCTL=answer

ここでの answer は以下のいずれかになります。

- *はい -root 以外のユーザーはこのデバイスを制御できます。*
- *No -root 以外のユーザーはこのデバイスを制御することはできません。*

11.2.2. System z 上の Linux の特定の ifcfg オプション

SUBCHANN336= <read_device_bus_id > , < write_device_bus_id > , < data_device_bus_id>

&lt ;read_device_bus_id >、< write_device_bus_id >、および < data_device_bus_id > は、ネットワークデバイスを表す 3 つのデバイスバス ID です。

PORTNAME=myname;

ここで、myname は Open Systems Adapter(OSA)ポート名または LAN Channel Station(LCS)ポート番号になります。

CTCPROT=answer

ここでの answer は以下のいずれかになります。

- *0 - 互換性モード、仮想マシンの TCP/IP（IBM S/390 および IBM System z オペレーティングシステム以外の Linux 以外のピアと使用）これはデフォルトのモードです。*
- *1 - Linux ピアに使用される拡張モード。*
- *3 - S/390 および IBM System z オペレーティングシステムの互換性モード。*

このディレクティブは、NETTYPE ディレクティブと併用されます。NETTYPE='ctc' の CTC プロトコルを指定します。デフォルトは 0 です。

OPTION='answer'

「answer」 は、有効な sysfs 属性とその値の引用符で囲まれた文字列です。現在、Red Hat Enterprise Linux インストーラーはこれを使用して、QETH デバイスのレイヤーモード(layer2)と、相対ポート番号(portno)を設定します。以下に例を示します。

```
OPTIONS='layer2=1 portno=0'
```

11.2.3. System z の Linux に必要な ifcfg オプション

NETTYPE=answer

ここでの answer は以下のいずれかになります。

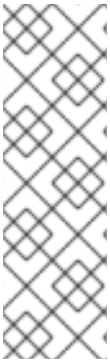
- *C TC - チャンネル間の通信ポイントツーポイントの TCP/IP または TTY の場合。*
- *lcs: LAN Channel Station(LCS)*
- *qeth: QETH（QDIO イーサネット）これは、デフォルトのネットワークインターフェースです。物理または仮想の OSA カードおよび HiperSockets デバイスをサポートするのに推奨される方法です。*

11.2.4. チャンネルボンディングインターフェース

Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux を使用すると、管理者は ボンディングカーネルモジュールとチャンネルボンディング インターフェースと呼ばれる特別なネットワークインターフェースを使用して、複数のネットワークインターフェースを 1 つの チャンネル にバインドできます。このチャンネルボンディングにより、複数のネットワークインターフェースが 1 つとして機能できるようになり、また同時に帯域幅が増加し、冗長性を提供します。

警告

ネットワークスイッチを使わずにケーブルの直接接続を使用すると、ボンディングはサポートされません。本章で説明されているフェイルオーバーメカニズムは、ネットワークスイッチがないと予想どおりに機能しません。詳細についてはナレッジベースの記事『ボンディングは、クロスオーバーケーブルを使用したダイレクトコレクションをサポートしますか?』を参照してください。



注記

active-backup、balance-tlb および balance-alb の各モードはスイッチの特定の設定を必要としません。他のボンディングモードでは、スイッチがリンクを集約するように設定する必要があります。たとえば、Cisco スイッチでは Modes 0、2、および 3 に EtherChannel を必要としますが、Mode 4 には LACP と EtherChannel が必要となります。kernel-doc パッケージでスイッチおよび bonding.txt ファイルで提供されるドキュメントを参照してください（「その他のリソース」を参照）。

11.2.4.1. ボンディングカーネルモジュールがインストールされているかの確認

Red Hat Enterprise Linux 6 では、ボンディングモジュールはデフォルトで読み込まれません。root で以下のコマンドを実行してこのモジュールを読み込みます。

```
~]# modprobe --first-time bonding
```

モジュールが実行されておらず、読み込まれたことを示す視覚的な出力はありません。このアクティベーションは、システム再起動後は維持されません。永続的なモジュールの読み込みの説明は、「永続的なモジュールの読み込み」 を参照してください。BONDING_OPTS ディレクティブを使用した適切な設定ファイルがあれば、ボンディングモジュールは必要に応じて読み込まれるので、別個に読み込む必要はないことに留意してください。

モジュールについての情報を表示するには、以下のコマンドを実行します。

```
~]$ modinfo bonding
```

コマンドオプションの詳細は、modprobe(8) man ページを参照してください。モジュールの読み込みおよびアンロードの詳細は、31章カーネルモジュールの使用 を参照してください。

11.2.4.2. チャンネルボンディングインターフェースの作成

チャンネルボンディングインターフェースを作成するには、/etc/sysconfig/network-scripts/ ディレクトリーに ifcfg-bondN という名前のファイルを作成し、N をそのインターフェースの番号 0 などに置き換えます。

ファイルの内容は、イーサネットインターフェースなど、ボンディングされるインターフェースタイプと同じになります。唯一の違いは、DEVICE ディレクティブは ボンディングNで、N はインターフェースの数に置き換えてください。NM_CONTROLLED ディレクティブを追加して、NetworkManager がこのデバイスを設定しないようにします。

例11.1 ifcfg-bond0 インターフェース設定ファイルの例

以下は、チャンネルボンディングインターフェース設定ファイルの例です。

```
DEVICE=bond0
IPADDR=192.168.1.1
NETMASK=255.255.255.0
ONBOOT=yes
BOOTPROTO=none
USERCTL=no
NM_CONTROLLED=no
BONDING_OPTS="bonding parameters separated by spaces"
```

ボンディングの MAC アドレスは、スレーブとなる最初のインターフェースから取得されます。また、必要に応じて HWADDR ディレクティブを使用して指定することもできます。NetworkManager でこのインターフェースを制御する場合は、NMI_CONTROLLED=no ディレクティブを削除するか、これを yes に設定し、TYPE=Bond および BONDING_MASTER=yes を追加します。

チャンネルボンディングインターフェースの作成後に、設定ファイルに MASTER ディレクティブおよび SLAVE ディレクティブを追加して、バインドするネットワークインターフェースを設定する必要があります。チャンネルボンディング各インターフェースの設定ファイルはほぼ同じです。

例11.2 ifcfg-ethX ボンディングインターフェース設定ファイルの例

2 つのイーサネットインターフェースがチャネルボンディングを使用している場合には、eth0 と eth1 の両方は以下のようになります。

```
DEVICE=ethX
BOOTPROTO=none
ONBOOT=yes
```

```
MASTER=bond0
SLAVE=yes
USERCTL=no
NM_CONTROLLED=no
```

この例では、X をインターフェースの数値値に置き換えます。

インターフェースを設定したら、network サービスを再起動してボンディングを起動します。これは単なる空の設定ファイルです。root で以下のコマンドを実行します。

```
~]# service network restart
```

ボンディングのステータスを確認するには、/proc/ ファイルを表示するには、

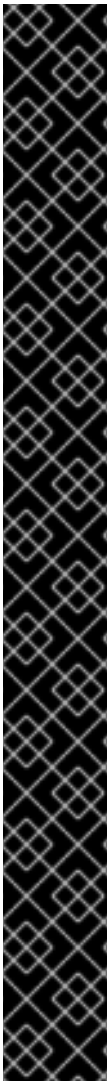
```
cat /proc/net/bonding/bondN
```

の形式でコマンドを実行します。

```
~]$ cat /proc/net/bonding/bond0
Ethernet Channel Bonding Driver: v3.6.0 (September 26, 2009)

Bonding Mode: load balancing (round-robin)
MII Status: down
MII Polling Interval (ms): 0
Up Delay (ms): 0
Down Delay (ms): 0
```

ボンディングモジュールの設定に関する指示およびアドバイスとボンディングパラメーターの一覧については、「チャンネルボンディングの使用」を参照してください。

Red Hat Enterprise Linux 6.3 では、ボンディングへの対応が NetworkManager に追加されました。NM_CONTROLLED および NM_BOND_VLAN_ENABLED ディレクティブの説明は、「イーサネットインターフェース」 を参照してください。



ボンディングモジュールパラメーターを指定する場所

Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;LinuxRed Hat Enterprise Linux 6nbsp;6 では、ボンディングカーネルモジュールのインターフェース固有のパラメーターは、ifcfg-bondN インターフェースファイルの ボンディングパラメーター のスペース区切りリストとして指定する必要があります。/etc/modprobe.d/bonding .conf のボンディング に固有のオプションは、または非推奨の /etc/modprobe.conf ファイルで指定し ない でください。

max_bonds パラメーターはインターフェース固有ではないため、必要に応じて、以下のように /etc/modprobe.d/bonding.conf で指定する必要があります。

```
options bonding max_bonds=1
```

ただし、BONDING_OPTS ディレクティブで ifcfg-bondN ファイルを使用する場合は、max_bonds パラメーターを設定し ない でください。このディレクティブにより、必要に応じてネットワークスクリプトがボンディングインターフェースを作成します。

/etc/modprobe.d/bonding.conf への変更は、モジュールが次回読み込まれるまで反映されないことに注意してください。実行中のモジュールは、最初にアンロードする必要があります。モジュールの読み込みおよびアンロードの詳細は、31章カーネルモジュールの使用 を参照してください。

11.2.4.2.1. 複数のボンド作成

Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;LinuxRed Hat Enterprise Linux 6nbsp;6 では、ボンディングインターフェースごとに BONDING_OPTS ディレクティブを含むチャンネルボンディングインターフェースが作成されます。この設定方法を使うと、複数のボンディングデバイスに異なる設定をすることが可能になります。複数のチャンネルボンディングインターフェースを作成するには、以下の手順に従います。

- *BONDING_OPTS ディレクティブがある複数の ifcfg-bondN ファイルを作成します。このディレクティブを使うと、ネットワークスクリプトが必要に応じてボンディングインターフェースを作成するようになります。*
- *ボンディングされるインターフェース設定ファイルを作成、または既存のものを編集し、SLAVE ディレクティブをこれに含めます。*
- *ボンディングされるスレーブインターフェースを MASTER ディレクティブでチャンネルボンディングインターフェースに割り当てます。*

例11.3 複数の ifcfg-bondN インターフェース設定ファイルの例

以下は、チャンネルボンディングインターフェース設定ファイルの例です。

```
DEVICE=bondN
IPADDR=192.168.1.1
NETMASK=255.255.255.0
ONBOOT=yes
BOOTPROTO=none
USERCTL=no
NM_CONTROLLED=no
BONDING_OPTS="bonding parameters separated by spaces"
```

この例では、N をインターフェースの番号に置き換えます。たとえば、2 つのボンディングを作成するには、ifcfg-bond0 と ifcfg-bond1 の 2 つの設定ファイルを作成します。

例11.2「ifcfg-ethX ボンディングインターフェース設定ファイルの例」 に従ってボンディングするインターフェースを作成し、MASTER=bondN ディレクティブを使用して必要に応じてボンドインターフェースに割り当てます。たとえば上記の例では、ボンドあたり 2 つのインターフェースが必要だとすると、2 つのボンドに 4 つのインターフェース設定ファイルを作成し、最初の 2 つを MASTER=bond0 を使って割り当て、残りの 2 つを MASTER=bond1 を使って割り当てます。

11.2.5. ボンディングを介した VLAN の設定

このセクションでは、サーバーとイーサネットスイッチの 2 つのイーサネットリンクで構成されるボンディングで VLAN を設定する方法を説明します。スイッチには、別のサーバーへの 2 つ目のボンディングがあります。最初のサーバーの設定は、基本的に IP アドレスとは異なる設定と同じであるように表示されます。

警告

ネットワークスイッチを使わずにケーブルの直接接続を使用すると、ボンディングはサポートされません。本章で説明されているフェイルオーバーメカニズムは、ネットワークスイッチがないと予想どおりに機能しません。詳細についてはナレッジベースの記事『ボンディングは、クロスオーバーケーブルを使用したダイレクトコレクションをサポートしますか?』を参照してください。

注記

active-backup、balance-tlb および balance-alb の各モードはスイッチの特定の設定を必要としません。他のボンディングモードでは、スイッチがリンクを集約するように設定する必要があります。たとえば、Cisco スイッチでは Modes 0、2、および 3 に EtherChannel を必要としますが、Mode 4 には LACP と EtherChannel が必要となります。kernel-doc パッケージでスイッチおよび bonding.txt ファイルで提供されるドキュメントを参照してください（「その他のリソース」を参照）。

サーバーで利用可能なインターフェースを確認します。

```
~]$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST> mtu 1500 qdisc pfifo_fast state DOWN qlen 1000
    link/ether 52:54:00:19:28:fe brd ff:ff:ff:ff:ff:ff
3: eth1: <BROADCAST,MULTICAST> mtu 1500 qdisc pfifo_fast state DOWN qlen 1000
    link/ether 52:54:00:f6:63:9a brd ff:ff:ff:ff:ff:ff
```

手順11.1 サーバーでのインターフェースの設定

1. *eth0 を使用してスレーブインターフェースを設定します。*

```
~]# vi /etc/sysconfig/network-scripts/ifcfg-eth0
NAME=bond0-slave0
DEVICE=eth0
TYPE=Ethernet
BOOTPROTO=none
ONBOOT=yes
```

```
MASTER=bond0
SLAVE=yes
NM_CONTROLLED=no
```

NAME ディレクティブの使用は任意です。これは、nm-connection-editor、nm-applet などの GUI インターフェースにより表示されます。

2. eth1 を使用してスレーブインターフェースを設定します。

```
~]# vi /etc/sysconfig/network-scripts/ifcfg-eth1
NAME=bond0-slave1
DEVICE=eth1
TYPE=Ethernet
BOOTPROTO=none
ONBOOT=yes
MASTER=bond0
SLAVE=yes
NM_CONTROLLED=no
```

NAME ディレクティブの使用は任意です。これは、nm-connection-editor、nm-applet などの GUI インターフェースにより表示されます。

3. チャンネルボンディングインターフェース ifcfg-bond0 を設定します。

```
~]# vi /etc/sysconfig/network-scripts/ifcfg-bond0
NAME=bond0
DEVICE=bond0
BONDING_MASTER=yes
TYPE=Bond
IPADDR=192.168.100.100
NETMASK=255.255.255.0
ONBOOT=yes
BOOTPROTO=none
BONDING_OPTS="mode=active-backup miimon=100"
NM_CONTROLLED=no
```

NAME ディレクティブの使用は任意です。これは、nm-connection-editor、nm-applet などの GUI インターフェースにより表示されます。この例では、リンク監視の詳細は、「ボンディングモジュールのディレクティブ」 セクションを参照してください。

4. サーバー上のインターフェースのステータスを確認します。

```
~]$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
```

```
      valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state
UP qlen 1000
   link/ether 52:54:00:19:28:fe brd ff:ff:ff:ff:ff:ff
   inet6 fe80::5054:ff:fe19:28fe/64 scope link
      valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state
UP qlen 1000
   link/ether 52:54:00:f6:63:9a brd ff:ff:ff:ff:ff:ff
   inet6 fe80::5054:ff:fef6:639a/64 scope link
      valid_lft forever preferred_lft forever
```

手順11.2 インターフェースとの競合の解決

スレーブとして設定されたインターフェースは、IPv6 のリンクローカルアドレス（ fe80の起動）とは別に IP アドレスを割り当てないでください。予期しない IP アドレスがある場合は、ONBOOT が yes に設定されている別の設定ファイルが存在する可能性があります。

1. *この場合は、以下のコマンドを実行して、競合の原因となる可能性のあるすべての ifcfg ファイルを一覧表示します。*

```
~]$ grep -r "ONBOOT=yes" /etc/sysconfig/network-scripts/ | cut -f1 -d":" | xargs grep -E
"IPADDR|SLAVE"
/etc/sysconfig/network-scripts/ifcfg-lo:IPADDR=127.0.0.1
```

上記は、新規インストールで想定される結果を示します。ONBOOT ディレクティブと IPADDR または SLAVE ディレクティブの両方を含むファイルが表示されます。たとえば、ifcfg-eth1 ファイルが正しく設定されていない場合、表示は以下のようになります。

```
~]# grep -r "ONBOOT=yes" /etc/sysconfig/network-scripts/ | cut -f1 -d":" | xargs grep -E
"IPADDR|SLAVE"
/etc/sysconfig/network-scripts/ifcfg-lo:IPADDR=127.0.0.1
/etc/sysconfig/network-scripts/ifcfg-eth1:SLAVE=yes
/etc/sysconfig/network-scripts/ifcfg-eth1:IPADDR=192.168.55.55
```

2. *見つかったその他の設定ファイルはバックアップのために別のディレクトリーに移動するか、HWADDR ディレクティブを使用して別のインターフェースに割り当てる必要があります。競合を解決すると、インターフェースが 「ダウン」 して再度 「起動」 するか、root としてネットワークサービスを再起動します。*

```
~]# service network restart
Shutting down interface bond0:                          [  OK  ]
```

```
Shutting down loopback interface:                          [  OK  ]
Bringing up loopback interface:                            [  OK  ]
Bringing up interface bond0:  Determining if ip address 192.168.100.100 is already in
use for device bond0...
                                                           [  OK  ]
```

NetworkManager を使用している場合は、この時点で再起動して不要な IP アドレスを忘れることがあります。root で、以下を実行します。

```
~]# service NetworkManager restart
```

手順11.3 サーバーのボンディングの確認

1. root としてサーバー上でボンディングを起動します。

```
~]# ifup /etc/sysconfig/network-scripts/ifcfg-bond0
Determining if ip address 192.168.100.100 is already in use for device bond0...
```

2. サーバー上のインターフェースのステータスを確認します。

```
~]$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,SLAVE,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
master bond0 state UP qlen 1000
    link/ether 52:54:00:19:28:fe brd ff:ff:ff:ff:ff:ff
3: eth1: <BROADCAST,MULTICAST,SLAVE,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
master bond0 state UP qlen 1000
    link/ether 52:54:00:f6:63:9a brd ff:ff:ff:ff:ff:ff
4: bond0: <BROADCAST,MULTICAST,MASTER,UP,LOWER_UP> mtu 1500 qdisc
noqueue state UP
    link/ether 52:54:00:19:28:fe brd ff:ff:ff:ff:ff:ff
    inet 192.168.100.100/24 brd 192.168.100.255 scope global bond0
    inet6 fe80::5054:ff:fe19:28fe/64 scope link
       valid_lft forever preferred_lft forever
```

eth0 と eth1 は master bond0 state UP で、bond0 のステータスは MASTER,UP であることに注意してください。

3.
ボンディング設定の詳細を表示します。

```
~]$ cat /proc/net/bonding/bond0
Ethernet Channel Bonding Driver: v3.6.0 (September 26, 2009)

Bonding Mode: transmit load balancing
Primary Slave: None
Currently Active Slave: eth0
MII Status: up
MII Polling Interval (ms): 100
Up Delay (ms): 0
Down Delay (ms): 0

Slave Interface: eth0
MII Status: up
Speed: 100 Mbps
Duplex: full
Link Failure Count: 0
Permanent HW addr: 52:54:00:19:28:fe
Slave queue ID: 0

Slave Interface: eth1
MII Status: up
Speed: 100 Mbps
Duplex: full
Link Failure Count: 0
Permanent HW addr: 52:54:00:f6:63:9a
Slave queue ID: 0
```

4.
サーバーのルートを確認します。

```
~]$ ip route
192.168.100.0/24 dev bond0  proto kernel  scope link  src 192.168.100.100
169.254.0.0/16 dev bond0  scope link  metric 1004
```

手順11.4 サーバー上での VLAN の設定

重要

本ガイドの書き込み時に、ボンディングにはスレーブがあり、VLAN インターフェースを起動する前にそれらを稼働させることが重要です。「」書き込んだ時点では、スレーブなしで VLAN インターフェースをボンディングに追加することはできません。Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;LinuxRed Hat Enterprise Linux 6nbsp;6 では、ONPARENT ディレクティブを yes に設定することは、ボンディングが起動する前に VLAN インターフェースが起動しないようにすることが重要です。これは、VLAN 仮想デバイスは親の MAC アドレスを取得し、NIC がスレーブになると、ボンディングはその MAC アドレスをその NIC の MAC アドレスに変更します。

注記

VLAN 仮想デバイスは、親の新規 MAC アドレスに一致するように MAC アドレスを変更できないため、fail_over_mac=follow オプションが指定されたボンディングで VLAN スレーブを設定することはできません。この場合、トラフィックは間違ったソースの MAC アドレスで送信されます。

一部の古いネットワークインターフェースカード、ループバックインターフェース、Wimax カード、および一部の Infiniband デバイスは VLAN チャレンジです。つまり、 VLAN に対応できません。通常、デバイスは VLAN ヘッダーと VLAN に関連付けられた MTU サイズが大きい場合に対応できないためです。

1. *VLAN インターフェースファイル bond0.192 を作成します。*

```
~]# vi /etc/sysconfig/network-scripts/ifcfg-bond0.192
DEVICE=bond0.192
NAME=bond0.192
BOOTPROTO=none
ONPARENT=yes
IPADDR=192.168.10.1
NETMASK=255.255.255.0
VLAN=yes
NM_CONTROLLED=no
```

2. *root として VLAN インターフェースを起動します。*

```
~]# ifup /etc/sysconfig/network-scripts/ifcfg-bond0.192
Determining if ip address 192.168.10.1 is already in use for device bond0.192...
```

3.
ネットワークスイッチでの VLAN タグ付けの有効化スイッチのドキュメントを参照して、必要な設定を確認してください。

4.
サーバー上のインターフェースのステータスを確認します。

```
~]# ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,SLAVE,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
master bond0 state UP qlen 1000
    link/ether 52:54:00:19:28:fe brd ff:ff:ff:ff:ff:ff
3: eth1: <BROADCAST,MULTICAST,SLAVE,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
master bond0 state UP qlen 1000
    link/ether 52:54:00:f6:63:9a brd ff:ff:ff:ff:ff:ff
4: bond0: <BROADCAST,MULTICAST,MASTER,UP,LOWER_UP> mtu 1500 qdisc
noqueue state UP
    link/ether 52:54:00:19:28:fe brd ff:ff:ff:ff:ff:ff
    inet 192.168.100.100/24 brd 192.168.100.255 scope global bond0
    inet6 fe80::5054:ff:fe19:28fe/64 scope link
       valid_lft forever preferred_lft forever
5: bond0.192@bond0: <BROADCAST,MULTICAST,MASTER,UP,LOWER_UP> mtu 1500
qdisc noqueue state UP
    link/ether 52:54:00:19:28:fe brd ff:ff:ff:ff:ff:ff
    inet 192.168.10.1/24 brd 192.168.10.255 scope global bond0.192
    inet6 fe80::5054:ff:fe19:28fe/64 scope link
       valid_lft forever preferred_lft forever
```

インターフェースの一覧に bond0.192@bond0 があり、ステータスが MASTER,UP になっていることに注意してください。

5.
サーバーのルートを確認します。

```
~]$ ip route
192.168.100.0/24 dev bond0  proto kernel  scope link  src 192.168.100.100
192.168.10.0/24 dev bond0.192  proto kernel  scope link  src 192.168.10.1
169.254.0.0/16 dev bond0  scope link  metric 1004
169.254.0.0/16 dev bond0.192  scope link  metric 1005
```

これで VLAN インターフェース bond0.192 を参照する 192.168.10.0/24 ネットワークのルートがある点に注意してください。

2 番目のサーバーの設定

同じサブネットとは異なる IP アドレスを使用して、2 番目のサーバーの設定手順を繰り返します。

ボンディングが起動し、ネットワークスイッチが想定どおりに機能しているかどうかをテストします。

```
~]$ ping -c4 192.168.100.100
PING 192.168.100.100 (192.168.100.100) 56(84) bytes of data.
64 bytes from 192.168.100.100: icmp_seq=1 ttl=64 time=1.35 ms
64 bytes from 192.168.100.100: icmp_seq=2 ttl=64 time=0.214 ms
64 bytes from 192.168.100.100: icmp_seq=3 ttl=64 time=0.383 ms
64 bytes from 192.168.100.100: icmp_seq=4 ttl=64 time=0.396 ms

--- 192.168.100.100 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3002ms
rtt min/avg/max/mdev = 0.214/0.586/1.353/0.448 ms
```

VLAN のテスト

ネットワークスイッチが VLAN 用に設定されていることをテストするには、最初のサーバーの VLAN インターフェースに ping を試行します。

```
~]# ping -c2 192.168.10.1
PING 192.168.10.1 (192.168.10.1) 56(84) bytes of data.
64 bytes from 192.168.10.1: icmp_seq=1 ttl=64 time=0.781 ms
64 bytes from 192.168.10.1: icmp_seq=2 ttl=64 time=0.977 ms
--- 192.168.10.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 0.781/0.879/0.977/0.098 ms
```

パケットロスは、すべてが正しく設定され、VLAN および基盤のインターフェースが 「up」 になっていることが示唆されません。

オプションのステップ

- *必要に応じて、ネットワークケーブルを 1 度に 1 つずつ削除および置き換えて、追加のテストを実行し、フェイルオーバーが想定どおりに機能することを確認します。ethtool ユーティリティーを使用して、どのインターフェースがどのケーブルに接続されているかを確認します。例 :*

  ```
  ethtool --identify ifname integer
  ```

  integer は、ネットワークインターフェースで LED をフラッシュする回数に置き換えます。

-

ボンディングモジュールは STP に対応していないため、ネットワークスイッチからの BPDU パケット送信を無効にすることを検討してください。

- 設定した接続以外のシステムがネットワークにリンクされていない場合は、スイッチポートを直接送受信できるようにすることを検討してください。たとえば、Cisco スイッチでは、portfast コマンドを使用して行います。

11.2.6. ネットワークブリッジ

ネットワークブリッジは、MAC アドレスに基づいてネットワーク間のトラフィックを転送するリンク層デバイスであるため、レイヤー 2 デバイスとも呼ばれます。各ネットワークに接続されているホストを学習して、ビルドする MAC アドレスのテーブルに基づいて、転送の決定を行います。Linux ホスト内では、ソフトウェアブリッジを使ってハードウェアをエミュレートすることができます。たとえば、仮想化アプリケーション内で NIC を 1 つ以上の仮想 NIC と共有するなどです。このケースは例として示します。

ネットワークブリッジを作成するには、/etc/sysconfig/network-scripts/ ディレクトリーに ifcfg-brN という名前のファイルを作成し、N をそのインターフェースの番号 0 などに置き換えます。

ファイルのコンテンツは、イーサネットインターフェースなどブリッジされるインターフェースがどのようなタイプでも類似したものになります。相違点は、以下のとおりです。

- DEVICE ディレクティブは、brN 形式の引数としてインターフェース名を指定しています。N はインターフェースの数に置き換えます。

- TYPE ディレクティブには、引数 Bridge が指定されています。このディレクティブは、デバイスタイプと、引数が大文字/小文字を区別するかを決定します。

- ブリッジインターフェースの設定ファイルに IP アドレスが追加され、物理インターフェースには MAC アドレスのみが含まれるようになりました。

- 追加のディレクティブ DELAY=0 が加えられ、ブリッジがトラフィックを監視し、ホストの位置を学習し、フィルタリング機能の基になる MAC アドレステーブルを構築する間に、ブリッジが待機することを回避します。ルーティングループが可能でない場合は、デフォルトの 15 秒遅延は不要です。

- NetworkManager がファイルを変更しないように、NM_CONTROLLED=no をイーサネットインターフェースに追加する必要があります。また、NetworkManager の今後のバー

ジョンがブリッジ設定に対応している場合は、ブリッジ設定ファイルに追加することもできます。

以下は、静的 IP アドレスを使用したブリッジインターフェース設定ファイルの例です。

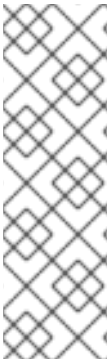
例11.4 ifcfg-br0 インターフェース設定ファイルの例

```
DEVICE=br0
TYPE=Bridge
IPADDR=192.168.1.1
NETMASK=255.255.255.0
ONBOOT=yes
BOOTPROTO=none
NM_CONTROLLED=no
DELAY=0
```

ブリッジを完成するには、別のインターフェースを作成するか既存のインターフェースを修正して、これをブリッジインターフェースに向けます。以下は、ブリッジインターフェースを参照するイーサネットインターフェース設定ファイルの例です。/etc/sysconfig/network-scripts/ifcfg-ethXで物理インターフェースを設定します。X は、以下のように、特定のインターフェースに対応する一意の番号に置き換えます。

例11.5 ifcfg-ethX インターフェース設定ファイルのサンプル

```
DEVICE=ethX
TYPE=Ethernet
HWADDR=AA:BB:CC:DD:EE:FF
BOOTPROTO=none
ONBOOT=yes
NM_CONTROLLED=no
BRIDGE=br0
```



注記

DEVICE ディレクティブでは、デバイスの種類を判断しないため、ほとんどすべてのインターフェース名を使用できます。一般的に使用される他の名前に は、tapset、ダミー、ボンディング などが含まれます。TYPE=Ethernet は必須ではありません。TYPE ディレクティブが設定されていない場合、デバイスはイーサネットデバイスとして処理されます（その名前が別のインターフェース設定ファイルに明示的にマッチしていない限り）。

ネットワークインターフェースの設定ファイルで使用されるディレクティブとオプションを確認するには、「インターフェース設定ファイル」 を参照してください。

警告

リモートホストでブリッジを設定し、設定する物理 NIC 経由でそのホストに接続されている場合は、続行する前に接続が失われる影響を検討してください。サービスを再起動する際には接続が失われ、エラーが発生すると接続を再確立することができない場合があります。コンソールもしくは帯域外のアクセスが推奨されます。

以下のように、変更を有効にするために、ネットワークサービスを再起動します。

```
service network restart
```

11.2.6.1. ボンドを使ったネットワークブリッジ

ボンディングされた 2 つ以上のイーサネットインターフェースで形成されたネットワークブリッジの例を示します。ボンディングインターフェースの設定ファイルに慣れていない場合は、を参照してください。 「チャンネルボンディングインターフェース」

ボンディングする 2 つ以上のイーサネットインターフェースの設定ファイルを、以下のように作成または編集します。

```
DEVICE=ethX
TYPE=Ethernet
USERCTL=no
SLAVE=yes
MASTER=bond0
BOOTPROTO=none
HWADDR=AA:BB:CC:DD:EE:FF
NM_CONTROLLED=no
```

注記

ethX をインターフェース名として使用することは一般的ですが、ほとんどすべての名前を使用することもできます。tap、dummy、bond などの名前は一般的に使用されます。

インターフェース設定ファイル /etc/sysconfig/network-scripts/ifcfg-bond0 を以下のように作成もしくは編集します。

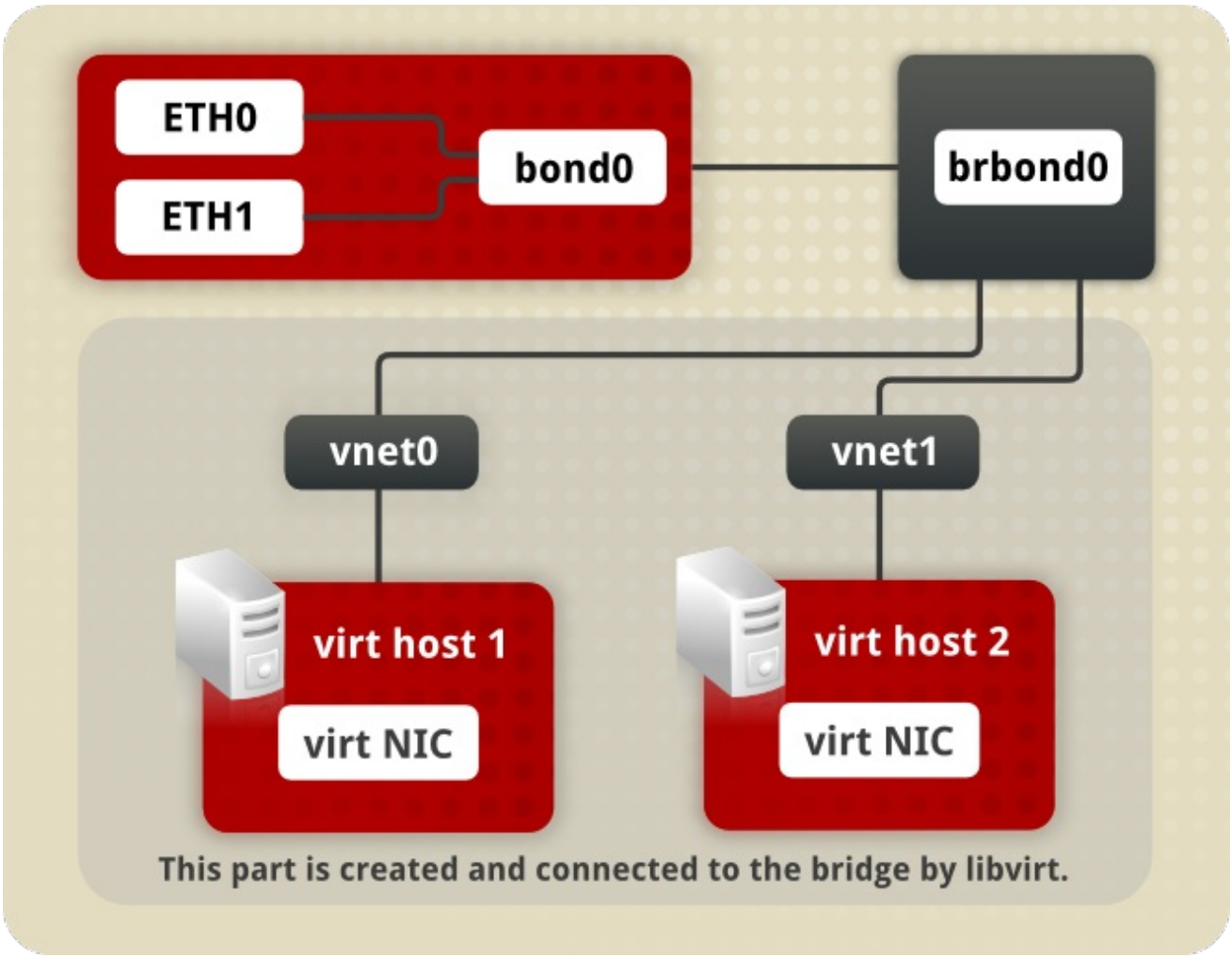
```
DEVICE=bond0
ONBOOT=yes
BONDING_OPTS='mode=1 miimon=100'
BRIDGE=br0
NM_CONTROLLED=no
```

ボンディングモジュールの設定に関する指示およびアドバイスとボンディングパラメーターの一覧については、「チャンネルボンディングの使用」を参照してください。

以下のように、1 つのインターフェース設定ファイル /etc/sysconfig/network-scripts/ifcfg-br0 を作成または編集します。

```
DEVICE=br0
ONBOOT=yes
TYPE=Bridge
IPADDR=192.168.1.1
NETMASK=255.255.255.0
NM_CONTROLLED=no
```

図11.1 2 つのボンディングされたイーサネットインターフェースで構成されるネットワークブリッジ。



[D]

MASTER=bond0 ディレクティブを持つ 2 つ以上のインターフェース設定ファイルができました。これは、DEVICE=bond0 ディレクティブを含む /etc/sysconfig/network-scripts/ifcfg-bond0 という名前の設定ファイルを参照します。次にこの ifcfg-bond0 は、IP アドレスを含む /etc/sysconfig/network-scripts/ifcfg-br0 設定ファイルをポイントし、ホスト内の仮想ネットワークのインターフェースとして機能します。

新しいインターフェースまたは最近設定されたインターフェースを起動するには、root で以下の形式のコマンドを実行します。

```
ifup device
```

または、以下のように Networking サービスを再起動して、変更を有効にします。

```
~]# service network restart
```

11.2.6.2. ボンディング VLAN を使用したネットワークブリッジ

NIC に障害が発生した場合でも、ゲスト用に別のサブネットを使用する予定の仮想化サーバーでは、ボンディング、VLAN、ブリッジを組み合わせることがよくあります。この設定の例が指定されます。ベースとなるデバイスの代わりに VLAN でブリッジを作成すると、ゲストのインターフェースを設定することなく、VLAN タグ付けをホストから完全に処理できるようになります。

1. 「「ボンディングを介した VLAN の設定」」で説明するように、ボンディングおよび VLAN が設定されていることを確認します。

2. ブリッジの設定ファイル ifcfg-br0 を作成します。

```
~]# vi /etc/sysconfig/network-scripts/ifcfg-br0
DEVICE=br0
ONBOOT=yes
TYPE=Bridge
IPADDR=192.168.10.1
NETMASK=255.255.255.0
NM_CONTROLLED=no
```

3. 先の例で VLAN の設定ファイル ifcfg-bond0.192 を調整して、新たに作成された br0 をマスターとして使用します。

```
~]# vi /etc/sysconfig/network-scripts/ifcfg-bond0.192
DEVICE=bond0.192
BOOTPROTO=none
ONPARENT=yes
#IPADDR=192.168.10.1
#NETMASK=255.255.255.0
VLAN=yes
NM_CONTROLLED=no
BRIDGE=br0
```

4. 新しいインターフェースまたは最近設定されたインターフェースを起動するには、root で以下の形式のコマンドを実行します。

```
ifup device
```

-

または、以下のように **Networking** サービスを再起動して、変更を有効にします。

```
~]# service network restart
```

11.2.7. 802.1Q VLAN タグの設定

1. 必要に応じて、**root** で以下のコマンドを実行し、**VLAN 8021q** モジュールを起動します。

```
~]# modprobe --first-time 8021q
```

モジュールが実行されておらず、読み込まれたことを示す視覚的な出力はありません。正しい設定ファイルを指定すると、**VLAN 8021q** モジュールが必要に応じてロードされるため、個別に読み込む必要はありません。

2. **/etc/sysconfig/network-scripts/ifcfg-ethX**で物理インターフェースを設定します。**X** は、以下のように、特定のインターフェースに対応する一意の番号に置き換えます。

```
DEVICE=ethX
TYPE=Ethernet
BOOTPROTO=none
ONBOOT=yes
```

3. **/etc/sysconfig/network-scripts** で **VLAN** インターフェース設定を行います。設定ファイルには、物理インターフェースと **.** 文字、**VLAN ID** 番号を指定する必要があります。たとえば、**VLAN ID** が **192** で、物理インターフェースが **eth0** の場合、設定ファイルのファイル名は **ifcfg-eth0.192** になります。

```
DEVICE=ethX.192
BOOTPROTO=none
ONBOOT=yes
IPADDR=192.168.1.1
NETMASK=255.255.255.0
USERCTL=no
NETWORK=192.168.1.0
VLAN=yes
```

同じインターフェース **eth0** に **2** 番目の **VLAN**（例： **VLAN ID 193**）を設定する必要があ

る場合には、VLAN 設定の詳細で eth0 .193 という名前の新規ファイルを追加します。

4. 変更を反映するために、ネットワークサービスを再起動します。これには、root で以下のコマンドを発行します。

```
~]# service network restart
```

11.2.8. エイリアスとクローンファイル

あまり使用されない 2 つのタイプのインターフェース設定ファイルは、エイリアス と クローン ファイルです。ip ユーティリティーが同じインターフェースに複数のアドレスの割り当てに対応するようになったため、複数のアドレスを同じインターフェースにバインドする方法を使用する必要がなくなりました。ip コマンドによるアドレスの割り当ては、複数のアドレス割り当てのために繰り返すことができます。以下に例を示します。

```
~]# ip address add 192.168.2.223/24 dev eth1
~]# ip address add 192.168.4.223/24 dev eth1
~]# ip addr
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen
1000
    link/ether 52:54:00:fb:77:9e brd ff:ff:ff:ff:ff:ff
    inet 192.168.2.223/24 scope global eth1
    inet 192.168.4.223/24 scope global eth1
```

ip ユーティリティーのコマンドは、アップストリームのパッケージ名の後に iproute2 と呼ばれる場合があり、man ip(8) ページに記載されています。Red Hat Enterprise Linux 6 のパッケージ名は iproute です。

注記

Red Hat Enterprise Linux 6 では、NetworkManager が ifcfg エイリアスファイルを読み込み、エイリアス名をアドレスラベルとして使用し、そのアドレスをマスターインターフェースに割り当てます。たとえば、ifcfg-eth0 ファイルおよび ifcfg-eth0:1 ファイルが存在する場合、NetworkManager はエイリアスファイルの DEVICE 行を読み取り、これをアドレスラベルとして保存します。エイリアスではなくセカンダリーアドレスの使用が推奨されます。

新規インストールでは、NetworkManager の IPv4 または IPv6 タブで Manual メソッドを選択し、同じインターフェースに複数の IP アドレスを割り当てる必要があります。このツールの使用方法は、10章NetworkManager を参照してください。

複数のアドレスを単一のインターフェースにバインドするために使用されるエイリアスインターフェース設定ファイル。ifcfg-if-name:alias-value 命名スキームを使用します。

たとえば、ifcfg-eth0:0 ファイルは、DEVICE=eth 0:0 と 10.0.0.2 の静的 IP アドレスを指定するように設定できます。これは、ifcfg-eth0 で DHCP 経由でその IP 情報を受け取るように設定されているイーサネットインターフェースのエイリアスとして機能します。この設定では、eth0 は動的 IP アドレスにバインドされますが、同じ物理ネットワークカードは、固定の 10.0.0 .2 IP アドレスを使用して要求を受け取ることができます。

警告

エイリアスインターフェースは DHCP をサポートしません。

クローンインターフェース設定ファイルは、ifcfg-if-name-clone-name の命名規則を使用する必要があります。エイリアスファイルは既存のインターフェースの複数のアドレスを許可しますが、クローンファイルを使用してインターフェースの追加オプションを指定します。たとえば、eth0 と呼ばれる標準の DHCP イーサネットインターフェースは、以下のようになります。

```
DEVICE=eth0
ONBOOT=yes
BOOTPROTO=dhcp
```

USERCTL ディレクティブのデフォルト値は指定されない場合は指定されないため、ユーザーはこのインターフェースを稼働させたり、ダウンしたりできません。ユーザーがインターフェースを制御できるようにするには、ifcfg-eth0 を ifcfg-eth0 -user にコピーし、ifcfg-eth0-user に次の行を追加します。

```
USERCTL=yes
```

これにより、ifcfg-eth0 と ifcfg- eth0 -user の設定オプションが組み合わされているため、/sbin/ifup eth0-user コマンドを使用して eth0 インターフェースを起動することができます。これは非常に基本的な例ですが、この方法はさまざまなオプションやインターフェースで使用できます。

グラフィカルツールを使用して、エイリアスとクローンインターフェース設定ファイルを作成することはできなくなりました。ただし、本セクションの最初で説明しているように、複数の IP アドレスを同じインターフェースに直接割り当てることができるため、この方法を使用する必要がなくなりました。新規インストールでは、NetworkManager の IPv4 または IPv6 タブで Manual メソッドを選択

し、同じインターフェースに複数の IP アドレスを割り当てる必要があります。このツールの使用方法は、10章NetworkManager を参照してください。

11.2.9. 診断インターフェース

ダイアルアップ接続を介してインターネットに接続する場合は、そのインターフェースに設定ファイルが必要になります。

PPP インターフェースファイルの名前は、以下の形式を使用して指定されます。

ifcfg-pppX

X は、特定のインターフェースに対応する一意の番号になります。

PPP インターフェース設定ファイルは、W vdial 時に自動的に 作成されます。または Kppp を使用して、ダイアアップアカウントを作成します。このファイルを手動で作成して編集することもできます。

以下は、一般的な /etc/sysconfig/network-scripts/ifcfg-ppp0 ファイルです。

```
DEVICE=ppp0
NAME=test
WVDIALSECT=test
MODEMPORT=/dev/modem
LINESPEED=115200
PAPNAME=test
USERCTL=true
ONBOOT=no
PERSIST=no
DEFROUTE=yes
PEERDNS=yes
DEMAND=no
IDLETIMEOUT=600
```

Serial Line Internet Protocol (SLIP)は別のダイヤアップインターフェースですが、あまり頻繁に使用されます。SLIP ファイルには、ifcfg-sl0 などのインターフェース設定ファイルの名前があります。

このファイルで使用可能な他のオプションには以下が含まれます。

DEFROUTE=answer

ここでの answer は以下のいずれかになります。

- *はい: このインターフェースをデフォルトルートとして設定します。*
- *no: このインターフェースをデフォルトルートとして設定しません。*

DEMAND=answer

ここでの answer は以下のいずれかになります。

- *はい：このインターフェースを使用すると、誰かが使用しようとしたときに、pppd が接続を開始できるようになります。*
- *No: このインターフェースに対して接続を手動で確立する必要があります。*

IDLETIMEOUT=value

value は、インターフェースが一意を切断するまでのアイドルアクティビティーの秒数です。

INITSTRING=string

ここでの string は、モデムデバイスに渡される初期化文字列です。このオプションは主に SLIP インターフェースで使用されます。

LINESPEED=value

value は、デバイスのボーレートです。使用できる標準値には、57600、38400、19200、および 9600 が含まれます。

MODEMPORT=device

device は、インターフェースの接続を確立するために使用されるシリアルデバイスの名前です。

MTU=value

value は、インターフェースの 最大転送単位 (MTU)設定です。MTU は、フレームが受け入れ可能なデータの最大数（ヘッダー情報をカウントしません）を指します。重複状況によっては、これを 576 の値に設定すると、パケットがドロップされ、接続のスループットが大幅に向上します。

NAME=name

name は、ダイアルアップ接続設定のコレクションに指定されたタイトルへの参照です。

PAPNAME=name

name は、リモートシステムへの接続を許可する パスワード認証プロトコル (PAP)交換時に指定したユーザー名です。

PERSIST=answer

ここでの answer は以下のいずれかになります。

- *はい: このインターフェースは、モードがハングアップした後に非アクティブであっても、いつでもアクティブにする必要があります。*
- *No: このインターフェースは、常にアクティブな状態を維持しないでください。*

REMIP=address

ここでの address は、リモートシステムの IP アドレスになります。通常、これは未指定のままです。

WVDIALSECT=name

ここでの name は、このインターフェースを /etc/wvdial.conf のダイア設定に関連付けます。このファイルには、ダイアル対象の電話番号と、インターフェースに関するその他の重要な情報が含まれています。

11.2.10. その他のインターフェース

その他の一般的なインターフェース設定ファイルには、以下が含まれます。

ifcfg-lo

ローカル ループバックインターフェース は、多くの場合、テストで使用され、同じシステムを参照する IP アドレスを必要とする各種アプリケーションで使用されます。ループバックデバイスに送信されるデータは、ホストのネットワーク層に即座に返されます。

IFCFG-LO スクリプトを手動で編集しないでください。

ループバックインターフェーススクリプト /etc/sysconfig/network-scripts/ifcfg-lo は手動で編集しないでください。これを実行すると、システムが正常に動作しなくなる可能性があります。

ifcfg-irlan0

infrared インターフェース を使用すると、ラップトップやプリンターなどのデバイス間の情報に対してインフラストラクチャーリンクを流れることができます。これは、イーサネットデバイスと同様に機能しますが、ピアツーピア接続で一般的に発生する点が異なります。

ifcfg-plip0

Parallel Line Interface Protocol (PLIP)接続は、並列ポートを使用するのを除き、イーサネットデバイスとほぼ同じように機能します。

System z 上の Linux 用のインターフェース設定ファイルには、以下が含まれます。

ifcfg-hsiN

HiperSockets インターフェースは、IBM System z オフィス上の z/VM ゲスト仮想マシンおよび論理パーティション(LPAR)内の高速の TCP/IP 通信のインターフェースです。

11.3. インターフェース制御スクリプト

インターフェース制御スクリプトは、システムインターフェースをアクティブおよび非アクティブにします。/etc/sysconfig/network-scripts/ ディレクトリーにある制御スクリプトを呼び出す主なインターフェース制御スクリプトは、/sbin/ifdown と /sbin/ifup の 2 つです。

ifup インターフェーススクリプトおよび ifdown インターフェーススクリプトは、/sbin/ ディレクトリー内のスクリプトへのシンボリックリンクです。これらのスクリプトのいずれかが呼び出されると、以下のようにインターフェースの値を指定する必要があります。

```
ifup eth0
```

IFUP インターフェーススクリプトおよび IFDOWN インターフェーススクリプトの使用

ifup インターフェーススクリプトおよび ifdown インターフェーススクリプトは、ユーザーがネットワークインターフェースを起動し、停止するために使用する唯一のスクリプトです。

以下のスクリプトは参照の目的でのみ説明されています。

ネットワークインターフェースを起動するプロセスで、さまざまなネットワーク初期化タスクを実行するのに使用する 2 つのファイルは /etc/rc.d/init.d/functions および /etc/sysconfig/network-scripts/network-functions です。詳細は、「ネットワーク機能ファイル」 を参照してください。

インターフェースが指定されており、要求を実行しているユーザーがインターフェースを制御できることを確認したら、正しいスクリプトによりインターフェースが起動しているか、ダウンします。以下は、/etc/sysconfig/network-scripts/ ディレクトリーにある一般的なインターフェース制御スクリプトです。

ifup-aliases

複数の IP アドレスがインターフェースに関連付けられている場合に、インターフェース設定ファイルから IP エイリアスを設定します。

ifup-ippp および ifdown-ippp

ISDN インターフェースを上下に移動します。

ifup-ipv6 および ifdown-ipv6

IPv6 インターフェースを up および down にします。

ifup-plip

PLIP インターフェースを起動します。

ifup-plusb

ネットワーク接続用に USB インターフェースを起動します。

ifup-post および ifdown-post

インターフェースの起動または停止後に実行するコマンドが含まれます。

ifup-ppp および ifdown-ppp

PP インターフェースを 上 下にします。

ifup-routes

デバイスを起動するときにデバイスの静的ルートを追加します。

ifdown-sit および ifup-sit

IPv4 接続内での IPv6 トンネルの起動と終了に関連する関数呼び出しが含まれます。

ifup-wireless

ワイヤレスインターフェースを起動します。

ネットワークスクリプトを削除または変更する場合は注意してください。

/etc/sysconfig/network-scripts/ ディレクトリーのスクリプトを削除または変更すると、インターフェースの接続が元に戻的または失敗する可能性があります。上級ユーザーのみが、ネットワークインターフェースに関連するスクリプトを変更する必要があります。

ネットワークスクリプトをすべて同時に操作する最も簡単な方法は、以下のコマンドにあるように、ネットワークサービス(/etc/rc.d/init.d/network)で /sbin/service コマンドを使用することです。

```
/sbin/service network action
```

ここでは、アクション の 開始、停止、または 再起動 が可能です。

設定されているデバイスと現在アクティブなネットワークインターフェースの一覧を表示するには、次のコマンドを実行します。

```
/sbin/service network status
```

11.4. 静的ルートおよびデフォルトのゲートウェイ

静的ルートは、デフォルトゲートウェイを通過するべきでないか、そうでないトラフィック用です。ルーティングは、しばしば、ルーティング専用のネットワーク上で、デバイスにより処理されます(ただし、デバイスはルーティングを行うように設定できます)。したがって、Red Hat Enterprise Linux サーバーまたはクライアントで静的ルートを設定する必要がない場合もしばしばあります。例外は、暗号化された VPN トンネルを通過する必要があるトラフィックや、コストやセキュリティー上の理由から、特定のルートを通過する必要があるトラフィックが含まれます。デフォルトゲートウェイは、ローカルネットワーク宛ではなく、ルーティングテーブルで優先ルートが指定されていないすべてのトラフィックに適用されます。デフォルトゲートウェイは、従来は専用のネットワークルーターです。

コマンドラインを使用した静的ルートの設定

静的ルートが必要な場合は、ip route add コマンドを使用してルーティングテーブルに追加し、ip route del コマンドを使用して削除します。より頻繁に使用される ip route コマンドには以下の形式が使用されます。

```
ip route [ add | del | change | append | replace ] destination-address
```

- *オプションおよび形式の詳細は、ip-route(8) man ページを参照してください。*

オプションを指定せずに ip route コマンドを使用して、IP ルーティングテーブルを表示します。以下に例を示します。

```
~]$ ip route
default via 192.168.122.1 dev eth0  proto static  metric 1024
192.168.122.0/24 dev ens9  proto kernel  scope link  src 192.168.122.107
192.168.122.0/24 dev eth0  proto kernel  scope link  src 192.168.122.126
```

ホストアドレス（つまり単一の IP アドレス）に静的ルートを追加するには、root でコマンドを実行します。

```
~]# ip route add 192.0.2.1 via 10.0.0.1 [dev ifname]
```

192. 0.2.1 はドット付き 10 進数表記のホストの IP アドレスで、10.0.0.1 は次のホップアドレス、ifname は終了インターフェースであり、次のホップになります。

ネットワークに静的ルートを追加するには、つまり IP アドレスの範囲を表す IP アドレスに静的なルートを追加するには、root で以下のコマンドを発行します。

```
~]# ip route add 192.0.2.0/24 via 10.0.0.1 [dev ifname]
```

ここでの 192.0.2.0 はドット形式 10 進法での宛先ネットワークの IP アドレスに、/24 はネットワークプレフィックスになります。ネットワークプレフィックスは、サブネットマスク内の有効なビット数です。ネットワークアドレスにスラッシュ、ネットワークプレフィックス長を続けるこの形式は、classless inter-domain routing (CIDR) 表記と呼ばれることもあります。

静的ルート設定は、インターフェースごとに /etc/sysconfig/network-scripts/route-インターフェース ファイルに保存できます。たとえば、次は静的ルートです。 eth0 インターフェースは /etc/sysconfig/network-scripts/route-eth0 ファイルに保存されます。route-interface ファイルには、ip コマンド引数と network/netmask ディレクティブの 2 つの形式があります。これについては、以下で説明します。

ip route コマンドに関する詳細情報は、ip-route(8) man ページを参照してください。

デフォルトゲートウェイの設定

デフォルトゲートウェイは、ネットワークスクリプトにより決定されます。これは、最初に

/etc/sysconfig/network を解析し、「up」 状態のインターフェースについてネットワークインターフェイス ifcfg ファイルを解析します。ifcfg ファイルは数字の小さい順に解析され、最後に読み取られる GATEWAY ディレクティブがルーティングテーブルのデフォルトルートを作成するために使用されます。

そのため、デフォルトのルートは GATEWAY ディレクティブを使用して示され、グローバルに、またはインターフェース固有の設定ファイルで指定できます。ゲートウェイをグローバルに指定すると、静的ネットワーク環境では、特に複数のネットワークインターフェースが存在する場合には、いくつかの利点があります。一貫して適用されると、障害検索が簡単になります。グローバルオプションである GATEWAYDEV ディレクティブもあります。複数のデバイスで GATEWAY を指定し、1 つのインターフェースが GATEWAYDEV ディレクティブを使用している場合は、そのディレクティブが優先されます。このオプションは、インターフェースがダウンし、障害検出が複雑になる可能性がある場合に予期しない結果をもたらす可能性があるため、推奨されません。

NetworkManager がモバイルホストを管理しているという動的なネットワーク環境では、ゲートウェイ情報はインターフェース固有である可能性が高く、DHCP による割り当てに任せるのが最善です。NetworkManager においてゲートウェイに達する出口インターフェースの選択に影響を及ぼす必要がある特別なケースでは、デフォルトゲートウェイに進まないインターフェースに ifcfg ファイルの DEFROUTE=no コマンドを利用します。

グローバルデフォルトゲートウェイ設定は /etc/sysconfig/network ファイルに保存されます。このファイルは、すべてのネットワークインターフェースのゲートウェイおよびホスト情報を指定します。このファイルおよび許可されるディレクティブの詳細は、「/etc/sysconfig/network」 を参照してください。

11.5. IFCFG ファイルでの静的ルートの設定

コマンドプロンプトで ip コマンドを使用して設定した静的ルートは、システムがシャットダウンまたは再起動すると失われます。静的ルートの設定を、システムを再起動した後も持続するようにするには、/etc/sysconfig/network-scripts/ ディレクトリーに保存されているインターフェース別の設定ファイルに追加する必要があります。ファイル名は、route-ifname の形式にする必要があります。設定ファイルで使用するコマンドには、2 つのタイプがあります。「IP コマンド引数形式を使用した静的ルート」 で説明されているように、「ネットワーク/マスクのディレクティブ形式」 で説明されているように、ip コマンドおよび Network/Netmask 形式が使用されます。

11.5.1. IP コマンド引数形式を使用した静的ルート

インターフェースごとの設定ファイル（例： /etc/sysconfig/network-scripts/route-eth0 ）で必要な場合は、最初の行でデフォルトゲートウェイへのルートを定義します。これは、ゲートウェイが DHCP 経由で設定されておらず、/etc/sysconfig/network ファイルでグローバルに設定されていない場合にのみ必要です。

```
default via 192.168.1.1 dev interface
```

ここでの 192.168.1.1 は、デフォルトゲートウェイの IP アドレスになります。interface は、デフォルトゲートウェイに接続されている、または到達可能なインターフェースになります。dev オプションは省略できます。これはオプションです。この設定は、/etc/sysconfig/network ファイルの設定よりも優先されます。

リモートネットワークへのルートが必要な場合は、静的ルートは以下のように指定できます。各行は、個別のルートとして解析されます。

> *10.10.10.0/24 via 192.168.1.1 [dev interface]*

ここでの 10.10.10.0/24 は、リモートもしくは宛先ネットワークのネットワークアドレスおよびプレフィックス長です。アドレス 192.168.1.1 は、リモートネットワークに続く IP アドレスです。ネクストホップアドレス の方が好ましいですが、出口インターフェースのアドレスでも機能します。「ネクストホップ」とは、ゲートウェイやルーターなどリンクのリモート側を意味します。dev オプションを使用して、終了 インターフェース を指定できますが、必須ではありません。必要に応じて静的ルートを追加します。

以下は、ip コマンド引数形式を使用した route-interface ファイルの例です。デフォルトゲートウェイは 192.168.0.1 です。 eth0 リース行または WAN 接続が 192.168.0.10 で利用できます。2 つの静的ルートは、10.10.10.0/24 ネットワークおよび 172.16.1.10/32 ホストに到達するためのものです。

```
default via 192.168.0.1 dev eth0
10.10.10.0/24 via 192.168.0.10 dev eth0
172.16.1.10/32 via 192.168.0.10 dev eth0
```

上記の例では、ローカルの 192.168.0.0/24 ネットワークに向かうパケットはそのネットワークに接続されているインターフェースに移動します。10.10.10.0/24 ネットワークおよび 172.16.1.10/32 ホストに向かうパケットは、192.168.0.10 に移動します。既知でないリモートネットワークに向かうパケットはデフォルトゲートウェイを使用するので、デフォルトルートが適切でない場合は、静的ルートはリモートネットワークもしくはホスト用のみに設定すべきです。ここでのリモートとは、システムに直接繋がれていないネットワークやホストを指します。

出口インターフェースの指定は、オプションです。特定のインターフェースからトラフィックを強制的に締め出したい場合は、これが便利です。たとえば、VPN の場合、リモートネットワークへのトラフィックが通過するように強制できます。 tun0 インターフェースが宛先ネットワークとは別のサブネットにある場合でも、インターフェース。

重複するデフォルトゲートウェイ

デフォルトゲートウェイが DHCP からすでに割り当てられている場合、IP コマンドの引数の形式により、起動時に 2 つのエラーか、ifup コマンドを使用して down 状態からインターフェースを起動する際に、「RTNETLINK answers」という 2 つのエラーが発生することがあります。 ファイルが存在するか、「Error:」のいずれかが重複するか、「X.X.X.X.X」はガベージコレクションです。X.X.X.X はゲートウェイか、異なる IP アドレスになります。デフォルトゲートウェイを使用して別のネットワークへの別のルートがある場合も、これらのエラーが発生する可能性があります。これらのエラーはいずれも無視しても問題ありません。

11.5.2. ネットワーク/マスクのディレクティブ形式

ネットワーク/ネットマスクディレクティブの形式を route-interface ファイルに使用することも可能です。以下は、ネットワーク/ネットマスク形式のテンプレートで、後に説明が続きます。

```
ADDRESS0=10.10.10.0
NETMASK0=255.255.255.0
GATEWAY0=192.168.1.1
```

- *ADDRESS0=10.10.10.0 は、到達するリモートネットワークまたはホストのネットワークアドレスです。*
- *NETMASK0=255.255.255.0 は、ADDRESS0=10.10.10.0 で定義されているネットワークアドレスのネットマスクです。*
- *GATEWAY0=192.168.1.1 は、ADDRESS0=10.10.10.0 への到達に使用可能なデフォルトゲートウェイまたは IP アドレスです。*

以下は、ネットワーク/ネットマスクディレクティブの形式を使用した route-interface ファイルの例です。デフォルトゲートウェイは 192.168.0.1 ですが、専用回線または WAN 接続が 192.168.0.10 で利用可能です。2 つの静的ルートは、10.10.10.0/24 および 172.16.1.0/24 のネットワークに到達するためのものです。

```
ADDRESS0=10.10.10.0
NETMASK0=255.255.255.0
GATEWAY0=192.168.0.10
ADDRESS1=172.16.1.10
NETMASK1=255.255.255.0
GATEWAY1=192.168.0.10
```

後に続く静的ルートは、順番に番号付けされる必要があり、いずれの値もスキップしてはいけませ

ん。例: ADDRESS0、ADDRESS1、ADDRESS2 など。

11.6. IPV6 トークンインターフェース識別子の設定

ネットワークでは一般的には、サーバーには静的アドレスが与えられ、通常、アドレスに障害が発生したり、不足したりする可能性がある DHCP サーバーに依存しないように手動で設定されます。IPv6 プロトコルでは、ステートレスアドレス自動設定 (SLAAC)が導入され、クライアントが DHCPv6 サーバーに依存せずにアドレスを割り当てることができます。SLAAC はインターフェースハードウェアに基づいて IPv6 アドレスを派生するため、ハードウェアが変更され、関連する SLAAC がアドレスの変更に関連する SLAAC の生成時にサーバーに使用することはできません。IPv6 環境では、ネットワーク接頭辞が変更された場合や、システムが新しい場所に移動した場合は、接頭辞の変更により、手動で構成された静的アドレスを編集する必要があります。

これらの問題に対処するために、IETF ドラフト『Tokenised IPv6 識別子』がカーネルに実装され、ip ユーティリティーに対応する機能が追加されました。これにより、管理者が指定するトークンに基づいて、IPv6 アドレスの 64 ビットインターフェース識別子の部分がルーター広告 (RA)から取得できるようになります。つまり、ネットワークインターフェースのハードウェアが変更されると、アドレスの低い 64 ビットが変更されず、システムが別のネットワークに移動した場合に、ルーター広告からネットワーク接頭辞が自動的に取得されるため、手動での編集は必要ありません。

トークン化された IPv6 識別子を使用するようにインターフェースを設定するには、root で以下の形式のコマンドを実行します。

```
~]# ip token set ::1a:2b:3c:4d/64 dev eth4
```

::1a:2b:3c:4d/64 は使用するトークンに置き換えます。この設定は永続的ではありません。これを永続化するには、コマンドを init スクリプトに追加します。「インターフェース制御スクリプト」を参照してください。

メモリーが可能なトークンを使用できますが、有効な 16 進数の範囲に制限されます。たとえば、従来ポート 53 の DNS サーバーでは、::53/ 64 のトークンを使用できます。

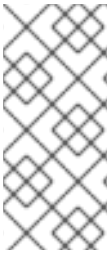
設定した IPv6 トークンすべてを表示するには、以下のコマンドを実行します。

```
~]$ ip token
    token :: dev eth0
    token :: dev eth1
    token :: dev eth2
    token :: dev eth3
    token ::1a:2b:3c:4d dev eth4
```

特定のインターフェースに設定された IPv6 トークンを表示するには、以下のコマンドを実行します。

```
~]$ ip token get dev eth4
    token ::1a:2b:3c:4d dev eth4
```

トークンをインターフェースに追加すると、以前に割り当てられたトークンが置き換えられ、そこから派生したアドレスが無効になることに注意してください。新しいトークンを指定すると、新しいアドレスが生成され、適用されますが、このプロセスで他のアドレスは変更されません。つまり、新しいトークン識別子は、他の IP アドレスではなく、既存のトークン化された識別子のみを置き換えます。



注記

複製アドレス検出(DAD)メカニズムが問題を解決できないため、同じトークンを複数のシステムまたはインターフェースに追加しないでください。トークンが設定されると、マシンをリブートしない限り、クリアやリセットを行うことはできません。

11.7. ネットワーク機能ファイル

Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux は、インターフェースを起動するために使用される重要な共通機能を含むいくつかのファイルを使用します。各インターフェースコントロールファイルにこれらの関数を含めるのではなく、必要に応じて呼び出されるいくつかのファイルでグループ化されます。

/etc/sysconfig/network-scripts/network-functions ファイルには、最も一般的に使用される IPv4 機能が含まれています。これは、多くのインターフェース制御スクリプトで役に立ちます。これらの機能には、インターフェースのステータスの変更に関する情報、ホスト名の設定、ゲートウェイデバイスの検出、特定のデバイスがダウンしているかどうかの確認、デフォルトルートの追加などを要求した実行中のプログラムへの問い合わせが含まれます。

IPv6 インターフェースに必要な機能は IPv4 インターフェースとは異なるため、/etc/sysconfig/network-scripts/network-functions-ipv6 ファイルは、特にこの情報を保持するために存在します。このファイルの機能は、静的 IPv6 ルートの設定および削除、トンネルの作成/削除、インターフェースへの IPv6 アドレスの追加と削除、インターフェース上の IPv6 アドレスの存在のテストを行います。

11.8. ETHTOOL

ethtool は、ネットワークインターフェースカード (NIC)を設定するユーティリティーです。この

ユーティリティーを使用すると、多くのネットワークデバイス（特にイーサネットデバイス）で速度、ポート、オートネゴシエーション、PCI の場所、チェックサムオフロードなどの設定をクエリーおよび変更できます。

ここでは、ethtool コマンドがよく知られていない便利なコマンドとともに使用する簡単な選択を紹介します。ethtool -h のタイプの完全なリスト、または man ページの ethtool(8) を参照してください。より包括的な一覧と説明は、 を参照してください。最初の 2 つの例は情報クエリー で、さまざまな形式のコマンドの使用を表示します。

最初は、コマンド構造です。

```
ethtool [option...] devname
```

ここでの option は none またはそれ以上のオプションであり、devname はネットワークインターフェースカード(NIC)になります。以下に例を示します。 eth0 または em1.

ethtool

デバイス名がオプションとしてのみの ethtool コマンドは、指定されたデバイスの現在の設定を出力するために使用されます。以下の形式を取ります。

```
ethtool devname
```

ここで、devname はご自分の NIC に置き換えます。以下に例を示します。 eth0 または em1.

一部の値は、コマンドが root として実行された場合にのみ取得できます。以下のコマンドが root として実行された場合の出力の例を以下に示します。

```
~]# ethtool em1
Settings for em1:
 Supported ports: [ TP ]
 Supported link modes:   10baseT/Half 10baseT/Full
                         100baseT/Half 100baseT/Full
                         1000baseT/Full
 Supported pause frame use: No
 Supports auto-negotiation: Yes
 Advertised link modes:  10baseT/Half 10baseT/Full
                         100baseT/Half 100baseT/Full
                         1000baseT/Full
 Advertised pause frame use: No
 Advertised auto-negotiation: Yes
```

```
Speed: 1000Mb/s
Duplex: Full
Port: Twisted Pair
PHYAD: 2
Transceiver: internal
Auto-negotiation: on
MDI-X: on
Supports Wake-on: pumbg
Wake-on: g
Current message level: 0x00000007 (7)
       drv probe link
Link detected: yes
```

引数の短い形式または長い形式で以下のコマンドを実行し、指定されたネットワークデバイスに関連するドライバー情報についてクエリーします。

```
ethtool -i, --driver devname
```

devname は、ネットワークインターフェースカード(NIC)に置き換えます。以下に例を示します。eth0 または em1.

以下は出力例です。

```
~]$ ethtool -i em1
driver: e1000e
version: 2.0.0-k
firmware-version: 0.13-3
bus-info: 0000:00:19.0
supports-statistics: yes
supports-test: yes
supports-eeprom-access: yes
supports-register-dump: yes
```

以下は、デバイスを識別し、リセットするコマンドオプションの一覧です。通常の -short 形式および --long 形式になります。

--statistics

--statistics または -S は、NIC およびドライバーの統計について指定されたネットワークデバイスをクエリーします。以下の形式を取ります。

```
-S, --statistics devname
```

*ここで、**devname** はご自分の **NIC** に置き換えます。*

--identify

--identify または -p オプションは、Operator が sight によってアダプターを簡単に特定できるように、アダプター固有のアクションを開始します。通常、これには、指定されたネットワークポートで 1 つ以上の LED をブリンクする必要があります。以下の形式を取ります。

```
-p, --identify devname integer
```

ここでの integer は、アクションを実行する時間（秒単位）です。

また、devname はご自分の NIC です。

--show-time-stamping

--show-time-stamping または -T オプションは、指定されたネットワークデバイスにタイムスタンプのパラメーターについてクエリーします。以下の形式を取ります。

```
-T, --show-time-stamping devname
```

*ここで、**devname** はご自分の **NIC** に置き換えます。*

--show-offload

--show-features または --show-offload または -k オプションは、指定されたネットワークデバイスに、プロトコルオフロードおよびその他の機能についてクエリーします。以下の形式を取ります。

```
-k, --show-features, --show-offload devname
```

*ここで、**devname** はご自分の **NIC** に置き換えます。*

--test

--test または -t オプションは、ネットワークインターフェースカードでテストを実行するために使用されます。以下の形式を取ります。

```
-t, --test devname word
```

ここでの 単語 は以下のいずれかになります。

- **オフライン: 包括的なテストを実行します。サービスが中断されます。**
- **オンライン: テストの縮小されたセットを実行します。サービスが中断されないでください。**
- **external_lb: ループバックケーブルに適合しながら、ループバックテストを含むテストの完全セットを実行します。**

また、devname はご自分の NIC です。

指定したネットワークデバイスの設定の一部またはすべての設定を変更するには、-s または --change オプションが必要です。以下のオプションはすべて、-s または --change オプションが指定されている場合にのみ適用されます。分かりやすくするために、ここでは省略します。

これらの設定を永続化するには、ETHTOOL_OPTS ディレクティブを使用します。インターフェース設定ファイルで使用して、ネットワークインターフェースの起動時に必要なオプションを設定できます。このディレクティブの使用方法は、「イーサネットインターフェース」 を参照してください。

--offload

--features、または --offload オプション、または -K オプションは、オフロードパラメーターと、指定したネットワークデバイスのその他の機能を変更します。以下の形式を取ります。

```
-K, --features, --offload devname feature boolean
```

feature は組み込み機能またはカーネルが提供する機能です。

boolean は ON または OFF のいずれかです。

また、devname はご自分の NIC です。

man ページの ethtool(8)に は、ほとんどの機能が記載されています。機能セットは NIC ドライバーに依存するため、man ページに記載されていない機能についてはドライバーのドキュメントを参照してください。

--speed

--speed オプションは、1 秒あたりのメガバイト(Mb/s)の速度を設定するために使用されます。速度値を省略すると、対応しているデバイス速度が表示されます。以下の形式を取ります。

```
--speed number devname
```

ここでの number は、1 秒あたりのメガバイト(Mb/s)の速度です。

また、devname はご自分の NIC です。

--duplex

--duplex オプションは、操作の送受信モードを設定するのに使用します。以下の形式を取ります。

```
--duplex word devname
```

ここでの 単語 は以下のいずれかになります。

- *half: half-duplex モードを設定します。通常、ハブに接続するときに使用されます。*
- *full: 完全 duplex モードを設定します。通常、スイッチまたは別のホストに接続する*

場合に使用します。

また、devname はご自分の NIC です。

--port

--port オプションは、デバイスポートの選択に使用されます。以下の形式を取ります。

```
--port value devname
```

value は以下のいずれかになります。

- TP - Twisted-Pair ケーブルをメディアとして使用するイーサネットインターフェース。
- A ui - Attachment Unit Interface(AUI)。通常はハブで使用されます。
- Bnc: BNC コネクターおよび共存ケーブルを使用するイーサネットインターフェース。
- MII: Media Independent Interface(MII)を使用するイーサネットインターフェース。
- Fibre: メディアに Optical Fibre を使用するイーサネットインターフェース。

また、devname はご自分の NIC です。

--autoneg

--autoneg オプションは、ネットワーク速度とモード（full-duplex または half-duplex モード）の自動ネゴシエーションを制御するために使用されます。オートネゴシエーションが有効な場合は、-r, --negotiate オプションを使用して、ネットワーク速度と操作のモードの再ネゴシエーションを開始できます。--a, --show-pause オプションを使用して自動ネゴシエーションの状態を表示できます。

以下の形式を取ります。

```
--autoneg value devname
```

value は以下のいずれかになります。

- はい: ネットワーク速度と操作モードの自動ネゴシエーションを許可します。
- No: ネットワーク速度および操作モードの自動ネゴシエーションを許可しません。

また、devname はご自分の NIC です。

--advertise

--advertise オプションは、自動ネゴシエーション用に公開される操作の速度およびモード（duplex モード）を設定するために使用されます。引数は、表11.1「ethtool アドバタイズオプション： 動作の速度とモード」 の 1 つ以上の 16 進数値です。

以下の形式を取ります。

```
--advertise option devname
```

ここでの option は、以下の表の 16 進数の値の 1 つまたは複数で、devname はご自分の NIC になります。

表11.1 ethtool アドバタイズオプション： 動作の速度とモード

16 進値	速度	duplex モード	IEEE standard?
0x001	10	半分	○

16 進値	速度	duplex モード	IEEE standard?
0x002	10	Full	○
0x004	100	半分	○
0x008	100	Full	○
0x010	1000	半分	
0x020	1000	Full	○
0x8000	2500	Full	○
0x1000	10000	Full	○
0x20000	20000MLD2	Full	
0x20000	20000MLD2	Full	
0x40000	20000KR2	Full	

--phyad

--phyad オプションを使用して、物理アドレスを変更します。多くの場合は MAC またはハードウェアアドレスと呼ばれますが、このコンテキストは物理アドレスと呼ばれます。

以下の形式を取ります。

```
--phyad physical_address devname
```

physical_address は 16 進数形式の物理アドレスで、devname はご自分の NIC に置き換えます。

--xcvr

--xcvr オプションは、transceiver タイプを選択するために使用されます。現在、「内部」 および 「外部」 のみを指定できます。他のタイプが追加される可能性があります。

以下の形式を取ります。

```
--xcvr word devname
```

ここでの 単語 は以下のいずれかになります。

- *internal: 内部トランザクションを使用します。*
- *external: 外部トランザクションを使用します。*

また、devname はご自分の NIC です。

--wol

--wol オプションは、「Wake-on-LAN」 オプションを設定します。すべてのデバイスがこれをサポートしているわけではありません。このオプションの引数は、有効にするオプションを指定する文字の文字列です。

以下の形式を取ります。

```
--wol value devname
```

value は、以下のいずれかになります。

- P - PHY アクティビティーでウェイクします。
- U - ユニキャストメッセージの Wake。
- m - マルチキャストメッセージの Wake。
- b - ブロードキャストメッセージの場合
- g - Wake-on-Lan; 「マジックパケット」の受信時にウェイクします。
- s - Wake-on-Lan のパスワードを使用してセキュリティー機能を有効にします。
- D - Wake-on-Lan を無効にし、すべての設定を消去します。

また、devname はご自分の NIC です。

--sopass

--sopass オプションを使用して 「SecureOn」 パスワードを設定します。このオプションの引数は、イーサネット MAC 16 フォーマットの 6 バイトでなければなりません (xx:yy:zz:aa:bb:cc)。

以下の形式を取ります。

```
--sopass xx:yy:zz:aa:bb:cc devname
```

ここで、xx:y:zz:aa:bb:cc は MAC アドレスと同じ形式のパスワードで、devname はご自分の NIC です。

--msglvl

--msglvl オプションは、名前または数字でドライバーの message-type フラグを設定するのに使用します。これらのタイプフラグの正確な意味はドライバーによって異なります。

以下の形式を取ります。

```
--msglvl message_type devname
```

ここで、message_type は以下のいずれかになります。

- *メッセージタイプ名（プレーンテキスト）。*
- *メッセージタイプを示す 16 進数。*

また、devname はご自分の NIC です。

定義されたメッセージタイプ名と数字を以下の表に示します。

表11.2 ドライバーメッセージタイプ

メッセージタイプ	16 進値	説明
drv	0x0001	一般的なドライバーのステータス

メッセージタイプ	16 進値	説明
probe	0x0002	ハードウェアプローブ
リンク	0x0004	リンク状態
timer	0x0008	Periodic status check
ifdown	0x0010	停止中のインターフェース
ifup	0x0020	起動中のインターフェース
rx_err	0x0040	受信エラー
tx_err	0x0080	送信エラー
intr	0x0200	割り込み処理
tx_done	0x0400	送信の完了
rx_status	0x0800	受信の完了
pktdata	0x1000	パケットの内容

メッセージタイプ	16 進値	説明
hw	0x2000	ハードウェアのステータス
wol	0x4000	wake-on-LAN ステータス

11.9. NETCONSOLE の設定

netconsole カーネルモジュールにより、ネットワーク経由でカーネルメッセージを別のコンピューターに記録できます。これにより、ディスクロギングに失敗した場合やシリアルコンソールを使用できない場合に、カーネルのデバッグを行うことができます。

リストマシンの設定

netconsole ロギングメッセージの受信を有効にするには、rsyslog パッケージをインストールします。

```
]# yum install rsyslog
```

rsyslogd が 514/UDP ポートをリッスンし、ネットワークからメッセージを受信するように設定するには、/etc/rsyslog.conf の MODULES セクションの以下の行のコメントを解除します。

```
$ModLoad imudp
$UDPServerRun 514
```

rsyslogd サービスを再起動して、変更を有効にします。

```
]# service rsyslog restart
```

rsyslogd が 514/udp ポートでリッスンしていることを確認するには、次のコマンドを使用します。

```
]# netstat -l | grep syslog
udp        0      0 *:syslog                    *:*
udp        0      0 *:syslog                    *:*
```

netstat -l 出力の 0 *:syslog 値は、/etc/services ファイルで定義されているデフォルトの netconsole ポートでリッスンしていることを意味します。

```
]$ cat /etc/services | grep syslog
syslog          514/udp
syslog-conn     601/tcp              # Reliable Syslog Service
syslog-conn     601/udp              # Reliable Syslog Service
syslog-tls      6514/tcp             # Syslog over TLS
```

送信元マシンの設定

Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;LinuxRed Hat Enterprise Linux 6nbsp;6 では、netconsole は、initscripts パッケージに含まれる /etc/sysconfig/netconsole ファイルを使用して設定されます。このパッケージはデフォルトでインストールされ、netconsole サービスも提供します。

送信マシンを設定するには、たとえば、syslogd サーバーの IP アドレスに一致するように、/etc/sysconfig/netconsole ファイルの SYSLOGADDR 変数の値を設定します。

```
SYSLOGADDR=192.168.0.1
```

netconsole サービスを再起動して、変更を適用します。次に、chkconfig コマンドを使用して、次回の再起動後に netconsole サービスが自動的に起動するようにします。

```
]# service netconsole restart
Initializing netconsole                    [  OK  ]
]# chkconfig netconsole on
```

デフォルトでは、rsyslogd サーバーは、/var/log/messages のクライアントからの netconsole メッセージ、または rsyslog.conf で指定されたファイルに書き込みます。

注記

rsyslogd と netconsole が別のポートを使用するように設定するには、/etc/rsyslog.conf の以下の行を必要なポート番号に変更します。

```
$UDPServerRun <PORT>
```

送信元マシンで、/etc/sysconfig/netconsole ファイルの以下の行をアンコメントして編集します。

```
SYSLOGPORT=514
```

netconsole 設定の詳細およびトラブルシューティングのヒントについては、Netconsole カーネル

のドキュメント を参照してください。

11.10. その他のリソース

以下は、ネットワークインターフェースの詳細を説明するリソースです。

インストールされているドキュメント

- /usr/share/doc/initscripts-version/sysconfig.txt: 本章で説明されていない IPv6 オプションを含む、ネットワーク設定ファイルの利用可能なオプションに関するガイドです。

オンラインリソース

- http://linux-ip.net/gl/ip-cref/ - 本書には、ip コマンドに関する情報が記載されています。これはルーティングテーブルを操作するのに使用できます。

- Red Hat Access Labs: Red Hat Access Labs には、「Network Bonding Helper」 が同梱されています。

その他の参考資料

- 付録E proc ファイルシステム - sysctl ユーティリティーと /proc/ ディレクトリー内の仮想ファイルを記述します。これには、ネットワークパラメーターや統計が含まれます。

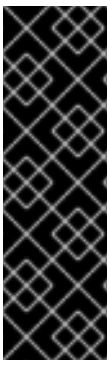
パート V. インフラストラクチャーサービス

ここでは、サービスおよびデーモンの設定方法、認証の設定、およびリモートログインの有効化方法について説明します。

第12章 サービスおよびデーモン

システムでセキュリティーを維持することが非常に重要です。このタスクの 1 つの方法として、システムサービスへのアクセスを慎重に管理することです。システムは、特定のサービスへのオープンアクセスを提供する必要がある場合があります（Web サーバーを実行している場合は httpd など）。ただし、サービスを提供する必要がない場合は、バグの不正使用の影響を最小限に抑えるために、サービスをオフにする必要があります。

本章では、ランレベルの概念と、デフォルトの設定方法を説明します。また、これらのランレベルごとに実行するサービスの設定についても説明し、service コマンドを使用して、コマンドラインでサービスの起動、停止、再起動を行う方法を説明します。



システムのセキュリティーを維持します。

新しいサービスへのアクセスを許可する場合は、ファイアウォールと SELinux の両方も設定する必要がある点に注意してください。新しいサービスを設定する際にコミットされる最も一般的な間違いの 1 つは、必要なファイアウォール設定と SELinux ポリシーを実装してアクセスを許可することです。詳細は『Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;LinuxRed Hat Enterprise Linux 6nbsp;6 『Security Guide』』を参照してください。

12.1. デフォルトのランレベルの設定

ランレベル は状態（または モード ）で、このランレベルが選択されている場合に実行されるサービスにより定義されます。番号付きのランレベルは 7 つあります（0からインデックス付けされます）。

表12.1 Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux のランレベル

ランレベル	説明
0	システムの停止に使用されます。このランレベルは予約されており、変更することはできません。
1	シングルユーザーモードで実行するために使用されます。このランレベルは予約されており、変更することはできません。
2	デフォルトでは使用されません。自由に定義できます。
3	コマンドラインユーザーインターフェースを使用して、完全なマルチユーザーモードで実行するために使用されます。
4	デフォルトでは使用されません。自由に定義できます。

ランレベル	説明
5	グラフィカルユーザーインターフェースを使用して、完全なマルチユーザーモードで実行するために使用されます。
6	システムを再起動するために使用されます。このランレベルは予約されており、変更することはできません。

稼働しているランレベルを確認するには、次のコマンドを実行します。

```
~]$ runlevel
N 5
```

runlevel コマンドは、以前のランレベルと現在のランレベルを表示します。この場合、番号は 5 になります。これは、グラフィカルユーザーインターフェースを使用して、完全なマルチユーザーモードで実行されていることを意味します。

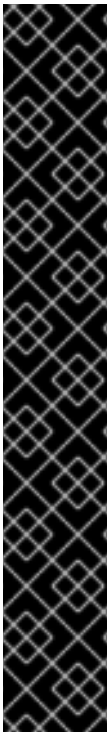
デフォルトのランレベルは、/etc/inittab ファイルを変更して変更できます。このファイルには、以下のようなファイルの末尾付近の行が含まれています。

```
id:5:initdefault:
```

これを行うには、このファイルを root として編集し、この行の番号を必要な値に変更します。この変更は、次回システムを再起動すると有効になります。

12.2. サービスの設定

システムの起動時に開始するサービスを設定できるようにするため、Red Hat Enterprise Linuxnbsp;Hat Enterprise Red Hat Enterprise Linuxnbsp;Linux には、サービス 設定 グラフィカルアプリケーション、ntsysv テキストユーザーインターフェース、および chkconfig コマンドラインツールが含まれます。



IRQBALANCE サービスの有効化

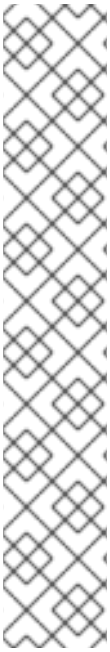
POWER アーキテクチャーで最適なパフォーマンスを確保するために、irqbalance サービスを有効にすることが推奨されます。多くの場合、このサービスは Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;LinuxRed Hat Enterprise Linux 6nbsp;6 のインストール時に実行されるようにインストールおよび設定されます。irqbalance が root として実行されていることを確認するには、シェルプロンプトで以下を入力します。

```
~]# service irqbalance status
irqbalance (pid  1234) is running...
```

グラフィカルユーザーインターフェースを使用してサービスを有効にして実行する方法は、「Service 設定ユーティリティーの使用」 を参照してください。コマンドラインでこれらのタスクを実行する方法は、それぞれ 「chkconfig ユーティリティーの使用」 および 「サービスの実行」 を参照してください。

12.2.1. Service 設定ユーティリティーの使用

Service Configuration ユーティリティーは、Red Hat が開発したグラフィカルアプリケーションで、特定のランレベルで起動するサービスや、メニューからの起動、停止、再起動を行います。ユーティリティーを起動するにはパネルから System → Administration → Services を選択するか、シェルプロンプトで system-config-services コマンドを入力します。



注記

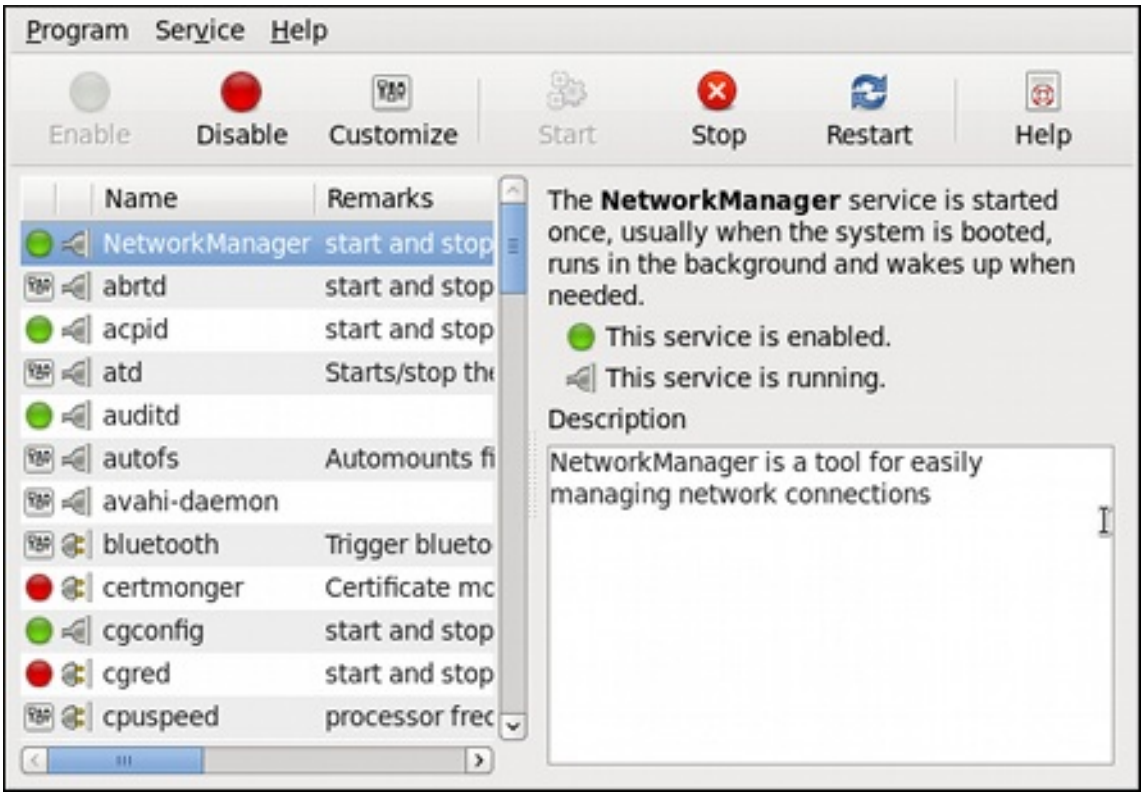
system-config-services ユーティリティーは、system-config-services パッケージで提供されます。これは、お使いの Red Hat Enterprise Linux バージョンにはデフォルトでインストールされない可能性があります。これを確認するには、まず以下のコマンドを実行します。

```
~]$ rpm -q system-config-services
```

パッケージがデフォルトでインストールされていない場合は、root で以下のコマンドを実行して手動でインストールします。

```
~]# yum install system-config-services
```

図12.1 Service Configuration ユーティリティー



[D]

このユーティリティーは、利用可能なすべてのサービス（ /etc/rc.d/init.d/ ディレクトリーのサービス）と、xinetdが制御するサービスの一覧とその説明と現在のステータスを表示します。使用されるアイコンの完全リストと、その意味の説明は、表12.2「考えられるサービス状態」 を参照してください。

すでに認証されていない限り、変更を加えたときにスーパーユーザーパスワードを入力するように求められます。

表12.2 考えられるサービス状態

アイコン	説明
 *[D]*	*サービスが有効になっている。*

アイコン	説明
 [D]	サービスが無効になっています。
 [D]	このサービスは、選択したランレベルでのみ有効になります。
 [D]	サービスが実行している。
 [D]	サービスが停止している。
 [D]	サービスに問題があります。
 [D]	サービスのステータスは不明です。

12.2.1.1. サービスの有効化および無効化

サービスを有効にするには、一覧から選択し、ツールバーの Enable ボタンをクリックするか、メインメニューから Service → Enable を選択します。

サービスを無効にするには、一覧から選択し、ツールバーの Disable ボタンをクリックするか、メインメニューから Service → Disable を選択します。

12.2.1.2. サービスの起動、再起動、停止

サービスを起動するには、一覧からサービスを選択し、ツールバーの Start ボタンをクリックするか、メインメニューから Service → Start を選択します。このオプションはオンデマンドで開始するため、xinetd が制御するサービスには使用できません。

実行中のサービスを再起動するには、一覧からサービスを選択し、ツールバーの Restart ボタンをクリックするか、メインメニューから Service → Restart を選択します。このオプションは、自動的に起動および停止されるため、xinetd が制御するサービスには使用できません。

サービスを停止するには、一覧から選択し、ツールバーの Stop ボタンをクリックするか、メインメニューから Service → Stop を選択します。このオプションは、ジョブの完了時に停止されるため、xinetd が制御するサービスには使用できません。

12.2.1.3. ランレベルの選択

特定のランレベルのサービスのみを有効にするには、リストからサービスを選択し、ツールバーの Customize ボタンをクリックするか、メインメニューから Service → Customize を選択します。次に、サービスを実行する各ランレベルの横にあるチェックボックスを選択します。このオプションは、xinetd が制御するサービスには使用できません。

12.2.2. ntsysv ユーティリティーの使用

ntsysv ユーティリティーは、単純なテキストユーザーインターフェースを使用してコマンドラインアプリケーションで、選択したランレベルで起動するサービスを設定します。ユーティリティーを起動するには、root でシェルプロンプトで ntsysv と入力します。

図12.2 ntsysv ユーティリティー

[D]

このユーティリティーは、利用可能なサービス（ /etc/rc.d/init.d/ ディレクトリーのサービス）と現在のステータスと、F1 を押して取得可能な説明を表示します。使用されているシンボルの一覧と、その意味の説明は、表12.3「考えられるサービス状態」 を参照してください。

表12.3 考えられるサービス状態

記号	説明
[*]	サービスが有効になっている。
[ ]	サービスが無効になっています。

12.2.2.1. サービスの有効化および無効化

サービスを有効にするには、上矢印キーおよび下矢印 キー を使用して一覧に移動し、Spacebar で選択します。アスタリスク(*)が括弧に表示されます。

サービスを無効にするには、上下の矢印キーと下矢印キーを使用してリストに移動し、そのステータスを Spacebar に切り替えます。 括弧内のアスタリスク(*)は消えます。

完了したら、Tab キーを使用して Ok ボタンに移動し、Enter を押して変更を確認します。ntsysv は、実際にサービスを開始または停止しないことに注意してください。サービスをすぐに起動または停

止する必要がある場合は、「サービスの起動」 の説明に従って service コマンドを使用します。

12.2.2.2. ランレベルの選択

デフォルトでは、ntsysv ユーティリティーは、現在のランレベルにのみ影響します。他のランレベルのサービスを有効または無効にするには、root で、設定する各ランレベルを表す追加の --level オプションの後に、0 から 6 の数字を付けてコマンドを実行します。

```
ntsysv --level runlevels
```

たとえば、ランレベル 3 および 5 を設定するには、以下を入力します。

```
~]# ntsysv --level 35
```

12.2.3. chkconfig ユーティリティーの使用

chkconfig ユーティリティーは、選択したサービスの起動に使用するランレベルを指定したり、現在の設定とともに利用可能なすべてのサービスを一覧表示したりできるコマンドラインツールです。リストを除き、このコマンドを使用するにはスーパーユーザー権限が必要です。

12.2.3.1. サービスの一覧表示

システムサービスの一覧（/etc/rc.d/init.d/ ディレクトリーからのサービス、xinetdが制御するサービス）を表示するには、type chkconfig --list を指定するか、引数なしで chkconfig を使用します。以下のような出力が表示されます。

```
~]# chkconfig --list
NetworkManager  0:off   1:off   2:on    3:on    4:on    5:on    6:off
abrtd           0:off   1:off   2:off   3:on    4:off   5:on    6:off
acpid           0:off   1:off   2:on    3:on    4:on    5:on    6:off
anamon          0:off   1:off   2:off   3:off   4:off   5:off   6:off
atd             0:off   1:off   2:off   3:on    4:on    5:on    6:off
auditd          0:off   1:off   2:on    3:on    4:on    5:on    6:off
avahi-daemon    0:off   1:off   2:off   3:on    4:on    5:on    6:off
... several lines omitted ...
wpa_supplicant  0:off   1:off   2:off   3:off   4:off   5:off   6:off

xinetd based services:
        chargen-dgram:  off
        chargen-stream: off
        cvs:            off
        daytime-dgram:  off
        daytime-stream: off
```

```
        discard-dgram:  off
... several lines omitted ...
        time-stream:    off
```

各行には、番号付きの各ランレベルのサービス名（オン または オフ）が続きます。たとえば、上記のリストでは NetworkManager はランレベル 2、3、4、5 で有効になっていますが、abrtd はランレベル 3 と 5 で実行されます。xinetd ベースのサービスは、最後に オン または off に一覧表示されます。

選択したサービスの現在の設定のみを表示するには、chkconfig --list の後にサービス名を指定します。

```
chkconfig --list service_name
```

たとえば、sshd サービスの現在の設定を表示するには、以下を入力します。

```
~]# chkconfig --list sshd
sshd            0:off   1:off   2:on    3:on    4:on    5:on    6:off
```

このコマンドを使用して、xinetd が管理するサービスの状態を表示することもできます。この場合、出力には、サービスが有効または無効なかどうかのみが表示されます。

```
~]# chkconfig --list rsync
rsync           off
```

12.2.3.2. サービスの有効化

ランレベル 2、3、4、および 5 でサービスを有効にするには、root で次のコマンドを実行します。

```
chkconfig service_name on
```

たとえば、4 つのランレベルで httpd サービスを有効にするには、以下を入力します。

```
~]# chkconfig httpd on
```

特定のランレベルでのみサービスを有効にするには、--level オプションに続いて、サービスを実行する各ランレベルを表す 0 から 6 の数字を追加します。

```
chkconfig service_name on --level runlevels
```

たとえば、ランレベル 3 および 5 で abrtd サービスを有効にするには、以下を入力します。

```
~]# chkconfig abrtd on --level 35
```

次回、これらのランレベルのいずれかを入力すると、サービスが起動します。サービスをすぐに起動する必要がある場合は、「サービスの起動」 の説明に従って service コマンドを使用します。

xinetd が管理するサービスで操作する場合は --level オプションは使用し ない でください。これはサポートされていないためです。たとえば、rsync サービスを有効にするには、以下を入力します。

```
~]# chkconfig rsync on
```

xinetd デーモンが実行している場合は、手動でデーモンを再起動しなくても、このサービスはすぐに有効になります。

12.2.3.3. サービスの無効化

ランレベル 2、3、4、および 5 でサービスを無効にするには、root で次のコマンドを実行します。

```
chkconfig service_name off
```

たとえば、4 つのランレベルで httpd サービスを無効にするには、以下を入力します。

```
~]# chkconfig httpd off
```

特定のランレベルでのみサービスを無効にするには、--level オプションの後に 0 から 6 の数字を付けて、サービスを実行したく ない 各ランレベルを追加します。

```
chkconfig service_name off --level runlevels
```

たとえば、ランレベル 2 および 4 で abrtd を無効にするには、以下を入力します。

```
~]# chkconfig abrtd off --level 24
```

次回、これらのランレベルのいずれかを入力すると、このサービスは停止します。サービスをすぐに停止する必要がある場合は、「サービスの停止」 の説明に従って service コマンドを使用します。

xinetd が管理するサービスで操作する場合は --level オプションは使用し ない でください。これはサポートされていないためです。たとえば、rsync サービスを無効にするには、以下を入力します。

```
~]# chkconfig rsync off
```

xinetd デーモンが実行している場合は、手動でデーモンを再起動しなくても、このサービスは直ちに無効になります。

12.3. サービスの実行

service ユーティリティーを使用すると、/etc/init.d/ ディレクトリーからサービスを起動、停止、または再起動できます。

12.3.1. サービスステータスの決定

サービスの現在のステータスを確認するには、シェルプロンプトで以下を入力します。

```
service service_name status
```

たとえば、httpd サービスのステータスを確認するには、以下を入力します。

```
~]# service httpd status
httpd (pid  7474) is running...
```

一度に利用可能なすべてのサービスのステータスを表示するには、service コマンドに --status-all オプションを指定して実行します。

```
~]# service --status-all
abrt (pid  1492) is running...
acpid (pid  1305) is running...
atd (pid  1540) is running...
auditd (pid  1103) is running...
automount (pid 1315) is running...
Avahi daemon is running
cpuspeed is stopped
... several lines omitted ...
wpa_supplicant (pid  1227) is running...
```

「Service 設定ユーティリティーの使用」 で説明されているように、Service Configuration ユー

ティリティーを使用することもできます。

12.3.2. サービスの起動

サービスを起動するには、root で次のコマンドを実行します。

```
service service_name start
```

たとえば、httpd サービスを起動するには、以下を入力します。

```
~]# service httpd start
Starting httpd:                                [  OK  ]
```

12.3.3. サービスの停止

実行中のサービスを停止するには、root で次のコマンドを実行します。

```
service service_name stop
```

たとえば、httpd サービスを停止するには、以下を入力します。

```
~]# service httpd stop
Stopping httpd:                                [  OK  ]
```

12.3.4. サービスの再開

サービスを再起動するには、root で次のコマンドを実行します。

```
service service_name restart
```

たとえば、httpd サービスを再起動するには、以下を入力します。

```
~]# service httpd restart
Stopping httpd:                                [  OK  ]
Starting httpd:                                [  OK  ]
```

12.4. その他のリソース

12.4.1. インストールされているドキュメント

- *chkconfig(8): chkconfig ユーティリティーの man ページです。*
- *ntsysv(8)- ntsysv ユーティリティーの man ページです。*
- *service(8): service ユーティリティーの man ページです。*
- *system-config-services(8)- system-config-services ユーティリティーの man ページです。*

12.4.2. 関連書籍

『Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;LinuxRed Hat Enterprise Linux 6nbsp;6 Security Guide』

Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;LinuxRed Hat Enterprise Linux 6nbsp;6 のセキュリティーを保護するためのガイドこれには、ファイアウォールの設定方法や、SELinux の設定に関する役立つ情報が含まれています。

第13章 認証の設定

認証 は、ユーザーを特定し、システムで検証される方法です。認証プロセスでは、ユーザー名とパスワードなど、ある程度の ID および認証情報を提示する必要があります。その後、認証情報は、システムの一部のデータストアに保存されている情報と比較されます。Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux では、認証設定ツールは LDAP などのユーザー認証情報に使用するデータストアの種類を設定するのに役立ちます。

シングルサインオンの便宜上や、Red Hat Enterprise Linuxnbsp;Hat Enterprise Red Hat Enterprise Linuxnbsp;Linux は中央のデーモンを使用して、多数の異なるデータストアのユーザー認証情報を保存できます。SSSD(System Security Services Daemon)は、LDAP、Kerberos、および外部アプリケーションと対話して、ユーザーの認証情報を検証できます。Authentication Configuration Tool は、NIS、Winbind、および LDAP とともに SSSD を設定して、認証処理とキャッシュを組み合わせることができます。

13.1. システム認証の設定

ユーザーが Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux システムにログインすると、そのユーザーはユーザー ID を確立するためにいくつかの 認証 情報を提示します。次に、システムは、設定された認証サービスに対してこれらの認証情報を確認します。認証情報が一致し、ユーザーアカウントがアクティブであれば、ユーザーは 認証されます。（ユーザーが認証されると、ユーザーが実行できるものを判別するためにアクセス制御サービスへ情報が渡されます。ユーザーがアクセスが 許可 されるリソースです。）

ユーザーがローカルシステムに配置できるか、ローカルシステムが LDAP や Kerberos などのリモートシステムのユーザーデータベースを参照できます。

システムは、ユーザー認証を確認するために、有効なアカウントデータベースの一覧を設定している。Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux では、認証設定ツールには、ユーザーデータストアを設定するための GUI とコマンドラインオプションの両方があります。

ローカルシステムは、ユーザー情報にさまざまなデータストアを使用できます。たとえば、Lightweight Directory Access Protocol (LDAP)、ネットワーク情報サービス(NIS)、および Winbind があります。さらに、LDAP データストアと NIS データストアの両方が Kerberos を使用してユーザーを認証することができます。

重要

インストール時またはセキュリティーレベル(Security Level Configuration Tool)で中程度または高レベルのセキュリティーレベルが設定されている場合、ファイアウォールは NIS 認証を防ぎます。ファイアウォールに関する詳しい情報は、『『セキュリティーガイド』の「ファイアウォール」セクションを参照して』ください。

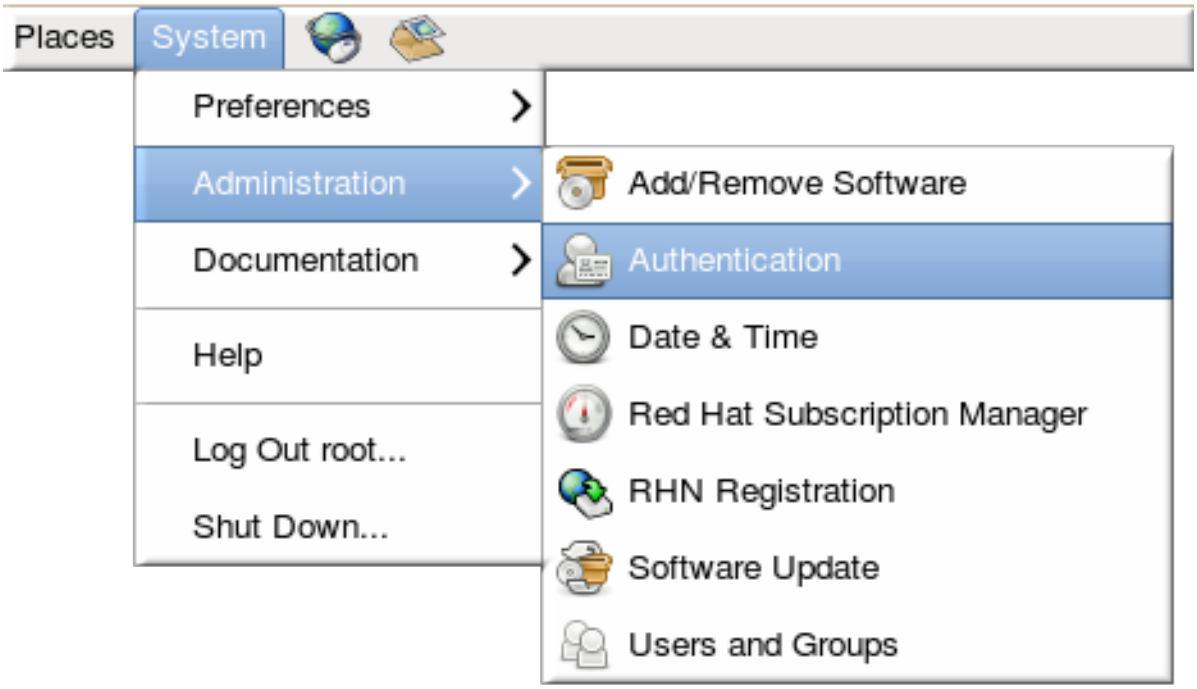
13.1.1. Authentication Configuration Tool UI の起動

1. *root でシステムにログインします。*

2. *システム を開きます。*

3. *Administration メニューを選択します。*

4. *Authentication 項目を選択します。*



または、system-config-authentication コマンドを実行します。

重要

Authentication Configuration Tool UI が閉じられると、変更はすぐに有効になります。

認証 ダイアログボックスには、設定タブが 2 つあります。

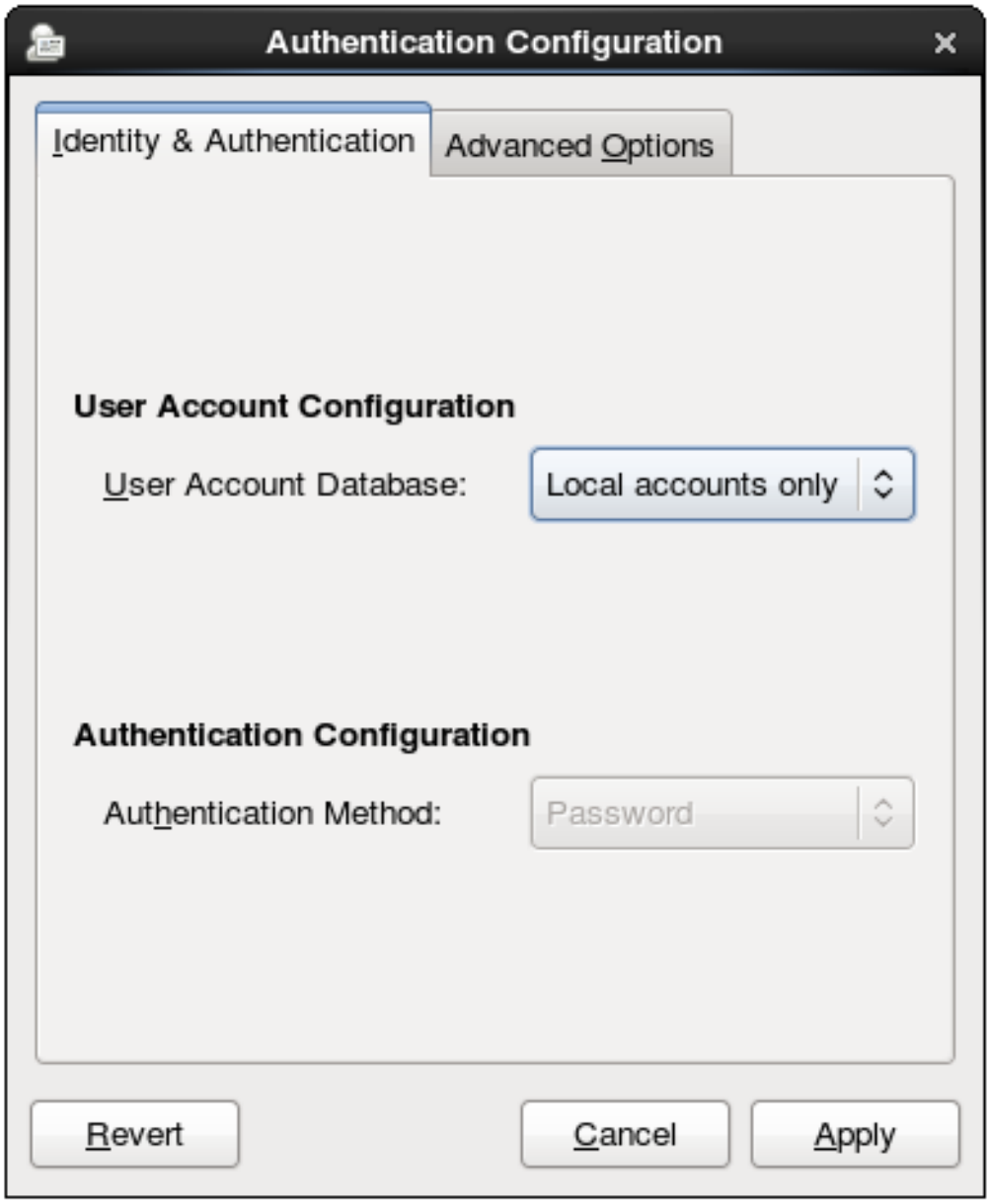
- ID および認証。アイデンティティーストアとして使用するリソース (ユーザー ID と対応する認証情報が保存されるデータリポジトリー) を設定します。

- 高度なオプション。スマートカードやフィンガープリントなどのパスワードや証明書以外の認証方法を許可します。

13.1.2. 認証用のアイデンティティーストアの選択

Identity & Authentication タブは、ユーザーの認証方法を設定します。デフォルトでは、ローカルシステム認証を使用します。つまり、ユーザーとパスワードは、ローカルシステムのアカウントに対して確認されます。Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux マシンは、LDAP、NIS、Winbind などのユーザーおよび認証情報を含む外部リソースを使用することもできます。

図13.1 ローカル認証



13.1.2.1. LDAP 認証の設定

openldap-clients パッケージまたは sssd パッケージは、ユーザーデータベースの LDAP サーバーを設定するのに使用されます。デフォルトでは、両方のパッケージがデフォルトでインストールされます。

1. *「Authentication Configuration Tool UI の起動」 のとおりに、認証設定ツールを開きます。*

2. *User Account Database ドロップダウンメニューで LDAP を選択します。*

3.

LDAP サーバーへの接続に必要な情報を設定します。

- *LDAP 検索ベース DN は、ユーザーディレクトリーのルートサフィックスまたは 識別名 (DN) を提供します。ID/認証に使用されるユーザーエントリーはすべて、この親エントリーの下に存在します。例：* **ou=people,dc=example,dc=com**

  このフィールドは任意です。これが指定されていない場合、System Security Services Daemon(SSSD)は、LDAP サーバーの設定エントリーの namingContexts および defaultNamingContext 属性を使用して検索ベースの検出を試みます。

- *LDAP サーバー は LDAP サーバーの URL を提供します。これには通常、* **ldap://ldap.example.com:389** *などの LDAP サーバーのホスト名およびポート番号の*

両方が必要です。

URL ldaps:// にセキュアなプロトコルを入力すると、Download CA Certificate ボタンが有効になります。

- TLS を使用して接続を暗号化する かどうかは、Start TLS を使用して LDAP サーバーへの接続を暗号化するかどうかを設定します。これにより、標準ポートを介したセキュアな接続が可能になります。

TLS を選択すると、Download CA Certificate ボタンが有効になります。これは、発行した認証局から LDAP サーバーの発行元の CA 証明書を取得します。CA 証明書は、プライバシーにより強化されたメール(PEM)形式である必要があります。

重要

サーバー URL がセキュアなプロトコル(ldaps)を使用する場合は、TLS を使用して接続を暗号化する ことはできませ ん。このオプションは、標準ポートでセキュアな接続を開始する Start TLS を使用します。セキュアなポートが指定されている場合は、TLS の代わりに SSL などのプロトコルを使用する必要があります。

4. 認証方法を選択します。LDAP は、簡単なパスワード認証または Kerberos 認証を許可します。

Kerberos の使用については、「LDAP または NIS 認証での Kerberos の使用」 を参照してください。

LDAP パスワード オプションは、PAM アプリケーションで LDAP 認証を使用します。このオプションには、LDAP サーバーに接続するためにセキュアな(ldaps://)URL または TLS オプションのいずれかが必要です。

13.1.2.2. NIS 認証の設定

1. ypbind パッケージをインストールします。これは NIS サービスに必要ですが、デフォルトではインストールされません。

```
~]# yum install ypbind
```

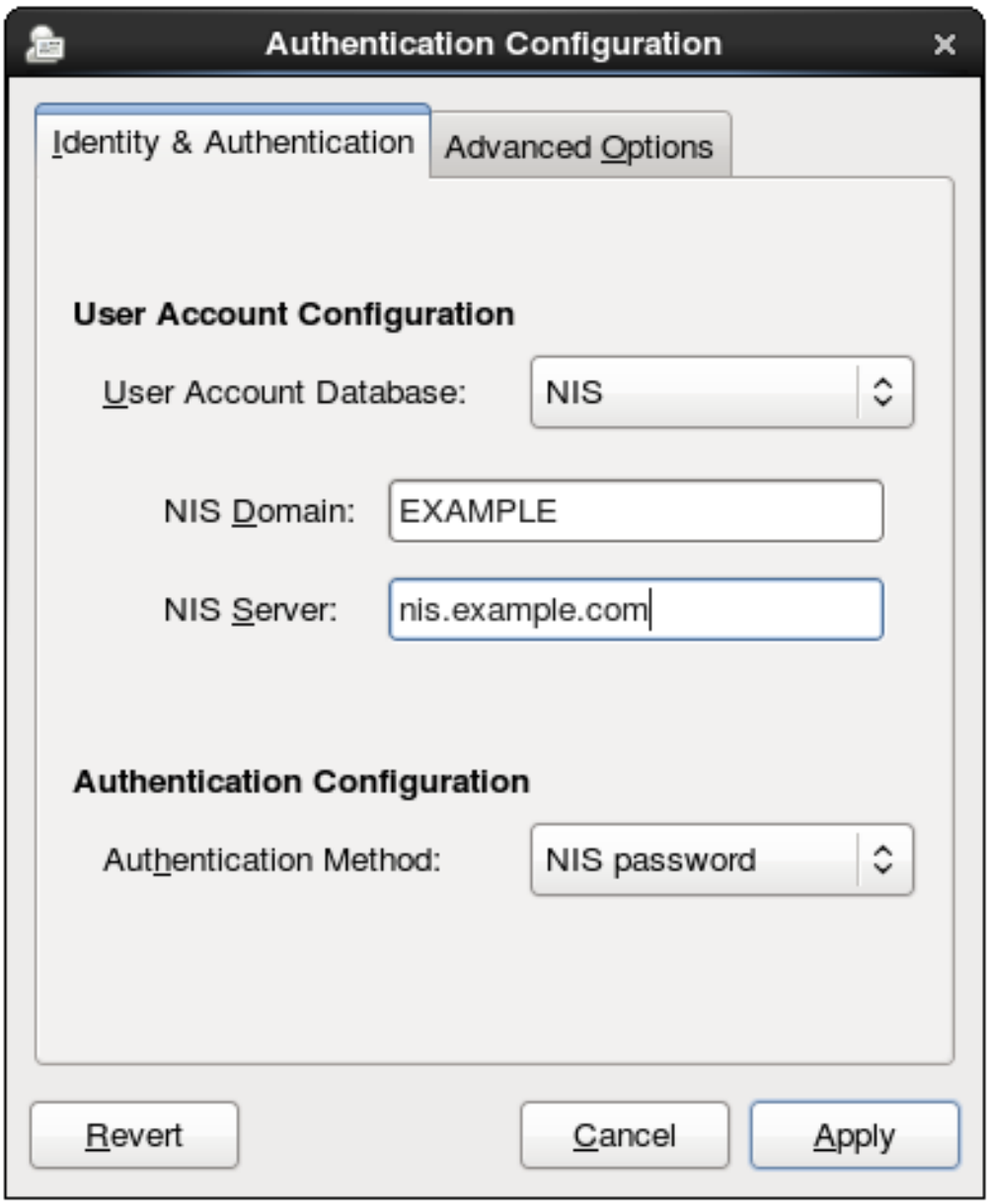
ypbind サービスをインストールすると、ポートマップと yp bind サービスが開始し、システムの起動時に開始できるようになります。

2.

「Authentication Configuration Tool UI の起動」 のとおりに、認証設定ツールを開きます。

3.

User Account Database ドロップダウンメニューで NIS を選択します。



4.

NIS サーバーに接続する情報を設定します。つまり、NIS ドメイン名およびサーバーホスト名になります。NIS サーバーが指定されていない場合、authconfig デーモンは NIS サーバーをスキャンします。

5.

認証方法を選択します。NIS は、簡単なパスワード認証または Kerberos 認証を許可します。

Kerberos の使用については、「LDAP または NIS 認証での Kerberos の使用」 を参照してください。

NIS の詳細は、『『セキュリティーガイド』』 の「NIS のセキュア化」セクションを参照してください。

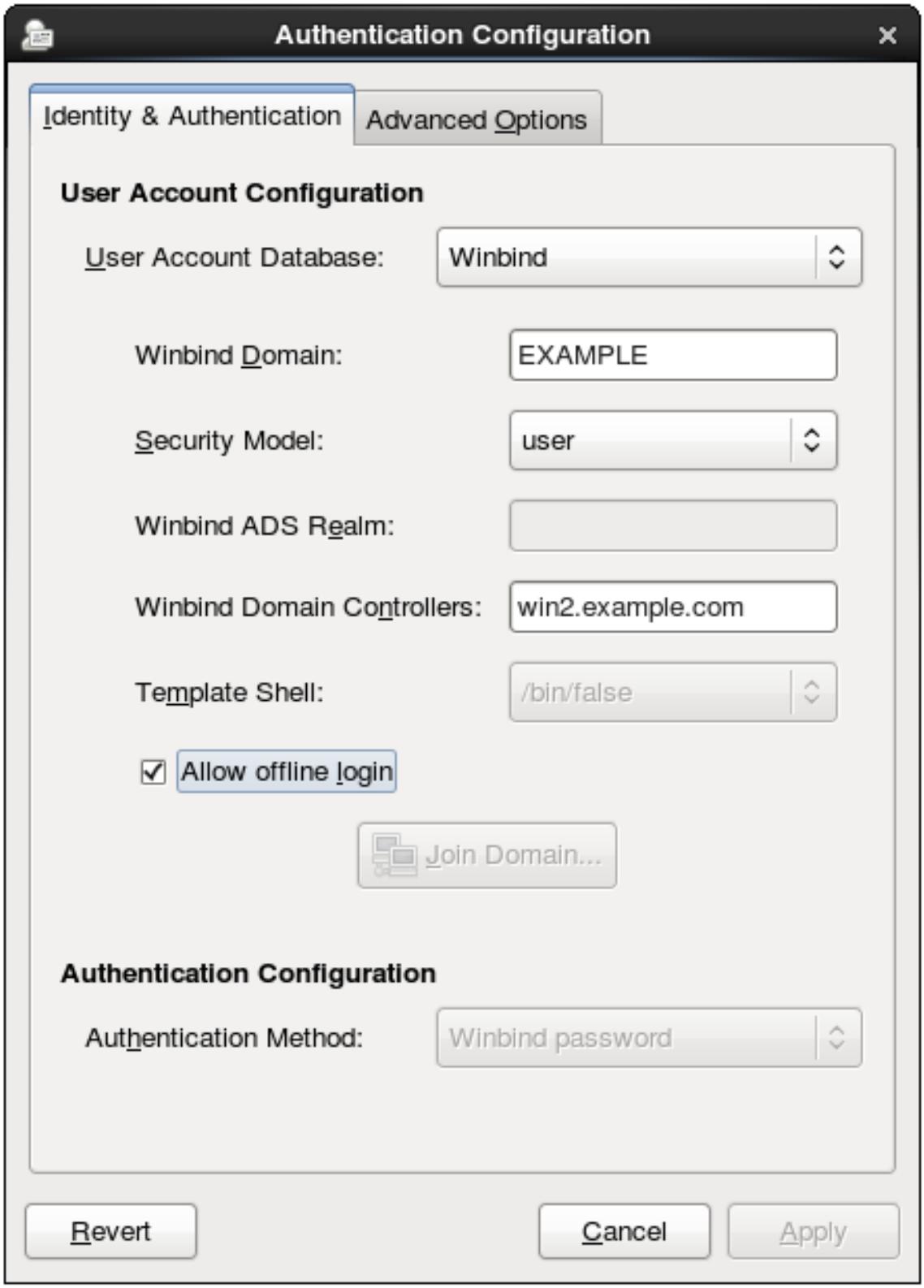
13.1.2.3. Winbind 認証の設定

1. samba-winbind パッケージをインストールします。これは、Samba サービスの Windows 統合機能に必要ですが、デフォルトではインストールされません。

```
~]# yum install samba-winbind
```

2. 「Authentication Configuration Tool UI の起動」 のとおりに、認証設定ツールを開きます。

3. User Account Database ドロップダウンメニューで Winbind を選択します。



4. ***Microsoft Active Directory ドメインコントローラーへの接続に必要な情報を設定します。***

   - ***Winbind ドメイン は、接続する Windows ドメインを提供します。***

これは、DOMAIN などの Windows 2000 形式である必要があります。

- *セキュリティーモデル は、Samba クライアントに使用するセキュリティーモデルを設定します。authconfig は、以下の 4 種類のセキュリティーモデルをサポートします。*

  - *ads は、Samba が Active Directory Server レルムのドメインメンバーとして動作するように設定します。このモードで実行するには、krb5-server パッケージをインストールして、Kerberos を適切に設定する必要があります。また、コマンドラインを使用して Active Directory Server に参加する場合は、以下のコマンドを使用する必要があります。*

    ```
    net ads join
    ```

  - *ドメイン には、Windows サーバーなど、Windows プライマリーまたはバックアップドメインコントローラーで認証することで、Samba がユーザー名/パスワードを検証します。*

  - *サーバー には、Windows サーバーなどの別のサーバーで認証することで、ローカルの Samba サーバーがユーザー名/パスワードを検証します。サーバーの認証を試みると、システムは user モードで認証を試行します。*

  - *ユーザー には、有効なユーザー名およびパスワードでクライアントにログインする必要があります。このモードは、暗号化されたパスワードに対応します。*

    ユーザー名の形式は **domain\user** *(例: EXAMPLE\jsmith) である必要があります。*

注記

指定したユーザーが Windows ドメインに存在することを確認するには、常に Windows 2000 形式の形式を使用してバックスラッシュ (\)文字をエスケープしてください。以下に例を示します。

```
~]# getent passwd domain\\user
DOMAIN\user:*:16777216:16777216:Name Surname:/home/DOMAIN/user:/bin/bash
```

以下はデフォルトのオプションになります。

- Winbind ADS レルム は、Samba サーバーが参加する Active Directory レルムを提供します。これは、ads セキュリティーモデルとのみ使用されます。
- winbind ドメインコントローラーは、使用するドメインコントローラーを提供します。ドメインコントローラーの詳細は、「ドメインコントローラー」 を参照してください。
- テンプレート シェルは、Windows ユーザーアカウント設定に使用するログインシェルを設定します。
- オフラインログインを許可する と、認証情報をローカルキャッシュに保存できます。システムがオフライン時に、ユーザーがシステムリソースに認証を試みるとキャッシュが参照されます。

Winbind サービスの詳細は、「Samba デーモンと関連サービス」 を参照してください。

Winbind およびトラブルシューティングのヒントの詳細は、Red Hat カスタマーポータル の ナレッジベース を参照してください。

また、Red Hat Access Labs ページには、Red Hat Enterprise Linux を Active Directory に接続

するのに役立つ smb.conf ファイルの一部を生成する Winbind Mapper ユーティリティーが含まれます。

13.1.2.4. LDAP または NIS 認証での Kerberos の使用

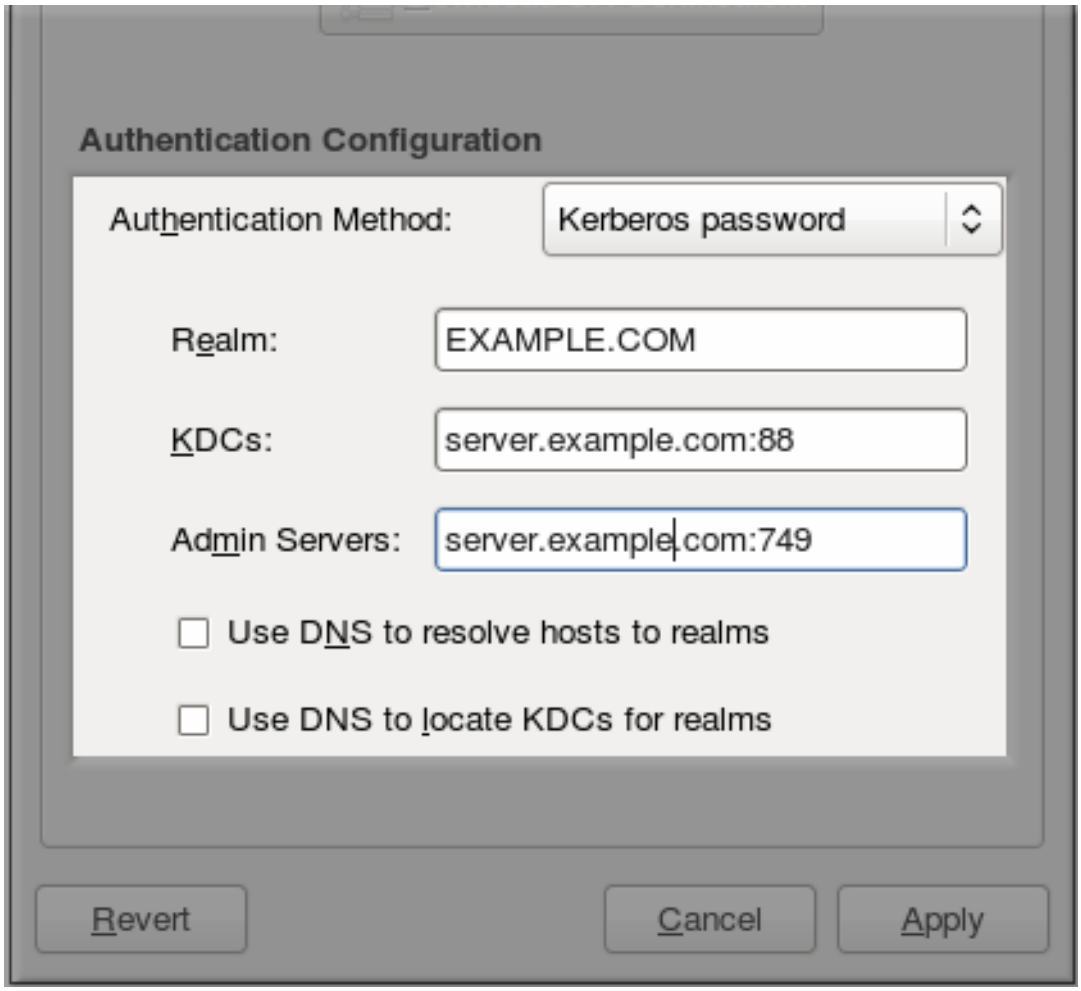
LDAP および NIS の認証ストアは、Kerberos 認証方法をサポートします。Kerberos の使用には、以下のような利点があります。

- これは、標準ポートを介した接続を許可する一方で、通信にセキュリティーレイヤーを使用します。
- SSSD で認証情報キャッシュを自動的に使用します。これにより、オフラインログインが可能になります。

Kerberos 認証を使用するには、krb5-libs および krb5-workstation パッケージが必要です。

Authentication Method ドロップダウンメニューの Kerberos パスワード オプションは、Kerberos レルムへの接続に必要なフィールドを自動的に開きます。

図13.2 Kerberos フィールド



- *レルム は、Kerberos サーバーのレルム名を指定します。レルムは、Kerberos を使用するネットワークで、1 つ以上の キー配布センター (KDC) と、潜在的に多数のクライアントで構成されます。*

- *KDC は、Kerberos チケットを発行するサーバーのコンマ区切りリストを提供します。*

- *管理サーバー は、レルムで kadmind プロセスを実行する管理サーバーのリストを提供します。*

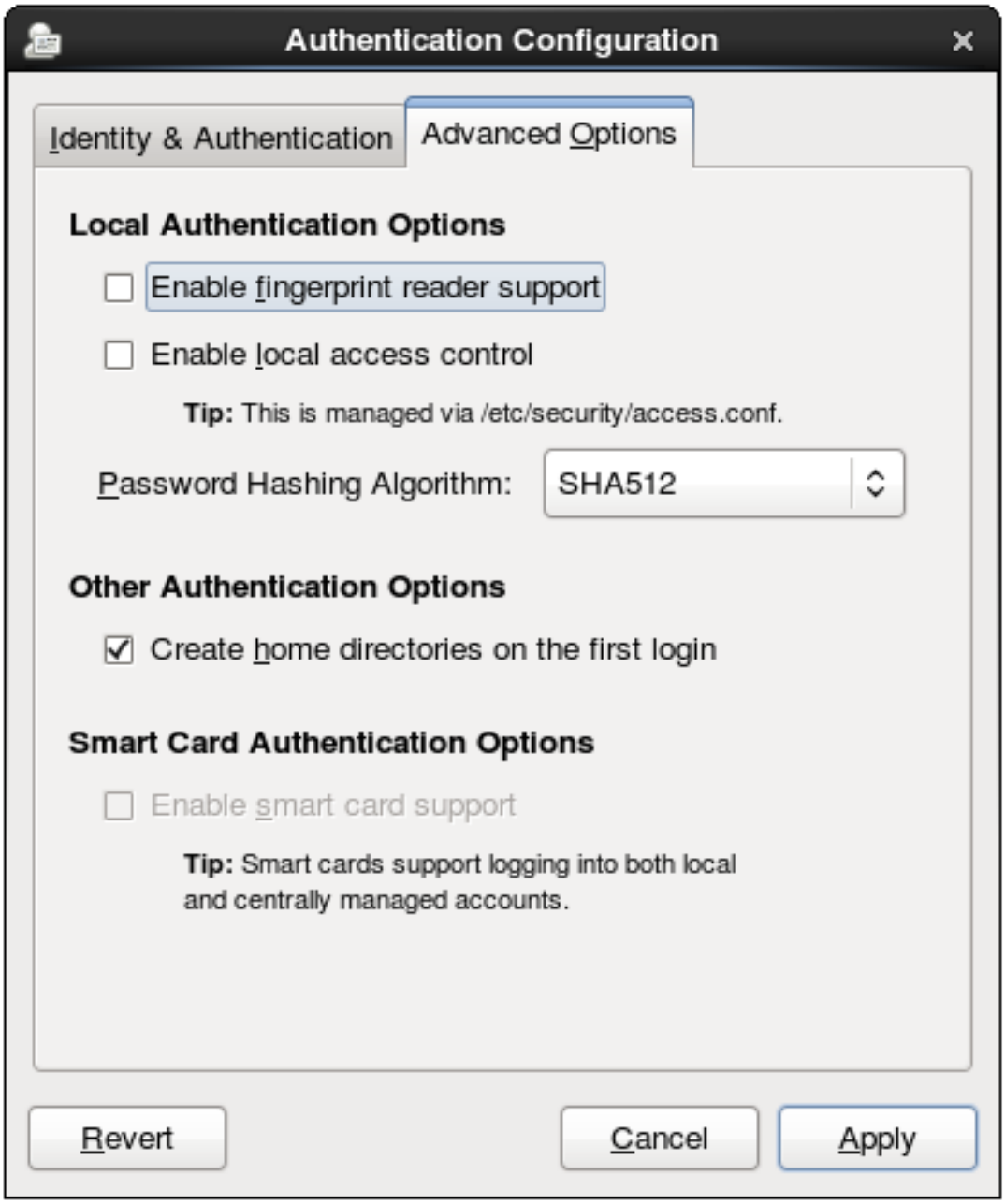
- *必要に応じて、DNS を使用してサーバーのホスト名を解決し、レルム内で追加の KDC を見つけます。*

Kerberos の詳細は、『Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;LinuxRed Hat Enterprise Linux 6nbsp;6 『Managing Single Sign-On and Smart Cards』 』の「Using Kerberos」を参照してください。

13.1.3. 代替認証機能の設定

Authentication Configuration Tool は、アイデンティティーストアとは別に、認証の動作に関連する設定も設定します。これには、完全に異なる認証方法（フィンガープリントスキャンとスマートカード）またはローカル認証ルールが含まれます。これらの代替認証オプションは、Advanced Options タブで設定します。

図13.3 高度な設定



13.1.3.1. フィンガープリント認証の使用

適切なハードウェアが利用可能な場合は、フィンガープリントリーダーのサポート オプションを使用すると、他の認証情報に加えて、フィンガープリントスキャンを使用してローカルユーザーの認証を行うことができます。

13.1.3.2. ローカル認証パラメーターの設定

ローカル認証オプション エリアには、ローカルシステムで認証の動作を定義する 2 つのオプションがあります。

- ローカルアクセス制御を有効 にすると、/etc/security/access.conf ファイルに、ローカルユーザー承認ルールを確認するように指示します。

- パスワードハッシュアルゴリズムは 、 ローカルに保存されたパスワードの暗号化に使用するハッシュアルゴリズムを設定します。

13.1.3.3. スマートカード認証の有効化

適切なスマートカードリーダーが利用可能な場合、システムは他のユーザー認証情報ではなくスマートカード（または トークン）を受け入れて認証できます。

Enable smart card support オプションを選択すると、スマートカード認証の動作を定義できます。

- カード削除アクション は、アクティブなセッション中にカードがカードリーダーから削除されるときにシステムに応答する方法を指示します。システムは削除を無視し、ユーザーが通常通りにリソースにアクセスできるようにしたり、スマートカードが提供されるまでシステムが直ちにロックできます。

- ログ インにスマートカード が必要であるか、ログインに許可されているか、スマートカードのログインが必要です。このオプションを選択すると、認証の他のすべての方法がすぐにブロックされます。

警告

スマートカードを使用してシステムに正常に認証されるまで、このオプションを選択しないでください。

スマートカードを使用するには、pam_pkcs11 パッケージが必要です。

13.1.3.4. ユーザーホームディレクトリーの作成

オプション（初回ログイン時にホームディレクトリーの作成）があり、ユーザーの初回ログイン時にホームディレクトリーを自動的に作成します。

このオプションは、LDAP など、集中管理されているアカウントでは利点があります。ただし、自動マウントなどのシステムがユーザーのホームディレクトリーの管理に使用される場合、このオプションは選択しないでください。

13.1.4. コマンドラインでの認証設定

authconfig コマンドラインツールは、スクリプトに渡される設定に従って、システム認証に必要な設定ファイルおよびサービスをすべて更新します。UI で設定できるすべての ID および認証設定オプションを許可すると共に、authconfig ツールを使用してバックアップおよびキックスタートファイルを作成することもできます。

authconfig オプションの完全なリストは、help 出力と man ページを参照してください。

13.1.4.1. authconfig を使用するためのヒント

authconfig を実行する際に注意すべき事項があります。

- すべてのコマンドで、--update オプションまたは --test オプションのいずれかを使用します。コマンドを正常に実行するには、以下のオプションのいずれかが必要です。--update を使用すると、設定の変更が書き込まれます。--test は変更を stdout に出力しますが、設定の変更は適用されません。

- 有効化オプションには、それぞれ対応する無効化オプションがあります。

13.1.4.2. LDAP ユーザーストアの設定

LDAP アイデンティティーストアを使用するには、--enableldap を使用します。LDAP を認証ソースとして使用する場合は、--enableldapauth を使用して、LDAP サーバー名、ユーザーサフィックス

のベース DN、TLS を使用するかどうか (オプション) など、必要な接続情報を使用します。authconfig コマンドには、ユーザーエントリーの RFC 2307bis スキーマを有効または無効にするオプションもありますが、認証設定 UI では不可能です。

プロトコル (ldap または ldaps) およびポート番号を含む完全な LDAP URL を使用するようにしてください。--enableldaptls オプションでセキュアな LDAP URL (ldaps) を使用 しないでください。

```
authconfig --enableldap --enableldapauth --
ldapserver=ldap://ldap.example.com:389,ldap://ldap2.example.com:389 --
ldapbasedn="ou=people,dc=example,dc=com" --enableldaptls --
ldaploadcacert=https://ca.server.example.com/caCert.crt --update
```

LDAP パスワード認証に --ldapauth を使用する代わりに、LDAP ユーザーストアで Kerberos を使用できます。これらのオプションは、「Kerberos 認証の設定」 で説明されています。

13.1.4.3. NIS ユーザーストアの設定

NIS アイデンティティーストアを使用するには、--enablenis を使用します。Kerberos パラメーターが明示的に設定されていない限り、NIS 認証が自動的に使用されるため、Kerberos 認証 (「Kerberos 認証の設定」)が使用されます。NIS サーバーおよび NIS ドメインを特定するためのパラメーターのみ。これを使用しないと、authconfig サービスは NIS サーバーのネットワークをスキャンします。

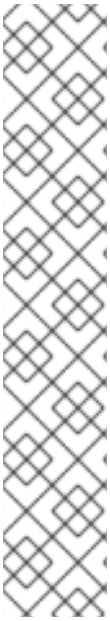
```
authconfig --enablenis --nisdomain=EXAMPLE --nisserver=nis.example.com --update
```

13.1.4.4. Winbind ユーザーストアの設定

Windows ドメインには複数のセキュリティーモデルがあり、ドメインで使用されるセキュリティーモデルによってローカルシステムの認証設定が決定します。

ユーザーおよびサーバーのセキュリティーモデルの場合、Winbind の設定にはドメイン（またはワークグループ）名とドメインコントローラーのホスト名のみが必要です。

```
authconfig --enablewinbind --enablewinbindauth --smbsecurity=user|server  --
enablewinbindoffline --smbservers=ad.example.com --smbworkgroup=EXAMPLE --update
```



注記

ユーザー名の形式は **domain\user** *(例: EXAMPLE\jsmith) である必要があります。*

指定したユーザーが Windows ドメインに存在することを確認するには、常に Windows 2000 形式の形式を使用してバックスラッシュ(\)文字をエスケープしてください。以下に例を示します。

```
~]# getent passwd domain\\user DOMAIN\user:*:16777216:16777216:Name Surname:/home/DOMAIN/user:/bin/bash
```

ads および domain セキュリティーモデルの場合、Winbind は、テンプレートシェルおよびレルム (adsのみ)への追加の設定を可能にします。以下に例を示します。

```
authconfig --enablewinbind --enablewinbindauth --smbsecurity ads  --enablewinbindoffline --smbservers=ad.example.com --smbworkgroup=EXAMPLE --smbrealm EXAMPLE.COM --winbindtemplateshell=/bin/sh --update
```

Windowsベースの認証とWindowsユーザーアカウントの情報を構成するためのオプションは他にもたくさんあります。たとえば、名前の形式、ユーザー名とともにドメイン名を要求するかどうか、UIDの範囲などです。これらのオプションは authconfig ヘルプに一覧表示されます。

13.1.4.5. Kerberos 認証の設定

LDAP および NIS の両方で、Kerberos 認証をネイティブ認証メカニズムの代わりに使用できます。少なくとも、Kerberos認証を使用するには、レルム、KDC、および管理サーバーを指定する必要があります。また、DNS を使用してクライアント名を解決し、追加の管理サーバーを検索するオプションもあります。

```
authconfig NIS or LDAP options --enablekrb5 --krb5realm EXAMPLE --krb5kdc kdc.example.com:88,server.example.com:88 --krb5adminserver server.example.com:749 --enablekrb5kdcdns --enablekrb5realmdns --update
```

13.1.4.6. ローカル認証設定の構成

また、認証設定ツールは、ホームディレクトリーの作成、パスワードハッシュアルゴリズムの設定、承認などのセキュリティーに関連するユーザー設定を制御することもできます。これらの設定は、アイデンティティー/ユーザーストア設定とは独立して行われます。

たとえば、ユーザーのホームディレクトリーを作成するには、次のコマンドを実行します。

```
authconfig --enablemkhomedir --update
```

ユーザーパスワードの暗号化に使用するハッシュアルゴリズムを設定または変更するには、以下を実行します。

```
authconfig --passalgo=sha512 --update
```

13.1.4.7. フィンガープリント認証の設定

フィンガープリントリーダーのサポートを有効にするオプションは 1 つあります。このオプションは、単独または他の authconfig 設定 (LDAP ユーザーストアなど) と併用できます。

```
~]# authconfig --enablefingerprint --update
```

13.1.4.8. スマートカード認証の設定

システムでスマートカードを使用するために必要なのは 、 --enablesmartcard オプションを設定することだけです。

```
~]# authconfig --enablesmartcard --update
```

スマートカードの他の設定オプションには、その他にもデフォルトのスマートカードモジュールの変更、スマートカードの削除時にシステムの動作の設定、ログイン用のスマートカードの要求などの設定オプションがあります。

たとえば、次のコマンドは、スマートカードが削除された場合にすぐにユーザーをロックアウトするようにシステムに指示します（スマートカードが削除された場合、1 の設定は無視します）。

```
~]# authconfig --enablesmartcard --smartcardaction=0 --update
```

スマートカード認証が正常に設定およびテストされたら、シンプルなパスワードベースの認証ではなく、ユーザーにスマートカード認証を要求するようにシステムを設定できます。

```
~]# authconfig --enablerequiresmartcard --update
```

警告

スマートカードを使用してシステムに正常に認証されるまで --enablerequiresmartcard オプションを使用しないでください。それ以外の場合は、ユーザーがシステムにログインできなくなる可能性があります。

13.1.4.9. キックスタートファイルと設定ファイルの管理

--update オプションは、設定の変更ですべての設定ファイルを更新します。いくつかの代替オプションがあり、動作が若干異なります。

- *--kickstart は、更新された設定をキックスタートファイルに書き込みます。*
- *--test は、全設定を標準出力(stdout)に変更しますが、設定ファイルは編集されません。*

さらに、authconfig を使用して以前の設定をバックアップおよび復元できます。すべてのアーカイブは /var/lib/authconfig/ ディレクトリーの一意のサブディレクトリーに保存されます。たとえば、--savebackup オプションは、バックアップディレクトリーを 2011-07-01 として指定します。

```
~]# authconfig --savebackup=2011-07-01
```

これにより、/var/lib/authconfig/backup-2011-07-01 ディレクトリーの下にあるすべての認証設定ファイルのバックアップが作成されます。

保存されたバックアップは、--restorebackup オプションを使用して設定を復元し、手動で保存した設定の名前を指定します。

```
~]# authconfig --restorebackup=2011-07-01
```

また、authconfig は、 (--update オプションを使用して) 変更を適用する前に設定のバックアップを作成します。設定は、--restorelastbackup オプションを使用して完全バックアップを指定することなく、最新の自動バックアップから復元できます。

13.1.5. カスタムホームディレクトリーの使用

LDAP ユーザーが /home にないホームディレクトリーを持っていて、ユーザーが最初にログインしたときにホームディレクトリーを作成するようにシステムが構成されている場合、これらのディレクトリーは誤った権限で作成されます。

1. /home ディレクトリーから、ローカルシステムで作成されたホームディレクトリーに、正しい SELinux コンテキストおよびパーミッションを適用します。以下に例を示します。

```
~]# semanage fcontext -a -e /home /home/locale
```

2. システムに oddjob-mkhomedir パッケージをインストールします。

   このパッケージは、認証設定ツールがホームディレクトリーの作成に使用する pam_oddjob_mkhomedir.so ライブラリーを提供します。デフォルトの pam_mkhomedir.so ライブラリーとは異なり、pam_oddjob_mkhomedir.so ライブラリーでは、SELinux ラベルを作成できます。

   Authentication Configuration Tool は、利用可能な場合は pam_oddjob_mkhomedir.so ライブラリーを自動的に使用します。それ以外の場合、デフォルトでは pam_mkhomedir.so が使用されます。

3. oddjobd サービスが実行中であることを確認します。

4. 「代替認証機能の設定」 にあるように、Authentication Configuration Tool を再度実行し、ホームディレクトリーを有効にします。

ホームディレクトリーの設定を変更する前にホームディレクトリーが作成された場合は、パーミッションと SELinux コンテキストを修正します。以下に例を示します。

```
~]# semanage fcontext -a -e /home /home/locale
# restorecon -R -v /home/locale
```

13.2. SSSD での認証情報の使用およびキャッシュ

SSSD(System Security Services Daemon)は、さまざまな ID プロバイダーおよび認証プロバイダーへのアクセスを提供します。

13.2.1. SSSD について

システム認証の多くは、ローカルで設定されているので、サービスは、ローカルユーザーストアをチェックして、ユーザーと認証情報を判断する必要があります。SSSD の機能により、ローカルサービスが SSSD のローカルキャッシュをチェックできますが、キャッシュはさまざまなリモート アイデンティティープロバイダー（LDAP ディレクトリー、Identity Management ドメイン、Active Directory、Kerberos レルムなど）から取得できます。

SSSD はこれらのユーザーおよび認証情報をキャッシュするため、ローカルシステム また はアイデンティティープロバイダーがオフラインになると、サービスに対するユーザーの認証情報が引き続き利用できます。

SSSD は、ローカルクライアントと設定されたデータストアとの間に仲介されます。この関係には、管理者にとって多くの利点があります。

- ID/認証サーバーの負荷を削減します。すべてのクライアントサービスが識別サーバーに直接アクセスしようとするのではなく、すべてのローカルクライアントは、識別サーバーに接続したり、そのキャッシュをチェックできる SSSD に問い合わせることができます。

- オフライン認証を許可します。SSSD は、必要に応じて、リモートサービスから取得するユーザー ID および認証情報のキャッシュを維持できます。これにより、リモート識別サーバーがオフラインの場合やローカルマシンがオフラインであっても、ユーザーはリソースに対して正常に認証できます。

- 単一ユーザーアカウントの使用 リモートユーザーには、ローカルシステム用のユーザーアカウントと組織システム用のユーザーアカウントなど、2 つ（または複数）ユーザーアカウントが頻繁にあります。これは、仮想プライベートネットワーク(VPN)に接続するために必要です。SSSD はキャッシュおよびオフライン認証をサポートするため、リモートユーザーはローカルマシンに認証し、SSSD がネットワーク認証情報を維持することでネットワークリソースに接続できます。

その他のリソース

本章では、SSSD でのサービスとドメインの設定の基本を説明しますが、これは包括的なリソースではありません。SSSD の各機能エリアには、その他の多くの設定オプションを利用できます。オプションの完全な一覧を表示するには、特定の機能エリアの man ページを参照してください。

一般的な man ページのいくつかは、表13.1「SSSD man ページのサンプリング」 に記載されています。sssd(8) man ページの「See Also」セクションには、SSSD の man ページの完全なリストもあります。

表13.1 SSSD man ページのサンプリング

機能エリア	man ページ
一般的な設定	sssd.conf(8)
sudo Services	sssd-sudo
LDAP ドメイン	sssd-ldap
Active Directory ドメイン	sssd-ad sssd-ldap
Identity Management（IdM または IPA）ドメイン	sssd-ipa sssd-ldap
ドメインの Kerberos 認証	sssd-krb5
OpenSSH キー	sss_ssh_authorizedkeys sss_ssh_knownhostsproxy
キャッシュのメンテナンス	sss_cache (cleanup) sss_useradd、sss_usermod、sss_userdel、 sss_seed（ユーザーキャッシュエントリー管理）

13.2.2. sssd.conf ファイルの設定

SSSD サービスおよびドメインは、.conf ファイルで設定されます。デフォルトでは、これは /etc/sssd/sssd.conf になりますが、SSSD はインストール後に設定されていないため、このファイルは手動で作成および設定する必要があります。

13.2.2.1. sssd.conf ファイルの作成

SSSD 設定ファイルには、以下の 3 つの部分があります。

- *[SSSD]、一般的な SSSD プロセスと運用設定用。基本的には、それぞれの設定済みサービス、ドメイン、および設定パラメーターを一覧表示します。*

- [service_name]、*サポートされている各システムサービスの設定オプションの場合（を参照）* *「SSSD およびシステムサービス」*

- [domain_type/DOMAIN_NAME]、*設定済みの各アイデンティティープロバイダーの設定オプションの場合*

重要

サービスはオプションですが、SSSD サービスを起動する前に少なくとも 1 つのアイデンティティープロバイダードメインを設定する必要があります。

例13.1 シンプルな sssd.conf ファイル

```
[sssd]
domains = LOCAL
services = nss
config_file_version = 2

[nss]
filter_groups = root
filter_users = root

[domain/LOCAL]
id_provider = local
auth_provider = local
access_provider = permit
```

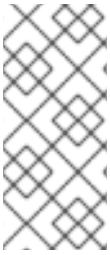
[sssd] セクションには、3 つの重要なパラメーターがあります。

- *ドメイン は、SSSD がアイデンティティープロバイダーとして使用する sssd.conf で設定されているすべてのドメインを一覧表示します。ドメインが domains キーに記載されていない場合は、configuration セクションがある場合でも SSSD では使用されません。*

- *サービス は、SSSD を使用する sssd.conf で設定されるすべてのシステムサービスを一*

覧表示します。SSSD が起動すると、設定された各システムサービスについて対応する SSSD サービスが起動します。サービスが services キーに記載されていない場合は、configuration セクションがある場合でも SSSD では使用されません。

- config_file_version は、ファイル形式の期待値を設定する設定ファイルのバージョンを設定します。最新の SSSD バージョンの場合、これはバージョン 2 です。



注記

サービスまたは ドメイン が sssd.conf ファイルで設定されている場合でも、[sssd] セクションにそれぞれサービスまたはドメインが表示されていない限り、SSSD はその サービス またはドメインと対話しません。

その他の設定パラメーターは sssd.conf の man ページに記載されています。

各サービスとドメインパラメーターは、本章と man ページの該当する設定セクションで説明しています。

13.2.2.2. カスタム設定ファイルの使用

デフォルトでは、sssd プロセスでは、設定ファイルは /etc/sssd/sssd.conf であることを前提としています。

別のファイルは、sssd コマンドで -c オプションを使用して SSSD に渡すことができます。

```
~]# sssd -c /etc/sssd/customfile.conf --daemon
```

13.2.3. SSSD の起動および停止

重要

SSSD を初めて起動する前に、少なくとも 1 つのドメインを設定します。「SSSD および ID プロバイダー（ドメイン）」を参照してください。

service コマンドまたは /etc/init.d/sssd スクリプトのいずれかが SSSD を起動できます。以下に例を示します。

```
~]# service sssd start
```

デフォルトでは、SSSD は自動的に起動するように設定されていません。この動作を変更するには、以下の 2 つの方法があります。

- authconfig コマンドで SSSD を有効にします。

```
~]# authconfig --enablesssd --enablesssdauth --update
```

- chkconfig コマンドを使用して、SSSD プロセスを開始リストに追加します。

```
~]# chkconfig sssd on
```

13.2.4. SSSD およびシステムサービス

SSSD およびその関連サービスは、sssd.conf ファイルで設定されます。[sssd] セクションは、sssd が services ディレクティブ内で開始する際にアクティブで、起動する必要がある サービスも一覧表示します。

SSSD は、複数のシステムサービスに認証情報キャッシュを提供できます。

- sssd_nss モジュールから name サービス要求に応答する Name Service Switch(NSS)プロバイダーサービス。これは、SSSD 設定の [nss] セクションで設定されます。

  この操作は、「サービスの設定: NSS」 で説明しています。

- sssd_pam モジュールを介して PAM 対話を管理する PAM プロバイダーサービスこの設定は、設定の [pam] セクションで設定されます。

  この操作は、「サービスの設定: PAM」 で説明しています。

- SSSD が known_hosts ファイルおよびその他のキー関連の設定を管理する方法を定義する SSH プロバイダーサービス。OpenSSH で SSSD を使用する方法は、「サービスの設定：OpenSSH およびキャッシュされたキー」 に記載されています。

- *LDAP サーバーに接続して設定されたマウント場所を取得する autofs プロバイダーサービス。これは、設定ファイルの [domain/NAME] セクションの LDAP アイデンティティープロバイダーの一部として設定されます。*

  この操作は、「サービスの設定： autofs」 で説明しています。

- *LDAP サーバーに接続して、設定された sudo ポリシーを取得する sudo プロバイダーサービスこれは、設定ファイルの [domain/NAME] セクションの LDAP アイデンティティープロバイダーの一部として設定されます。*

  この操作は、「サービスの設定： sudo」 で説明しています。

- *SSSD が Kerberos で機能して Active Directory ユーザーおよびグループを管理する方法を定義する PAC レスポンダーサービス。これは、特に 「ドメインの作成： Active Directory」 で説明されているように、ドメインを使用した Active Directory アイデンティティープロバイダーの管理に含まれます。*

13.2.5. サービスの設定: NSS

SSSD は NSS モジュール sssd_nss を提供します。これは、SSSD を使用してユーザー情報を取得するように指示します。NSS 設定には SSSD モジュールへの参照が含まれ、SSSD 設定は SSSD が NSS と対話する方法を設定します。

NSS サービスマップおよび SSSD

Name Service Switch(NSS)は、複数の設定および名前解決サービスを検索するための中央設定を提供します。NSS は、システム ID とサービスを設定ソースでマッピングする方法を 1 つ提供します。

SSSD は、NSS と NSS を複数のタイプの NSS マップのプロバイダーサービスとして機能します。

- *パスワード（パスワード）*
- *ユーザーグループ（シャドウ）*
- *グループ（グループ）*

- *Netgroups (netgroups)*

- *サービス（サービス）*

手順13.1 SSSD を使用するように NSS サービスを設定する

NSS は、任意のサービスマップと全サービスマップに複数の ID と設定プロバイダーを使用できます。デフォルトでは、サービスにシステムファイルを使用します。SSSD を含めるには、nss_sss モジュールを希望のサービスタイプに追加する必要があります。

1. *認証設定ツールを使用して SSSD を有効にします。これにより、nsswitch.conf ファイルが SSSD をプロバイダーとして使用するように自動的に設定されます。*

```
~]# authconfig --enablesssd --update
```

これにより、パスワード、シャドウ、グループ、および netgroups サービスが SSSD モジュールを使用するようにマップされます。

```
passwd:     files sss
shadow:     files sss
group:      files sss

netgroup:   files sss
```

2. *SSSD が authconfig で有効になっている場合、サービスマップはデフォルトでは有効になっていません。このマップを追加するには、nsswitch.conf ファイルを開き、sss モジュールを サービス マップに追加します。*

```
~]# vim /etc/nsswitch.conf

...
services: file sss
...
```

手順13.2 NSS と連携する SSSD の設定

SSSD が NSS 要求の処理に使用するオプションおよび設定は、[nss] services セクションで SSSD 設定ファイルで設定されます。

1. ***sssd.conf ファイルを開きます。***

```
~]# vim /etc/sssd/sssd.conf
```

2. ***NSS が SSSD と連携するサービスの 1 つとしてリストされていることを確認します。***

```
[sssd]
config_file_version = 2
reconnection_retries = 3
sbus_timeout = 30
services = nss, pam
```

3. ***[nss] セクションで、NSS パラメーターを変更します。これらについては、表 13.2「SSSD [nss] 設定パラメーター」 に一覧表示されます。***

```
[nss]
filter_groups = root
filter_users = root
reconnection_retries = 3
entry_cache_timeout = 300
entry_cache_nowait_percentage = 75
```

4. ***SSSD を再起動します。***

```
~]# service sssd restart
```

表13.2 SSSD [nss] 設定パラメーター

パラメーター	値の形式	説明
entry_cache_nowait_percentage	整数	キャッシュを更新する前に **sssd_nss** がキャッシュされたエントリーを返す期間を指定します。このパラメーターをゼロ(**0**)に設定すると、エントリーキャッシュの更新が無効になります。 これにより、次の更新の前にある期間が次の期間になる前にエントリーキャッシュが要求されると、バックグラウンドでエントリーを更新するよう自動的にエントリーキャッシュが設定されます。たとえば、間隔が 300 秒でキャッシュの割合が 75 の場合、要求が 225 秒 - 75% の間隔でエントリーキャッシュの更新が開始されます。 このオプションに許可される値は 0 から 99 です。これにより、**entry_cache_timeout** の値に基づいてパーセンテージが設定されます。デフォルト値は 50% です。

パラメーター	値の形式	説明
entry_negative_timeout	整数	**sssd_nss** が **負** のキャッシュヒットをキャッシュする期間（秒単位）を指定します。負のキャッシュヒットは、存在しないエントリーを含む、無効なデータベースエントリーのクエリーです。
filter_users、filter_groups	文字列	特定のユーザーが NSS データベースから取得されないように SSSD に指示します。これは、**root** などのシステムアカウントに特に便利です。
filter_users_in_groups	Boolean	グループ検索の実行時に **filter_users** リストに一覧表示されているユーザーがグループメンバーシップに表示されるかどうかを設定します。**FALSE** に設定すると、グループの検索はそのグループのメンバーであるすべてのユーザーを返します。指定されていない場合、デフォルト値は **true** で、グループメンバーの一覧をフィルターします。
debug_level	整数、0 - 9	デバッグロギングレベルを設定します。

NSS の互換性モード

NSS 互換性(compat)モード は、/etc/passwd ファイルの追加エントリーをサポートし、netgroup のユーザーまたはメンバーがシステムにアクセスできるようにします。

NSS の互換性モードが SSSD で機能できるようにするには、/etc/nsswitch.conf ファイルに以下のエントリーを追加します。

```
passwd: compat
passwd_compat: sss
```

NSS の互換性モードを有効にすると、以下の passwd エントリーがサポートされます。

- ***+user -user***

  ネットワーク情報システム(NIS)マップから指定した ユーザー を include(+)または exclude(-)します。

- ***+@netgroup -@netgroup***

NIS マップから指定した netgroup 内の全ユーザーを追加(+)または exclude(-)します。

- +

以前の NIS マップから除外したユーザー以外は、全ユーザーを除外します。

NSS の互換性モードの詳細は、nsswitch.conf(5) man ページを参照してください。

13.2.6. サービスの設定: PAM

警告

PAM 設定ファイルの間違いにより、ユーザーがシステムから完全にロックされる可能性があります。変更を実行する前に常に設定ファイルのバックアップを作成し、変更を元に戻すことができるようにセッションを開いた状態にします。

SSSD は、PAM モジュール sssd_pam を提供します。これにより、SSSD を使用してユーザー情報を取得するようシステムに指示します。PAM 設定には SSSD モジュールへの参照が含まれ、SSSD 設定は SSSD が PAM と対話する方法を設定します。

手順13.3 PAM の設定

1. authconfig を使用して、システム認証に SSSD を有効にします。

```
# authconfig --update --enablesssd --enablesssdauth
```

これにより、PAM 設定が自動的に更新され、すべての SSSD モジュールを参照できます。

```
#%PAM-1.0
# This file is auto-generated.
# User changes will be destroyed the next time authconfig is run.
auth        required      pam_env.so
auth        sufficient    pam_unix.so nullok try_first_pass
auth        requisite     pam_succeed_if.so uid >= 500 quiet
```

```
auth sufficient pam_sss.so use_first_pass
auth        required      pam_deny.so

account     required      pam_unix.so
account     sufficient    pam_localuser.so
account     sufficient    pam_succeed_if.so uid < 500 quiet
account [default=bad success=ok user_unknown=ignore] pam_sss.so
account     required      pam_permit.so

password    requisite     pam_cracklib.so try_first_pass retry=3
password    sufficient    pam_unix.so sha512 shadow nullok try_first_pass use_authtok
password sufficient pam_sss.so use_authtok
password    required      pam_deny.so

session     optional      pam_keyinit.so revoke
session     required      pam_limits.so
session     [success=1 default=ignore] pam_succeed_if.so service in crond quiet
use_uid
session sufficient pam_sss.so
session     required      pam_unix.so
```

これらのモジュールは、必要に応じてステートメントを 含める ように設定できます。

2. sssd.conf ファイルを開きます。

```
# vim /etc/sssd/sssd.conf
```

3. PAM が SSSD と連携するサービスの 1 つとしてリストされていることを確認します。

```
[sssd]
config_file_version = 2
reconnection_retries = 3
sbus_timeout = 30
services = nss, pam
```

4. [pam] セクションで、PAM パラメーターを変更します。これらについては、表 13.3「SSSD [pam] 設定パラメーター」 に一覧表示されます。

```
[pam]
reconnection_retries = 3
offline_credentials_expiration = 2
offline_failed_login_attempts = 3
offline_failed_login_delay = 5
```

5. ***SSSD を再起動します。***

```
~]# service sssd restart
```

表13.3 SSSD [pam] 設定パラメーター

パラメーター	値の形式	説明
offline_credentials_expiration	整数	認証プロバイダーがオフラインの場合にキャッシュされたログインを許可する期間を日数単位で設定します。この値は、最後に成功したオンラインログインから測定されます。指定されていない場合、デフォルトでゼロ(**0**)に設定されます。これは無制限になります。
offline_failed_login_attempts	整数	認証プロバイダーがオフラインの場合に失敗したログイン試行の数を設定します。指定されていない場合、デフォルトでゼロ(**0**)に設定されます。これは無制限になります。
offline_failed_login_delay	整数	ユーザーが失敗したログイン試行制限にヒットした場合にログイン試行を防ぐ期間を設定します。ゼロ(**0**)に設定すると、プロバイダーがオフラインの場合に、試行に失敗した制限に達すると、ユーザーは認証できません。オンライン認証に成功すると、オフライン認証を再度有効にできます。指定されていない場合、デフォルトは**5**(5)に設定されます。

13.2.7. サービスの設定： autofs

自動マウント、LDAP、および SSSD

自動マウントマップは通常、マップ、マウントディレクトリー、およびファイルサーバー間の関係を定義するフラットファイルです。（自動マウントは『ストレージ管理ガイド』 で説明しています。）

たとえば、ディレクトリー pub をホストする nfs.example.com という名前のファイルサーバーがあり、自動マウントが /shares/ ディレクトリーにディレクトリーをマウントするように設定されているとします。そのため、マウント場所は /shares/pub です。すべてのマウントは auto.master ファイルに一覧表示されます。このファイルは、さまざまなマウントディレクトリーとそれらのマウントディレクトリーを設定するファイルを識別します。次に、auto.shares ファイルは各ファイルサーバーと、/shares/ ディレクトリーにあるマウントディレクトリーを識別します。関係は、以下のように表示できます。

```
    auto.master
_____________/_____________
/                          /
/                          /
```

```
/shares/        auto.shares
                |
        |
        |
            nfs.example.com:pub
```

次に、すべてのマウントポイントは 2 つの異なるファイル（最低でも auto.master と auto）で定義されます。ファイルに関係なく、これらのファイルは各ローカル自動マウントプロセスで利用できる必要があります。

大規模な環境用に管理者が管理する方法の 1 つは、自動マウント設定を中央の LDAP ディレクトリーに保存し、各ローカルシステムがその LDAP ディレクトリーを指すように設定することです。つまり、更新が単一の場所でのみ行われ、新しいマップはローカルシステムによって自動的に認識されます。

automount-LDAP 設定の場合、自動マウントファイルは LDAP エントリーとして保存され、必要な自動マウントファイルに変換されます。各要素は LDAP 属性に変換されます。

LDAP エントリーは以下のようになります。

```
# container entry
dn: cn=automount,dc=example,dc=com
objectClass: nsContainer
objectClass: top
cn: automount

# master map entry
dn: automountMapName=auto.master,cn=automount,dc=example,dc=com
objectClass: automountMap
objectClass: top
automountMapName: auto.master

# shares map entry
dn: automountMapName=auto.shares,cn=automount,dc=example,dc=com
objectClass: automountMap
objectClass: top
automountMapName: auto.shares

# shares mount point
dn: automountKey=/shares,automountMapName=auto.master,cn=automount,dc=example,dc=com
objectClass: automount
objectClass: top
automountKey: /shares
automountInformation: auto.shares

# pub mount point
dn: automountKey=pub,automountMapName=auto.shares,cn=automount,dc=example,dc=com
objectClass: automount
```

```
objectClass: top
automountKey: pub
automountInformation: filer.example.com:/pub
description: pub
```

次に、スキーマ要素は、以下のような構造に一致します（RFC 2307 スキーマを使用）。

```
            auto.master
                objectclass: automountMap
                filename attribute: automountMapName
  ____________________________|_____________________________
  |                                       |
  |                                       |
/shares/                                 auto.shares
objectclass: automount                        objectclass: automountMap
mount point name attribute: automountKey      filename attribute: automountMapName
map name attribute: automountInformation          |
                                  |
                                          |
                                    nfs.example.com:pub
                                    objectclass: automount
                                    mount point name attribute: automountKey
                                    fileserver attribute: automountInformation
```

autofs は、これらのスキーマ要素を使用して自動マウント設定を取得します。/etc/sysconfig/autofs ファイルは、自動マウントエンティティーに使用される LDAP サーバー、ディレクトリーの場所、およびスキーマ要素を識別します。

```
LDAP_URI=ldap://ldap.example.com
SEARCH_BASE="cn=automount,dc=example,dc=com"
MAP_OBJECT_CLASS="automountMap"
ENTRY_OBJECT_CLASS="automount"
MAP_ATTRIBUTE="automountMapName"
ENTRY_ATTRIBUTE="automountKey"
VALUE_ATTRIBUTE="automountInformation"
```

自動マウント設定を LDAP ディレクトリーにポイントするのではなく、SSSD を参照するように設定できます。SSSD は、自動マウントが必要とするすべての情報を保存し、ユーザーがディレクトリーのマウントを試みると、その情報は SSSD にキャッシュされます。LDAP サーバーへの接続数を減らすことで、フェイルオーバー、サービス検出、タイムアウトなど、設定にはいくつかの利点があります。最も重要な点として、SSSD を使用すると、LDAP サーバーがオフラインであってもクライアントがディレクトリーを正常にマウントできるように、すべてのマウント情報を キャッシュすることができます。

手順13.4 SSSD での autofs サービスの設定

1. autofs パッケージおよび sssd-common パッケージがインストールされていることを確認します。

2. ***sssd.conf ファイルを開きます。***

```
~]# vim /etc/sssd/sssd.conf
```

3. ***autofs サービスを、SSSD が管理するサービスの一覧に追加します。***

```
[sssd]
services = nss,pam,autofs
....
```

4. ***[autofs] サービス設定を新しいセクションを作成します。このセクションは空白のままにすることができます。負のキャッシュヒットのタイムアウトについては、設定可能なオプションが 1 つのみあります。***

ただし、このセクションでは、SSSD が autofs サービスを認識し、デフォルト設定を提供する必要があります。

```
[autofs]
```

5. ***自動マウント情報は、SSSD 設定の設定済みの LDAP ドメインから読み取られるため、LDAP ドメインが利用可能でなければなりません。追加の設定がない場合、設定は自動マウント情報の RFC 2307 スキーマおよび LDAP 検索ベース(ldap_search_base)にデフォルト設定されます。これはカスタマイズが可能です。***

- ***ディレクトリータイプ autofs_provider。デフォルトは id_provider の値に設定されます。値が*** **none** ***の場合は、ドメインの autofs を明示的に無効にします。***
- ***検索ベース ldap_autofs_search_base。***
- ***マップエントリー ldap_autofs_map_object_classを認識するために使用するオブジェクトクラス***
- ***マップ名 ldap_autofs_map_nameを認識するために使用する属性です。***

- マウントポイントエントリー ldap_autofs_entry_object_classを認識するために使用するオブジェクトクラス

- マウントポイント名 ldap_autofs_entry_keyを認識するために使用する属性です。

- マウントポイントの追加設定情報に使用する属性である ldap_autofs_entry_value

以下に例を示します。

```
[domain/LDAP]
...
autofs_provider=ldap
ldap_autofs_search_base=cn=automount,dc=example,dc=com
ldap_autofs_map_object_class=automountMap
ldap_autofs_entry_object_class=automount
ldap_autofs_map_name=automountMapName
ldap_autofs_entry_key=automountKey
ldap_autofs_entry_value=automountInformation
```

6. sssd.conf ファイルを保存して閉じます。

7. nsswitch.conf ファイルを編集し、ldap から sss に変更して、SSSD で自動マウントマップ情報を検索するように autofs を設定します。

```
# vim /etc/nsswitch.conf

automount: files sss
```

8. SSSD を再起動します。

```
# service sssd restart
```

13.2.8. サービスの設定： sudo

sudo、LDAP、および SSSD について

sudo ルールは sudoers ファイルで定義されます。これは、一貫性を保つためにすべてのマシンに個別に分散する必要があります。

大規模な環境用に管理者が管理する方法の 1 つに、sudo 設定を中央の LDAP ディレクトリーに保存し、各ローカルシステムがその LDAP ディレクトリーを指すように設定することです。つまり、更新を 1 つの場所で行うだけで、新しいルールはローカルシステムによって自動的に認識されます。

sudo-LDAP 設定では、各 sudo ルールは LDAP エントリーとして保存され、sudo ルールのコンポーネントは LDAP 属性で定義されます。

sudoers ルールは以下のようになります。

```
Defaults    env_keep+=SSH_AUTH_SOCK
...
%wheel        ALL=(ALL)       ALL
```

LDAP エントリーは以下のようになります。

```
# sudo defaults
dn: cn=defaults,ou=SUDOers,dc=example,dc=com
objectClass: top
objectClass: sudoRole
cn: defaults
description: Default sudoOptions go here
sudoOption: env_keep+=SSH_AUTH_SOCK

# sudo rule
dn: cn=%wheel,ou=SUDOers,dc=example,dc=com
objectClass: top
objectClass: sudoRole
cn: %wheel
sudoUser: %wheel
sudoHost: ALL
sudoCommand: ALL
```

注記

SSSD は、sudoHost 属性の値に応じてローカルシステムに適用される sudo ルールのみをキャッシュします。つまり、sudoHost の値が ALL に設定されていると、ホスト名と一致する正規表現、システムの netgroup と一致する正規表現を使用するか、システムのホスト名、完全修飾ドメイン名、または IP アドレスと一致させます。

sudo サービスは、LDAP サーバーを参照し、その LDAP エントリーからルール設定を取得するように設定できます。sudo 設定を LDAP ディレクトリーにポイントするのではなく、SSSD を参照するように設定できます。SSSD は、sudo が必要とするすべての情報を保存し、ユーザーが sudo関連操作を試行するたびに、最新の sudo 設定を SSSD ディレクトリーからプルできます。ただし、SSSD は、

LDAP サーバーがオフラインであっても、その集中 LDAP 設定を使用してタスクを実行できるように、すべての sudo タイルもキャッシュし ます。

手順13.5 SSSD を使用した sudo の設定

SSSD sudo 設定オプションはすべて、man ページの sssd-ldap(5) に記載されています。

1. sssd-common パッケージがインストールされていることを確認します。

```
~]$ rpm -q sssd-common
```

2. sssd.conf ファイルを開きます。

```
~]# vim /etc/sssd/sssd.conf
```

3. SSSD が管理するサービス一覧に sudo サービスを追加します。

```
[sssd]
services = nss,pam,sudo
....
```

4. 新しい [sudo] サービス設定セクションを作成します。このセクションは空白のままにすることができます。1 つの設定可能なオプションは 1 つしかなく、sudo がピリオドの前または後に評価するための設定可能です。

ただし、このセクションでは、SSSD が sudo サービスを認識し、デフォルト設定を提供する必要があります。

```
[sudo]
```

5. sudo 情報は、SSSD 設定の設定済みの LDAP ドメインから読み取られるため、LDAP ドメインが利用可能でなければなりません。LDAP プロバイダーの場合、これらのパラメーターが必要です。

- ディレクトリータイプ sudo_provider。これは常に ldap です。

- 検索ベースである ldap_sudo_search_base。

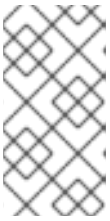
- LDAP サーバーの URI ldap_uri。

以下に例を示します。

```
[domain/LDAP]
id_provider = ldap

sudo_provider = ldap
ldap_uri = ldap://example.com
ldap_sudo_search_base = ou=sudoers,dc=example,dc=com
```

Identity Management（IdM または IPA）プロバイダーの場合、サーバーへの接続時に Kerberos 認証を実行するために必要な追加のパラメーターがあります。

```
[domain/IDM]
id_provider = ipa
ipa_domain = example.com
ipa_server = ipa.example.com
ldap_tls_cacert = /etc/ipa/ca.crt

sudo_provider = ldap
ldap_uri = ldap://ipa.example.com
ldap_sudo_search_base = ou=sudoers,dc=example,dc=com
ldap_sasl_mech = GSSAPI
ldap_sasl_authid = host/hostname.example.com
ldap_sasl_realm = EXAMPLE.COM
krb5_server = ipa.example.com
```



注記

Identity Management プロバイダーの sudo_provider タイプは ldap です。

6. sudo ルールキャッシュの更新に使用する間隔を設定します。

特定のシステムユーザーの キャッシュは常にチェックされ、そのユーザーがタスクを実行するたびに更新されます。ただし、SSSD は、ローカルシステムに関連するすべてのルールをキャッシュします。この完全なキャッシュは 2 つの方法で更新されます。

- 最後に完全更新からのルールのみの変更（ldap_sudo_smart_refresh_interval、秒単位の時間）、デフォルトは 15 分です。

- 完全にキャッシュ全体をダンプし、LDAP サーバーの現在のすべてのルール（ldap_sudo_full_refresh_interval、時間（秒単位））にプルします。デフォルトは 6 時間です。

これらの 2 つの更新間隔は別々に設定されます。以下に例を示します。

```
[domain/LDAP]
...
ldap_sudo_full_refresh_interval=86400
ldap_sudo_smart_refresh_interval=3600
```

注記

SSSD は、ローカルシステムに適用される sudo ルールのみをキャッシュします。つまり、sudoHost の値が ALL に設定されていると、ホスト名と一致する正規表現、システムの netgroup と一致する正規表現を使用するか、システムのホスト名、完全修飾ドメイン名、または IP アドレスと一致させます。

7. 必要に応じて、sudo ルールに使用されるスキーマを変更する値を設定します。

スキーマ要素は ldap_sudorule_* 属性で設定されます。デフォルトでは、スキーマ要素はすべて sudoers.ldap で定義されたスキーマを使用します。これらのデフォルトはほぼすべてのデプロイメントで使用されます。

8. sssd.conf ファイルを保存して閉じます。

9. nsswitch.conf ファイルを編集して sss の場所を追加して、SSSD でルール設定を検索するように sudo を設定します。

```
~]# vim /etc/nsswitch.conf

sudoers: files sss
```

10. SSSD を再起動します。

```
~]# service sssd restart
```

13.2.9. サービスの設定： OpenSSH およびキャッシュされたキー

OpenSSH は、2 つのシステム間で安全な、暗号化された接続を作成します。あるマシンが別のマシンに対して認証してアクセスを許可します。認証は、サーバー接続またはそのマシンのユーザー用のマシン自体になります。OpenSSH の詳細は、14章OpenSSH を参照してください。

この認証は、認証ユーザーまたはマシンを識別する 公開鍵と秘密鍵のペア を使用して実行されます。マシンにアクセスしようとしているリモートマシンまたはユーザーがキーペアを提示します。その後、ローカルマシンはそのリモートエンティティーを信頼するかどうかを選択します。信頼される場合、そのリモートマシンの公開鍵は known_hosts ファイルまたは authorized_keys のリモートユーザーに保存されます。リモートマシンまたはユーザーが再度認証を試みるたびに、ローカルシステムは最初に known_hosts ファイルまたは authorized_keys ファイルをチェックして、リモートエンティティーが認識され、信頼できるかどうかを確認します。存在する場合には、アクセスが許可されます。

最初の問題は、これらのアイデンティティーを確実に検証する際に生じます。

known_hosts ファイルは、マシン名、その IP アドレス、および公開鍵のトリプレットです。

```
server.example.com,255.255.255.255 ssh-rsa
AbcdEfg1234ZYX098776/AbcdEfg1234ZYX098776/AbcdEfg1234ZYX098776=
```

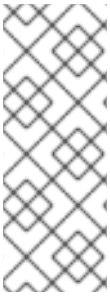
known_hosts ファイルは、さまざまな理由ですぐに古くなる可能性があります。たとえば、IP アドレスを介して DHCP サイクルを使用しているシステム、新しい鍵を定期的に再発行することも、仮想マシンまたはサービスをオンラインおよび削除することもできます。これにより、ホスト名、IP アドレス、およびキートリプレットが変更されます。

管理者は、セキュリティーを維持するために、現在の known_hosts ファイルを消去して維持する必要があります。（またはシステムユーザーは、提示されるマシンと鍵を受け入れることで、キーベースのセキュリティーのセキュリティー上のメリットを否定します。）

さらに、マシンとユーザーの両方について問題として、キーをスケーラブルな方法で配布しています。マシンは、暗号化されたセッションの確立の一環として鍵を送信できますが、ユーザーはキーを事前に指定する必要があります。鍵を継続的に伝播して更新するのは、管理作業が困難です。

最後に、SSH キーとマシン情報はローカルでのみ維持されます。known_hosts ファイルが均一に更新されていないため、一部のシステムで認識され、他のシステムで信頼されないマシンまたはユーザーが存在する可能性があります。

SSSD の目的は、認証情報をキャッシュとしてサーバーすることです。これには、マシンおよびユーザーの SSH 公開鍵の認証情報キャッシュとして機能することが含まれます。OpenSSH は、キャッシュされた鍵を確認するように SSSD を参照するように設定されています。SSSD は Red Hat Linux の Identity Management(IPA)ドメインを ID として使用し、Identity Management は実際に公開鍵とホスト情報を保存します。



注記

Identity Management ドメインに登録または結合した Linux マシンのみが、OpenSSH の鍵キャッシュとして SSSD を使用できます。その他の Unix マシンおよび Windows マシンは、known_hosts ファイルで通常の認証メカニズムを使用する必要があります。

ホストキーに SSSD を使用するように OpenSSH の設定

OpenSSH は、ユーザー固有の設定ファイル(~/.ssh/config)またはシステム全体の設定ファイル(/etc/ssh/ssh_config)で設定されます。ユーザーファイルはシステム設定よりも優先され、パラメーターで最初に取得した値が使用されます。このファイルのフォーマットおよび規則は、14章OpenSSH で説明されています。

ホストキーを管理するために、SSSD には sss_ssh_knownhostsproxy のツールがあり、2 つの操作を実行します。

1. *Identity Management サーバーから公開鍵キーを取得し、/var/lib/sss/pubconf/known_hosts ファイルに保存するように SSSD に要求します。*

2. *ソケット（デフォルト）またはプロキシーコマンドのいずれかを使用して、ホストマシンとの接続を確立します。*

このツールの形式は以下のとおりです。

```
sss_ssh_knownhostsproxy [-d sssd_domain] [-p ssh_port] HOST [PROXY_COMMAND]
```

表13.4 sss_ssh_knownhostsproxy オプション

短い引数	長い引数	説明
	HOSTNAME	チェックおよび接続するホストのホスト名を指定します。OpenSSH 設定ファイルでは、トークンである **%h** を指定できます。

短い引数	長い引数	説明
	PROXY_COMMAND	SSH クライアントへの接続に使用するプロキシーコマンドを渡します。これは、**ssh -o ProxyCommand=値** の実行に似ています。このオプションは、コマンドラインまたは別のスクリプトを使用して **sss_ssh_knownhostsproxy** を実行する場合に使用されますが、OpenSSH 設定ファイルには必要ありません。
-d **sssd_domain**	--domain **sssd_domain**	指定のドメインのエントリーで公開鍵のみを検索します。指定のない場合は、SSSD は設定済みのすべてのドメインで鍵を検索します。
-p **port**	--port **port**	このポートを使用して SSH クライアントに接続します。デフォルトでは、これはポート 22 です。

この SSSD ツールを使用するには、ssh_config または ~/.ssh/config ファイルに 2 つのパラメーターを追加または編集します。

- ***SSH クライアント(ProxyCommand)への接続に使用するコマンドを指定します。これは sss_ssh_knownhostsproxy で、必要な引数とホスト名を使用します。***
- ***SSSD ホストファイルの場所を指定します（グローバル既知のホストファイル）。***

たとえば、設定されたすべての SSSD ドメインで公開鍵を検索し、ポートとホストを指定して接続します。

```
ProxyCommand /usr/bin/sss_ssh_knownhostsproxy -p %p %h
GlobalKnownHostsFile /var/lib/sss/pubconf/known_hosts
```

ユーザーキーに SSSD を使用するように OpenSSH の設定

SSSD は、OpenSSH にユーザーの公開鍵を提供できます。鍵は、sss_ssh_authorizedkeys ツールの出力から直接 SSH デーモン sshd により読み取られ、ファイルに保存されません。

外部プログラムからユーザーの公開鍵を読み取るように sshd を設定するには（この場合は sss_ssh_authorizedkeys ツール）、/etc/ssh/sshd_config ファイルの AuthorizedKeysCommand ディレクティブを使用します。

sss_ssh_authorizedkeys ツールは、Identity Management(IPA)ドメインのユーザーエントリーから SSH 公開鍵を取得し、OpenSSH authorized_keys 形式で出力できます。コマンドの形式は以下の

とおりです。

```
sss_ssh_authorizedkeys [-d sssd_domain] USER
```

表13.5 sss_ssh_authorizedkeys オプション

短い引数	長い引数	説明
	USER	公開鍵を取得するユーザー名またはアカウント名。OpenSSH 設定ファイルでは、これはトークン **%u** で表示できます。
-d **sssd_domain**	--domain **sssd_domain**	指定のドメインのエントリーで公開鍵のみを検索します。指定のない場合は、SSSD は設定済みのすべてのドメインで鍵を検索します。

この機能は、以下のように /etc/ssh/sshd_config で設定されます。

```
AuthorizedKeysCommand /usr/bin/sss_ssh_authorizedkeys
AuthorizedKeysCommandRunAs nobody
```

これおよび他のオプションは、man ページの sshd_config(5) に記載されています。変更を反映するには、sshd サービスを再起動する必要があります。

13.2.10. SSSD および ID プロバイダー（ドメイン）

SSSD は ドメインを 認識します。ドメインは、さまざまな外部データソースに関連付けられた SSSD 設定ファイル内のエントリーです。ドメインは、アイデンティティープロバイダー（ユーザー情報用）の組み合わせで、オプションで認証（認証要求用）などの他のプロバイダーやパスワードの変更などの他の操作を組み合わせたものです。（すべての操作が単一のドメインまたはサーバー内で実行される場合、アイデンティティープロバイダーはすべての操作にも使用できます。）

SSSD は、異なる LDAP アイデンティティープロバイダー（OpenLDAP、Red Hat Directory Server、Microsoft Active Directory など）と連携し、ネイティブの LDAP 認証、Kerberos 認証、またはプロバイダー固有の認証プロトコル（Active Directory など）を使用できます。

ドメイン設定は、アイデンティティープロバイダー、認証プロバイダー、およびそれらのプロバイダー の情報にアクセスするための特定の設定を定義します。ID プロバイダーおよび認証プロバイダーにはいくつかのタイプがあります。

- 一般的な LDAP サーバーの場合

- *Active Directory（LDAP プロバイダータイプの拡張）*
- *Identity Management（LDAP プロバイダータイプの拡張）*
- *ローカル SSSD データベースのローカル（ローカル）*
- *Proxy*
- *Kerberos（認証プロバイダーのみ）*

ID プロバイダーおよび認証プロバイダーは、ドメインエントリーで異なる組み合わせで設定できます。可能な組み合わせは、表13.6「アイデンティティーストアおよび認証タイプの組み合わせ」 に記載されています。

表13.6 アイデンティティーストアおよび認証タイプの組み合わせ

識別プロバイダー	認証プロバイダー
Identity Management(LDAP)	Identity Management(LDAP)
Active Directory(LDAP)	Active Directory(LDAP)
Active Directory(LDAP)	Kerberos
LDAP	LDAP
LDAP	Kerberos
proxy	LDAP
proxy	Kerberos
proxy	proxy

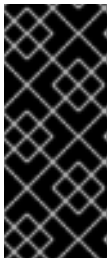
ドメインエントリー自体に加えて、SSSD がクエリーするドメイン一覧にドメイン名を追加する必要があります。以下に例を示します。

```
[sssd]
```

```
domains = LOCAL,Name
...

[domain/Name]
id_provider = type
auth_provider = type
provider_specific = value
global = value
```

グローバル 属性は、キャッシュやタイムアウト設定など、どのタイプのドメインでも利用できます。それぞれの ID および認証プロバイダーには、独自の必須およびオプションの設定パラメーターがあります。

表13.7 一般的な [domain] 設定パラメーター

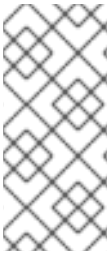
パラメーター	値の形式	説明
id_provider	文字列	このドメインに使用するデータバックエンドを指定します。サポート対象のアイデンティティーバックエンドは以下のとおりです。 ● ldap ● ipa（Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux の Identity Management） ● AD(Microsoft Active Directory) ● **nss_nis** などのレガシー NSS プロバイダーの場合はプロキシー。プロキシー ID プロバイダーを使用するには、**proxy_lib_name** オプションで設定されるレガシー NSS ライブラリーを正常に指定する必要があります。 ● ローカル（SSSD 内部ローカルプロバイダー）
auth_provider	文字列	ドメインに使用される認証プロバイダーを設定します。このオプションのデフォルト値は **id_provider** の値です。サポートされる認証プロバイダーは ldap、ipa、ad、krb5(Kerberos)、proxy、および none です。
min_id,max_id	整数	**オプション:**ドメインの UID および GID 範囲を指定します。ドメインにその範囲外のエントリーが含まれている場合は無視されます。**min_id** のデフォルト値は **1** です。**max_id** のデフォルト値は **0**（無制限）です。  **重要** デフォルトの **min_id** 値は、すべての種類のアイデンティティープロバイダーで同じです。LDAP ディレクトリーが UID 番号を使用していると、ローカルの **/etc/passwd** ファイルのユーザーとの競合が発生する可能性があります。このような競合を回避するには、可能な限り **min_id** を **1000** 以上に設定します。

パラメーター	値の形式	説明
cache_credentials	Boolean	**オプション:**ローカルの SSSD ドメインデータベースキャッシュにユーザーの認証情報を保存するかどうかを指定します。このパラメーターのデフォルト値は **false** です。LOCAL ドメイン以外のドメインではこの値を **true** に設定して、オフライン認証を有効にします。
entry_cache_timeout	整数	**オプション:**SSSD が **正** のキャッシュヒットをキャッシュする期間（秒単位）を指定します。正のキャッシュヒットはクエリーに成功します。
use_fully_qualified_names	Boolean	**オプション:**このドメインへの要求に完全修飾ドメイン名を必要とするかどうかを指定します。**true** に設定すると、このドメインへのすべての要求は完全修飾ドメイン名を使用する必要があります。また、要求からの出力に完全修飾名が表示されることも意味します。完全修飾ユーザー名への要求を制限すると、SSSD はユーザー名が競合しているユーザーを持つドメインを区別できます。 ***use_fully_qualified_names*** が **false** に設定されている場合、リクエストの完全修飾名を使用できますが、出力には簡素化されたバージョンのみが表示されます。 SSSD は、レルム名ではなく、ドメイン名に基づいた名前のみを解析できます。ただし、ドメインとレルムの両方に同じ名前を使用できます。

13.2.11. ドメインの作成： LDAP

LDAP ドメインは、SSSD が LDAP ディレクトリーをアイデンティティープロバイダーとして使用すること（およびオプションで認証プロバイダーとして使用する）ことを意味します。SSSD は、主要なディレクトリーサービスを複数サポートします。

- ***Red Hat Directory Server***
- ***OpenLDAP***
- ***Identity Management（IdM または IPA）***
- ***Microsoft Active Directory 2008 R2***



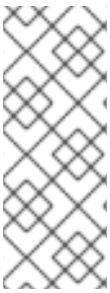
注記

一般的な LDAP アイデンティティープロバイダーで利用可能なパラメーターはすべて、LDAP プロバイダーのサブセットである Identity Management および Active Directory アイデンティティープロバイダーでも利用できます。

LDAP ドメインを設定するためのパラメーター

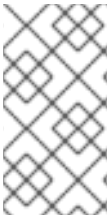
LDAP ディレクトリーは、アイデンティティープロバイダーと認証プロバイダーの両方として機能します。この設定には、LDAP サーバーのユーザーディレクトリーを特定して接続するための十分な情報が必要ですが、これらの接続パラメーターの定義方法は柔軟性があります。

LDAP サーバーへの接続に使用するユーザーアカウントを指定する場合や、パスワード操作に異なる LDAP サーバーを使用する場合など、より詳細な制御を可能にする他のオプションも使用できます。最も一般的なオプションは、表13.8「LDAP ドメイン設定パラメーター」 に記載されています。



注記

サーバー側のパスワードポリシーは常に、クライアント側から有効になっているポリシーよりも優先されます。たとえば、ldap_pwd_policy=shadow オプションを設定する場合、ユーザーの shadow LPAD 属性で定義したポリシーは、OpenLDAP サーバーでパスワードポリシーが有効であるかどうかには影響しません。



ヒント

その他の多くのオプションは、LDAP ドメイン設定 sssd-ldap(5) の man ページに記載されています。

表13.8 LDAP ドメイン設定パラメーター

パラメーター	説明
ldap_uri	SSSD が接続する LDAP サーバーの URI のコンマ区切りリストを指定します。一覧は優先順に指定されるため、一覧の最初のサーバーが最初に試行されます。追加のサーバーを一覧表示すると、フェイルオーバーの保護が提供されます。これは、DNS SRV レコードが指定されない場合は検出できます。

パラメーター	説明
ldap_search_base	LDAP ユーザー操作の実行に使用するベース DN を指定します。  重要 正しく使用しないと、ldap_search_base により SSSD ルックアップが失敗する可能性があります。 AD プロバイダーでは、ldap_search_base を設定する必要はありません。AD プロバイダーは必要な情報をすべて自動的に検出します。Red Hatnbsp;Hat は、この状況でパラメーターを設定せず、代わりに AD プロバイダーが検出するものに依存することを推奨します。

パラメーター	説明
ldap_tls_reqcert	TLS セッションで SSL サーバー証明書を確認する方法を指定します。4 つのオプションがあります。 • 証明書の要求を無効にし ない でください。 • 証明書を要求 できるよう にしますが、証明書が指定されていないか、不正な証明書が指定されていない場合でも、通常通りに続行されます。 • 証明書をリクエストして、証明書が指定されていない場合には通常続行されます。適切でない証明書を指定すると、セッションは終了します。 • デ マンと ハード のオプションは同じです。これには、有効な証明書が必要です。またはセッションを終了する必要があります。 デフォルトは hard です。
ldap_tls_cacert	SSSD が認識するすべての CA の CA 証明書が含まれるファイルに完全パスおよびファイル名を指定します。SSSD は、これらの CA が発行する証明書を受け入れます。 明示的に指定されていない場合は、OpenLDAP システムのデフォルトを使用します。

パラメーター	説明
ldap_referrals	SSSD が LDAP 参照を使用するかどうかを設定します。つまり、クエリーを LDAP データベースから別の LDAP データベースに転送します。SSSD は、データベースレベルおよびサブツリー参照をサポートします。同じ LDAP サーバー内の参照元では、SSSD はクエリーされるエントリーの DN を調整します。異なる LDAP サーバーを参照する参照の場合、SSSD は DN で完全に一致します。この値を true に設定すると参照元が有効になります。これがデフォルトです。 参照元の追跡試行に費やされた時間が原因で、全体のパフォーマンスに悪影響を及ぼす可能性があります。リファーラルチェックを無効にすると、パフォーマンスが大幅に向上します。

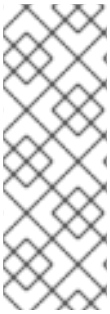
パラメーター	説明

ldap_schema	ユーザーエントリーの検索時に使用するスキーマのバージョンを設定します。これには、rfc2307、rfc2307bis、ad、または ipa を指定できます。デフォルトは rfc2307 です。
	
	RFC 2307 では、グループオブジェクトは多値属性 memberuid を使用します。これは、そのグループに属するユーザーの名前を一覧表示します。RFC 2307bis では、グループオブジェクトは、ユーザーまたはグループエントリーの完全な識別名(DN)を含む member 属性を使用します。RFC 2307bis は、member 属性を使用してネスト化されたグループを許可します。これらの異なるスキーマはグループメンバーシップに異なる定義を使用するため、SSSD で誤った LDAP スキーマを使用すると、適切なパーミッションが設定されていてもネットワークリソースの表示と管理に悪影響を及ぼす可能性があります。
	
	たとえば、RFC 2307bis の場合、ネスト化されたグループまたはプライマリー/セカンダリーグループを使用する場合にすべてのグループが返されます。
	
	`$ id`
	`uid=500(myserver) gid=500(myserver)`
	`groups=500(myserver),510(myothergroup)`
	
	SSSD が RFC 2307 スキーマを使用している場合は、プライマリーグループのみが返されます。
	
	この設定は、SSSD がグループメンバーを決定する方法にのみ影響します。実際のユーザーデータは変更されません。

パラメーター	説明
ldap_search_timeout	LDAP 検索がキャンセルされ、キャッシュされた結果が返されるまでの実行時間を秒単位で設定します。 LDAP 検索がタイムアウトすると、SSSD は自動的にオフラインモードに切り替わります。
ldap_network_timeout	接続試行に失敗した後に SSSD が LDAP サーバーのポーリングを試みる時間を秒単位で設定します。デフォルトは 6 秒です。
ldap_opt_timeout	サーバーから応答が受信されない場合に、同期 LDAP 操作を中止するまでの待機時間を秒単位で設定します。このオプションは、SASL バインドの場合の KDC との通信時にタイムアウトを制御します。デフォルトは 5 秒です。

LDAP ドメインの例

LDAP 設定は、特定の環境や SSSD の動作に応じて柔軟性が非常に高くなります。以下は LDAP ドメインの一般的な例ですが、SSSD 設定はこれらの例に限定されません。



注記

ドメインエントリーの作成に加えて、sssd.conf ファイルのクエリーを行う SSSD のドメイン一覧に新しいドメインを追加します。以下に例を示します。

```
domains = LOCAL,LDAP1,AD,PROXYNIS
```

例13.2 基本的な LDAP ドメイン設定

LDAP ドメインには 3 つの項目が必要です。

- LDAP サーバー
- 検索ベース
- セキュアな接続を確立する方法

最後の項目は LDAP 環境によって異なります。SSSD は機密情報を処理するため、安全な接続が必要です。この接続は専用の TLS/SSL 接続にすることも、Start TLS を使用できます。

専用の TLS/SSL 接続を使用すると LDAPS 接続を使用してサーバーに接続するため、ldap_uri オプションの一部として設定されます。

```
# An LDAP domain
[domain/LDAP]
cache_credentials = true

id_provider = ldap
auth_provider = ldap

ldap_uri = ldaps://ldap.example.com:636
ldap_search_base = dc=example,dc=com
```

TLS を使用するには、セキュアでないポートでセキュアな接続を確立するために証明書情報を入力する方法が必要です。これは、ldap_id_use_start_tls オプションを使用して Start TLS を使用し、ldap_tls_cacert オプションを使用して SSL サーバー証明書を発行した CA 証明書を特定します。

```
# An LDAP domain
[domain/LDAP]
cache_credentials = true

id_provider = ldap
auth_provider = ldap

ldap_uri = ldap://ldap.example.com
ldap_search_base = dc=example,dc=com
ldap_id_use_start_tls = true
ldap_tls_reqcert = demand
ldap_tls_cacert = /etc/pki/tls/certs/ca-bundle.crt
```

13.2.12. ドメインの作成： Identity Management(IdM)

Identity Management（IdM または IPA）アイデンティティープロバイダーは、汎用 LDAP プロバイダーの拡張です。LDAP プロバイダーのすべての設定オプションは、IdM プロバイダーと、SSSD が IdM ドメインのクライアントとして機能させ、IdM 機能を拡張することを可能にする追加のパラメーターで利用できます。

Identity Management は、ドメインの ID、認証、アクセス制御ルール、およびパスワードのプロバイダーとして機能します。*_providerまた、Identity Management には、SELinux ポリシー、自動マウント情報、ホストベースのアクセス制御を管理する独自のドメイン内に設定オプションがあります。

IdM ドメインの機能はすべて SSSD 設定に関連付けて、システムユーザーにこれらのセキュリティー関連のポリシーを適用し、キャッシュすることができます。

例13.3 基本的な IdM プロバイダー

LDAP プロバイダーなどの IdM プロバイダーは、ID、認証、アクセス制御などの複数のサービスを提供するように設定できます。

IdM サーバーでは、非常に便利な設定が 2 つあります（必須ではありません）。

- *デフォルトの RFC 2307 スキーマではなく、特定の IdM スキーマを使用します。*
- *クライアントが最初に IdM ドメインに接続するときに、Identity Management ドメインの DNS サーバーをこのクライアントの IP アドレスで更新するように SSSD を設定します。*

```
[sssd]
domains = local,example.com
...

[domain/example.com]
id_provider = ipa
ipa_server = ipaserver.example.com
ipa_hostname = ipa1.example.com
auth_provider = ipa
access_provider = ipa
chpass_provider = ipa

# set which schema to use
ldap_schema = ipa

# automatically update IdM DNS records
ipa_dyndns_update = true
```

Identity Management は、Linux ドメイン全体でユーザーのセキュリティーポリシーとアイデンティティーを定義して維持します。これには、アクセス制御ポリシー、SELinux ポリシーなどのルールが含まれます。IdM ドメインのこれらの要素の一部は、SSSD を IdM クライアントとして使用して SSSD と直接対話し、これらの機能は sssd.conf の IdM ドメインエントリーで管理できます。

ほとんどの設定パラメーターは、IdM が固定スキーマを使用するので、スキーマ要素の設定に関連している（ほとんどのデプロイメントでは関係ありません）、変更する必要はありません。実際、IdM

の機能にクライアント側の設定は必要ありません。しかし、動作を微調整すると便利な状況があります。

例13.4 SELinux を使用した IdM プロバイダー

IdM は、システムユーザーの SELinux ユーザーポリシーを定義して、SSSD の SELinux プロバイダーとして機能します。これは、selinux_provider パラメーターで設定されます。プロバイダーは id_provider の値にデフォルト設定されるため、SELinux ルールを サポート するように明示的に設定する必要はありません。ただし、SSSD で IdM プロバイダーの SELinux サポートを明示的に 無効 にするのに便利です。

```
selinux_provider = ipa
```

例13.5 ホストベースのアクセス制御のある IdM プロバイダー

IdM はホストベースのアクセス制御を定義し、ユーザーが接続または接続の試行に使用しているホストに基づいて、サービスまたはシステム全体へのアクセスを制限できます。このルールは、アクセスプロバイダーの動作の一部として SSSD によって評価および強制できます。

ホストベースのアクセス制御を有効にするには、Identity Management サーバーが少なくともアクセスプロバイダーである必要があります。

SSSD がホストベースのアクセス制御ルールを評価する方法は 2 つあります。

- SSSD は、ユーザーが IdM リソースに接続するために使用するマシン（ソースホスト）を評価できます。これはデフォルトでは無効になっているため、ルールのターゲットホスト部分のみが評価されます。
- SSSD は、指定した間隔で、キャッシュのホストベースのアクセス制御ルールを更新できます。

以下に例を示します。

```
access_provider = ipa
ipa_hbac_refresh = 120

# check for source machine rules; disabled by default
ipa_hbac_support_srchost = true
```

例13.6 レルム間の Kerberos 信頼を使用したアイデンティティー管理

Identity Management（IdM または IPA）は、Active Directory DNS ドメインと Kerberos レルムとの間で信頼できる関係で設定できます。これにより、Active Directory ユーザーは Linux システムのサービスおよびホストにアクセスできます。

SSSD には、レルム間の信頼で使用される 2 つの設定があります。

- *Kerberos チケットに必要なデータを追加するサービス*
- *サブドメインをサポートする設定*

Kerberos チケットデータ

Microsoft は、特権アクセス証明書 または MS-PAC と呼ばれる特別な承認構造を使用します。PAC は、Windows ドメインの他の Windows クライアントおよびサーバーへのエンティティーを識別する方法として Kerberos チケットに組み込まれます。

SSSD には、Kerberos チケットの追加データを生成する特別な PAC サービスがあります。Active Directory ドメインを使用する場合は、Windows ユーザーの PAC データを追加する必要がある場合があります。その場合は、SSSD で pac サービスを有効にします。

```
[sssd]
services = nss, pam, pac
...
```

Windows サブドメイン

通常、SSSD のドメインエントリーは単一のアイデンティティープロバイダーに直接対応します。ただし、IdM のレルム間の信頼を使用すると、IdM ドメインは別のドメインを信頼するため、ドメインは相互に透過的になります。SSSD はこの信頼関係に従うことができるので、IdM ドメインが設定されている場合は、SSSD のドメインセクションで設定せずに、SSSD により Windows ドメインが自動的に検索およびサポートされます。

これは、subdomains_provider パラメーターを IdM ドメインセクションに追加することで設定します。これは実際には任意のパラメーターです。サブドメインが検出されると、SSSD は ipa

プロバイダータイプを使用するようにデフォルト設定されます。ただし、このパラメーターを使用して、none の値を設定してサブドメインのフェッチを無効にすることもできます。

```
[domain/IDM]
...
subdomains_provider = ipa
get_domains_timeout = 300
```

13.2.13. ドメインの作成： Active Directory

Active Directory アイデンティティープロバイダーは、汎用 LDAP プロバイダーの拡張です。LDAP プロバイダーのすべての設定オプションは、Active Directory プロバイダーと、Active Directory とシステムユーザー間のユーザーアカウントおよびアイデンティティーマッピングに関連するいくつかの追加パラメーターで利用できます。

標準の LDAP サーバーと Active Directory サーバーには、いくつかの基本的な違いがあります。Active Directory プロバイダーを設定する場合は、特定の設定が必要な設定エリアがあります。

- Windows セキュリティー ID を使用する ID は、対応する Linux システムユーザー ID にマップされる必要があります。
- 検索では、範囲の取得拡張機能を考慮する必要があります。
- LDAP 参照にはパフォーマンスの問題が発生する可能性があります。

Active Directory Securiy ID および Linux ユーザー ID のマッピング

Windows と Linux がシステムユーザーを処理する方法と、Active Directory と標準の LDAPv3 ディレクトリーサービスで使用されるユーザースキーマでは、固有の構造的な違いがあります。SSSD で Active Directory アイデンティティープロバイダーを使用してシステムユーザーを管理する場合は、Active Directory 形式のユーザーを新しい SSSD ユーザーを調整する必要があります。これには 2 つの方法があります。

- Services for Unix を使用した Windows ユーザーおよびグループエントリーに POSIX 属性を挿入し、それらの属性を PAM/NSS にプルする
- SSSD で ID マッピングを使用した Active Directory セキュリティー ID(SID)と Linux で生成された UID 間のマップの作成

ID マッピングは、Active Directory で追加のパッケージや設定を必要としないため、ほとんどの環境の最も簡単なオプションです。

ID マッピングのメカニズム

Linux/Unix システムは、ローカルユーザー ID 番号およびグループ ID 番号を使用して、システム上のユーザーを特定します。これらの UID:GID 番号は 501:501 などの単純な整数です。この数は、大規模な Linux/Unix ドメインの一部であるシステムであっても、常にローカルで作成および管理されるため、簡単な数値です。

Microsoft Windows および Active Directory は、異なるユーザー ID 構造を使用して、ユーザー、グループ、およびマシンを特定します。各 ID は、セキュリティーバージョン、発行認証局タイプ、マシン、およびアイデンティティー自体を識別するさまざまなセグメントで構成されます。以下に例を示します。

```
S-1-5-21-3623811015-3361044348-30300820-1013
```

3 番目のブロックから 6 番目のブロックはマシン識別子です。

```
S-1-5-21-3623811015-3361044348-30300820-1013
```

最後のブロックは、特定のエンティティーを識別する RID（相対識別子 ）です。

```
S-1-5-21-3623811015-3361044348-30300820-1013
```

使用可能な ID 番号の範囲は常に SSSD に割り当てられます。（これはローカル範囲であるため、すべてのマシンで同じです。）

```
|_____________________________________|
|                     |
minimum ID                max ID
```

この範囲は 10,000 セクション（デフォルトでは）に分けられ、各セクションには 200,000 ID が割り当てられます。

```
| slice 1 | slice 2 |   ...   |
|_________|_________|_________|
|       |       |       |
minimum ID                max ID
```

新しい Active Directory ドメインが検出されると、SID がハッシュ化されます。次に、SSSD はハッシュと利用可能なセクションの数を使用して、Active Directory ドメインに割り当てる ID セクションを決定します。これは ID セクションの割り当てを確実に行うため、多くのクライアントマシンの同じ Active Directory ドメインに同じ ID 範囲が割り当てられます。

```
| Active  | Active  |         |
|Directory|Directory|         |
|domain 1 |domain 2 |   ...   |
|         |         |         |
| slice 1 | slice 2 |   ...   |
|_________|_________|_________|
|         |         |         |
minimum ID                  max ID
```

注記

ID セクションの割り当て方法は一貫していますが、ID マッピングは、クライアントマシンで Active Directory ドメインが発生している順序を基にしているため、すべての Linux クライアントマシン で ID 範囲の割り当ての一貫性がなくなることがあります。一貫性が必要な場合は、ID マッピングを無効にし、明示的な POSIX 属性の使用を検討してください。

ID マッピングパラメーター

ID マッピングは 2 つのパラメーターで有効にされており、もう 1 つは適切な Active Directory ユーザースキーマをロードできるようにします。

```
ldap_id_mapping = True
ldap_schema = ad
```

注記

ID マッピングを有効にすると、uidNumber 属性および gidNumber 属性は無視されます。これにより、手動で割り当てられた値を防ぐことができます。いずれか の値を手動で割り当てる必要がある場合は、すべて の値を手動で割り当て、ID マッピングを無効にする必要があります。

ユーザーのマッピング

Active Directory ユーザーがローカルシステムサービスに初めてログインしようとすると、そのユーザーのエントリーが SSSD キャッシュに作成されます。リモートユーザーは、システムユーザーと同じように設定されます。

- ユーザーのシステム UID が、そのドメインの SID と ID 範囲に基づいて作成されます。

- GID は、UID と同じユーザー用に作成されます。
- ユーザー用にプライベートグループが作成されます。
- sssd.conf ファイルのホームディレクトリーの形式に基づいて、ホームディレクトリーが作成されます。
- シェルは、システムのデフォルト、または sssd.conf ファイルの設定に従って作成されます。
- ユーザーが Active Directory ドメイン内のグループに所属する場合は、SID を使用して、SSSD により、Linux システムのこれらのグループにユーザーが追加されます。

Active Directory ユーザーと範囲の取得検索

Microsoft Active Directory には属性 MaxValRange があります。この属性は、多値属性の値の数に制限を設定します。これは、検索拡張機能を 取得する範囲 です。基本的に、これは複数検索を実行し、すべての一致が返されるまで、指定した範囲内で結果のサブセットを返します。

たとえば、member 属性の検索を行う場合、各エントリーには複数の値があり、その属性を持つエントリーが複数ある可能性があります。2000 に一致する結果（またはそれ以上）がある場合、MaxValRange は一度に表示される値を制限します。これは、値の範囲です。指定される属性には追加のフラグセットがあり、結果が置かれているセットの範囲を表示します。

```
attribute:range=low-high:value
```

たとえば、検索では 100 から 500 になります。

```
member;range=99-499: cn=John Smith...
```

詳細は、Microsoft のドキュメント を参照して http://msdn.microsoft.com/en-us/library/cc223242.aspx ください。

SSSD は、追加の設定なしに、Active Directory プロバイダーを使用した範囲の取得をサポートします。

ただし、ldap_user_search_base などの検索を設定するのに使用できる LDAP プロバイダー属性の一部は、範囲の取得では実行されません。Active Directory プロバイダードメインで検索ベースを設定し、どの検索で範囲の取得をトリガーするかを考慮してください。

パフォーマンスおよび LDAP 参照

参照元の追跡試行に費やされた時間が原因で、全体のパフォーマンスに悪影響を及ぼす可能性があります。Active Directory アイデンティティープロバイダーで参照のchasing を使用すると、特にパフォーマンス低下が低くなります。リファーラルチェックを無効にすると、パフォーマンスが大幅に向上します。

LDAP 参照はデフォルトで有効になっているため、LDAP ドメイン設定で明示的に無効にする必要があります。以下に例を示します。

```
ldap_referrals = false
```

他のプロバイダータイプとしての Active Directory

Active Directory は ID プロバイダーとして使用し、アクセス、パスワード、および認証プロバイダーとして使用できます。

汎用 LDAP プロバイダー設定には、Active Directory プロバイダーの設定に使用できるオプションは複数あります。ad 値の使用は、パラメーターと値を自動的にプルし、Active Directory の特定のプロバイダーを設定する短い形です。たとえば、access_provider = ad を使用して、明示的な LDAP プロバイダーパラメーターを使用して、Active Directory アクセスプロバイダーをこの設定に拡張します。

```
access_provider = ldap
ldap_access_order = expire
ldap_account_expire_policy = ad
```

手順13.6 Active Directory アイデンティティープロバイダーの設定

Active Directory は、ドメインの ID、認証、アクセス制御ルール、およびパスワードのプロバイダーとして機能します。*_providerさらに、デフォルトの RFC 2307 を使用するのではなく、ユーザーおよびグループのエントリー用にネイティブの Active Directory スキーマを読み込むこともできます。

1. Active Directory システムおよび Linux システムの両方に適切に構成されている環境があることを確認します。

   - 特に SSSD でサービス検出を使用する場合は、名前解決を適切に設定する必要があります。

- *Kerberos が正常に機能するには、両方のシステムのクロックが同期されている必要があります。*

2. *Linux システムを Active Directory クライアントとして設定し、Active Directory ドメイン内に登録します。これは、Linux システムで Kerberos サービスおよび Samba サービスを設定して行います。*

   a. *Active Directory Kerberos レルムを使用するように Kerberos を設定します。*

      i. *Kerberos クライアント設定ファイルを開きます。*

```
~]# vim /etc/krb5.conf
```

      ii. *[logging] セクションおよび [libdefaults] セクションを設定し、Active Directory レルムに接続するようにします。*

```
[logging]
 default = FILE:/var/log/krb5libs.log

[libdefaults]
 default_realm = EXAMPLE.COM
 dns_lookup_realm = true
 dns_lookup_kdc = true
 ticket_lifetime = 24h
 renew_lifetime = 7d
 rdns = false
 forwardable = false
```

      SSSD で自動検出を使用しない場合は、[realms] セクションおよび [domain_realm] セクションも設定し、Active Directory サーバーを明示的に定義します。

   b. *Active ディレクトリーサーバーに接続するように Samba サーバーを設定します。*

      i. *Samba 設定ファイルを開きます。*

```
~]# vim /etc/samba/smb.conf
```

ii.

[global] セクションに Active Directory ドメイン情報を設定します。

```
[global]
   workgroup = EXAMPLE
   client signing = yes
   client use spnego = yes
   kerberos method = secrets and keytab
   log file = /var/log/samba/%m.log
   password server = AD.EXAMPLE.COM
   realm = EXAMPLE.COM
   security = ads
```

c.

Linux マシンを Active Directory ドメインに追加します。

i.

Windows 管理ユーザーの Kerberos 認証情報を取得します。

```
~]# kinit Administrator
```

ii.

net コマンドを使用してマシンをドメインに追加します。

```
~]# net ads join -k
Joined 'server' to dns domain 'example.com'
```

これにより、新しい keytab ファイル /etc/krb5.keytab が作成されます。

システムのキーを一覧表示し、ホストプリンシパルが存在することを確認します。

```
~]# klist -k
```

3.

authconfig を使用して、システム認証に SSSD を有効にします。

```
# authconfig --update --enablesssd --enablesssdauth
```

4.

例13.7「Active Directory 2008 R2 ドメイン」 および 例13.8「ID マッピングのある Active Directory 2008 R2 ドメイン」 に示されるように、Active Directory ドメインを SSSD

設定の ID プロバイダーとして設定します。

5. SSH サービスを再起動して、新しい PAM 設定を読み込みます。

```
~]# service sshd restart
```

6. 設定ファイルの変更後に SSSD を再起動します。

```
~]# service sssd restart
```

例13.7 Active Directory 2008 R2 ドメイン

```
~]# vim /etc/sssd/sssd.conf

[sssd]
config_file_version = 2
domains = ad.example.com
services = nss, pam

...

[domain/ad.example.com]
id_provider = ad
ad_server = ad.example.com
ad_hostname = ad.example.com
auth_provider = ad
chpass_provider = ad
access_provider = ad

# defines user/group schema type
ldap_schema = ad

# using explicit POSIX attributes in the Windows entries
ldap_id_mapping = False

# caching credentials
cache_credentials = true

# access controls
ldap_access_order = expire
ldap_account_expire_policy = ad
ldap_force_upper_case_realm = true

# performance
ldap_referrals = false
```

ID マッピングに重要なパラメーターは、Active Directory スキーマ(ldap_schema)と ID マッピン

グを明示的に有効にする必要があります(ldap_id_mapping)。

例13.8 ID マッピングのある Active Directory 2008 R2 ドメイン

```
~]# vim /etc/sssd/sssd.conf

[sssd]
config_file_version = 2
domains = ad.example.com
services = nss, pam

...

[domain/ad.example.com]
id_provider = ad
ad_server = ad.example.com
ad_hostname = ad.example.com
auth_provider = ad
chpass_provider = ad
access_provider = ad

# defines user/group schema type
ldap_schema = ad

# for SID-UID mapping
ldap_id_mapping = True

# caching credentials
cache_credentials = true

# access controls
ldap_access_order = expire
ldap_account_expire_policy = ad
ldap_force_upper_case_realm = true

# performance
ldap_referrals = false
```

Active Directory ドメインの潜在的な設定属性は、sssd-ldap(5)および sssd-ad(5) の man ページに記載されています。

13.2.14. ドメインの設定： LDAP プロバイダーとしての Active Directory(Alternative)

Active Directory はタイプ固有のアイデンティティープロバイダーとして設定できますが、Kerberos 認証プロバイダーを使用して純粋な LDAP プロバイダーとして設定することもできます。

手順13.7 LDAP プロバイダーとしての Active Directory の設定

1.

SSSD は SASL を使用して Active Directory サーバーに接続することが推奨されます。つまり、ローカルホストには Linux ホストの Windows ドメイン のサービスキータブが必要です。

このキータブは Samba を使用して作成できます。

a.

Active Directory レルムを使用するように /etc/krb5.conf ファイルを設定します。

```
[logging]
 default = FILE:/var/log/krb5libs.log

[libdefaults]
 default_realm = AD.EXAMPLE.COM
 dns_lookup_realm = true
 dns_lookup_kdc = true
 ticket_lifetime = 24h
 renew_lifetime = 7d
 rdns = false
 forwardable = false

[realms]
# Define only if DNS lookups are not working
# AD.EXAMPLE.COM = {
#  kdc = server.ad.example.com
#  admin_server = server.ad.example.com
#  master_kdc = server.ad.example.com
# }

[domain_realm]
# Define only if DNS lookups are not working
# .ad.example.com = AD.EXAMPLE.COM
# ad.example.com = AD.EXAMPLE.COM
```

b.

Samba 設定ファイル /etc/samba/smb.conf を設定して、Windows Kerberos レルムを参照します。

```
[global]
   workgroup = EXAMPLE
   client signing = yes
   client use spnego = yes
   kerberos method = secrets and keytab
   log file = /var/log/samba/%m.log
   password server = AD.EXAMPLE.COM
   realm = EXAMPLE.COM
   security = ads
```

c.
Kerberos を初期化するには、root で以下のコマンドを入力します。

```
~]# kinit Administrator@EXAMPLE.COM
```

d.
次に net ads コマンドを実行して管理者プリンシパルとしてログインします。この管理者アカウントにはマシンを Windows ドメインに追加するのに十分な権限が必要ですが、ドメイン管理者権限は必要ありません。

```
~]# net ads join -U Administrator
```

e.
net ads を再度実行して、ホストマシンをドメインに追加します。これは、ホストプリンシパル(host/FQDN)で行うか、必要に応じて NFS サービス(nfs/FQDN)を使用して実行できます。

```
~]# net ads join createupn="host/rhel-server.example.com@AD.EXAMPLE.COM" -
U Administrator
```

2.
Unix パッケージのサービスが Windows サーバーにインストールされていることを確認します。

3.
SSSD で使用する Windows ドメインを設定します。

a.
Windows マシンで Server Manager を開きます。

b.
Active Directory ドメインサービス ロールを作成します。

c.
ad.example.com などの新しいドメインを作成します。

d.
Active Directory Domain Services ロールに UNIX サービス用の Identity Management を追加します。Unix NIS ドメインを、設定のドメイン名として使用します。

4.
Active Directory サーバーで、Linux ユーザーのグループを作成します。

a.
Administrative Tools を開き、Active Directory Users および Computers を選択し

ます。

b.
Active Directory ドメイン ad.example.com を選択します。

c.
Users タブで右クリックし、Create a New Group を選択します。

d.
新しいグループ unixusers という名前を付け、保存します。

e.
unixusers グループエントリーをダブルクリックして、Users タブを開きます。

f.
Unix Attributes タブを開きます。

g.
NIS ドメインを、ad.example.com に設定し、必要に応じてグループ ID(GID)番号を設定する NIS ドメインに設定します。

5.
Unix グループに含まれるようにユーザーを設定します。

a.
Administrative Tools を開き、Active Directory Users および Computers を選択します。

b.
Active Directory ドメイン ad.example.com を選択します。

c.
Users タブで右クリックし、Create a New User を選択します。

d.
新しいユーザー aduser という名前を付け、ユーザーが 次回のログオンと Lock アカウント の チェックボックスが選択されていない ことを確認します。

次に、ユーザーを保存します。

e.
aduser ユーザーエントリーをダブルクリックして、Unix Attributes タブを開きます。Unix 設定が Active Directory ドメインと unixgroup グループと一致することを確認

します。

- Active Directory ドメイン用に作成された NIS ドメイン
- UID
- ログインシェルの /bin/bash
- ホームディレクトリーの /home/aduserへのホームディレクトリー
- プライマリーグループ名（ unixusersへ）

ヒント

大規模なディレクトリーのパスワード検索には、要求ごとに数秒かかる場合があります。最初のユーザールックアップは、LDAP サーバーへの呼び出しです。インデックスのない検索はリソース集約型が多いため、サーバーはディレクトリー内のすべてのエントリーが一致するかどうかをチェックするため、インデックス化された検索よりも時間がかかります。ユーザー検索を迅速化するには、SSSD が検索する属性をインデックス化します。

- uid
- uidNumber
- gidNumber
- gecos

6. Linux システムで、SSSD ドメインを設定します。

```
~]# vim /etc/sssd/sssd.conf
```

LDAP プロバイダーパラメーターの完全なリストは、sssd-ldap(5) man ページを参照してください。

例13.9 Unix のサービスがある Active Directory 2008 R2 ドメイン

```
[sssd]
config_file_version = 2
domains = ad.example.com
services = nss, pam

...

[domain/ad.example.com]
cache_credentials = true

# for performance
ldap_referrals = false

id_provider = ldap
auth_provider = krb5
chpass_provider = krb5
access_provider = ldap

ldap_schema = rfc2307bis

ldap_sasl_mech = GSSAPI
ldap_sasl_authid = host/rhel-server.example.com@AD.EXAMPLE.COM

#provide the schema for services for unix
ldap_schema = rfc2307bis

ldap_user_search_base = ou=user accounts,dc=ad,dc=example,dc=com
ldap_user_object_class = user
ldap_user_home_directory = unixHomeDirectory
ldap_user_principal = userPrincipalName

# optional - set schema mapping
# parameters are listed in sssd-ldap
ldap_user_object_class = user
ldap_user_name = sAMAccountName

ldap_group_search_base = ou=groups,dc=ad,dc=example,dc=com
ldap_group_object_class = group

ldap_access_order = expire
ldap_account_expire_policy = ad
ldap_force_upper_case_realm = true
ldap_referrals = false

krb5_realm = AD-REALM.EXAMPLE.COM
# required
krb5_canonicalize = false
```

7.
SSSD を再起動します。

```
~]# service sssd restart
```

13.2.15. ドメインオプション： ユーザー名の形式の設定

SSSD が実行する主なアクションの 1 つは、ローカルシステムユーザーをリモート ID プロバイダーのアイデンティティーにマッピングすることです。SSSD は、ユーザー名とドメインバックエンド名の組み合わせを使用してログイン ID を作成します。

SSSD は、異なるドメインに属する限り、同じユーザー名を持つ異なるユーザーを認識できます。たとえば、SSSD は ldap.example.com ドメインで jsmith と jsmith の両方を正常に認証 でき ます。

完全なユーザー名の作成に使用される名前の形式は、設定の [sssd] セクションでユニバーサルで定義され、各ドメインセクションで個別に定義することができます（任意）。

さまざまなサービスのユーザー名（LDAP、Samba、Active Directory、Identity Management、ローカルシステムなど）にはそれぞれ異なる形式があります。SSSD がユーザー名/ドメイン名セットを識別するために使用する式は、異なる形式で名前を解釈できる必要があります。この式は re_expression パラメーターで設定されます。

グローバルデフォルトでは、このフィルターは name@domain 形式の名前を作成します。

```
(?P<name>[^@]+)@?(?P<domain>[^@]*$)
```

注記

正規表現の形式は Python 構文です。

ドメイン部分は、アイデンティティープロバイダーのドメイン名に基づいて自動的に指定できます。そのため、ユーザーは jsmith としてログインでき、ユーザーが LOCAL ドメインに属する場合（例）、ユーザー名は SSSD によって jsmith@LOCAL として解釈されます。

ただし、他のアイデンティティープロバイダーには他の形式が含まれる場合があります。たとえば、Samba には非常に厳密な形式があるため、ユーザー名は DOMAIN\username の形式と一致する必要があります。Samba の場合は、正規表現は以下のようになります。

```
(?P<domain>[^\\]*?)\\?(?P<name>[^\\]+$)
```

Active Directory などの一部のプロバイダーは、複数の異なる名前形式をサポートします。たとえば、Active Directory および Identity Management は、デフォルトで 3 つの異なる形式をサポートします。

- username
- username@domain.name
- DOMAIN\username

Active Directory プロバイダーおよび Identity Management プロバイダーのデフォルト値は、3 つの名前形式すべてを可能にするより複雑なフィルターです。

```
(((?P<domain>[^\\]+)\\(?P<name>.+$))|((?P<name>[^@]+)@(?P<domain>.+$))|(^(?P<name>[^@\\]+)$))
```

注記

jsmith@ldap.example.com などの完全修飾名で情報を要求すると、適切なユーザーアカウントが常に返されます。異なるドメインに同じユーザー名を持つユーザーが複数ある場合は、ユーザー名のみを指定すると、ルックアップ順にドメインが最初に表示されるユーザーが返されます。

re_expression はユーザー名の形式を設定するのに最も重要な方法ですが、他のアプリケーションで便利なオプションが 2 つあります。

デフォルトのドメイン名値

1 つ目は、すべてのユーザーで使用されるデフォルトのドメイン名(default_domain_suffix)を設定します。（これは、[sssd] セクションでのみ利用できるグローバル設定です。） 複数のドメインが設定され、1 つのドメインのみがユーザーデータを格納する場合があり、もう 1 つがホストまたはサービス ID に使用される場合があります。デフォルトのドメイン名を設定すると、ユーザーはドメイン名のみでログインでき、ドメイン名（プライマリードメイン外のユーザーに必要）を指定できません。

```
[sssd]
...
default_domain_suffix = USERS.EXAMPLE.COM
```

- 出力のフルネーム形式

他のパラメーターは、ユーザー名の 解釈 を定義するのではなく、識別された名前を 出力 する方法を定義するのではなく、re_expression に関連します。full_name_format パラメーターは、ユーザー名とドメイン名（決定後に）を表示する方法を設定します。

```
full_name_format = %1$s@%2$s
```

13.2.16. ドメインオプション： オフライン認証の有効化

ユーザー ID は常にキャッシュされ、ドメインサービスに関する情報も常にキャッシュされます。ただし、ユーザーの 認証情報 はデフォルトでキャッシュされません。つまり、SSSD は認証要求に対してバックエンドアイデンティティープロバイダーを常にチェックします。アイデンティティープロバイダーがオフラインまたは利用できない場合は、これらの認証要求を処理する方法がないため、ユーザー認証が失敗する可能性があります。

オフラインの認証情報キャッシュ を有効にすると、（ログインに成功すると）SSSD キャッシュにユーザーアカウントの一部として認証情報を保存します。そのため、アイデンティティープロバイダーが利用できない場合でも、ユーザーは保存された認証情報を使用して引き続き認証できます。オフライン認証情報のキャッシングは主に個別のドメインエントリーで設定されますが、認証情報のキャッシングはローカルの PAM サービスとリモートドメインと対話するため、PAM サービスセクションで設定できる任意の設定があります。

```
[domain/EXAMPLE]
cache_credentials = true
```

オプションのパラメーターは、これらの認証情報の有効期限が切れる際に設定されます。有効期限は、アカウントや認証情報が指定されたユーザーがローカルサービスに無期限にアクセスできなくなる可能性があるためです。

認証情報の有効期限は、システムの認証要求を処理する PAM サービスで設定されます。

```
[sssd]
services = nss,pam
...

[pam]
offline_credentials_expiration = 3
...

[domain/EXAMPLE]
cache_credentials = true
...
```

offline_credentials_expiration は、ユーザーの単一の認証情報エントリーがキャッシュに保持されるというログインに成功した後の日数を設定します。これをゼロ(0)に設定すると、エントリーが永久に保持されます。

認証情報キャッシュとは関係ありませんが、各ドメインには、個々のユーザーおよびサービスが期限切れになる際の設定オプションがあります。

- *account_cache_expiration は、ログインに成功すると、ユーザーアカウントの全エントリーが SSSD キャッシュから削除されるまでの日数を設定します。これは、個別のオフライン認証情報キャッシュの有効期限と同じか、またはそれ以上である必要があります。*

- *entry_cache_timeout は、SSSD が ID プロバイダーから更新された情報を要求する前に、キャッシュに保存されているすべてのエントリーに有効期間を秒単位で設定します。group、service、netgroup、sudo、autofs エントリーには個別のキャッシュタイムアウトパラメーターがあります。これらは sssd.conf の man ページに記載されています。デフォルトの時間は 5400 秒（90 分）です。*

以下に例を示します。

```
[sssd]
services = nss,pam
...

[pam]
offline_credentials_expiration = 3
...

[domain/EXAMPLE]
cache_credentials = true
account_cache_expiration = 7
entry_cache_timeout = 14400
...
```

13.2.17. ドメインオプション： パスワードの有効期限の設定

パスワードポリシーは通常、有効期限を設定し、その後パスワードを交換する必要があります。パスワードの有効期限ポリシーは、アイデンティティープロバイダーを介してサーバー側で評価され、PAM サービスを介して SSSD で警告を処理し、表示することができます。

パスワードの有効期限の警告を表示する方法は 2 つあります。

- *pam_pwd_expiration_warning パラメーターは、パスワードの有効期限の前に警告を表示する前に、すべてのドメインのグローバルデフォルト設定を定義します。これは PAM サービスに設定されます。*

- *pwd_expiration_warning パラメーターは、パスワードの有効期限の前に警告を表示する前に、ドメインごとの設定を定義します。*

ドメインレベルのパスワード有効期限の警告を使用する場合は、認証プロバイダー (auth_provider)もドメインに設定する必要があります。

以下に例を示します。

```
[sssd]
services = nss,pam
...

[pam]
pam_pwd_expiration_warning = 3
...

[domain/EXAMPLE]
id_provider = ipa
auth_provider = ipa
pwd_expiration_warning = 7
```

警告を表示するには、パスワードの有効期限の警告をサーバーから SSSD に送信する必要があります。サーバーからパスワード警告が送信されていない場合は、パスワードの有効期限が SSSD で設定された期間内にある場合でも、SSSD を介してメッセージは表示されません。

パスワード有効期限の警告が SSSD で設定されていない場合や、0 に設定されていない場合は、SSSD パスワードの警告フィルターが適用されず、サーバー側のパスワード警告が表示されます。

注記

パスワード警告がサーバーから送信される限り、PAM またはドメインパスワードの有効期限は、バックエンドアイデンティティープロバイダーのパスワード警告設定を上書きします。たとえば、警告期間が 28 日に設定され、SSSD の PAM サービスが 7 日に設定されているバックエンドアイデンティティープロバイダーについて考えてみましょう。プロバイダーは 28 日以降の SSSD に警告を送信しますが、SSSD 設定で指定したパスワードの有効期限に従い、警告が 7 日までローカルで表示されません。

パスワード以外の認証についてのパスワードの有効期限の警告

デフォルトでは、パスワードの有効期限は、ユーザーが認証中にパスワードを入力する場合にのみ検証されます。ただし、たとえば SSH ログイン時など、パスワード以外の認証方法が使用されている場合でも、有効期限チェックを実行し、警告を表示するように SSSD を設定できます。

パスワード以外の認証方法でパスワード有効期限の警告を有効にするには、以下を実行します。

1. *sssd.conf ファイルで access_provider パラメーターが ldap に設定されていることを確認します。*

2. *sssd.conf で ldap_pwd_policy パラメーターが設定されていることを確認してください。多くの場合、適切な値は shadow です。*

3. *sssd.conf の ldap_access_order パラメーターに、以下の pwd_expire_* の値のいずれかを追加します。パスワードが期限切れになる場合は、これらの値の 1 つが有効期限の警告のみを表示します。また、以下を追加しています。*

   - *pwd_expire_policy_reject は、パスワードの有効期限が切れている場合にユーザーがログインできないようにします。*
   - *pwd_expire_policy_warn を使用すると、パスワードが期限切れであってもユーザーはログインできます。*
   - *pwd_expire_policy_renew により、ユーザーが期限切れのパスワードでログインしようとすると、すぐにパスワードを変更するように求められます。*

   以下に例を示します。

```
[domain/EXAMPLE]
access_provider = ldap
ldap_pwd_policy = shadow
ldap_access_order = pwd_expire_policy_warn
```

ldap_access_order の使用方法とその値の詳細は、sssd-ldap(5) の man ページを参照してください。

13.2.18. ドメインオプション： DNS サービス検出の使用

RFC 2782 で定義された DNS サービス検出により、アプリケーションが特定タイプの特定サービスに対して指定のドメインの SRV レコードを確認でき、そのタイプで検出されたサーバーが返されます。

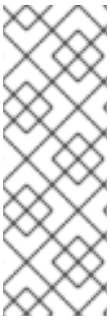
SSSD では、ID プロバイダーおよび認証プロバイダーは（IP アドレスまたはホスト名で）明示的に定義するか、サービス検出を使用して動的に検出できます。プロバイダーサーバーが一覧にない場合は（たとえば、id_provider = ldap に対応する ldap_uri パラメーターなしで設定されている場合）、検出は自動的に使用されます。

DNS 検出クエリーの形式は以下のとおりです。

```
_service._protocol.domain
```

たとえば、example.com ドメインで TCP を使用して LDAP サーバーのスキャンは以下のようになります。

```
_ldap._tcp.example.com
```



注記

サービス検出を使用するすべてのサービスについて、特別な DNS レコードを DNS サーバーに追加します。

```
_service._protocol._domain TTL priority weight port hostname
```

SSSD の場合、サービス種別はデフォルトで LDAP であり、ほぼすべてのサービスは TCP を使用します（ただし、UDP で開始する Kerberos を除く）。サービス検出を有効にするには、ドメイン名のみが必要になります。デフォルトではマシンホスト名のドメイン部分を使用しますが、別のドメインを指定することもできます（ dns_discovery_domain パラメーターを使用）。

そのため、デフォルトでは 1 つの例外を除き、サービス検出に追加の設定を行う必要はありません。パスワード変更プロバイダーはデフォルトでサーバー検出が無効になり、サービスタイプを設定して明示的に有効にする必要があります。

```
[domain/EXAMPLE]
...
chpass_provider = ldap
```

```
ldap_chpass_dns_service_name = ldap
```

設定は必要ありませんが、別の DNS ドメイン(dns_discovery_domain)を使用するか、スキャンする別のサービスタイプを設定してサーバー検出をカスタマイズすることができます。以下に例を示します。

```
[domain/EXAMPLE]
id _provider = ldap

dns_discovery_domain = corp.example.com
ldap_dns_service_name = ldap

chpass_provider = krb5
ldap_chpass_dns_service_name = kerberos
```

最後に、サービス検出はバックアップサーバーでは使用されません。これは、プロバイダーのプライマリーサーバーにのみ使用されます。つまり、検出を使用して最初にサーバーを特定でき、SSSD はバックアップサーバーを使用するようにフォールバックできます。プライマリーサーバーに検出を使用するには、_srv_ をプライマリーサーバーの値として使用し、バックアップサーバーを一覧表示します。以下に例を示します。

```
[domain/EXAMPLE]
id _provider = ldap
ldap_uri = _srv_
ldap_backup_uri = ldap://ldap2.example.com

auth_provider = krb5
krb5_server = _srv_
krb5_backup_server = kdc2.example.com

chpass_provider = krb5
ldap_chpass_dns_service_name = kerberos
ldap_chpass_uri = _srv_
ldap_chpass_backup_uri = kdc2.example.com
```

注記

サービス検出はバックアップサーバーとは使用できません。プライマリーサーバーのみを使用できます。

DNS ルックアップがホスト名の IPv4 アドレスが返されない場合、SSSD は障害を返す前に IPv6 アドレスの検索を試行します。これにより、非同期リゾルバーが正しいアドレスを識別することのみが保証されます。

ホスト名の解決の動作は、sssd.conf 設定ファイルの lookup family order オプションで設定され

ます。

13.2.19. ドメインオプション： 証明書のサブジェクト名での IP アドレスの使用（LDAP のみ）

サーバー名の代わりに ldap_uri オプションで IP アドレスを使用すると、TLS/SSL 接続が失敗する可能性があります。TLS/SSL 証明書には、IP アドレスではなくサーバー名が含まれます。ただし、証明書の サブジェクトの別名 フィールドを使用して、サーバーの IP アドレスを含めることができます。これにより、IP アドレスを使用した正常な接続が可能になります。

手順13.8 証明書のサブジェクト名での IP アドレスの使用

1. 既存の証明書を証明書要求に変換します。署名鍵(-signkey)は、最初に証明書を発行した CA の発行者のキーです。外部 CA でこれを行う場合は、別の PEM ファイルが必要です。証明書が自己署名されている場合は、証明書自体になります。以下に例を示します。

```
openssl x509 -x509toreq -in old_cert.pem -out req.pem -signkey key.pem
```

自己署名証明書の場合：

```
openssl x509 -x509toreq -in old_cert.pem -out req.pem -signkey old_cert.pem
```

2. /etc/pki/tls/openssl.cnf 設定ファイルを編集して、[v3_ca] セクションにサーバーの IP アドレスを追加します。

```
subjectAltName = IP:10.0.0.10
```

3. 生成された証明書要求を使用して、指定した IP アドレスで新規の自己署名証明書を生成します。

```
openssl x509 -req -in req.pem -out new_cert.pem -extfile ./openssl.cnf -extensions v3_ca -
signkey old_cert.pem
```

-extensions オプションは、証明書で使用する拡張機能を設定します。適切なセクションを読み込むには、v3_ca にする必要があります。

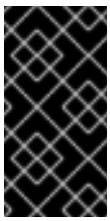
4. old_cert.pem ファイルから new_cert.pem ファイルに秘密鍵ブロックをコピーし、1 つのファイルに関連情報をすべて保持します。

nss-tools パッケージが提供する certutil ユーティリティーを使用して証明書を作成する場合は、certutil が証明書作成の DNS サブジェクト代替名をサポートすることに注意してください。

13.2.20. ドメインの作成： プロキシー

SSSD を使用するプロキシーは、中間設定であるリレーです。SSSD はそのプロキシーサービスに接続し、そのプロキシーが指定のライブラリーを読み込みます。これにより、SSSD は使用できない一部のリソースを使用できます。たとえば、SSSD は認証プロバイダーとして LDAP および Kerberos のみをサポートしますが、プロキシーを使用すると、SSSD はフィンガープリントスキャナーやスマートカードなどの別の認証方法を使用できます。

表13.9 プロキシードメイン設定パラメーター

パラメーター	*説明*
proxy_pam_target	*PAM を認証プロバイダーとしてプロキシーする必要があるターゲットを指定します。PAM ターゲットは、デフォルトの PAM ディレクトリー /etc/pam.d/ に PAM スタック情報が含まれるファイルです。* *これは、認証プロバイダーのプロキシーに使用されます。*  *重要* *プロキシー PAM スタックに pam_sss.so が再帰的に追加 されない ようにします。*
proxy_lib_name	*ID 要求をプロキシー処理する既存の NSS ライブラリーを指定します。* *これは、アイデンティティープロバイダーのプロキシーに使用されます。*

例13.10 プロキシー ID および Kerberos 認証

プロキシーライブラリーは、proxy_lib_name パラメーターを使用して読み込まれます。このライブラリーは、指定の認証サービスと互換性がある限りすべて使用できます。Kerberos 認証プロバイダーの場合は、NIS などの Kerberos 互換ライブラリーである必要があります。

```
[domain/PROXY_KRB5]
auth_provider = krb5
krb5_server = kdc.example.com
krb5_realm = EXAMPLE.COM

id_provider = proxy
proxy_lib_name = nis
cache_credentials = true
```

例13.11 LDAP アイデンティティーおよびプロキシー認証

プロキシーライブラリーは、proxy_pam_target パラメーターを使用して読み込まれます。このライブラリーは、指定のアイデンティティープロバイダーと互換性のある PAM モジュールである必要があります。たとえば、以下は LDAP で PAM フィンガープリントモジュールを使用します。

```
[domain/LDAP_PROXY]
id_provider = ldap
ldap_uri = ldap://example.com
ldap_search_base = dc=example,dc=com

auth_provider = proxy
proxy_pam_target = sssdpamproxy
cache_credentials = true
```

SSSD ドメインを設定したら、指定した PAM ファイルが設定されていることを確認してください。この例では、ターゲットは sssdpamproxy であるため、/etc/pam.d/sssdpamproxy ファイルを作成し、PAM/LDAP モジュールを読み込みます。

```
auth        required      pam_frprint.so
account     required      pam_frprint.so
password    required      pam_frprint.so
session     required      pam_frprint.so
```

例13.12 プロキシー ID および認証

SSSD には、アイデンティティープロキシーと認証プロキシーの両方を持つドメインを使用できます。指定される設定はプロキシー設定、認証 PAM モジュールの proxy_pam_target、NIS や LDAP などのサービスの proxy_lib_name のみです。

この例は可能な設定を示していますが、これは現実的な設定ではありません。LDAP がアイデンティティーと認証に使用される場合、ID プロバイダーと認証プロバイダーの両方が、プロキシーではなく LDAP 設定に設定する必要があります。

```
[domain/PROXY_PROXY]
auth_provider = proxy
id_provider = proxy
proxy_lib_name = ldap
proxy_pam_target = sssdproxyldap
cache_credentials = true
```

SSSD ドメインを追加したら、システム設定を更新して、プロキシーサービスを設定します。

1. *pam_ldap.so モジュールを必要とする /etc/pam.d/sssdproxyldap ファイルを作成します。*

```
auth        required      pam_ldap.so
account     required      pam_ldap.so
password    required      pam_ldap.so
session     required      pam_ldap.so
```

2. *nss-pam-ldapd パッケージがインストールされていることを確認します。*

```
~]# yum install nss-pam-ldapd
```

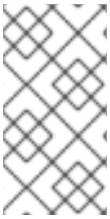
3. *LDAP ディレクトリー情報が含まれるように、/etc/nslcd.conf ファイル（LDAP ネームサービスデーモンの設定ファイル）を編集します。*

```
uid nslcd
gid ldap
uri ldaps://ldap.example.com:636
base dc=example,dc=com
ssl on
tls_cacertdir /etc/openldap/cacerts
```

13.2.21. ドメインの作成： Kerberos 認証

LDAP およびプロキシーのアイデンティティープロバイダーは、共に個別の Kerberos ドメインを使用して認証を提供できます。Kerberos 認証プロバイダーを設定するには、キー配布センター (KDC) と Kerberos ドメインが必要です。すべてのプリンシパル名は、指定のアイデンティティープロバイ

ダーで利用可能でなければなりません。そうでない場合は、SSSD は username@REALM 形式を使用してプリンシパルを構築します。



注記

Kerberos は認証のみを提供でき、アイデンティティーデータベースを提供することはできません。

SSSD は、Kerberos KDC も Kerberos kadmin サーバーであることを前提としています。ただし、実稼働環境では一般的に KDC の複数の読み取り専用レプリカと、単一の kadmin サーバーのみがあります。krb5_kpasswd オプションを使用して、パスワードの変更サービスが実行されている場所、またはデフォルト以外のポートで実行しているかどうかを指定します。krb5_kpasswd オプションが定義されていない場合は、SSSD は Kerberos KDC を使用してパスワードを変更します。

基本的な Kerberos 設定オプションは、表13.10「Kerberos 認証設定パラメーター」 に記載されています。man ページの sssd-krb5(5) には、Kerberos 設定オプションの詳細情報が記載されています。

例13.13 基本的な Kerberos 認証

```
# A domain with identities provided by LDAP and authentication by Kerberos
[domain/KRBDOMAIN]
id_provider = ldap
chpass_provider = krb5
ldap_uri = ldap://ldap.example.com
ldap_search_base = dc=example,dc=com
ldap-tls_reqcert = demand
ldap_tls_cacert = /etc/pki/tls/certs/ca-bundle.crt

auth_provider = krb5
krb5_server = kdc.example.com
krb5_backup_server = kerberos.example.com
krb5_realm = EXAMPLE.COM
krb5_kpasswd = kerberos.admin.example.com
krb5_auth_timeout = 15
```

例13.14 Kerberos チケット更新オプションの設定

Kerberos 認証プロバイダー（他のタスクとともに）は、ユーザーおよびサービスのチケット保証チケット(TGT)を要求します。これらのチケットは、チケットプリンシパル（ユーザー）がアクセスするなど、特定のサービスに対して動的に他のチケットを生成するために使用されます。

ユーザープリンシパルに最初に付与される TGT は、チケットの有効期間（デフォルトでは設定

された KDC で設定されたもの）に対してのみ有効です。その後、チケットを更新または拡張することができません。ただし、チケットを更新しないと、操作中にサービスにアクセスしようとし、チケットの有効期限が切れると、一部のサービスで問題が発生する可能性があります。

Kerberos チケットはデフォルトでは更新できませんが、krb5_renewable _lifetime パラメーターおよび krb5_renew_ interval パラメーターを使用してチケットの 更新を有効にできます。

チケットの有効期間は、krb5_lifetime パラメーターで SSSD で設定されます。これは、単一のチケットが有効である期間を指定し、KDC の任意の値を上書きします。

チケット更新自体は krb5_renewable_lifetime パラメーターで有効になっ ています。このパラメーターは、チケットの最大有効期間を設定し、すべての更新をカウントします。

たとえば、チケットの有効期間は 1 時間に設定され、更新可能な期間は 24 時間に設定されます。

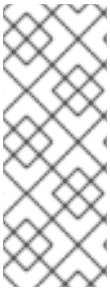
```
krb5_lifetime = 1h
krb5_renewable_lifetime = 1d
```

つまり、チケットは 1 時間ごとに有効期限が切れ、1 日に継続的に更新できます。

有効期間および更新可能な有効期間の値は、秒(s)、分(m)、時間(h)、または日(d)のいずれかになります。

もう 1 つのオプション - チケットの更新にも設定する必要があります。krb5_renew_interval パラメーターです。これは、チケットを更新する必要があるかどうかを確認するために SSSD がチェックする頻度を設定します。チケットの有効期間の半分（設定が重要）、チケットは自動的に更新されます。（この値は常に秒単位です。）

```
krb5_lifetime = 1h
krb5_renewable_lifetime = 1d
krb5_renew_interval = 60s
```



注記

krb5_renewable_lifetime 値が設定されていないか、krb5_renew_interval パラメーターが設定されていないか、ゼロ(0)に設定されている場合、チケットの更新が無効になります。チケットの更新 を有効にするには、krb5_renewable_lifetime および krb5_renew_interval の両方が必要になります。

表13.10 Kerberos 認証設定パラメーター

パラメーター	説明
chpass_provider	パスワード変更操作に使用するサービスを指定します。これは、認証プロバイダーと同じであることを前提としています。Kerberos を使用するには、これを krb5 に設定します。
krb5_server	SSSD が接続する IP アドレスまたはホスト名でプライマリー Kerberos サーバーを指定します。
krb5_backup_server	プライマリーサーバーが利用できない場合に SSSD が接続する Kerberos サーバーの IP アドレスまたはホスト名のコンマ区切りリストを指定します。一覧は優先順に指定されるため、一覧の最初のサーバーが最初に試行されます。 1 時間後に、SSSD は krb5_server パラメーターで指定されたプライマリーサービスへの再接続を試みます。 KDC または kpasswd サーバーのサービス検出を使用する場合、SSSD はまず UDP を接続プロトコルとして指定する DNS エントリーを検索し、その後に TCP にフォールバックします。
krb5_realm	KDC が提供する Kerberos レルムを特定します。
krb5_lifetime	指定された有効期間(s)、分(m)、時間(h)、または日(d)で Kerberos チケットを要求します。
krb5_renewable_lifetime	更新可能な Kerberos チケットを、秒(s)、分(m)、時間(h)、または日(d)で指定した合計有効期間でリクエストします。

パラメーター	説明
krb5_renew_interval	SSSD がチケットを更新するかどうかをチェックする時間を秒単位で設定します。チケットは、有効期間が半分以上になると自動的に更新されます。このオプションがないと、またはゼロに設定されていると、自動チケットの更新が無効になります。
krb5_store_password_if_offline	Kerberos 認証プロバイダーがオフラインの場合にユーザーパスワードを保存するかどうかを設定し、プロバイダーがオンラインに戻る際にそのキャッシュを使用してチケットを要求するかどうかを設定します。デフォルトは false で、パスワードを保存しません。
krb5_kpasswd	変更パスワードサービスが KDC で実行していない場合は、使用する別の Kerberos kadmin サーバーを一覧表示します。

パラメーター	説明
krb5_ccname_template	ユーザーの認証情報キャッシュの保存に使用するディレクトリーを指定します。これはテンプレート化でき、以下のトークンがサポートされます。 • %u （ユーザーのログイン名） • %u （ユーザーのログイン UID） • %p （ユーザーのプリンシパル名） • %r （レルム名） • %h （ユーザーのホームディレクトリー） • %d （ krb5ccache_dir パラメーターの値） • %p: SSSD クライアントのプロセス ID。 • %%、リテラルパーセント記号(%) • XXXXXX: テンプレートの最後に、一意のファイル名を安全に作成するように指示する文字列 以下に例を示します。 krb5_ccname_template = FILE:%d/krb5cc_%U_XXXXXX

パラメーター	説明
krb5_ccachedir	認証情報キャッシュを保存するディレクトリーを指定します。これは、%d および %P を除き、krb5_ccname_template と同じトークンを使用して、テンプレート 化できます。%u、%U、%p、または %h を使用すると、SSSD は各ユーザーにプライベートディレクトリーを作成します。それ以外の場合は、パブリックディレクトリーを作成します。
krb5_auth_timeout	オンライン認証またはパスワードの変更要求が中断されるまでの時間を秒単位で指定します。可能な場合は、認証要求はオフラインで続行されます。デフォルトは 15 秒です。

13.2.22. ドメインの作成： アクセス制御

SSSD は、ドメイン設定の基本的なアクセス制御を提供します。これにより、シンプルなユーザーを許可/拒否リストまたは LDAP バックエンド自体を使用できます。

Simple アクセスプロバイダーの使用

Simple Access Provider は、ユーザー名またはグループのリストに基づいてアクセスを許可または拒否します。

Simple Access Provider は、特定のマシンへのアクセスを制限する方法です。たとえば、ある会社がラップトップを使用する場合、Simple Access Provider を使用して、同じ認証プロバイダーに対して別のユーザーが認証した場合でも、特定のユーザーまたは特定のグループのみへのアクセスを制限できます。

最も一般的なオプションは simple_allow_users と simple_allow_groups で、特定のユーザー（特定のユーザーまたはグループメンバーのいずれか）に明示的にアクセスを許可し、他のすべてのユーザーへのアクセスを拒否します。拒否リストを作成することも可能です（明示的なユーザーのみへのアクセスを制限し、他のすべてのアクセスを暗黙的に許可します）。

Simple Access プロバイダーは、以下の 4 つのルールに従って、アクセスが付与されるべきユーザーを決定します。

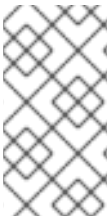
- allow および deny リストの両方が空の場合は、アクセスが許可されます。
- 一覧を指定すると、許可ルールが最初に評価され、次にルールを拒否します。実際には、拒否ルールが許可ルールよりも優先されることを意味します。

- 許可リストを指定すると、リストでない限り、すべてのユーザーは拒否されます。

- 拒否リストのみを指定すると、リストでない限り、すべてのユーザーがアクセスが許可されます。

この例では、2 人のユーザーと IT グループに属するユーザーへのアクセスを付与します。暗黙的に、他のすべてのユーザーは拒否されます。

```
[domain/example.com]
access_provider = simple
simple_allow_users = jsmith,bjensen
simple_allow_groups = itgroup
```



注記

SSSD の LOCAL ドメインは、アクセスプロバイダーとして simple をサポートしません。

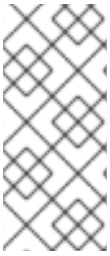
その他のオプションは sssd-simple の man ページに記載されていますが、これはほとんど使用されません。

アクセスフィルターの使用

LDAP サーバー、Active Directory サーバー、または Identity Management サーバーは、ドメインのアクセス制御ルールを提供できます。関連するオプション（LDAP の場合はldap_access_filter、AD の場合は ad_access_filter ）は、指定したホストへのアクセスが付与されるユーザーを指定します。ユーザーフィルターを使用するか、すべてのユーザーがアクセスを拒否する必要があります。以下の例を参照してください。

```
[domain/example.com]
access_provider = ldap
ldap_access_filter = memberOf=cn=allowedusers,ou=Groups,dc=example,dc=com
```

```
[domain/example.com]
access_provider = ad
ad_access_filter = memberOf=cn=allowedusers,ou=Groups,dc=example,dc=com
```



注記

LDAP アクセスプロバイダーのオフラインキャッシュは、ユーザーの最後にオンラインログインの試行が成功したかどうかを判断するために制限されます。最後のログイン時にアクセスが付与されたユーザーは、オフライン時でもアクセスが許可されます。

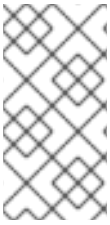
SSSD は、エントリーの authorizedService または host 属性で結果を確認することもできます。実際、ユーザーエントリーおよび設定に応じて、すべてのオプション (LDAP フィルター、authorizedService、および host) を評価できます。ldap_access_order パラメーターは、評価すべき順に、使用するアクセス制御の手法をすべて表示します。

```
[domain/example.com]
access_provider = ldap
ldap_access_filter = memberOf=cn=allowedusers,ou=Groups,dc=example,dc=com
ldap_access_order = filter, host, authorized_service
```

認可されたサービスまたは許可されたホストの評価に使用するユーザーエントリーの属性をカスタマイズできます。追加のアクセス制御パラメーターは sssd-ldap(5) の man ページに一覧表示されます。

13.2.23. ドメインの作成： プライマリーサーバーとバックアップサーバー

ドメインの ID プロバイダーおよび認証プロバイダーは自動フェイルオーバー用に設定できます。SSSD は、最初に指定のプライマリーサーバーに接続を試みます。そのサーバーに到達できない場合は、SSSD はリストされたバックアップサーバーを通過します。



注記

SSSD は、接続が再確立されるまで 30 秒ごとにプライマリーサーバーに接続し、バックアップからプライマリーに切り替えます。

サービスエリアのすべてには、プライマリーサーバーおよびバックアップサーバーのオプション設定があります。[3].

表13.11 プライマリーサーバーパラメーターとセカンダリーサーバーパラメーター

サービスエリア	プライマリーサーバー属性	バックアップサーバーの属性
LDAP アイデンティティープロバイダー	ldap_uri	ldap_backup_uri

サービスエリア	プライマリーサーバー属性	バックアップサーバーの属性
Active Directory アイデンティティープロバイダー	ad_server	ad_backup_server
Identity Management（IdM または IPA）アイデンティティープロバイダー	ipa_server	ipa_backup_server
Kerberos 認証プロバイダー	krb5_server	krb5_backup_server
Kerberos 認証プロバイダー	krb5_server	krb5_backup_server
パスワード変更プロバイダー	ldap_chpass_uri	ldap_chpass_backup_uri

プライマリーサーバーとして設定できるサーバーは 1 つだけです。（オプション）プライマリーサーバーはホスト名ではなく _srv_ を使用して、サービス検出に設定できます。） 複数のバックアップサーバーをコンマ区切りリストで設定できます。バックアップサーバーの一覧は優先順に行われるため、最初にリストアップされたサーバーが最初に試行されます。

```
[domain/EXAMPLE]
id_provider = ad
ad_server = ad.example.com
ad_backup_server = ad1.example.com, ad-backup.example.com
```

13.2.24. SSSD ユーティリティーのインストール

SSSD キャッシュ、ユーザーエントリー、およびグループエントリーを処理する追加のツールは sssd-tools パッケージに含まれます。このパッケージは必須ではありませんが、ユーザーアカウントの管理に役立つインストールに役立ちます。

```
~]# yum install sssd-tools
```

注記

sssd-tools パッケージは、Optional サブスクリプションチャンネルで提供されます。Red Hat 追加チャンネルの詳細は、「Optional および Supplementary リポジトリーの追加」 を参照してください。

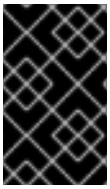
13.2.25. SSSD および UID および GID 番号

ユーザーが作成されると、useradd などのシステムツールを使用して、Red Hat Identity Management またはその他のクライアントツールを使用して、ユーザー ID 番号とグループ ID 番号が

自動的に割り当てられます。

ユーザーがシステムまたはサービスにログインすると、SSSD は、関連する UID/GID 番号でそのユーザー名をキャッシュします。次に、UID 番号がユーザーの ID キーとして使用されます。ユーザー名が同じで UID が異なるユーザーがシステムにログインすると、SSSD は名前が競合する 2 つの異なるユーザーとして処理します。

つまり、SSSD は UID 番号の変化を認識しません。SSSD は、異なる UID が指定された既存ユーザーとしてではなく、別の新規ユーザーとして解釈します。既存のユーザーの UID 番号が変更されると、そのユーザーは SSSD、関連のサービスやドメインにログインできなくなります。また、これは ID 情報に SSSD を使用するクライアントアプリケーションにも影響があり、競合が発生したユーザーは検索されず、これらのアプリケーションにアクセスできなくなります。



重要

SSSD では、UID/GID の変更に対応していません。

何らかの理由で UID/GID 番号が変更された場合は、そのユーザーが再ログインする前に、ユーザーの SSSD キャッシュを消去する必要があります。以下に例を示します。

```
~]# sss_cache -u jsmith
```

SSSD キャッシュのクリーンアップは、「SSSD キャッシュのパージ」 で説明されています。

13.2.26. ローカルシステムユーザーの作成

ユーザーがログインして追加するのを待たずに、ユーザーを SSSD データベースに閲覧すると便利な場合もあります。

注記

ユーザーアカウントを手動で追加するには、sssd-tools パッケージをインストールする必要があります。

新たなシステムユーザーを作成する場合、SSSD のローカル ID プロバイダードメインにユーザーを作成できます。これは、新しいシステムユーザーの作成や、SSSD 設定のトラブルシューティング、特別なグループまたはネスト化されたグループの作成に役立ちます。

sss_useradd コマンドを使用して、新しいユーザーを追加できます。

最も基本的なコマンド sss_useradd コマンドには、新しいユーザー名のみが必要です。

```
~]# sss_useradd jsmith
```

UID や GID、ホームディレクトリー、ユーザーが属するグループなど、アカウントに属性を設定するのに使用できる他のオプション （sss_useradd(8)にリストされている）もあります。

```
~]# sss_useradd --UID 501 --home /home/jsmith --groups admin,dev-group jsmith
```

13.2.27. キックスタート中の SSSD キャッシュへのユーザーのシード

注記

ユーザーアカウントを手動で追加するには、sssd-tools パッケージをインストールする必要があります。

SSSD では、リモートドメインのユーザーは、そのアイデンティティーがアイデンティティープロバイダーから取得されるまでローカルシステムでは使用できません。ただし、ユーザー ID がネットワーク上にある場合、一部のネットワークインターフェースはユーザーがログインするまで利用できません。このような場合は、適切なドメインに関連付けられたユーザー ID で SSSD キャッシュを確認し、ユーザーがローカルでログインし、適切なインターフェースをアクティブにすることができます。

これは、sss_seed ユーティリティーを使用して行います。

```
sss_seed --domain EXAMPLE.COM --username testuser --password-file /tmp/sssd-pwd.txt
```

このユーティリティーには、少なくともユーザー名、ドメイン名、およびパスワードを特定するオプションが必要です。

- --domain は、SSSD 設定からのドメイン名を提供します。このドメインは、SSSD 設定にすでに存在している必要があります。
- --username は、ユーザーアカウントの短縮名です。

- *--password-file: seed エントリーの一時パスワードが含まれるファイルのパスおよび名前。ユーザーアカウントが SSSD キャッシュにすでに存在する場合は、このファイルの一時パスワードは SSSD キャッシュに保存されているパスワードを上書きします。*

追加のアカウント設定オプションは、man ページの sss_seed(8) に記載されています。

これはほとんどの場合、キックスタートまたは自動セットアップの一部として実行されるため、SSSD を有効にし、SSSD ドメインを設定してパスワードファイルを作成します。以下に例を示します。

```
function make_sssd {
cat <<- _EOF_
[sssd]
domains = LOCAL
services = nss,pam

[nss]

[pam]

[domain/LOCAL]
id_provider = local
auth_provider = local
access_provider = permit

_EOF_
}

make_sssd >> /etc/sssd/sssd.conf

authconfig --enablesssd --enablesssdauth --update

function make_pwdfile {
cat <<1 _EOF_
password
_EOF_
}

make_pwdfile >> /tmp/sssd-pwd.txt

sss_seed --domain EXAMPLE.COM --username testuser --password-file /tmp/sssd-pwd.txt
```

13.2.28. SSSD キャッシュの管理

SSSD は、同じタイプのドメインと異なるタイプのドメインを定義できます。SSSD はドメインごとに個別のデータベースファイルを維持します。つまり、各ドメインには独自のキャッシュがあります。このキャッシュファイルは /var/lib/sss/db/ ディレクトリーに保存されます。

SSSD キャッシュのパージ

ドメインのアイデンティティープロバイダーに LDAP 更新が行われるため、キャッシュをクリアして新しい情報を迅速に再読み込みする必要がある場合があります。

キャッシュの purge ユーティリティー sss_cache は、ユーザー、ドメイン、またはグループの SSSD キャッシュにレコードを無効にします。現在のレコードを無効にすると、キャッシュがアイデンティティープロバイダーから更新されたレコードを取得するよう強制するため、変更はすぐに実行できます。

注記

このユーティリティーは、sssd パッケージの SSSD に含まれています。

最も一般的なものは、キャッシュをクリアし、すべてのレコードを更新するために使用されます。

```
~]# sss_cache -E
```

sss_cache コマンドは、特定のドメインのキャッシュされたエントリーをすべて削除することもできます。

```
~]# sss_cache -Ed LDAP1
```

管理者が特定のレコード（ユーザー、グループ、または netgroup）が更新されていることが分かっている場合は、sss_cache はその特定のアカウントのレコードをパージし、残りのキャッシュをそのまま残すことができます。

```
~]# sss_cache -u jsmith
```

表13.12 一般的な sss_cache オプション

短い引数	長い引数	説明
-E	--everything	sudo ルールを除くキャッシュされたエントリーをすべて無効にします。
-d **name**	--domain **name**	指定のドメイン内でのみ、ユーザー、グループ、およびその他のエントリーのキャッシュエントリーを無効にします。
-G	--groups	すべてのグループレコードを無効にします。**-g** も使用する場合は、**-G** が優先されます。**-g** は無視されます。

短い引数	長い引数	説明
-g **name**	--group **name**	指定のグループのキャッシュエントリーを無効にします。
-N	--netgroups	すべての netgroup キャッシュレコードのキャッシュエントリーを無効にします。**-n** も使用する場合は、**-N** が優先されます。**-n** は無視されます。
-n **name**	--netgroup **name**	指定した netgroup のキャッシュエントリーを無効にします。
-U	--users	すべてのユーザーレコードのキャッシュエントリーを無効にします。**-u** オプションも使用する場合は、**-U** が優先されます。**-u** は無視されます。
-u **name**	--user **name**	指定のユーザーのキャッシュエントリーを無効にします。

ドメインキャッシュファイルの削除

すべてのキャッシュファイルの名前がドメインに対して付けられます。たとえば、exampleldap という名前のドメインの場合、キャッシュファイルの名前は cache_exampleldap.ldb です。

キャッシュファイルを削除する場合は注意してください。この操作には大きな影響があります。

- ***キャッシュファイルを削除すると、識別とキャッシュされた認証情報の両方など、すべてのユーザーデータが削除されます。したがって、システムがオンラインになっていず、ドメインのサーバーに対してユーザー名で認証できる場合を除き、キャッシュファイルを削除しないでください。認証情報キャッシュがないと、オフライン認証は失敗します。***

- ***別のアイデンティティープロバイダーを参照するように設定を変更すると、元のプロバイダーからキャッシュされたエントリーがタイムアウトするまで、SSSD は両方のプロバイダーのユーザーを認識します。***

  キャッシュをパージするとこれを回避することができますが、新しいプロバイダーに別のドメイン名を使用することが推奨されます。SSSD を再起動すると、新しい名前で新しいキャッシュファイルが作成され、古いファイルは無視されます。

13.2.29. SSSD のダウングレード

ダウングレード時 — SSSD のバージョンをダウングレードするか、オペレーティングシステム自体をダウングレードする場合、既存の SSSD キャッシュを削除する必要があります。キャッシュが削除さ

れないと、SSSDプロセスは停止しますが、PIDファイルは残ります。SSSD ログは、キャッシュバージョンが認識されないため、関連付けられたドメインのいずれかに接続できないことを示します。

```
(Wed Nov 28 21:25:50 2012) [sssd] [sysdb_domain_init_internal] (0x0010): Unknown DB
version [0.14], expected [0.10] for domain AD!
```

その後、ユーザーは認識されなくなり、ドメインサービスおよびホストに対して認証できなくなります。

SSSD バージョンをダウングレードしてから以下を行います。

1. 既存のキャッシュデータベースファイルを削除します。

```
~]# rm -rf /var/lib/sss/db/*
```

2. SSSD プロセスを再起動します。

```
~]# service sssd restart
Stopping sssd:                                [FAILED]
Starting sssd:                               [  OK  ]
```

13.2.30. SSSD での NSCD の使用

SSSD は、NSCD デーモンで使用するように設計されていません。SSSD は NSCD と直接競合しませんが、両サービスを使用すると、特にエントリーがキャッシュされる期間において予期しない動作が発生する可能性があります。

問題の最も一般的な証拠は NFS と競合します。Network Managerを使用してネットワーク接続を管理する場合、ネットワークインターフェイスが起動するまでに数分かかる場合があります。この間、さまざまなサービスが起動しようとします。これらのサービスがネットワークが稼働している前に起動し、DNS サーバーが利用できる場合には、これらのサービスは必要な正引きまたは逆引き DNS エントリーを特定できません。これらのサービスは、誤った、または空の resolv.conf ファイルを読み取る可能性があります。このファイルは、通常 1 回だけ読み取るため、このファイルへの変更は自動的に適用されません。これにより、サービスを手動で再起動しない限り、NSCD サービスが実行されているマシンで NFS ロックが失敗する可能性があります。

この問題を回避するには、/etc/nscd.conf ファイルでホストとサービスのキャッシュを有効にし、passwd、グループ、および netgroup エントリーの SSSD キャッシュに依存します。

/etc/nscd.conf ファイルを変更します。

```
enable-cache hosts yes
enable-cache passwd no
enable-cache group no
enable-cache netgroup no
```

ホストの要求に回答する NSCD により、これらのエントリーは NSCD によりキャッシュされ、ブートプロセス中に NSCD によって返されます。その他のエントリーはすべて SSSD により処理されます。

13.2.31. SSSD のトラブルシューティング

- *「SSSD ドメインのデバッグログの設定」*
- *「SSSD ログファイルの確認」*
- *「SSSD 設定に関する問題」*

SSSD ドメインのデバッグログの設定

各ドメインは、独自のデバッグログレベルを設定します。ログレベルを増やすと、SSSD またはドメイン設定の問題に関する詳細情報が提供されます。

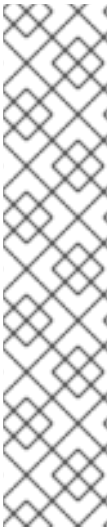
ログレベルを変更するには、sssd.conf ファイルの各セクションに debug_level パラメーターを設定し、追加のログを生成します。以下に例を示します。

```
[domain/LDAP]
cache_credentials = true
debug_level = 9
```

表13.13 デバッグログレベル

レベル	説明
0	致命的な障害。SSSDの起動を妨げる、またはSSSDの実行を停止させるもの。
1	重大なエラー。SSSD を強制終了しないものの、少なくとも 1 つの主要な機能が適切に機能していないことを示すエラー。

レベル	説明
2	深刻なエラー。特定の要求または操作が失敗したことを示すエラー。
3	マイナーな障害。これらのエラーが浸透して、2の動作不良の原因となるのです。
4	構成設定。
5	関数データ。
6	操作関数のメッセージを追跡します。
7	内部制御関数のメッセージのトレース。
8	対象の関数内部変数の内容。
9	非常に低いレベルのトレース情報。



注記

1.8 よりも古い SSSD のバージョンでは、[sssd] セクションにデバッグログレベルをグローバルに設定できます。今回のリリースより、各ドメインおよびサービスは独自のデバッグログレベルを設定する必要があります。

グローバル SSSD デバッグログレベルを SSSD 設定ファイルの各設定領域にコピーするには、sssd_update_debug_levels.py スクリプトを使用します。

```
python -m SSSDConfig.sssd_update_debug_levels.py
```

SSSD ログファイルの確認

SSSD は、/var/log/sssd/ ディレクトリーにある操作に関する情報を報告するログファイルを使用します。SSSD は、各ドメインのログファイルと sssd_pam.log および sssd_nss.log ファイルを生成します。

さらに、/var/log/secure ファイルは認証の失敗と、失敗の原因を記録します。

SSSD 設定に関する問題
問：
SSSD が起動に失敗する

答：

SSSD では、デーモンを起動する前に、必要なすべてのエントリーで設定ファイルを適切に設定する必要があります。

SSSD では、サービスが起動する前に、最低でもドメインを適切に設定する必要があります。ドメインがないと、SSSD を起動すると、ドメインが設定されていないエラーが返されます。

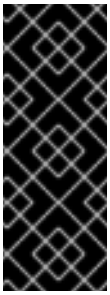
```
# sssd -d4

[sssd] [ldb] (3): server_sort:Unable to register control with rootdse!
[sssd] [confdb_get_domains] (0): No domains configured, fatal error!
[sssd] [get_monitor_config] (0): No domains configured.
```

/etc/sssd/sssd.conf ファイルを編集し、最低でも 1 つのドメインを作成します。

SSSD は、開始する前に、少なくとも 1 つ以上の利用可能なサービスプロバイダーも必要です。問題がサービスプロバイダー設定にある場合、エラーメッセージはサービスが設定されていないことを示します。

```
[sssd] [get_monitor_config] (0): No services configured!
```

/etc/sssd/sssd.conf ファイルを編集し、1 つ以上のサービスプロバイダーを設定します。



重要

SSSD では、サービスプロバイダーを /etc/sssd/sssd.conf ファイルの単一の services エントリーでコンマ区切りリストとして設定する必要があります。サービスが複数のエントリーに一覧表示されます。最後のエントリーのみが SSSD によって認識されます。

問：

「id」または「getent group」を持つグループメンバーを持つグループは表示されません。

答：

これは、sssd.conf の [domain/DOMAINNAME] セクションに誤った ldap_schema 設定が原因である可能性があります。

SSSD は RFC 2307 および RFC 2307bis スキーマタイプをサポートします。デフォルトでは、SSSD はより一般的な RFC 2307 スキーマを使用します。

RFC 2307 と RFC 2307bis の相違点は、グループメンバーシップが LDAP サーバーに保存される方法です。RFC 2307 サーバーでは、グループメンバーは、メンバーであるユーザーの名前が含まれる多値 memberuid 属性として保存されます。RFC2307bis サーバーでは、グループメンバーは、このグループのメンバーであるユーザーまたはグループの DN を含む多値 member または uniqueMember 属性として保存されます。RFC2307bis を使用すると、ネストされたグループも保守できます。

グループルックアップが情報が返されない場合は、以下を行います。

1. *ldap_schema を rfc2307bis に設定します。*
2. *Delete /var/lib/sss/db/cache_DOMAINNAME.ldb.*
3. *SSSD を再起動します。*

これが機能しない場合は、この行を sssd.conf に追加します。

```
ldap_group_name = uniqueMember
```

次に、キャッシュを削除し、再度 SSSD を再起動します。

問：

認証は LDAP に対して失敗します。

答：

認証を実行するには、SSSD で通信チャネルを暗号化する必要があります。これは、sssd.conf が標準プロトコル (ldap://) 経由で接続するように設定されていると、Start TLS で通信チャネルの暗号化を試みます。sssd.conf がセキュアなプロトコル (ldaps://) に接続するように設定されている場合、SSSD は SSL を使用します。

つまり、LDAP サーバーは SSL または TLS で実行する必要があります。標準のLDAPポート（389）でTLSを有効にするか、セキュアLDAPSポート（636）でSSLを有効にする必要があります。SSLまたはTLSのいずれかを使用する場合、LDAPサーバーも有効な証明書の信頼で構成する必要があります。

無効な証明書の信頼は、LDAPに対する認証に関する最も一般的な問題の1つです。クライアントがLDAPサーバー証明書を適切に信頼していない場合、接続を検証できず、SSSDはパスワードの送信を拒否します。LDAP プロトコルでは、パスワードをプレーンテキストで LDAP サーバーに送信する必要があります。暗号化されていない接続でプレーンテキストでパスワードを送信することは、セキュリティーの問題です。

証明書が信頼されていない場合は、TLS 暗号化を開始できなかったことを示す syslog メッセージが書き込まれます。証明書設定は、SSSD とは別に LDAP サーバーにアクセスできるかどうかを確認してテストできます。たとえば、以下は、test.example.com への TLS 接続を介して匿名バインドをテストします。

```
$ ldapsearch -x -ZZ -h test.example.com -b dc=example,dc=com
```

証明書信頼が適切に設定されていない場合、テストは以下のエラーを出して失敗します。

```
ldap_start_tls: Connect error (-11) additional info: TLS error -8179:Unknown code ___f 13
```

証明書を信頼するには、次のコマンドを実行します。

1. *LDAP サーバー証明書に署名するために使用される認証局の公開 CA 証明書のコピーを取得してローカルシステムに保存します。*

2. *ファイルシステムの CA 証明書を参照する sssd.conf ファイルに行を追加します。*

   ```
   ldap_tls_cacert = /path/to/cacert
   ```

3. *LDAP サーバーが自己署名証明書を使用する場合は、sssd.conf ファイルから ldap_tls_reqcert 行を削除します。*

   このパラメーターにより、SSSD が CA 証明書により発行された証明書を信頼するように指示します。これは、自己署名の CA 証明書を使用するセキュリティーリスクになります。

問：

非標準ポートで LDAP サーバーへの接続に失敗します。

答：

SELinux を Enforcing モードで実行する場合は、クライアントの SELinux ポリシーを変更して、標準以外のポートで LDAP サーバーに接続する必要があります。以下に例を示します。

```
# semanage port -a -t ldap_port_t -p tcp 1389
```

問：

NSS がユーザー情報を返すことができません

答：

これは通常、SSSD が NSS サービスに接続できないことを意味します。

NSS が実行していることを確認します。

```
# service sssd status
```

NSS が実行している場合、プロバイダーが /etc/sssd/sssd.conf ファイルの [nss] セクションで適切に設定されていることを確認します。特に、filter_users 属性および filter_groups 属性を確認します。

NSS が SSSD が使用するサービスの一覧に含まれていることを確認します。

/etc/nsswitch.conf ファイルの設定を確認します。

問：

NSS が間違ったユーザー情報を返す

答：

検索が正しくないユーザー情報を返した場合は、別のドメインでユーザー名が競合していないことを確認してください。複数のドメインがある場合は、/etc/sssd/sssd.conf ファイルで use_fully_qualified_domains 属性を true に設定します。これは、同じ名前の異なるドメインの異なるユーザーを区別します。

問：

ローカルの SSSD ユーザーのパスワードを設定すると、パスワードが 2 回要求されます。

答：

ローカルの SSSD ユーザーのパスワードを変更しようとすると、パスワードを 2 回求められる場合があります。

```
[root@clientF11 tmp]# passwd user1000
Changing password for user user1000.
New password:
Retype new password:
New Password:
Reenter new Password:
passwd: all authentication tokens updated successfully.
```

これは、PAM 設定が間違っているためです。/etc/pam.d/system-auth ファイルで use_authtok オプションが正しく設定されていることを確認します。

問：

Identity Management(IPA)プロバイダーで sudo ルールを使用しようとすると、sudo が適切に設定されていない場合でも sudo ルールは見つかりません。

答：

SSSD クライアントは Identity Management サーバーに正常に認証でき、sudo ルールの LDAP ディレクトリーを適切に検索します。ただし、ルールが存在しないことを示します。たとえば、ログで以下を行います。

```
(Thu Jun 21 10:37:47 2012) [sssd[be[ipa.test]]] [sdap_sudo_load_sudoers_process] (0x0400):
Receiving sudo rules with base [ou=sudoers,dc=ipa,dc=test]
(Thu Jun 21 10:37:47 2012) [sssd[be[ipa.test]]] [sdap_sudo_load_sudoers_done] (0x0400):
Received 0 rules
(Thu Jun 21 10:37:47 2012) [sssd[be[ipa.test]]] [sdap_sudo_purge_sudoers] (0x0400): Purging
SUDOers cache of user's [admin] rules
(Thu Jun 21 10:37:47 2012) [sssd[be[ipa.test]]] [sysdb_sudo_purge_byfilter] (0x0400): No rules
matched
(Thu Jun 21 10:37:47 2012) [sssd[be[ipa.test]]] [sysdb_sudo_purge_bysudouser] (0x0400): No
rules matched
(Thu Jun 21 10:37:47 2012) [sssd[be[ipa.test]]] [sdap_sudo_load_sudoers_done] (0x0400):
Sudoers is successfuly stored in cache
(Thu Jun 21 10:37:47 2012) [sssd[be[ipa.test]]] [be_sudo_handler_reply] (0x0200): SUDO
Backend returned: (0, 0, Success)
```

SSSD に Identity Management プロバイダーを使用する場合、SSSD は Kerberos/GSS-API を使用して基礎となる LDAP ディレクトリーへの接続を試行します。ただし、SSSD は LDAP サーバーへの匿名接続を使用して sudo ルールを取得します。つまり、SSSD は、デフォルト設定で Identity Management サーバーから sudo ルールを取得できません。

Kerberos/GSS-API 接続での sudo ルールの取得をサポートするには、sssd.conf の ID プロバイダー設定で認証メカニズムとして GSS-API を有効にします。以下に例を示します。

```
[domain/ipa.example.com]
id_provider = ipa
ipa_server = ipa.example.com
```

```
ldap_tls_cacert = /etc/ipa/ca.crt

sudo_provider = ldap
ldap_uri = ldap://ipa.example.com
ldap_sudo_search_base = ou=sudoers,dc=ipa,dc=example,dc=com
ldap_sasl_mech = GSSAPI
ldap_sasl_authid = host/hostname.ipa.example.com
ldap_sasl_realm = IPA.EXAMPLE.COM
krb5_server = ipa.example.com
```

問：

大規模なディレクトリーのパスワード検索には、要求ごとに数秒かかる場合があります。どのように改善できるのでしょうか？

答：

最初のユーザールックアップは、LDAP サーバーへの呼び出しです。インデックスのない検索はリソース集約型が多いため、サーバーはディレクトリー内のすべてのエントリーが一致するかどうかをチェックするため、インデックス化された検索よりも時間がかかります。ユーザー検索を迅速化するには、SSSD が検索する属性をインデックス化します。

uid

uidNumber

gidNumber

gecos

問：

Active Directory アイデンティティープロバイダーは sssd.conf ファイルで適切に設定されていますが、SSSD は接続に失敗し、GSS-API エラーが出されます。

答：

SSSD は、ホスト名を使用して Active Directory プロバイダーのみに接続できます。ホスト名が指定されていない場合、SSSD クライアントはホストの IP アドレスを解決できず、認証に失敗します。

たとえば、以下の設定を使用します。

```
[domain/ADEXAMPLE]
```

```
debug_level = 0xFFF0
id_provider = ad
ad_server = 255.255.255.255
ad_domain = example.com
krb5_canonicalize = False
```

SSSD クライアントはこの GSS-API 失敗を返し、認証要求に失敗します。

```
(Fri Jul 27 18:27:44 2012) [sssd[be[ADTEST]]] [sasl_bind_send] (0x0020): ldap_sasl_bind failed
(-2)[Local error]
(Fri Jul 27 18:27:44 2012) [sssd[be[ADTEST]]] [sasl_bind_send] (0x0080): Extended failure
message: [SASL(-1): generic failure: GSSAPI Error: Unspecified GSS failure.  Minor code may
provide more information (Cannot determine realm for numeric host address)]
```

このエラーを回避するには、ad_server を Active Directory ホストの名前に設定します。

問：

SSSD が中央認証用に設定されましたが、アプリケーション（Firefox、Adobe など）が複数起動しません。

答：

64 ビットシステムでも、32 ビットのアプリケーションでは、パスワードと ID キャッシュにアクセスするために使用する SSSD のバージョンが 32 ビットである必要があります。32 ビットバージョンの SSSD が利用できない場合は、システムが SSSD キャッシュを使用するように設定されている場合、32 ビットアプリケーションは起動に失敗する可能性があります。

たとえば、Firefox はパーミッション拒否エラーで失敗できます。

```
Failed to contact configuration server. See http://www.gnome.org/projects/gconf/
for information. (Details -  1: IOR file '/tmp/gconfd-somebody/lock/ior'
not opened successfully, no gconfd located: Permission denied 2: IOR
file '/tmp/gconfd-somebody/lock/ior' not opened successfully, no gconfd
located: Permission denied)
```

Adobe Reader の場合は、現在のシステムユーザーが認識されていないことを示すエラーが表示されます。

```
~]$ acroread
(acroread:12739): GLib-WARNING **: getpwuid_r(): failed due to unknown
user id (366)
```

他のアプリケーションは同様のユーザーまたはパーミッションエラーが表示される可能性があります。

問：

SSSD は、削除された自動マウントの場所を表示します。

答：

自動マウントの場所の SSSD キャッシュは、場所が後で変更または削除しても持続します。SSSD で autofs 情報を更新するには、次のコマンドを実行します。

1. *「SSSD キャッシュのパージ」 の説明に従って、autofs キャッシュを削除します。*

2. *「SSSD の起動および停止」 にあるように、SSSD を再起動します。*

---

[3]
サービスの特定サーバーが設定されていない場合、ほとんどのサービスはアイデンティティープロバイダーサーバーにデフォルト設定されます。

第14章 OPENSSH

SSH (Secure Shell)は、クライアント/サーバーアーキテクチャーを使用する 2 つのシステム間でのセキュアな通信を容易にし、ユーザーがリモートでサーバーホストシステムにログインできるようにするプロトコルです。FTP、Telnet、rlogin などの他のリモート通信プロトコルとは異なり、SSH はログインセッションを暗号化するため、侵入者が接続して暗号化されていないパスワードを収集するのが困難になります。

ssh プログラムは、telnet や rsh などのリモートホストへのログインに使用される、旧式で、セキュリティー保護が十分でない端末アプリケーションを置き換えるように設計されています。scp と呼ばれる関連プログラムが、ホスト間でファイルをコピーするように設計された rcp などの古いプログラムに代わるものです。このような旧式アプリケーションは、クライアントとサーバーとの間で送信するパスワードを暗号化しないため、可能な限り使用しないようにしてください。リモートシステムへのログインにセキュアな方法を使用することで、クライアントシステムとリモートホストの両方に対するリスクが軽減されます。

Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux には、一般的な OpenSSH パッケージ openssh と、OpenSSH サーバー、openssh-server パッケージ、およびクライアント openssh-clients パッケージが含まれます。

14.1. SSH プロトコル

14.1.1. SSH を使用する理由

潜在的な侵入者は、ネットワークトラフィックの中断、傍受、経路変更を可能にする様々なツールを自由に駆使して、システムに侵入します。一般的には、これらの脅威は以下のとおり分類できます。

2 つのシステム間の通信の傍受

攻撃者は、ネットワーク上で通信を行う二者の間のどこかに潜み、両者間で渡される情報をコピーしている可能性があります。攻撃者は情報を傍受して保持する、または情報を改ざんして元の受信者に送信する場合があります。

通常、この攻撃は パケットスニファー を使用して実行されます。これは、ネットワークを通過する各パケットをキャプチャーしてその内容を分析する一般的なネットワークユーティリティーです。

特定のホストの偽装

攻撃者のシステムは、送信の対象となる受信者を装うように設定されます。このストラテジーが機能すると、ユーザーのシステムは間違ったホストと通信していることに気づかないままとなり

ます。

この攻撃は、DNS ポイズニング として知られる手法、または IP スプーフィング と呼ばれる手法を使用して実行できます。前者の場合、侵入者はクラックされた DNS サーバーを使用して、クライアントシステムを不当に複製されたホストへ指定します。後者の場合、侵入者は、信頼されたホストから送信されたように見せかけた偽装ネットワークパケットを送信します。

いずれの手法でも、潜在的な機密情報を傍受することが可能です。その傍受が悪意のある理由で行われる場合には、多大な損害をもたらしかねません。リモートシェルログインとファイルコピー用に SSH を使用すると、こうしたセキュリティーの脅威を大幅に軽減できます。これは、SSH クライアントとサーバーがデジタル署名を使用してそれぞれの ID を確認するためです。さらに、クライアントシステムとサーバーシステムとの間の通信はすべて暗号化されます。各パケットはローカルシステムとリモートシステムのみに知られている鍵を使用して暗号化されるため、通信のいずれか一方の ID をスプーフィングする試みは成功しません。

14.1.2. 主な特長

SSH プロトコルは、以下のような保護手段を提供します。

対象のサーバーになりすますことができない

クライアントは、初回接続後に、以前接続したサーバーと同じサーバーに接続していることを確認できます。

認証情報の取得ができない

クライアントは、強力な 128 ビット暗号化を使用して、サーバーへ認証情報を送信します。

通信の傍受ができない

セッション中に送受信された全データは、128 ビット暗号化を使用して転送されるため、傍受された送信データの暗号解読と読み取りは非常に困難になります。

さらに、以下のようなオプションも提供されます。

ネットワーク上でグラフィカルアプリケーションを使用するセキュアな手段を提供する

クライアントは、X11 転送 と呼ばれる技術を使用して、サーバーから X11 (X Window System)アプリケーションを転送できます。ForwardX11Trusted オプションを yes に設定した場合、または -Y オプションで SSH を使用する場合は、X11 SECURITY 拡張制御が省略され、セキュリティー上の脅威となる可能性があることに注意してください。

セキュアでないプロトコルをセキュアにする手段を提供する

SSH プロトコルは、送受信するものをすべて暗号化します。SSH サーバーは、ポート転送 と呼ばれる技術を使用して、POP などのセキュアではないプロトコルをセキュアにし、システムとデータ全体のセキュリティーを強化できます。

セキュアなチャンネルを作成する

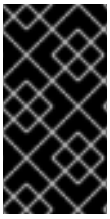
OpenSSH サーバーとクライアントは、サーバーマシンとクライアントマシンとの間のトラフィックに対して、仮想プライベートネットワークに似たトンネルを作成するように設定できます。

Kerberos 認証をサポートする

OpenSSH サーバーとクライアントは、Kerberos ネットワーク認証プロトコルの GSSAPI (Generic Security Services Application Program Interface)実装を使用して認証を行うように設定できます。

14.1.3. プロトコルのバージョン

現在、SSH にはバージョン 1 とバージョン 2 の 2 つの種類があります。Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux の OpenSSH スイートは、SSH バージョン 2 を使用します。これには、バージョン 1 の既知の不正使用の影響を受けない、強化された鍵交換アルゴリズムがあります。ただし、互換性の理由から、OpenSSH スイートはバージョン 1 の接続もサポートしますが、バージョン 1 はデフォルトで無効になっており、設定ファイルで有効にする必要があります。



重要

可能な限り、SSH バージョン 1 を使用せず、SSH バージョン 2 互換のサーバーとクライアントを使用してください。

14.1.4. SSH 接続のイベントシーケンス

以下に挙げる一連のイベントは、2 つのホスト間で行われる SSH 通信の整合性を保護するのに役立

ちます。

1. 暗号化ハンドシェイクが行われ、クライアントが正しいサーバーと通信していることを確認できます。

2. クライアントとリモートホストとの間の接続のトランスポート層が、対称暗号方式を使用して暗号化されます。

3. クライアントが、サーバーに対して自己認証します。

4. クライアントは、暗号化された接続でリモートホストと対話します。

14.1.4.1. トランスポート層

トランスポート層の主な役割は、認証時とその後の通信中に、2 つのホスト間の通信を簡単に安全でセキュアなものにすることです。トランスポート層は、データの暗号化と復号を処理し、データパケットの送受信時にその整合性を保護することでその役割を果たします。また、トランスポート層は、情報を圧縮して転送を高速にします。

SSH クライアントがサーバーに接続すると鍵情報が交換されるため、両システムでトランスポート層が適正に構築できます。以下は、こうした鍵情報の交換中に発生する手順です。

- 鍵を交換する
- 公開鍵暗号化アルゴリズムが決定する
- 対称暗号化アルゴリズムが決定する
- メッセージ認証アルゴリズムが決定する
- ハッシュアルゴリズムが決定する

鍵交換時に、サーバーは一意の ホストキー を使用してクライアントに対して自らを識別します。

クライアントがこの特定のサーバーと事前に通信しなかった場合、サーバーのホストキーはクライアントに認識されず、接続されません。OpenSSH は、ユーザーに、ホストの信頼性が確立されないことと、ユーザーがこれを受け入れるか拒否するかをユーザーに通知します。ユーザーは、新規ホストキーを受け入れる前に独立して検証する必要があります。その後の接続では、サーバーのホスト鍵がクライアントに保存されているバージョンと照合され、クライアントが実際に目的のサーバーと通信していることが信頼されます。その後、ホスト鍵が一致しなくなった場合は、接続前にクライアントに保存されたバージョンを削除する必要があります。

警告

新しい SSH サーバーの整合性を常に確認してください。攻撃者は初回コンタクト中に、認識されずにローカルシステムに目的の SSH サーバーを事前設定できます。新しい SSH サーバーの整合性を確認するには、最初の接続前、またはホスト鍵の不一致が発生した場合には、サーバー管理者に問い合わせてください。

SSH は、ほとんどすべての公開鍵アルゴリズムまたはエンコード形式に対応するように設計されています。初回の鍵交換で、交換に使用されるハッシュ値と共有秘密値が作成されると、2 つのシステムは新しい鍵とアルゴリズムの計算を直ちに開始して、認証と、今後の接続で送信されるデータを保護します。

所定の鍵とアルゴリズムを使用して一定量のデータ (正確な量は SSH 実装により異なる) が送信された後に、もう 1 回鍵交換が行われてハッシュ値と新しい共有秘密値の別のセットが生成されます。攻撃者がハッシュ値と共有秘密値を判別できたとしても、その情報が役に立つのは限られた時間のみです。

14.1.4.2. 認証

トランスポート層が、2 つのシステム間で情報を渡すためのセキュアなトンネルを構築すると、サーバーは、秘密鍵でエンコードされた署名の使用やパスワードの入力など、サポートされている別の認証方法をクライアントに伝えます。次に、クライアントが、対応しているいずれかの方法を使用して、サーバーに対して自己認証を試みます。

SSH サーバーとクライアントは、異なるタイプの認証を採用するように設定できるため、双方の制御が最適化されます。サーバーは、そのセキュリティーモデルに基づいて、対応する暗号化方法を決定できます。クライアントは、利用可能なオプションの中から、試行する認証方法の順番を選択できます。

14.1.4.3. チャネル

SSH トランスポート層での認証に成功すると、多重化と呼ばれる手法により複数のチャンネルが開きます。[4].これらの各チャンネルは、異なるターミナルセッションと、転送された X11 セッションの通信を処理します。

クライアントとサーバーの両方で、新しいチャンネルを作成できます。その後、各接続の両端に、別々の番号が割り当てられます。クライアントが新しいチャンネルを開こうとする際、要求と共にチャンネル番号を送信します。この情報はサーバーにより保存され、そのチャンネルに通信を移動するのに使用されます。これは、異なるタイプのセッションが相互に影響しないように、あるセッションの終了時にそのチャンネルが SSH による一次接続を停止せずに閉じることができるようにするためです。

また、チャンネルは フロー制御 もサポートしており、これにより、順序どおりにデータを送受信できます。この方法では、チャンネルが開いているというメッセージをクライアントが受信するまで、チャンネルでデータが送信されません。

クライアントが要求するサービスのタイプと、ユーザーがネットワークに接続される方法に応じて、クライアントとサーバーは、各チャンネルの特性を自動的にネゴシエートします。これにより、プロトコルの基本インフラストラクチャーを変更しなくても、異なるタイプのリモート接続を非常に柔軟に処理できます。

14.2. OPENSSH の設定

14.2.1. 設定ファイル

設定ファイルには、クライアントプログラム用（ssh、scp、および sftp）とサーバー用（ sshd デーモン）の異なる 2 つのセットがあります。

システム全体の SSH 設定情報は、表14.1「システム全体の設定ファイル」 の説明に従って /etc/ssh/ ディレクトリーに保存されます。ユーザー固有の SSH 設定情報は、表14.2「ユーザー固有の設定ファイル」 に記載されているように、ユーザーのホームディレクトリーの ~/.ssh/ に保存されます。

表14.1 システム全体の設定ファイル

ファイル	詳細
/etc/ssh/moduli	セキュアなトランスポート層を構築するのに非常に重要となる、Diffie-Hellman 鍵交換に使用される Diffie-Hellman グループが置かれています。SSH セッションの初めに鍵が交換される時、共有秘密値が作成されますが、どちらか一方の当事者だけでは決定できません。この値は、ホスト認証を行うのに使用されます。

ファイル	詳細
/etc/ssh/ssh_config	デフォルトの SSH クライアント設定ファイルです。~/.ssh/config が存在する場合は、これは上書きされることに注意してください。
/etc/ssh/sshd_config	sshd デーモンの設定ファイルです。
/etc/ssh/ssh_host_dsa_key	sshd デーモンが使用する DSA 秘密鍵です。
/etc/ssh/ssh_host_dsa_key.pub	sshd デーモンが使用する DSA 公開鍵です。
/etc/ssh/ssh_host_key	sshd デーモンが使用する SSH プロトコルのバージョン 1 用の RSA 秘密鍵です。
/etc/ssh/ssh_host_key.pub	sshd デーモンが使用する SSH プロトコルのバージョン 1 用の RSA 公開鍵です。
/etc/ssh/ssh_host_rsa_key	sshd デーモンが使用する SSH プロトコルのバージョン 2 用の RSA 秘密鍵です。
/etc/ssh/ssh_host_rsa_key.pub	sshd デーモンが使用する SSH プロトコルのバージョン 2 用の RSA 公開鍵です。
/etc/pam.d/sshd	sshd デーモンの PAM 設定ファイルです。
/etc/sysconfig/sshd	sshd サービスの設定ファイルです。

表14.2 ユーザー固有の設定ファイル

ファイル	詳細
~/.ssh/authorized_keys	サーバー用の認証済み公開鍵の一覧があります。クライアントがサーバーに接続すると、サーバーが、このファイル内に格納されている署名済み公開鍵を確認してクライアントを認証します。
~/.ssh/id_dsa	ユーザーの DSA 秘密鍵が含まれます。
~/.ssh/id_dsa.pub	ユーザーの DSA 公開鍵です。
~/.ssh/id_rsa	ssh が使用する SSH プロトコルのバージョン 2 用の RSA 秘密鍵です。

ファイル	詳細
~/.ssh/id_rsa.pub	ssh が使用する SSH プロトコルのバージョン 2 用の RSA 公開鍵です。
~/.ssh/identity	ssh が使用する SSH プロトコルのバージョン 1 用の RSA 秘密鍵です。
~/.ssh/identity.pub	ssh が使用する SSH プロトコルのバージョン 1 用の RSA 公開鍵です。
~/.ssh/known_hosts	ユーザーがアクセスする SSH サーバーの DSA ホストキーが含まれます。このファイルは、SSH クライアントが正しい SSH サーバーに接続することを確認するのに非常に重要になります。

SSH 設定ファイルに使用可能な各種ディレクティブの情報は、man ページの ssh_config(5)および sshd_config(5)を参照してください。

14.2.2. OpenSSH サーバーの起動

OpenSSH サーバーを実行するには、openssh-server がインストールされている必要があります（Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;Linux Red Hat Enterprise Linux 6nbsp;Linux 6nbsp;6 に新しいパッケージをインストールする方法については、「パッケージのインストール」 を参照してください）。

sshd デーモンを起動するには、シェルプロンプトで以下を入力します。

```
~]# service sshd start
```

実行中の sshd デーモンを停止するには、以下のコマンドを使用します。

```
~]# service sshd stop
```

システムの起動時にデーモンが自動的に起動するようにするには、以下を入力します。

```
~]# chkconfig sshd on
```

これにより、レベル 2、3、4、および 5 のサービスが有効になります。設定オプションの詳細

は、12章サービスおよびデーモン を参照してください。

システムを再インストールすると、新しい識別鍵のセットが作成される点に注意してください。したがって、再インストールの前にいずれかの OpenSSH ツールを使用してシステムに接続したことがあるクライアントには、以下のようなメッセージが表示されます。

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!     @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that the RSA host key has just been changed.
```

これを防ぐには、/etc/ssh/ ディレクトリーから関連ファイルをバックアップして（完全な一覧は 表14.1「システム全体の設定ファイル」 を参照）、システムを再インストールするたびに復元できます。

14.2.3. リモート接続に必要な SSH

SSH を本当の意味で有効なものにするためには、セキュリティー保護されていない接続プロトコルは使用しないことをお勧めします。それ以外の場合、ユーザーのパスワードは SSH を使用した 1 回のセッションで保護されても、その後に Telnet を使用してログインした時にのみキャプチャーされます。無効にするサービスには、telnet、rsh、rlogin、vsftpd などがあります。

これらのサービスを無効にするには、シェルプロンプトで以下のコマンドを入力します。

```
~]# chkconfig telnet off
~]# chkconfig rsh off
~]# chkconfig rlogin off
~]# chkconfig vsftpd off
```

ランレベルおよび一般的なサービスの設定に関する詳しい情報は、12章サービスおよびデーモン を参照してください。

14.2.4. キーベースの認証の使用

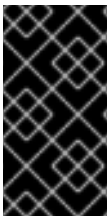
システムのセキュリティーをさらに強化するには、標準のパスワード認証を無効にして鍵ベースの認証を強制できます。これを行うには、vi、nano などのテキストエディターで /etc/ssh/sshd_config 設定ファイルを開き、PasswordAuthentication オプションを以下のように変更します。

```
PasswordAuthentication no
```

-

ssh、scp、または sftp を使用してクライアントマシンからサーバーに接続できるようにするには、以下の手順に従って認証鍵ペアを生成します。鍵はユーザーごとに別々に生成する必要がある点に注意してください。

Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;LinuxRed Hat Enterprise Linux 6nbsp;6 はデフォルトで SSH プロトコル 2 および RSA キーを使用します（詳細は 「プロトコルのバージョン」 を参照してください）。



重要

root として鍵ペアを生成しないでください。root のみがこれらの鍵を使用できるためです。

注記

システムを再インストールする前に、~/.ssh/ ディレクトリーをバックアップし、生成されたキーペアを保持します。root を含む、必要なユーザーの新しいシステムのホームディレクトリーに、バックアップデータをコピーします。

14.2.4.1. 鍵ペアの生成

以下の手順に従って、SSH プロトコルのバージョン 2 用の RSA 鍵ペアを生成します。

1. RSA 鍵ペアを生成するには、シェルプロンプトで次のコマンドを実行します。

```
~]$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/john/.ssh/id_rsa):
```

2. Enter キーを押して、新しく作成されたキーのデフォルトの場所(~/.ssh/id_rsa)を確認します。

3. パスフレーズを入力します。プロンプトが表示されたら再入力して確認します。セキュリティー上の理由により、アカウントのログイン時に使用するパスワードは使用しないでください。

この後、以下のようなメッセージが表示されます。

```
Your identification has been saved in /home/john/.ssh/id_rsa.
Your public key has been saved in /home/john/.ssh/id_rsa.pub.
The key fingerprint is:
e7:97:c7:e2:0e:f9:0e:fc:c4:d7:cb:e5:31:11:92:14 john@penguin.example.com
The key's randomart image is:
+--[ RSA 2048]----+
|            E.  |
|           . .  |
|            o . |
|             . .|
|        S .    . |
|         + o o ..|
|          * * +oo|
|           O +..=|
|           o*  o.|
+-----------------+
```

4. ~/.ssh/ ディレクトリーのパーミッションを変更します。

```
~]$ chmod 700 ~/.ssh
```

5. ~/.ssh/id_rsa.pub の内容を、接続するマシンの ~/.ssh/authorized_keys にコピーし、ファイルがすでに存在する場合はこれを最後に追加します。

6. 以下のコマンドを使用して、~/.ssh/authorized_keys ファイルのパーミッションを変更します。

```
~]$ chmod 600 ~/.ssh/authorized_keys
```

SSH プロトコルのバージョン 2 用の DSA キーペアを生成するには、次の手順を実行します。

1. シェルプロンプトで以下を入力して DSA 鍵ペアを生成します。

```
~]$ ssh-keygen -t dsa
Generating public/private dsa key pair.
Enter file in which to save the key (/home/john/.ssh/id_dsa):
```

2. Enter キーを押して、新しく作成された鍵のデフォルトの場所(~/.ssh/id_dsa)を確認します。

3.
パスフレーズを入力します。プロンプトが表示されたら再入力して確認します。セキュリティー上の理由により、アカウントのログイン時に使用するパスワードは使用しないでください。

この後、以下のようなメッセージが表示されます。

```
Your identification has been saved in /home/john/.ssh/id_dsa.
Your public key has been saved in /home/john/.ssh/id_dsa.pub.
The key fingerprint is:
81:a1:91:a8:9f:e8:c5:66:0d:54:f5:90:cc:bc:cc:27 john@penguin.example.com
The key's randomart image is:
+--[ DSA 1024]----+
|   .oo*o.        |
|  ...o Bo        |
| .. . + o.       |
|.  .   E o       |
| o..o   S        |
|. o= .           |
|. +              |
| .               |
|                 |
+-----------------+
```

4.
~/.ssh/ ディレクトリーのパーミッションを変更します。

```
~]$ chmod 700 ~/.ssh
```

5.
~/.ssh/id_dsa.pub の内容を、接続するマシンの ~/.ssh/authorized_keys にコピーし、ファイルがすでに存在する場合はこれを最後に追加します。

6.
以下のコマンドを使用して、~/.ssh/authorized_keys ファイルのパーミッションを変更します。

```
~]$ chmod 600 ~/.ssh/authorized_keys
```

SSH プロトコルのバージョン 1 用の RSA 鍵ペアを生成するには、以下の手順に従います。

1.
RSA 鍵ペアを生成するには、シェルプロンプトで次のコマンドを実行します。

-

```
~]$ ssh-keygen -t rsa1
Generating public/private rsa1 key pair.
Enter file in which to save the key (/home/john/.ssh/identity):
```

2. Enter を押して、新規作成された鍵のデフォルトの場所（つまり ~/.ssh/identity）を確認します。

3. パスフレーズを入力します。プロンプトが表示されたら再入力して確認します。セキュリティー上の理由から、アカウントへのログインに使用するパスワードと同じパスワードを使用しないでください。

この後、以下のようなメッセージが表示されます。

```
Your identification has been saved in /home/john/.ssh/identity.
Your public key has been saved in /home/john/.ssh/identity.pub.
The key fingerprint is:
cb:f6:d5:cb:6e:5f:2b:28:ac:17:0c:e4:62:e4:6f:59 john@penguin.example.com
The key's randomart image is:
+--[RSA1 2048]----+
|                 |
|    . .          |
|   o o           |
|    + o E        |
|   . o S         |
|      = +   .    |
|     . = . o . .|
|      . = o o..o|
|      .o o  o=o.|
+-----------------+
```

4. ~/.ssh/ ディレクトリーのパーミッションを変更します。

```
~]$ chmod 700 ~/.ssh
```

5. ~/.ssh/identity.pub の内容を、接続するマシンの ~/.ssh/authorized_keys にコピーし、ファイルがすでに存在する場合はこれを最後に追加します。

6. 以下のコマンドを使用して、~/.ssh/authorized_keys ファイルのパーミッションを変更します。

```
~]$ chmod 600 ~/.ssh/authorized_keys
```

システムにパスフレーズを記憶させる設定方法については 「ssh-agent の設定」 を参照してください。

重要

秘密鍵を任意のボディーと共有しないでください。これは個人使用のみを目的としています。

14.2.4.2. ssh-agent の設定

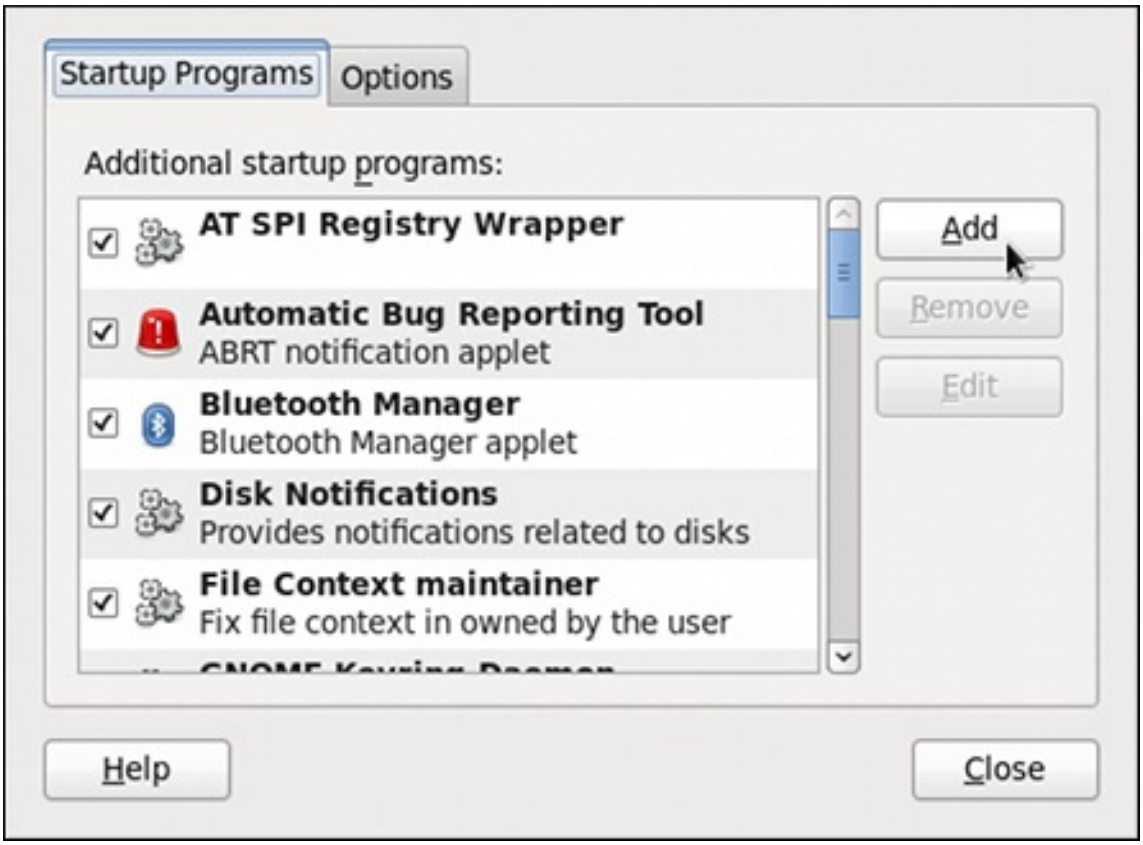
ssh-agent 認証エージェントを使用するとパスフレーズを保存することができるため、リモートマシンとの接続を開始する度にパスフレーズを入力する必要がなくなります。GNOME を実行している場合は、ログイン時には常にパスフレーズを求めるプロンプトを表示して、セッションを通してそのパスフレーズを記憶させておくように設定できます。それ以外の方法として、特定のシェルプロンプト用にパスフレーズを保存しておくことも可能です。

以下のステップに従って、GNOME セッション中にパスフレーズを保存します。

1. openssh-askpass パッケージがインストールされていることを確認します。Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux に新しいパッケージをインストールする方法の詳細は、 「パッケージのインストール」 を参照してください。

2. パネルから System → Preferences → Startup Applications を選択します。スタートアップアプリケーション設定 が起動し、利用可能な起動プログラムの一覧を含むタブがデフォルトで表示されます。

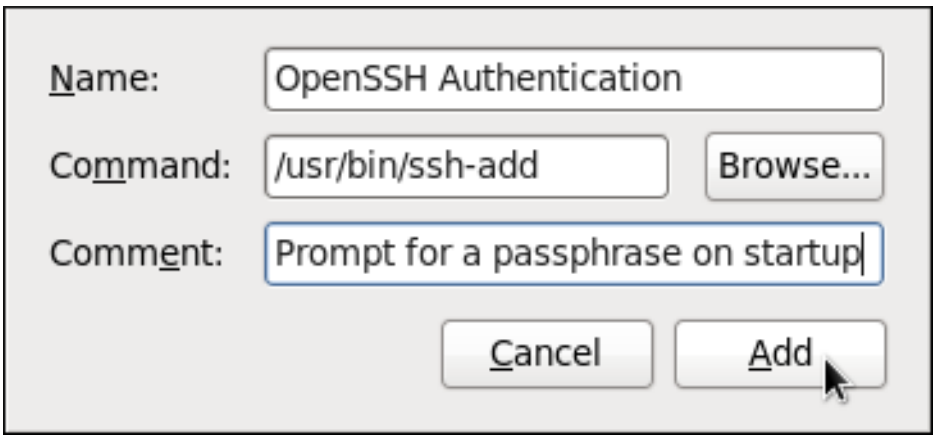
図14.1 自動起動するアプリの設定



[D]

3. *右側の 追加 ボタンをクリックして、コマンド フィールドに /usr/bin/ssh-add と入力します。*

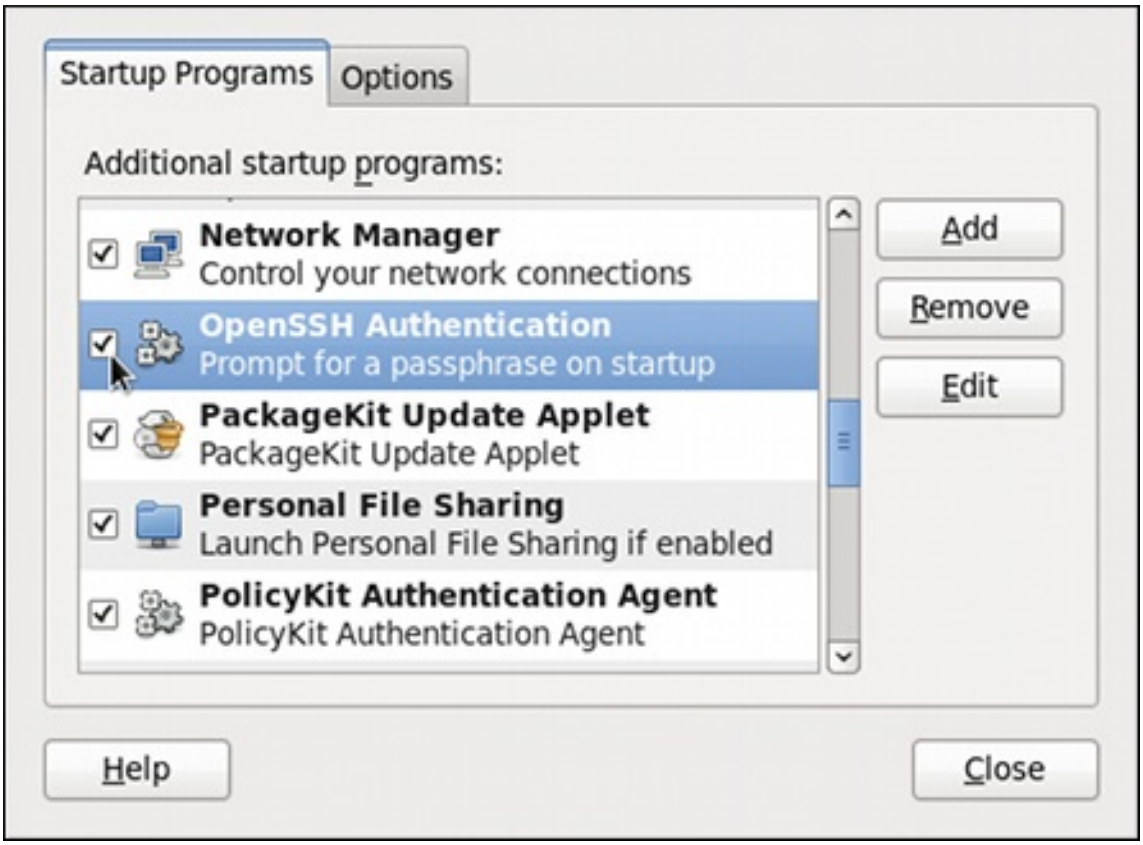
図14.2 新規アプリケーションの追加



[D]

4. *Add をクリックして、新たに追加した項目の横にあるチェックボックスが選択されていることを確認します。*

図14.3 アプリケーションの有効化

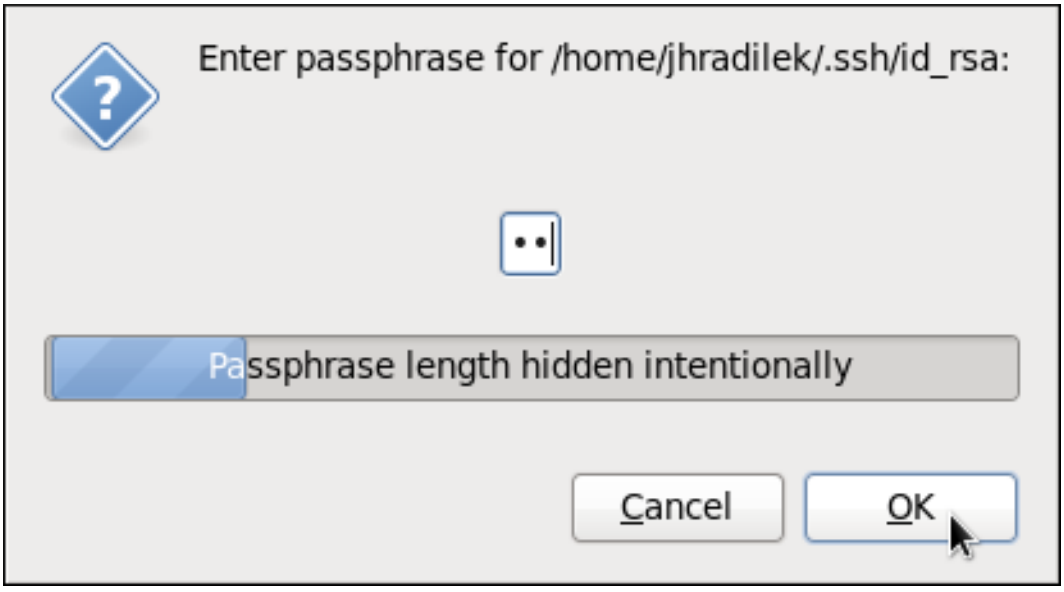


[D]

5.

一度ログアウトしてから再度ログインします。パスフレーズの入力を求めるダイアログボックスが表示されます。これ以降は、ssh、scp、または sftp によるパスワードの入力を要求されることはありません。

図14.4 パスフレーズの入力



[D]

特定のシェルプロンプト用のパスフレーズを保存するには、以下のコマンドを使用します。

```
~]$ ssh-add
Enter passphrase for /home/john/.ssh/id_rsa:
```

ログアウト時には、パスフレーズは記憶されない点に注意してください。仮想コンソールまたはターミナルウィンドウにログインする度にコマンドを実行する必要があります。

14.2.4.3. sshd に必要な認証方法が複数になる

セキュリティーを強化するために、SSH ではパスフレーズと公開鍵の両方など、正常にログインするための複数の認証方法が必要になる場合があります。/etc/ssh/sshd_config ファイルの RequiredAuthentications2 オプションを必要に応じて設定します。たとえば、以下を実行して行います。

```
~]# echo "RequiredAuthentications2 publickey,password" >> /etc/ssh/sshd_config
```

利用可能なオプションについての詳細は、man ページの sshd_config(5) を参照してください。

14.3. OPENSSH 証明書認証の使用

14.3.1. SSH 証明書の概要

認証に公開鍵暗号を使用するには、すべてのクライアントからクライアントがログインするすべてのサーバーに公開鍵をコピーする必要があります。このシステムは適切にスケーリングされず、管理者の負担になる可能性があります。認証 局(CA)から公開鍵を使用してクライアント証明 書を認証すると、複数のシステム間で鍵をコピーする必要がなくなります。X.509 公開鍵インフラストラクチャー証明書システムはこの問題に対する解決策を提供しますが、証明書署名を取得するために関連する企業を含む送受信プロセスがあります。別の方法として、OpenSSH は、単純な証明書および関連する CA インフラストラクチャーの作成に対応します。

OpenSSH 証明書には、公開鍵、ID 情報、および有効性制約が含まれます。これらは、ssh-keygen ユーティリティーを使用して標準の SSH 公開鍵で署名されます。証明書の形式は、/usr/share/doc/openssh-version/PROTOCOL.certkeys で説明されています。

ssh-keygen ユーティリティーは、ユーザーとホストの 2 種類の証明書に対応します。ユーザー証明書はサーバーにユーザーを認証しますが、ホスト証明書はサーバーホストをユーザーに対して認証します。ユーザーまたはホスト認証に使用する証明書については、sshd が CA 公開鍵を信頼するように設定する必要があります。

14.3.2. SSH 証明書のサポート

Red Hat Enterprise Linux 6.5 では、openssh-5.3p1-94.el6 パッケージで、新しい OpenSSH 証明書形式を使用したユーザーおよびホストの証明書認証のサポートが追加されました。必要に応じて、最新の OpenSSH パッケージがインストールされていることを確認するには、root で次のコマンドを実行します。

```
~]# yum install openssh
Package openssh-5.3p1-104.el6_6.1.i686 already installed and latest version
Nothing to do
```

14.3.3. SSH CA 証明書署名キーの作成

2 種類の証明書が必要です。ホスト証明書とユーザー証明書です。2 つの証明書を署名するために別の 2 つの鍵（ca_user_key および ca_host_key など）を使用することが適していますが、両方の証明書に署名するのに 1 つの CA キーのみを使用できます。また、別の鍵を使用する場合は手順を行う方が簡単なため、以下の例では別のキーを使用します。

ユーザー証明書を作成するためにユーザーの公開鍵に署名するコマンドの基本的なフォーマットは

```
ssh-keygen -s ca_user_key -I certificate_ID id_rsa.pub
```

です。-s は、証明書の署名に使用される秘密鍵を示します。-I は ID 文字列(certificate_ID)を示します。これは任意のアルファ数値値になります。これは、証明書にゼロ終端文字列として保存されます。certificate_ID は、証明書が識別に使用されるたびにログに記録され、証明書の取り消し時にも使用されます。長い値を設定すると、ログの読み取りが困難になります。そのため、ホスト証明書のホスト名とユーザー証明書のユーザー名を使用すると安全な選択肢になります。

ホストの公開鍵に署名してホスト証明書を作成するには、-h オプションを追加します。

```
ssh-keygen -s ca_host_key -I certificate_ID -h ssh_host_rsa_key.pub
```

ホストキーはデフォルトでシステムで生成され、キーを一覧表示するには、以下のようにコマンドを入力します。

```
~]# ls -l /etc/ssh/ssh_host*
-rw-------. 1 root root  668 Jul  9  2014 /etc/ssh/ssh_host_dsa_key
-rw-r--r--. 1 root root  590 Jul  9  2014 /etc/ssh/ssh_host_dsa_key.pub
-rw-------. 1 root root  963 Jul  9  2014 /etc/ssh/ssh_host_key
```

```
-rw-r--r--. 1 root root  627 Jul  9  2014 /etc/ssh/ssh_host_key.pub
-rw-------. 1 root root 1671 Jul  9  2014 /etc/ssh/ssh_host_rsa_key
-rw-r--r--. 1 root root  382 Jul  9  2014 /etc/ssh/ssh_host_rsa_key.pub
```

重要

CA 鍵を作成して、他の秘密鍵と同じように安全な場所に保存することが推奨されます。以下の例では、root ユーザーが使用されます。管理ユーザーアカウントを持つオフラインコンピューターを使用する実際の実稼働環境では、推奨しています。キーの長さに関するガイダンスは 『NIST Special Publication 800-131A』 を参照してください。

手順14.1 SSH CA 証明書署名鍵の生成

1. CA として指定されたサーバーで、署名証明書で使用する鍵を 2 つ生成します。これらは、他のすべてのホストが信頼する必要のあるキーです。ca_user_key および ca_host_key など、適切な名前を選択します。ユーザー証明書署名要求を生成するには、root で以下のコマンドを入力します。

```
~]# ssh-keygen -t rsa -f ~/.ssh/ca_user_key
Generating public/private rsa key pair.
Created directory '/root/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/ca_user_key.
Your public key has been saved in /root/.ssh/ca_user_key.pub.
The key fingerprint is:
11:14:2f:32:fd:5d:f5:e4:7a:5a:d6:b6:a0:62:c9:1f root@host_name.example.com
The key's randomart image is:
+--[ RSA 2048]----+
|       .+.      o|
|      . o     +.|
|     o + .   . o|
|      o + . . ..|
|       S . ... *|
|        . . . .*.|
|         = E  .. |
|        . o .    |
|           .     |
+-----------------+
```

以下のように、ホスト証明書署名要求(ca_host_key)を生成します。

```
~]# ssh-keygen -t rsa -f ~/.ssh/ca_host_key
Generating public/private rsa key pair.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/ca_host_key.
```

```
Your public key has been saved in /root/.ssh/ca_host_key.pub.
The key fingerprint is:
e4:d5:d1:4f:6b:fd:a2:e3:4e:5a:73:52:91:0b:b7:7a root@host_name.example.com
The key's randomart image is:
+--[ RSA 2048]----+
|           ..    |
|          . ....|
|       . . o +oo|
|      o .   o *o|
|       S     = .|
|            o. .|
|           *.E. |
|          +o=    |
|         .oo.    |
+-----------------+
```

必要に応じて、パーミッションが正しいことを確認します。

```
~]# ls -la ~/.ssh
total 40
drwxrwxrwx. 2 root root 4096 May 22 13:18 .
dr-xr-x---. 3 root root 4096 May  8 08:34 ..
-rw-------. 1 root root 1743 May 22 13:15 ca_host_key
-rw-r--r--. 1 root root  420 May 22 13:15 ca_host_key.pub
-rw-------. 1 root root 1743 May 22 13:14 ca_user_key
-rw-r--r--. 1 root root  420 May 22 13:14 ca_user_key.pub
-rw-r--r--. 1 root root  854 May  8 05:55 known_hosts
-r--------. 1 root root 1671 May  6 17:13 ssh_host_rsa
-rw-r--r--. 1 root root 1370 May  7 14:30 ssh_host_rsa-cert.pub
-rw-------. 1 root root  420 May  6 17:13 ssh_host_rsa.pub
```

2.

サーバーのホスト公開キーをホスト名、CA サーバーの完全修飾ドメイン名(FQDN)など、識別文字列と共に署名し、末尾の .. なしで CA サーバーの 完全修飾ドメイン名 (FQDN)を署名して、CA サーバー自体のホスト証明書を作成します。このコマンドは、

```
ssh-keygen -s ~/.ssh/ca_host_key -I certificate_ID -h -Z host_name.example.com -V -
start:+end /etc/ssh/ssh_host_rsa.pub
```

-Z オプションの形式を取ります。-Z オプションは、この証明書をドメイン内の特定のホストに制限します。-V オプションは有効期間を追加するため、強く推奨されます。有効期間は 1 年 5 週間にすることが意図されており、証明書を変更する時間と、証明書の有効期限が経過する期間などを考慮してください。

以下に例を示します。

```
~]# ssh-keygen -s ~/.ssh/ca_host_key -I host_name -h -Z host_name.example.com -V -
1w:+54w5d /etc/ssh/ssh_host_rsa.pub
```

```
Enter passphrase:
Signed host key /root/.ssh/ssh_host_rsa-cert.pub: id "host_name" serial 0 for
host_name.example.com valid from 2015-05-15T13:52:29 to 2016-06-08T13:52:29
```

14.3.4. SSH CA 公開鍵の配布と信頼

プロジェクトから証明書の認証されたログインを許可するホストは、ユーザーの証明書を認証するために、ユーザー証明書の署名に使用された CA の公開キーを信頼するように設定する必要があります。この例では、ca_user_key.pub です。

ca_user_key.pub 鍵を公開し、リモートユーザーがログインできるように必要なすべてのホストにダウンロードします。または、CA ユーザーの公開鍵をすべてのホストにコピーします。実稼働環境では、公開鍵をまず管理者アカウントにコピーすることを検討してください。secure copy コマンドを使用して、公開鍵をリモートホストにコピーすることができます。このコマンドは、

```
scp ~/.ssh/ca_user_key.pub root@host_name.example.com:/etc/ssh/
```

の形式を取ります。host_name は、ログインプロセス中に提示されるユーザーの証明書の認証に必要なサーバーのホスト名です。秘密鍵ではなく公開鍵をコピーしてください。たとえば、root で以下を実行します。

```
~]# scp ~/.ssh/ca_user_key.pub root@host_name.example.com:/etc/ssh/
The authenticity of host 'host_name.example.com (10.34.74.56)' can't be established.
RSA key fingerprint is fc:23:ad:ae:10:6f:d1:a1:67:ee:b1:d5:37:d4:b0:2f.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'host_name.example.com,10.34.74.56' (RSA) to the list of known
hosts.
root@host_name.example.com's password:
ca_user_key.pub                               100%  420     0.4KB/s   00:00
```

リモートユーザー認証の場合、CA キーは、cert-authority ディレクティブを使用するか、/etc/ssh/sshd_config ファイルの TrustedUserCAKeys ディレクティブを使用してグローバルに使用することで、~/.ssh/authorized_keys ファイルで信頼されるユーザー/authorized_keys としてマークできます。リモートホスト認証の場合、CA キーは、/etc/ssh/known_hosts ファイルまたは ~/.ssh/ssh_known_hosts ファイルのユーザーごとに信頼できるとマークできます。

手順14.2 ユーザー署名キーの信頼

- 1 つ以上の原則が記載されていて、設定がグローバルを有効にする場合は、/etc/ssh/sshd_config ファイルを以下のように編集します。

```
TrustedUserCAKeys /etc/ssh/ca_user_key.pub
```

sshd を再起動して変更を適用します。

```
~]# service sshd restart
```

不明なホストについての警告が表示されないようにするには、ユーザーのシステムが、ホスト証明書の署名に使用された CA の公開鍵を信頼する必要があります。この例では ca_host_key.pub です。

手順14.3 ホスト署名キーの信頼

1. ホスト証明書の署名に使用される公開鍵の内容を抽出します。たとえば、CA では以下のようになります。

```
cat ~/.ssh/ca_host_key.pub
ssh-rsa  AAAAB5Wm.== root@ca-server.example.com
```

2. サーバーの署名済みホスト証明書を信頼するようにクライアントシステムを設定するには、ca_host_key.pub の内容をグローバル known_hosts ファイルに追加します。これにより、新しいマシンが *.example.com ドメインで接続されるたびに、すべてのユーザーに対して、サーバーのホストが CA 公開鍵に対してアドバタイズされた証明書が自動的にチェックされます。root としてログインし、/etc/ssh/ssh_known_hosts ファイルを設定します。

```
~]# vi /etc/ssh/ssh_known_hosts
# A CA key, accepted for any host in *.example.com
@cert-authority *.example.com ssh-rsa AAAAB5Wm.
```

ssh-rsa AAAAB5Wm. は ca_host_key.pub の内容です。上記は、システムが CA サーバーのホスト公開鍵を信頼するように設定します。これにより、ホストが提示する証明書のグローバル認証ができるようになります。

14.3.5. SSH 証明書の作成

証明書とは、署名された公開鍵です。ユーザーおよびホストの公開鍵を CA サーバーの秘密鍵で署名するために CA サーバーにコピーする必要があります。

重要

多くのキーを CA にコピーして署名すると、一意に名前が付けられていない場合は混乱が生じる可能性があります。デフォルト名が常に使用されると、コピーされる最新のキーによって、以前のコピーされたキーが上書きされます。これは、1 つの管理者に許容可能なメソッドである可能性があります。以下の例では、デフォルト名が使用されます。実稼働環境では、簡単に認識できる名前を使用することを検討してください。
キーのコピー先となる管理ユーザーが所有する CA サーバーに、指定されたディレクトリーを設定することが推奨されます。これらのキーを root ユーザーの /etc/ssh/ ディレクトリーにコピーすることは推奨していません。以下の例では、key / という名前のディレクトリーを持つ admin という名前 のアカウントが使用されます。

管理者アカウント（この例では admin ）と、ユーザーのキーを受け取るディレクトリーを作成します。以下に例を示します。

```
~]$ mkdir keys
```

鍵をコピーできるようにパーミッションを設定します。

```
~]$ chmod o+w keys
ls -la keys
total 8
drwxrwxrwx. 2 admin admin 4096 May 22 16:17 .
drwx------. 3 admin admin 4096 May 22 16:17 ..
```

14.3.5.1. ホストを認証する SSH 証明書の作成

ホスト証明書に署名するコマンドの形式は

```
ssh-keygen -s ca_host_key -I host_name -h ssh_host_rsa_key.pub
```

になります。ホスト証明書の名前は ssh_host_rsa_key-cert.pub です。

手順14.4 ホスト証明書の生成

ユーザーへのホストを認証するには、ホストに公開鍵を生成し、CA で署名した CA サーバーに渡して、ホストへのログインを試行するユーザーに存在するようにホストに保存するために再度渡す必要があります。

1. *ホスト鍵は、システムで自動的に生成されます。一覧表示するには、以下のコマンドを入力します。*

```
~]# ls -l /etc/ssh/ssh_host*
-rw-------. 1 root root  668 May  6 14:38 /etc/ssh/ssh_host_dsa_key
-rw-r--r--. 1 root root  590 May  6 14:38 /etc/ssh/ssh_host_dsa_key.pub
-rw-------. 1 root root  963 May  6 14:38 /etc/ssh/ssh_host_key
-rw-r--r--. 1 root root  627 May  6 14:38 /etc/ssh/ssh_host_key.pub
-rw-------. 1 root root 1679 May  6 14:38 /etc/ssh/ssh_host_rsa_key
-rw-r--r--. 1 root root  382 May  6 14:38 /etc/ssh/ssh_host_rsa_key.pub
```

2.

選択した公開鍵を CA として指定されたサーバーにコピーします。たとえば、ホストからは以下のようになります。

```
~]# scp /etc/ssh/ssh_host_rsa_key.pub admin@ca-
server.example.com:~/keys/ssh_host_rsa_key.pub
The authenticity of host 'ca-server.example.com (10.34.74.58)' can't be established.
RSA key fingerprint is b0:e5:ea:b8:75:e2:f0:b1:fe:5b:07:39:7f:58:64:d9.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'ca-server.example.com,10.34.74.58' (RSA) to the list of
known hosts.
admin@ca-server.example.com's password:
ssh_host_rsa_key.pub                          100%  382     0.4KB/s   00:00
```

または、CA から以下を行います。

```
~]$ scp root@host_name.example.com:/etc/ssh/ssh_host_rsa_key.pub
~/keys/ssh_host_rsa_key.pub
```

3.

CA サーバーで、ホストの公開鍵に署名します。たとえば、root で以下を実行します。

```
~]# ssh-keygen -s ~/.ssh/ca_host_key -I host_name -h -Z host_name.example.com -V -
1d:+54w /home/admin/keys/ssh_host_rsa_key.pub
Enter passphrase:
Signed host key /home/admin/keys/ssh_host_rsa_key-cert.pub: id "host_name" serial
0 for host_name.example.com valid from 2015-05-26T12:21:54 to 2016-06-08T12:21:54
```

host_name は、証明書を必要とするシステムのホスト名です。

4.

証明書をホストにコピーします。CA の場合の例を以下に示します。

```
~]# scp /home/admin/keys/ssh_host_rsa_key-cert.pub
root@host_name.example.com:/etc/ssh/
root@host_name.example.com's password:
ssh_host_rsa_key-cert.pub                     100% 1384     1.5KB/s   00:00
```

5. ユーザーがログインプロセスの開始時に証明書をユーザーのシステムに提示するようにホストを設定します。root で、/etc/ssh/sshd_config ファイルを以下のように編集します。

```
HostCertificate /etc/ssh/ssh_host_rsa_key-cert.pub
```

6. sshd を再起動して、変更を有効にします。

```
~]# service sshd restart
```

7. ユーザーのシステムで、ユーザーが以前に設定したホストにログインしている場合は、~/.ssh/known_hosts ファイルからホストに属する鍵を削除します。ユーザーがホストにログインすると、ホストの信頼性についての警告が表示されなくなります。

クライアントシステムでホスト証明書をテストするには、手順14.3「ホスト署名キーの信頼」 の説明に従って、クライアントがグローバル /etc/ssh/known_hosts ファイルを設定しており、サーバーの公開鍵が ~/.ssh/known_hosts ファイルにないことを確認します。次に、リモートユーザーとして SSH 経由でサーバーへのログインを試みます。ホストの信頼性についての警告は表示されません。必要な場合は SSH コマンドに -v オプションを追加して、ロギング情報を確認します。

14.3.5.2. ユーザーの認証用の SSH 証明書の作成

ユーザーの証明書に署名するには、

```
ssh-keygen -s ca_user_key -I user_name -Z user_name -V -start:+end id_rsa.pub
```

の形式でコマンドを使用します。生成される証明書の名前は id_rsa-cert.pub です。

OpenSSH のデフォルト動作は、証明書で指定されたプリンシパルのいずれかがリモートユーザー名と一致する場合に、ユーザーがリモートユーザーとしてログインできることです。これは以下の方法

で調整できます。

- -Z オプションを使用して、署名プロセス中にユーザーの名前をさらに証明書に追加します。

```
-Z "name1[,name2,...]"
```

- ユーザーのシステムで、cert-authority ディレクティブを使用して ~/.ssh/authorized_keys ファイルに CA の公開鍵を追加し、以下のようにプリンシパル名を一覧表示します。

```
~]# vi ~/.ssh/authorized_keys
# A CA key, accepted for any host in *.example.com
@cert-authority principals="name1,name2" *.example.com ssh-rsa AAAAB5Wm.
```

- サーバーで、ユーザーまたはグローバルのいずれかで AuthorizedPrincipalsFile ファイルを作成し、ログインを許可されたユーザーのファイルに原則の名前を追加します。次に、/etc/ssh/sshd_config ファイルで AuthorizedPrincipalsFile ディレクティブを使用してファイルを指定します。

手順14.5 ユーザー証明書の生成

ユーザーをリモートホストに対して認証するには、ユーザーが公開鍵を生成し、CA サーバーに渡して CA サーバーが署名してから、ホストへのログイン時に使用するユーザーにより保存し直す必要があります。

1. クライアントシステムで、証明書を必要とするユーザーとしてログインします。利用可能なキーが以下のようにあるかどうかを確認します。

```
~]$ ls -l ~/.ssh/
```

適切な公開鍵が存在しない場合は、ディレクトリーがデフォルトのディレクトリーでなければ、それを生成してディレクトリーパーミッションを設定します。たとえば、以下のコマンドを入力します。

```
~]$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/user1/.ssh/id_rsa):
Created directory '/home/user1/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/user1/.ssh/id_rsa.
Your public key has been saved in /home/user1/.ssh/id_rsa.pub.
The key fingerprint is:
b1:f8:26:a7:46:87:c3:60:54:a3:6d:85:0d:60:fe:ce user1@host1.example.com
The key's randomart image is:
+--[ RSA 2048]----+
|    oo++.        |
|   o.o.o.        |
|  .o o .         |
|    oo . o       |
|   . oo.S        |
|     o=..        |
|    .Eo+         |
|     .=          |
|    ..           |
+-----------------+
```

デフォルトでは、ユーザーの鍵のディレクトリーパーミッションは drwx------ で、または 8 進数の 0700 です。必要に応じて、パーミッションが正しいことを確認します。

```
~]$ ls -la ~/.ssh
total 16
drwx------. 2 user1 user1 4096 May  7 12:37 .
drwx------. 3 user1 user1 4096 May  7 12:37 ..
-rw-------. 1 user1 user1 1679 May  7 12:37 id_rsa
-rw-r--r--. 1 user1 user1  421 May  7 12:37 id_rsa.pub
```

キー生成の例と、正しいディレクトリー権限の設定方法については、「キーベースの認証の使用」 を参照してください。

2.

選択した公開鍵を署名するには、CA として指定されたサーバーにコピーする必要があります。そのためには、secure copy コマンドを使用することができます。

```
scp ~/.ssh/id_protocol.pub admin@ca_server.example.com:~/keys/
```

ここで protocol はファイル名の一部で、キーの生成に使用するプロトコル（ rsa など）、admin は CA サーバーのアカウントであり、/keys/ は署名するキーを受信するディレクトリーの設定です。

選択した公開鍵を CA として指定されたサーバーにコピーします。以下に例を示します。

```
~]$ scp ~/.ssh/id_rsa.pub admin@ca-server.example.com:~/keys/
admin@ca-server.example.com's password:
id_rsa.pub                                    100%  421     0.4KB/s   00:00
```

手順14.3「ホスト署名キーの信頼」 の説明に従ってホスト署名キーを信頼するようにクライアントシステムを設定した場合は、リモートホストの信頼性についての警告は表示されません。

3. CA サーバーで、ユーザーの公開鍵に署名します。たとえば、root で以下を実行します。

```
~]# ssh-keygen -s ~/.ssh/ca_user_key -I user1 -Z user1 -V -1d:+54w
/home/admin/keys/id_rsa.pub
Enter passphrase:
Signed user key /home/admin/keys/id_rsa-cert.pub: id "user1" serial 0 for
host_name.example.com valid from 2015-05-21T16:43:17 to 2016-06-03T16:43:17
```

4. 作成された証明書を、システムのユーザーの ~/.ssh/ ディレクトリーにコピーします。以下に例を示します。

```
~]# scp /home/admin/keys/id_rsa-cert.pub user1@host_name.example.com:~/.ssh/
user1@host_name.example.com's password:
id_rsa-cert.pub                               100%  1498    1.5KB/s   00:00
```

5. 標準のファイル名と場所を使用する場合は、SSH デーモンは -cert.pub で終わるユーザー証明書を検索し、見つかった場合は自動的に使用するため、追加の設定は必要ありません。SSH バージョン 2 キーのデフォルトの場所とファイル名は ~/.ssh/id_dsa、~/.ssh/id_ecdsa および ~/.ssh/id_rsa の man ページで説明されています。これらの場所と命名規則を使用する場合は、sshd が証明書を提示できるように設定ファイルを編集する必要はありません。リモートシステムにログインする際に自動的に使用されます。この場合、ステップ 6 に進みます。

デフォルト以外のディレクトリーまたはファイルの命名規則を使用する必要がある場合は、root で、以下の行を /etc/ssh/ssh_config または ~/.ssh/config ファイルに追加します。

```
IdentityFile ~/path/key_file
```

これは秘密鍵名でなければならず、.pub や -cert .pub は使用しないことに注意してください。ファイルのパーミッションが正しいことを確認します。以下に例を示します。

```
~]$ ls -la ~/.ssh/config
```

```
-rw-rw-r--. 1 user1 user1 36 May 27 21:49 /home/user1/.ssh/config
chmod 700 ~/.ssh/config
~]$ ls -la ~/.ssh/config
-rwx------. 1 user1 user1 36 May 27 21:49 /home/user1/.ssh/config
```

これにより、CA ユーザーの証明書署名鍵を信頼するように設定されたリモートシステムにログインする際に、このシステムのユーザーがユーザー証明書で認証できるようになります。

6.

ユーザー証明書をテストするには、ユーザーのアカウントから SSH 経由でサーバーへのログインを試みます。指定した場合は、証明書のプリンシパルとしてリストされているユーザーとしてこれを実行する必要があります。パスワードの入力を求めるプロンプトはないはずです。必要な場合は SSH コマンドに -v オプションを追加して、ロギング情報を確認します。

14.3.6. PKCS#11 トークンを使用した SSH 証明書の署名

-D を使用してトークンライブラリーを提供し、-s オプションに公開半分を引数として指定することで、PKCS#11 トークンに保存されている CA キーを使用してホストキーに署名できます：

```
ssh-keygen -s ca_host_key.pub -D libpkcs11.so -I certificate_ID host_key.pub
```

いずれの場合も、certificate_ID は、証明書が認証に使用される際にサーバーによってログに記録される 「鍵識別子」 です。

証明書は、ユーザーまたはホスト名（プリンシパル）のセットでのみ有効になるように設定できます。デフォルトでは、生成された証明書はすべてのユーザーまたはホストに対して有効です。指定のプリンシパルセットの証明書を生成するには、以下のように -Z オプションを指定してコンマ区切りリストを使用します。

```
ssh-keygen -s ca_user_key.pub -D libpkcs11.so -I certificate_ID -Z user1,user2 id_rsa.pub
```

ホストの場合：

```
ssh-keygen -s ca_host_key.pub -D libpkcs11.so -I certificate_ID -h -Z host.domain
ssh_host_rsa_key.pub
```

有効性およびユーザー証明書の使用に関する追加の制限は、証明書オプションで指定できます。証明書オプションは、特定のソースアドレスから提示された場合にのみ SSH セッションの機能を無効にするか、特定のコマンドの使用を強制できます。有効な証明書オプションの一覧は、-O オプションの ssh-keygen(1) man ページを参照してください。

証明書は、特定の有効期間に対して有効になるように定義できます。-V オプションを使用すると、証明書の開始時間と終了時間を指定できます。例：

```
ssh-keygen -s ca_user_key -I certificate_ID id_rsa.pub -V "-1w:+54w5d"
```

この範囲外の時点で提示される証明書は有効とみなされません。デフォルトでは、証明書は UNIX Epoch から無期限に有効になります。

14.3.7. SSH CA 証明書の表示

証明書を表示するには、-L を使用してコンテンツを一覧表示します。たとえば、ユーザーの証明書の場合は、以下のようになります。

```
~]$ ssh-keygen -L -f ~/.ssh/id_rsa-cert.pub
/home/user1/.ssh/id_rsa-cert.pub:
        Type: ssh-rsa-cert-v01@openssh.com user certificate
        Public key: RSA-CERT 3c:9d:42:ed:65:b6:0f:18:bf:52:77:c6:02:0e:e5:86
        Signing CA: RSA b1:8e:0b:ce:fe:1b:67:59:f1:74:cd:32:af:5f:c6:e8
        Key ID: "user1"
        Serial: 0
        Valid: from 2015-05-27T00:09:16 to 2016-06-09T00:09:16
        Principals:
                user1
        Critical Options: (none)
        Extensions:
                permit-X11-forwarding
                permit-agent-forwarding
                permit-port-forwarding
                permit-pty
                permit-user-rc
```

ホスト証明書を表示するには、以下を実行します。

```
~]# ssh-keygen -L -f /etc/ssh/ssh_host_rsa_key-cert.pub
/etc/ssh/ssh_host_rsa_key-cert.pub:
        Type: ssh-rsa-cert-v01@openssh.com host certificate
        Public key: RSA-CERT 1d:71:61:50:05:9b:ec:64:34:27:a5:cc:67:24:03:23
        Signing CA: RSA e4:d5:d1:4f:6b:fd:a2:e3:4e:5a:73:52:91:0b:b7:7a
        Key ID: "host_name"
        Serial: 0
        Valid: from 2015-05-26T17:19:01 to 2016-06-08T17:19:01
        Principals:
                host_name.example.com
        Critical Options: (none)
        Extensions: (none)
```

14.3.8. SSH CA 証明書の取り消し

証明書が盗まれる場合、取り消す必要があります。OpenSSH は失効リストを配布するメカニズムを提供していませんが、依然として失効リストを作成し、他の方法で配布して、事前に作成して配布されるすべてのホストおよびユーザー証明書を変更できます。

鍵を取り消すには、それらを revoked_keys ファイルに追加し、以下のように sshd_config ファイルにファイル名を指定します。

```
RevokedKeys /etc/ssh/revoked_keys
```

: このファイルが読み取り可能でない場合は、全ユーザーに対して公開鍵認証が拒否されることに注意してください。

キーが取り消されているかどうかをテストするには、失効リストにキーが存在するようクエリーします。以下のようなコマンドを使用します。

```
ssh-keygen -Qf /etc/ssh/revoked_keys ~/.ssh/id_rsa.pub
```

ユーザーは、cert-authority ディレクティブを変更して known_hosts ファイルで 取り消しするように CA 証明書を取り消す ことができます。

14.4. OPENSSH クライアント

クライアントマシンから OpenSSH サーバーに接続するには、openssh-clients および openssh パッケージがインストールされている必要があります（Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux に新しいパッケージをインストールする方法については 「パッケージのインストール」 を参照してください）。

14.4.1. ssh ユーティリティーの使用

ssh ユーティリティーを使用すると、リモートマシンにログインしてそのマシン上でコマンドを実行することができます。これは、rlogin、rsh、および telnet プログラムに代わるセキュアな手段です。

telnet コマンドと同様に、以下のコマンドを使用してリモートマシンにログインします。

```
ssh hostname
```

たとえば、penguin.example.com という名前のリモートマシンにログインするには、シェルプロンプトで以下を入力します。

```
~]$ ssh penguin.example.com
```

これで、ローカルマシンで使用しているユーザー名でログインします。別のユーザー名を指定する場合には、以下の形式のコマンドを使用してください。

```
ssh username@hostname
```

たとえば、john として penguin.example.com にログインするには、以下を入力します。

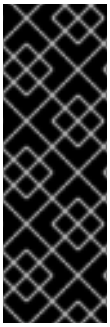
```
~]$ ssh john@penguin.example.com
```

初回接続時には、以下のようなメッセージが表示されます。

```
The authenticity of host 'penguin.example.com' can't be established.
RSA key fingerprint is 94:68:3a:3a:bc:f3:9a:9b:01:5d:b3:07:38:e2:11:0c.
Are you sure you want to continue connecting (yes/no)?
```

yes を入力して確定します。サーバーが既知ホストの一覧に追加されたことを知らせるメッセージと、パスワードの入力を求めるプロンプトが以下のように表示されます。

```
Warning: Permanently added 'penguin.example.com' (RSA) to the list of known hosts.
john@penguin.example.com's password:
```



重要

キーが変更された場合は、SSH サーバーのホストキーを更新します。クライアントは、サーバーのホストキーが ~/.ssh/known_hosts ファイルから削除されるまで接続を開始できないことをユーザーに通知します。SSH サーバーのシステム管理者に連絡して、サーバーが危険にさらされていないことを確認してから、最初にリモートマシンの名前が含まれる行を削除します。

パスワードを入力すると、リモートマシン用のシェルプロンプトが表示されます。

別の方法として、シェルプロンプトにログインせずに、ssh プログラムを使用してリモートマシン上でコマンドを実行することもできます。

```
ssh [username@]hostname command
```

たとえば、/etc/redhat-release ファイルは、Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux バージョンに関する情報を提供します。penguin.example.com でこのファイルの内容を表示するには、以下を入力します。

```
~]$ ssh john@penguin.example.com cat /etc/redhat-release
john@penguin.example.com's password:
Red Hat Enterprise Linux Server release 6.2 (Santiago)
```

正しいパスワードを入力すると、ユーザー名が表示され、ローカルのシェルプロンプトに戻ります。

14.4.2. scp ユーティリティーの使用

scp を使用すると、暗号化されたセキュアな接続でマシン間でファイルを転送できます。設計に関しては、rcp と非常に似ています。

ローカルファイルをリモートシステムへ転送するには、以下の形式でコマンドを使用します。

```
scp localfile username@hostname:remotefile
```

たとえば、taglist.vim を penguin.example.com という名前のリモートマシンに転送する場合は、シェルプロンプトで以下を入力します。

```
~]$ scp taglist.vim john@penguin.example.com:.vim/plugin/taglist.vim
john@penguin.example.com's password:
taglist.vim                                   100%  144KB 144.5KB/s   00:00
```

一度に複数のファイルを指定することも可能です。.vim/plugin/ の内容を、リモートマシン penguin.example.com の同じディレクトリーに転送するには、以下のコマンドを入力します。

```
~]$ scp .vim/plugin/* john@penguin.example.com:.vim/plugin/
```

```
john@penguin.example.com's password:
closetag.vim                                  100%   13KB  12.6KB/s   00:00
snippetsEmu.vim                               100%   33KB  33.1KB/s   00:00
taglist.vim                                   100%  144KB 144.5KB/s   00:00
```

リモートファイルをローカルシステムへ転送するには、以下の構文を使用します。

```
scp username@hostname:remotefile localfile
```

たとえば、.vimrc 設定ファイルをリモートマシンからダウンロードするには、以下を入力します。

```
~]$ scp john@penguin.example.com:.vimrc .vimrc
john@penguin.example.com's password:
.vimrc                                        100% 2233     2.2KB/s   00:00
```

14.4.3. sftp ユーティリティーの使用

sftp ユーティリティーを使用すると、セキュアでインタラクティブな FTP セッションを開くことができます。その設計では、ftp と似ていますが、暗号化された接続を使用します。

リモートシステムに接続するには、以下の形式でコマンドを使用します。

```
sftp username@hostname
```

たとえば、ユーザー名として john で penguin.example.com という名前のリモートマシンにログインするには、以下を入力します。

```
~]$ sftp john@penguin.example.com
john@penguin.example.com's password:
Connected to penguin.example.com.
sftp>
```

正しいパスワードを入力すると、プロンプトが表示されます。sftp ユーティリティーは、ftp で使用されるコマンドセットと同様のものを使用します（表14.3「利用可能な sftp コマンドの抜粋」を参照）。

表14.3 利用可能な sftp コマンドの抜粋

コマンド	詳細
ls [directory]	リモート ディレクトリー の内容を一覧表示します。指定がない場合は、デフォルトで現在の作業ディレクトリーが使用されます。
cd directory	リモートの作業ディレクトリーを ディレクトリー に変更します。
mkdir directory	リモート ディレクトリー を作成します。
rmdir path	リモート ディレクトリー を削除します。
put localfile [remotefile]	localfile をリモートマシンに転送します。
get remotefile [localfile]	remotefile をリモートマシンから転送します。

利用可能なコマンドの一覧は、sftp(1)の man ページを参照してください。

14.5. セキュアなシェルの追加

セキュアなコマンドラインインターフェースは、数多くある SSH の用途の中でも初歩的なものに過ぎません。十分な帯域幅があれば、X11 セッションは SSH チャンネル上で送信できます。あるいは、TCP/IP 転送を使用することで、以前はセキュリティー保護されていなかったシステム間のポート接続を特定の SSH チャンネルにマッピングすることができます。

14.5.1. X11 転送

SSH 接続上で X11 セッションを開始するには、以下の形式でコマンドを使用します。

```
ssh -Y username@hostname
```

たとえば、ユーザー名として john で penguin.example.com という名前のリモートマシンにログインするには、以下を入力します。

```
~]$ ssh -Y john@penguin.example.com
john@penguin.example.com's password:
```

セキュアなシェルプロンプトから X プログラムを実行すると、SSH クライアントとサーバーは新しいセキュアなチャンネルを作成し、X プログラムデータはそのチャンネル上で透過的にクライアントマ

シンに送信されます。

X11 転送は非常に便利なものです。たとえば、X11 転送を使用して、Printer Configuration ユーティリティーのセキュアかつインタラクティブなセッションを作成できます。これを行うには、ssh を使用してサーバーに接続し、以下のコマンドを入力します。

```
~]$ system-config-printer &
```

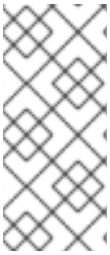
プリンター設定ツールが 表示され、リモートユーザーがリモートシステムで安全に印刷を設定できます。

X11 転送は、信頼できる転送と信頼できない転送間で区別されないことに注意してください。

14.5.2. ポート転送

SSH は、ポート転送によりセキュリティー保護されていない TCP/IP プロトコルをセキュアにすることができます。この手法を使用する場合、SSH サーバーは SSH クライアントをつなぐ暗号化された経路となります。

ポート転送は、クライアント上のローカルポートをサーバー上のリモートポートにマッピングすることで機能します。SSH ではサーバーの任意のポートをクライアント上の任意のポートにマッピングすることが可能です。このテクニックが機能するためにポート番号が一致する必要はありません。



注記

予約済みのポート番号を使用する場合には、1024 未満のポートをリッスンするようにポート転送を設定する場合は、root レベルのアクセスが必要であることに注意してください。

localhost で接続をリッスンする TCP/IP ポート転送チャンネルを作成するには、以下の形式でコマンドを使用します。

```
ssh -L local-port:remote-hostname:remote-port username@hostname
```

たとえば、暗号化された接続で POP3 を使用して、mail.example.com と呼ばれるサーバーでメールを確認するには、以下のコマンドを使用します。

```
~]$ ssh -L 1100:mail.example.com:110 mail.example.com
```

ポート転送チャンネルがクライアントマシンとメールサーバー間に配置されたら、POP3 メールクライアントに対し localhost 上のポート 1100 を使用して、新しいメールを確認するように指示します。クライアントシステムのポート 1100 に送信されたリクエストは、安全に mail.example.com サーバーに転送されます。

mail.example.com が SSH サーバーを実行しておらず、同じネットワーク上にある別のマシンの場合でも、SSH を使用して接続の一部をセキュアにすることができます。ただし、若干異なるコマンドが必要になります。

```
~]$ ssh -L 1100:mail.example.com:110 other.example.com
```

この例では、クライアントマシンのポート 1100 からの POP3 要求がポート 22 の SSH 接続を介して SSH サーバー other.example.com に転送されます。次に、other.example.com が mail.example.com のポート 110 に接続し、新規メールを確認します。この手法を使用する場合、クライアントシステムと other.example.com SSH サーバー間の接続のみがセキュアである点に注意してください。

ポート転送は、ネットワークのファイアウォール経由でセキュアに情報を取得する場合にも使用できます。ファイアウォールが標準ポート (ポート 22) 経由の SSH トラフィックを許可するよう設定されているものの、他のポートへのアクセスはブロックする場合でも、確立された SSH 接続にそのような通信をリダイレクトすることにより、ブロックされたポートを使用した 2 つのホスト間の接続は可能になります。

重要

この方法で接続を転送すると、クライアントシステムの任意のユーザーがそのサービスに接続できるため、接続のセキュリティーはクライアントシステムとしてのみ保護されます。クライアントシステムが危険にさらされると、攻撃者は転送されたサービスにアクセスすることもできます。

必要に応じて、/etc/ssh/sshd_config ファイルの AllowTcpForwarding 行に No パラメーターを指定して sshd サービスを再起動し、この機能を無効にします。

14.6. その他のリソース

OpenSSH および OpenSSL の詳細は、以下に記載のリソースを参照してください。

14.6.1. インストールされているドキュメント

- *sshd(8): sshd デーモンの man ページです。*
- *ssh(1): ssh クライアントの man ページです。*
- *scp(1)- scp ユーティリティーの man ページです。*
- *sftp(1)- sftp ユーティリティーの man ページです。*
- *ssh-keygen(1)- ssh-keygen ユーティリティーの man ページです。*
- *ssh_config(5): 利用可能な SSH クライアント設定オプションの詳細が記載された man ページです。*
- *sshd_config(5): man ページで、利用可能な SSH デーモン設定オプションが説明されています。*
- */usr/share/doc/openssh-バージョン: OpenSSH が対応するプロトコルの詳細情報が含まれます。*

14.6.2. 便利な Web サイト

http://www.openssh.com/

その他のドキュメント、よくある質問、メーリングリストへのリンク、バグレポートなどの役立つリソースを含む OpenSSH のホームページです。

http://www.openssl.org/

その他のドキュメント、よくある質問、メーリングリストへのリンクなどの役立つリソースを掲載した OpenSSL のホームページです。

---

[4]
多重接続は、共有されている共通のメディアで送信されるいくつかのシグナルで構成されます。

SSH により、異なるチャンネルが共通のセキュアな接続で送信されます。

第15章 TIGERVNC

Tiger vnc(Tiger Virtual Network Computing)は、グラフィカルデスクトップ共有用のシステムであり、他のコンピューターのリモート制御を可能にします。

Tiger vnc は、クライアントサーバープリンシパルで機能します。サーバー はその出力(vncserver)を共有し、クライアント (vncviewer)はサーバーに接続します。

15.1. VNC SERVER

vncserver は、VNC(Virtual Network Computing)デスクトップを起動するユーティリティーです。適切なオプションで Xvnc を実行し、VNC デスクトップでウィンドウマネージャーを起動します。vncserver を使用すると、どこからでも任意の数のクライアントがアクセス可能なマシンで個別のセッションを並行して実行できます。

15.1.1. VNC サーバーのインストール

TigerVNC サーバーをインストールするには、root で以下のコマンドを実行します。

```
~]# yum install tigervnc-server
```

15.1.2. VNC サーバーの設定

VNC サーバーでは、表示、ネットワークアドレス、ポート、セキュリティーの設定などに対するオプションのパラメーターを使用して、1 人または複数のユーザー用の画面を起動するよう設定できます(ユーザーのシステムがシステムに存在する場合)。

手順15.1 1 人のユーザー用に VNC サーバーの設定

- /etc/sysconfig/vncservers を編集し、次の形式で行を追加して、ユーザー名とディスプレイ番号を指定します。

  ```
  VNCSERVERS="display_number:user"
  ```

  VNC ユーザー名は、システムのユーザーに対応している必要があります。

  例15.1 ユーザーのディスプレイ番号の設定

たとえば、ユーザー joe のディスプレイ番号 3 を設定するには、設定ファイルを開いて編集します。

```
~]# vi /etc/sysconfig/vncservers
```

以下のように行を追加します。

```
VNCSERVERS="3:joe"
```

ファイルを保存してから閉じます。

上記の例では、ディスプレイ番号 3 とユーザー joe が設定されます。ワークステーションのメイン X ディスプレイは通常 0 と示されるため、ディスプレイ番号には 0 を使用しないでください。

手順15.2 複数のユーザーの VNC 表示の設定

- 複数のユーザーに VNC 表示を設定するには、/etc/sysconfig/vncservers を編集してユーザー名とディスプレイ番号を指定し、次の形式で行を追加します。

```
VNCSERVERS="display_number:user display_number:user"
```

VNC ユーザー名は、システムのユーザーに対応している必要があります。

例15.2 2 人のユーザーの表示番号の設定

たとえば、2 人のユーザーを設定するには、設定ファイルを開いて編集します。

```
~]# vi /etc/sysconfig/vncservers
```

以下のように行を追加します。

```
VNCSERVERS="3:joe 4:jill"
```

手順15.3 VNC 表示引数の設定

- 以下のように **VNCSERVERARGS** ディレクティブを使用して引数を追加して、**/etc/sysconfig/vncservers** ファイルに追加の設定を指定します。

```
VNCSERVERS="display_number:user display_number:user"
VNCSERVERARGS[display_number]="arguments"
```

表15.1 頻繁に使用される VNC サーバーパラメーター

VNCSERVERARGS	定義
-geometry	作成する VNC デスクトップのサイズを指定します。デフォルトは 1024x768 です。
-nolisten tcp	TCP(Transmission Control Protocol)を介した VNC サーバーへの接続を防止する
-localhost	セキュアなトンネルを介して実行する場合を除き、リモート VNC クライアントが接続しないようにする

その他のオプションについては、**Xvnc(1)** の **man** ページを参照してください。

例15.3 vncserver 引数の設定

上記の例では、2 人のユーザーに引数を追加するには、以下のように **/etc/sysconfig/vncservers** ファイルを編集します。

```
VNCSERVERS="3:joe 4:jill"
VNCSERVERARGS[1]="-geometry 800x600 -nolisten tcp -localhost"
VNCSERVERARGS[2]="-geometry 1920×1080 -nolisten tcp -localhost"
```

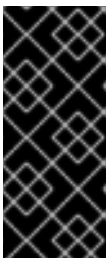
手順15.4 VNC ユーザーパスワードの設定

- **/etc/sysconfig/vncservers** ファイルで定義されたすべてのユーザーに **VNC** パスワードを設定するには、**root** で次のコマンドを実行します。

```
~]# vncpasswd
Password:
Verify:
```

ユーザーに **VNC** パスワードを個別に設定するには、以下を実行します。

```
~]# su - user
~]$ vncpasswd
Password:
Verify:
```



重要

保存されたパスワードは暗号化されていません。パスワードファイルへのアクセスが可能であれば、誰でもプレーンテキストのパスワードを見ることができます。

15.1.3. VNC サーバーの起動

VNC デスクトップを起動するには、vncserver ユーティリティーが使用されます。これは、Xvnc サーバーの起動プロセスを簡素化する Perl スクリプトです。適切なオプションで Xvnc を実行し、VNC デスクトップでウィンドウマネージャーを起動します。vncserver を開始する方法は 3 つあります。

- vncserver は、最初に利用可能なディスプレイ番号を選択し、そのディスプレイ番号で Xvnc を起動して、Xvnc セッションでデフォルトのウィンドウマネージャーを起動することができます。これらの手順はすべて 1 つのコマンドによって提供されます。

  ```
  ~]$ vncserver
  ```

  VNC パスワードが設定されていない場合は、コマンドの初回実行時に VNC パスワードを入力するように求められます。

- または、特定のディスプレイ番号を指定することもできます。

  ```
  vncserver :display_number
  ```

  vncserver はそのディスプレイ番号で Xvnc を起動しようとし、ディスプレイ番号が利用できない場合は終了するようにできます。

  以下に例を示します。

  ```
  ~]$ vncserver :20
  ```

- または、root で、/etc/sysconfig/vncservers 設定ファイルに設定したユーザーのディスプレイを使用して VNC サーバーを起動するには、次のコマンドを実行します。

```
~]# service vncserver start
```

システム起動時に vncserver サービスを自動的に有効にできます。ログインするたびに、vncserver は自動的に開始します。root で以下を実行します。

```
~]# chkconfig vncserver on
```

15.1.4. VNC セッションの終了

vncserver サービスの有効化と同様に、システム開始時に自動的にサービスの起動を無効にできます。

```
~]# chkconfig vncserver off
```

または、システムの実行中に、root で以下のコマンドを実行してサービスを停止できます。

```
~]# service vncserver stop
```

特定の表示を終了するには、ディスプレイ番号とともに -kill オプションを使用して vncserver を終了します。

例15.4 特定の表示の終了

たとえば、ディスプレイ番号 2 を終了するには、次のコマンドを実行します。

```
~]# vncserver -kill :2
```

例15.5 Xvnc プロセスの終了

VNC サービスを中断したり表示したりできない場合は、プロセス ID(PID)を使用して Xvnc セッションを終了します。プロセスを表示するには、以下を入力します。

```
~]$ service vncserver status
Xvnc (pid 4290 4189) is running...
```

プロセス 4290 を終了するには、root で次のコマンドを実行します。

```
~]# kill -s 15 4290
```

15.2. 既存のデスクトップの起動

デフォルトでは、ログインしているユーザーはディスプレイ 0 の X サーバーにより提供されたデスクトップを使用します。ユーザーは TigerVNC サーバー x0vncserver を使用してデスクトップを共有できます。

手順15.5 X デスクトップの共有

ログインしているユーザーのデスクトップを x0vncserver を使用して共有するには、以下の手順を実行します。

1. rootで次のコマンドを実行します。

```
~]# yum install tigervnc-server
```

2. ユーザーの VNC パスワードを設定します。

```
~]$ vncpasswd
Password:
Verify:
```

3. そのユーザーで以下のコマンドを入力します。

```
~]$ x0vncserver -PasswordFile=.vnc/passwd -AlwaysShared=1
```

ファイアウォールがポート 5900 への接続を許可するよう設定されている場合、リモートビューアーはディスプレイ 0 に接続し、ログインしているユーザーのデスクトップを表示できます。ファイアウォールの設定方法は 「VNC のためのファイアウォールの設定」 を参照してください。

15.3. VNC ビューアーの使用

VNC ビューアーは、VNC サーバーによって作成されたグラフィカルユーザーインターフェースを表示し、VNC サーバーをリモートで制御できるプログラムです。共有されているデスクトップは、デフォルトでシステムに直接ログインしたユーザーに表示されるデスクトップと同じではありません。VNC サーバーは、各ディスプレイ番号に対して一意のデスクトップを作成します。任意の数のクライアントが VNC サーバーに接続できます。

15.3.1. VNC ビューアーのインストール

root で TigerVNC クライアント vncviewer をインストールするには、以下のコマンドを実行します。

```
~]# yum install tigervnc
```

TigerVNC クライアントにはグラフィカルユーザーインターフェース(GUI)があり、vncviewer コマンドを入力すると起動できます。または、コマンドラインインターフェース(CLI)を使用して vncviewer を操作することもできます。vncviewer のパラメーター一覧を表示するには、コマンドラインで vncviewer -h と入力します。

15.3.2. VNC サーバーへの接続

VNC サーバーを設定すると、VNC サーバーを任意の VNC ビューアーに接続できます。

手順15.6 SSH を使用した VNC サーバーへの接続

1. *引数なしで vncviewer コマンドを入力すると、VNC Viewer: Connection Details ユーティリティーが表示されます。接続する VNC サーバーを求めるプロンプトが出されます。*

2. *必要な場合は、同じ画面への既存の VNC 接続の切断を回避するために、以下のようにデスクトップの共有を許可するオプションを選択します。*

   a. *Options (オプション) ボタンを選択します。*

   b. *Misc. タブを選択します。*

   c. *Shared （共有）ボタンを選択します。*

   d. *OK を選択してメインメニューに戻ります。*

3. *接続するアドレスとディスプレイ番号を入力します。*

   ```
   address:display_number
   ```

4. *Connect (接続) を押して VNC サーバー画面に接続します。*

5. *VNC パスワードを入力するよう求められます。これは、グローバルなデフォルトの VNC パスワードが設定されていない限り、ディスプレイ番号に対応するユーザーの VNC パスワードです。*

   VNC サーバーデスクトップを示すウィンドウが表示されます。これは通常のユーザーに表示されるデスクトップではなく、Xvnc デスクトップであることに注意してください。

手順15.7 CLI を使用した VNC サーバーへの接続

1. *引数としてアドレスとディスプレイ番号を指定して viewer コマンドを入力します。*

   ```
   vncviewer address:display_number
   ```

ここで、address は IP アドレスまたはホスト名です。

2. VNC パスワードを入力して自分自身を認証します。これは、グローバルなデフォルトの VNC パスワードが設定されていない限り、ディスプレイ番号に対応するユーザーの VNC パスワードです。

3. VNC サーバーデスクトップを示すウィンドウが表示されます。これは通常のユーザーに表示されるデスクトップではなく、Xvnc デスクトップであることに注意してください。

15.3.2.1. VNC のためのファイアウォールの設定

暗号化されていない接続を使用する場合は、ファイアウォールが接続を拒否する可能性があります。VNC プロトコルは、TCP パケットで転送される リモートフレームバッファー (RFB)です。必要な場合は、以下のように TCP プロトコルのポートを開きます。-via オプションを使用する場合、トラフィックはデフォルトで有効になっている SSH 経由でリダイレクトされます。

注記

VNC サーバーのデフォルトのポートは 5900 です。リモートデスクトップにアクセスできるポートに到達するには、デフォルトのポートとユーザーに割り当てられたディスプレイ番号の合計を計算します。たとえば、2 つ目のディスプレイは 2 + 5900 = 5902 のようになります。

手順15.8 lokkit を使用したポートを開く

lokkit コマンドは、コマンドラインを使用してポートを迅速に有効にする方法を提供します。

1. TCP のポート 5902 などの特定のポートを有効にするには、root で以下のコマンドを実行します。

```
~]# lokkit --port=5902:tcp --update
```

これにより、--disabled オプションで無効にされていない限り、ファイアウォールが再起動されることに注意してください。アクティブな接続は終了し、開始マシンでタイムアウトします。

2. 選択したポートが開いているかどうかを確認します。root で以下を入力します。

```
~]# iptables -L -n | grep 'tcp.*59'
ACCEPT     tcp  --  0.0.0.0/0     0.0.0.0/0    state NEW tcp dpt:5902
```

3. **VNC に使用するポート番号が分からない場合は、root で以下を入力します。**

```
~]# netstat -tnlp
tcp    0    0 0.0.0.0:6003    0.0.0.0:*    LISTEN    4290/Xvnc
tcp    0    0 0.0.0.0:5900    0.0.0.0:*    LISTEN    7013/x0vncserver
tcp    0    0 0.0.0.0:5902    0.0.0.0:*    LISTEN    4189/Xvnc
tcp    0    0 0.0.0.0:5903    0.0.0.0:*    LISTEN    4290/Xvnc
tcp    0    0 0.0.0.0:6002    0.0.0.0:*    LISTEN    4189/Xvnc
```

59XX を開始するポートは、VNC RFB プロトコル用です。60XX 以降のポートは、X windows プロトコル用です。

root で、ポートおよび Xvnc セッションに関連するユーザーを一覧表示するには、次のコマンドを実行します。

```
~]# lsof -i -P | grep vnc
Xvnc      4189    jane    0u  IPv6  27972    0t0  TCP *:6002 (LISTEN)
Xvnc      4189    jane    1u  IPv4  27973    0t0  TCP *:6002 (LISTEN)
Xvnc      4189    jane    6u  IPv4  27979    0t0  TCP *:5902 (LISTEN)
Xvnc      4290     joe    0u  IPv6  28231    0t0  TCP *:6003 (LISTEN)
Xvnc      4290     joe    1u  IPv4  28232    0t0  TCP *:6003 (LISTEN)
Xvnc      4290     joe    6u  IPv4  28244    0t0  TCP *:5903 (LISTEN)
x0vncserv 7013     joe    4u  IPv4  47578    0t0  TCP *:5900 (LISTEN)
```

手順15.9 エディターを使用したファイアウォールの設定

管理ツールを使用して複数のインストールのために設定ファイルを準備する際には、ファイアウォール設定ファイルを直接編集すると便利です。設定ファイルの間違いがある場合は、予期せぬ結果が発生し、エラーが発生する可能性があり、ファイアウォール設定が適用されないことに注意してください。したがって、編集後に /etc/sysconfig/system-config-firewall ファイルの詳細を確認します。

1. **ファイアウォールが許可するものを確認するには、root で次のコマンドを実行して、ファイアウォール設定ファイルを表示します。**

```
~]# less /etc/sysconfig/system-config-firewall
# Configuration file for system-config-firewall

--enabled
--service=ssh
```

この例では、デフォルトのインストールでファイアウォールは有効になっていますが、VNC ポートが通過するように設定されていません。

2. /etc/sysconfig/system-config-firewall で root として編集し、ファイアウォール設定ファイルに

```
--port=port_number:tcp
```

という行を追加します。たとえば、ポート 5902 を追加するには、次のコマンドを実行します。

```
~]# vi /etc/sysconfig/system-config-firewall
 # Configuration file for system-config-firewall

--enabled
--service=ssh
--port=5902:tcp
```

3. ファイアウォールが再読み込みされたり、システムが再起動されても、これらの変更は反映されないことに注意してください。/etc/sysconfig/system-config-firewall の設定を適用するには、root で以下のコマンドを実行します。

```
~]# lokkit --update
```

15.3.3. SSH を使用した VNC サーバーへの接続

VNC は、通信上の攻撃に対するセキュリティーのないクリアテキストネットワークプロトコルです。通信をセキュアにするには、-via オプションを指定してサーバークライアント接続を暗号化します。これにより、VNC サーバーとクライアントとの間に SSH トンネルが作成されます。

VNC サーバークライアント接続を暗号化するコマンドの形式は以下のとおりです。

```
vncviewer -via user@host:display_number
```

例15.6 -via オプションの使用

1. SSH を使用して VNC サーバーに接続するには、以下のようにコマンドを入力します。

```
$ vncviewer -via joe@192.168.2.101 127.0.0.1:3
```

2. プロンプトが表示されたら、パスワードを入力し、Enter を押して確認します。

3. リモートデスクトップのウィンドウが画面に表示されます。

SSH の使用の詳細は、14章OpenSSH を参照してください。

15.4. 関連情報

TigerVNC に関する詳細は、以下の資料を参照してください。

インストールされているドキュメント

- vncserver(1): VNC サーバーユーティリティーの man ページです。
- vncviewer(1): VNC ビューアーの man ページです。
- vncpasswd(1): VNC パスワードコマンドの man ページです。
- Xvnc(1): Xvnc サーバー設定オプションの man ページです。
- x0vncserver(1): 既存の X サーバーを共有する TigerVNC サーバーの man ページです。

パート VI. サーバー

ここでは、Web サーバーの設定方法やネットワーク上でのファイルおよびディレクトリーの共有方法など、サーバーに関連するさまざまなトピックについて説明します。

第16章 DHCP サーバー

DHCP (Dynamic Host Configuration Protocol: 動的ホスト構成プロトコル) は、クライアントマシンに TCP/IP 情報を自動的に割り当てるネットワークプロトコルです。各 DHCP クライアントは、一元的に配置された DHCP サーバーに接続します。このサーバーは、そのクライアントのネットワーク設定（IP アドレス、ゲートウェイ、DNS サーバーを含む）を返します。

16.1. DHCP を使用する理由

DHCP は、クライアントネットワークインターフェースの自動設定に役立ちます。クライアントシステムを設定する場合は、IP アドレス、ネットマスク、ゲートウェイ、または DNS サーバーを指定する代わりに DHCP を選択できます。クライアントはこの情報を DHCP サーバーから取得します。DHCP は、多数のシステムの IP アドレスを変更する場合に便利です。すべてのシステムを再設定する代わりに、サーバー上の 1 つの設定ファイルを新しい IP アドレス用に編集することのみが可能です。組織の DNS サーバーが変更されると、DHCP クライアントではなく、DHCP サーバーで変更が行われます。ネットワークを再起動するかクライアントを再起動すれば、変更が反映されます。

組織に機能的な DHCP サーバーがネットワークに正しく接続されている場合、ラップトップなどのモバイルコンピューターのユーザーは、これらのデバイスをオフィスからオフィスに移動できます。

16.2. DHCPV4 サーバーの設定

dhcp パッケージには、インターネットシステムの Consortium(ISC)DHCP サーバーが含まれます。まず、パッケージをスーパーユーザーとしてインストールします。

```
~]# yum install dhcp
```

dhcp パッケージをインストールすると、ファイル /etc/dhcp/dhcpd.conf が作成されます。これは、空の設定ファイルです。

```
~]# cat /etc/dhcp/dhcpd.conf
#
# DHCP Server Configuration file.
#   see /usr/share/doc/dhcp*/dhcpd.conf.sample
```

設定ファイルの例は、/usr/share/doc/dhcp-<バージョン>/dhcpd.conf.sample にあります。/etc/dhcp/dhcpd.conf を設定するする際に、このファイルを使用してください。詳細は以下で説明します。

DHCP は /var/lib/dhcpd/dhcpd.leases ファイルを使用してクライアントリースデータベースを保存します。詳細は、「リースデータベース」を参照してください。

16.2.1. 設定ファイル

DHCP サーバーを設定する最初のステップは、クライアントのネットワーク情報を格納する設定ファイルを作成することです。このファイルを使用して、クライアントシステムのオプションとグローバルオプションを宣言します。

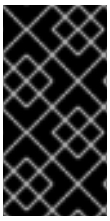
設定ファイルには追加のタブや空白行が含まれているため、簡単に書式を整えることができます。キーワードは大文字と小文字を区別せず、ハッシュ記号(#)で始まる行はコメントとみなされます。

設定ファイルのステートメントには、次のような 2 つのタイプがあります。

- パラメーター: タスクの実行方法、タスクを実行するかどうか、クライアントに送信するネットワーク設定のオプションを規定します。
- 宣言 - ネットワークトポロジの記述、クライアントの記述、クライアントのアドレス指定、宣言グループへのパラメーターグループの適用を行います。

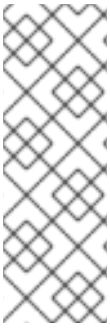
キーワードオプションから始まるパラメーターは、オプション と呼ばれます。これらのオプションは DHCP オプションを制御しますが、パラメーターはオプションではなく、DHCP サーバーの動作を制御する値を設定します。

中括弧({ })で囲まれたセクションの前に宣言されたパラメーター（オプションを含む）はグローバルパラメーターとみなされます。グローバルパラメーターは、これ以降のすべてのセクションに適用されます。



重要

設定ファイルを変更すると、コマンド service dhcpd restart で DHCP デーモンを再起動するまで変更は反映されません。



注記

DHCP 設定ファイルを変更し、毎回サービスを再起動する代わりに、omshell コマンドを使用して、DHCP サーバーへの接続、クエリー、および設定を変更するインタラクティブな方法を提供します。omshell を使用すると、DHCP サーバーの実行中でも変更を行うことができます。omshell の詳細については、omshell の man ページを参照してください。

例16.1「サブネットの宣言」 では、ルーター、subnet-mask、domain-search、domain-name-servers、および time-offset オプションは、以下に宣言された ホスト ステートメントに使用されます。

提供されるすべての サブネット、および DHCP サーバーが接続されているすべての サブネット について、サブネット 宣言が 1 つ必要です。これは、DHCP デーモンに対して、アドレスがその サブネット 上に存在することを認識する方法を示しています。サブネット にアドレスが動的に割り当てられない場合でも、サブネット ごとに subnet 宣言が必要です。

以下の例では、サブネットの DHCP クライアントごとにグローバルオプションがあり、範囲 が宣言されています。クライアントには、範囲 内の IP アドレスが割り当てられます。

例16.1 サブネットの宣言

```
subnet 192.168.1.0 netmask 255.255.255.0 {
        option routers                  192.168.1.254;
        option subnet-mask              255.255.255.0;
        option domain-search            "example.com";
        option domain-name-servers      192.168.1.1;
        option time-offset              -18000;     # Eastern Standard Time
 range 192.168.1.10 192.168.1.100;
}
```

動的 IP アドレスをサブネット内のシステムにリースする DHCP サーバーを設定するには、例16.2「Range パラメーター」 を実際の値で変更します。これにより、クライアントのデフォルトのリース時間、最大リース時間、ネットワークの設定値を宣言します。この例では、192.168.1.10 および 192.168.1.100 の 範囲 の IP アドレスをクライアントシステムに割り当てます。

例16.2 Range パラメーター

```
default-lease-time 600;
max-lease-time 7200;
option subnet-mask 255.255.255.0;
option broadcast-address 192.168.1.255;
option routers 192.168.1.254;
option domain-name-servers 192.168.1.1, 192.168.1.2;
```

```
option domain-search "example.com";
subnet 192.168.1.0 netmask 255.255.255.0 {
  range 192.168.1.10 192.168.1.100;
}
```

ネットワークインターフェースカードの MAC アドレスに基づいて IP アドレスをクライアントに割り当てるには、host 宣言内の ハードウェアイーサネット パラメーターを使用します。例16.3「DHCP を使用した静的 IP アドレス」 で説明されているように、ホスト apex 宣言は、MAC アドレス 00:A0:78:8E:9E:AA が常に IP アドレス 192.168.1.4 を受信するように指定します。

オプションのパラメーター host-name を使用して、クライアントにホスト名を割り当てることもできます。

例16.3 DHCP を使用した静的 IP アドレス

```
host apex {
  option host-name "apex.example.com";
  hardware ethernet 00:A0:78:8E:9E:AA;
  fixed-address 192.168.1.4;
}
```

同じ物理ネットワークを共有するすべてのサブネットは、例16.4「Shared-network 宣言」 に示されるように shared-network 宣言内で宣言する必要があります。shared-network 内のパラメーターですが、囲まれた subnet 宣言以外は、グローバルパラメーターとみなされます。shared-network の名前は、テストラボ環境内のすべてのサブネットを記述するために「test-lab」というタイトルを使用するなど、ネットワークの説明的なタイトルである必要があります。

例16.4 Shared-network 宣言

```
shared-network name {
   option domain-search           "test.redhat.com";
   option domain-name-servers     ns1.redhat.com, ns2.redhat.com;
   option routers                 192.168.0.254;
   #more parameters for EXAMPLE shared-network
   subnet 192.168.1.0 netmask 255.255.252.0 {
      #parameters for subnet
      range 192.168.1.1 192.168.1.254;
   }
   subnet 192.168.2.0 netmask 255.255.252.0 {
      #parameters for subnet
      range 192.168.2.1 192.168.2.254;
   }
}
```

例16.5「Group 宣言」 で説明されているように、group 宣言はグローバルパラメーターを宣言のグループに適用するために使用されます。たとえば、共有ネットワーク、サブネット、ホストをグループ化することができます。

例16.5 Group 宣言

```
group {
  option routers                192.168.1.254;
  option subnet-mask            255.255.255.0;
  option domain-search          "example.com";
  option domain-name-servers    192.168.1.1;
  option time-offset            -18000;     # Eastern Standard Time
  host apex {
    option host-name "apex.example.com";
    hardware ethernet 00:A0:78:8E:9E:AA;
    fixed-address 192.168.1.4;
  }
  host raleigh {
    option host-name "raleigh.example.com";
    hardware ethernet 00:A1:DD:74:C3:F2;
    fixed-address 192.168.1.6;
  }
}
```

サンプル設定ファイルの使用

提供される設定ファイルのサンプルを開始点として使用し、カスタム設定オプションを追加できます。このファイルを適切な場所にコピーするには、root で以下のコマンドを実行します。

```
~]# cp /usr/share/doc/dhcp-<version_number>/dhcpd.conf.sample
/etc/dhcp/dhcpd.conf
```

... ここで、 <version_number& gt; は DHCP バージョン番号になります。

オプションステートメントの完全なリストと、それらの操作は、dhcp-options の man ページを参照してください。

16.2.2. リースデータベース

DHCP サーバーでは、ファイル /var/lib/dhcpd/dhcpd.leases が DHCP クライアントのリースデータベースを保存します。このファイルは変更しないでください。最近割り当てられた各 IP アドレスの DHCP リース情報は、リースデータベースに自動的に保存されます。情報には、リースの長さ、IP アド

レスの割り当て、リースの開始日と終了日、リースの取得に使用されたネットワークインターフェースカードの MAC アドレスが含まれます。

リースデータベースの時刻はすべて、現地時間でなく協定世界時 (UTC) を使用します。

リースデータベースは、サイズが大きくなり過ぎるのを避けるために適宜再作成されます。最初に、すべての既知のリースは一時的なリースデータベースに保存されます。dhcpd.leases ファイルの名前は dhcpd.leases~ に変更され、一時的なリースデータベースが dhcpd.leases に書き込まれます。

DHCP デーモンを強制終了したり、リースデータベースがバックアップファイルに変更した後に、新規ファイルを書き込む前にシステムをクラッシュさせる可能性があります。この場合、dhcpd.leases ファイルは存在しませんが、サービスを起動する必要があります。この際、新規のリースファイルを作成しないでください。作成すると、それまでのリースはすべて失われ、多くの問題が発生します。これを解決する方法は、dhcpd.leases~ バックアップファイルの名前を dhcpd.leases に変更して、デーモンを起動することです。

16.2.3. サーバーの起動と停止

初めての DHCP サーバーの起動

DHCP サーバーを初めて起動すると、dhcpd.leases ファイルがなければ失敗します。ファイルがない場合は、touch /var/lib/dhcpd/dhcpd.leases コマンドを使用して作成します。

同じサーバーが DNS サーバーとして BIND を実行している場合でも、named サービスを開始すると dhcpd.leases ファイルが自動的にチェックされるため、この手順は必要ありません。

DHCP サービスを起動するには、コマンド /sbin/service dhcpd start を使用します。DHCP サーバーを停止するには、/sbin/service dhcpd stop コマンドを使用します。

デフォルトでは、DHCP サービスは起動時に起動しません。システムの起動時にデーモンが自動的に起動するように設定するには、12章サービスおよびデーモン を参照してください。

複数のネットワークインターフェースがシステムにアタッチされていて、DHCP サーバーをいずれかのインターフェースで起動する必要がある場合は、DHCP サーバーがそのデバイスでのみ起動するよ

うに設定します。/etc/sysconfig/dhcpd で、インターフェース名を DHCPDARGS の一覧に追加します。

```
# Command line options here
DHCPDARGS=eth0
```

これは、ネットワークカードが 2 つあるファイアウォールマシンで役立ちます。1 つのネットワークカードを DHCP クライアントとして設定すると、インターネットに IP アドレスを取得できます。他のネットワークカードは、ファイアウォールの内側にある内部ネットワーク用の DHCP サーバーとして使用できます。内部ネットワークに接続されたネットワークカードのみを指定するとユーザーはインターネット経由でデーモンに接続できないため、システムをよりセキュアにすることができます。

/etc/sysconfig/dhcpd で指定できるその他のコマンドラインオプションは次のとおりです。

- -p <portnum> - dhcpd がリッスンする UDP ポート番号を指定します。デフォルト値はポート 67 です。DHCP サーバーは、指定された UDP ポートよりも大きいポート番号で DHCP クライアントに応答を送信します。たとえば、デフォルトのポート 67 を使用する場合、サーバーはポート 67 でリクエストをリッスンし、ポート 68 にあるクライアントに応答します。ポートを指定して DHCP リレーエージェントを使用する場合は、DHCP リレーエージェントがリッスンするポートと同じポートを指定する必要があります。詳細は、「DHCP リレーエージェント」を参照してください。

- -f: フォアグラウンドプロセスとしてデーモンを実行します。これは主にデバッグ用に使用されます。

- -d - DHCP サーバーデーモンを標準のエラー記述子に記録します。これは主にデバッグ用に使用されます。このオプションを指定しないと、ログは /var/log/messages に書き込まれます。

- -cf <filename> - 設定ファイルの場所を指定します。デフォルトの場所は /etc/dhcp/dhcpd.conf です。

- -LF <filename> - リースデータベースファイルの場所を指定します。リースデータベースファイルがすでに存在する場合は、DHCP サーバーを起動するたびに同じファイルを使用することが非常に重要になります。このオプションは、実稼働環境以外のマシンでデバッグする目的にのみ使用することが強く推奨されます。デフォルトの場所は /var/lib/dhcpd/dhcpd.leases です。

- -q - デーモンの起動時に著作権に関するメッセージ全体を表示しません。

16.2.4. DHCP リレーエージェント

DHCP リレーエージェント(dhcrelay)では、DHCP サーバーを使用しないサブネットから、他のサブネット上の 1 つ以上の DHCP サーバーに、DHCP および BOOTP 要求のリレーを行うことができます。

DHCP クライアントが情報を要求すると、DHCP リレーエージェントは、DHCP リレーエージェントの起動時に指定した DHCP サーバーの一覧に要求を転送します。DHCP サーバーが応答を返すと、応答は元の要求を送信したネットワーク上でブロードキャストまたはユニキャストになります。

DHCP リレーエージェントは、INTERFACES ディレクティブを使用して /etc/sysconfig/dhcrelay で指定されていない限り、すべてのインターフェースの DHCP 要求をリッスンします。

DHCP リレーエージェントを起動するには、コマンド service dhcrelay start を使用します。

16.3. DHCPV4 クライアントの設定

DHCP クライアントを手動で設定するには、/etc/sysconfig/network ファイルを変更して、/etc/sysconfig/network-scripts ディレクトリー内の各ネットワークデバイスのネットワークおよび設定ファイルを有効にします。このディレクトリーでは、各デバイスに ifcfg-eth0 という名前の設定ファイルがなければなりません。ここで、eth0 はネットワークデバイス名です。

/etc/sysconfig/network-scripts/ifcfg-eth0 ファイルに以下の行が含まれていることを確認してください。

```
DEVICE=eth0
BOOTPROTO=dhcp
ONBOOT=yes
```

DHCP を使用するには、各デバイスに設定ファイルを設定します。

ネットワークスクリプトの他のオプションには以下が含まれます。

- DHCP_HOSTNAME: DHCP サーバーが IP アドレスを受信する前にクライアントがホスト名を指定する必要がある場合にのみこのオプションを使用します。

- *PEERDNS= <answer >。ここで、& lt;answer& gt; は以下のいずれかになります。*
  - *はい: サーバーの情報を使用して /etc/resolv.conf を変更します。これはデフォルトです。*
  - *No: /etc/resolv.conf は変更しないでください。*

グラフィカルインターフェースを使用する場合は、NetworkManager を使用して DHCP を使用するようにネットワークインターフェースを設定する手順は、10章NetworkManager を参照してください。

注記

プロトコルタイミング、リース要件、要求、動的 DNS サポート、エイリアス、クライアント側設定に上書き、追加、または追加するさまざまな値など、クライアント DHCP オプションの高度な設定は、man ページの dhclient および dhclient.conf を参照してください。

16.4. マルチホーム DHCP サーバーの設定

マルチホームの DHCP サーバーでは、複数のネットワーク（つまり複数のサブネット）が提供されます。本セクションでは、DHCP サーバーを設定して複数のネットワークを提供する方法、リッスンするネットワークインターフェースを選択する方法、およびネットワークを移動するシステムのネットワーク設定を定義する方法を説明します。

変更を行う前に、既存の /etc/sysconfig/dhcpd および /etc/dhcp/dhcpd.conf ファイルのバックアップを作成してください。

DHCP デーモンは、特に指定しない限り、すべてのネットワークインターフェースでリッスンします。/etc/sysconfig/dhcpd ファイルを使用して、DHCP デーモンがリッスンするネットワークインターフェースを指定します。以下の /etc/sysconfig/dhcpd 例は、DHCP デーモンが eth0 インターフェースおよび eth1 インターフェースをリッスンするように指定します。

```
DHCPDARGS="eth0 eth1";
```

システムにネットワークインターフェースカード eth0、eth1、および eth2 があり、DHCP デーモンが eth0 カードでリッスンすることのみが必要で、/etc/sysconfig/dhcpd で eth0 のみを指定する必要があります。

```
DHCPDARGS="eth0";
```

以下は、2 つのネットワークインターフェースを持つサーバーの場合、10.0.0.0/24 ネットワークに eth0 と 172.16.0.0/24 ネットワークの eth1 を持つサーバーの基本的な /etc/dhcp/dhcpd.conf ファイルです。複数の subnet 宣言で複数のネットワークに異なる設定を定義することができます。

```
default-lease-time 600;
max-lease-time 7200;
subnet 10.0.0.0 netmask 255.255.255.0 {
 option subnet-mask 255.255.255.0;
 option routers 10.0.0.1;
 range 10.0.0.5 10.0.0.15;
}
subnet 172.16.0.0 netmask 255.255.255.0 {
 option subnet-mask 255.255.255.0;
 option routers 172.16.0.1;
 range 172.16.0.5 172.16.0.15;
}
```

subnet 10.0.0.0 netmask 255.255.255.0;

DHCP サーバーが提供するすべてのネットワークに subnet 宣言が必要です。複数のサブネットには、複数の subnet 宣言が必要です。DHCP サーバーに subnet 宣言の範囲にネットワークインターフェースがない場合、DHCP サーバーはそのネットワークを提供しません。

subnet 宣言が 1 つしかなく、ネットワークインターフェースがそのサブネットの範囲にない場合、DHCP デーモンは起動に失敗し、以下のようなエラーが /var/log/messages に記録されます。

```
dhcpd: No subnet declaration for eth0 (0.0.0.0).
dhcpd: ** Ignoring requests on eth0.  If this is not what
dhcpd:    you want, please write a subnet declaration
dhcpd:    in your dhcpd.conf file for the network segment
dhcpd:    to which interface eth1 is attached. **
dhcpd:
dhcpd:
dhcpd: Not configured to listen on any interfaces!
```

option subnet-mask 255.255.255.0;

option subnet-mask オプションは、サブネットマスクを定義し、 subnet 宣言内の netmask 値を上書きします。簡単なケースでは、サブネットとネットマスクの値は同じです。

option routers 10.0.0.1;

option routers オプションは、サブネットのデフォルトゲートウェイを定義します。これは、

システムが異なるサブネット上の内部ネットワーク、さらには外部ネットワークに届くために必要です。

range 10.0.0.5 10.0.0.15;

range オプションは、利用可能な IP アドレスのプールを指定します。システムには、指定された IP アドレスの範囲からアドレスが割り当てられます。

詳細は、dhcpd.conf(5) の man ページを参照してください。

16.4.1. ホストの設定

変更を行う前に、既存の /etc/sysconfig/dhcpd および /etc/dhcp/dhcpd.conf ファイルのバックアップを作成してください。

複数ネットワークに対する単一システムの設定

以下の /etc/dhcp/dhcpd.conf 例は、2 つのサブネットを作成し、接続するネットワークに応じて、同じシステムの IP アドレスを設定します。

```
default-lease-time 600;
max-lease-time 7200;
subnet 10.0.0.0 netmask 255.255.255.0 {
 option subnet-mask 255.255.255.0;
 option routers 10.0.0.1;
 range 10.0.0.5 10.0.0.15;
}
subnet 172.16.0.0 netmask 255.255.255.0 {
 option subnet-mask 255.255.255.0;
 option routers 172.16.0.1;
 range 172.16.0.5 172.16.0.15;
}
host example0 {
 hardware ethernet 00:1A:6B:6A:2E:0B;
 fixed-address 10.0.0.20;
}
host example1 {
 hardware ethernet 00:1A:6B:6A:2E:0B;
 fixed-address 172.16.0.20;
}
```

host example0

host 宣言は、IP アドレスなどの単一システムの特定のパラメーターを定義します。複数のホ

ストに特定のパラメーターを設定するには、複数の host 宣言を使用します。

ほとんどの DHCP クライアントは host 宣言の名前を無視します。そのため、他の host 宣言に固有である限り、この名前は任意の名前にすることができます。複数のネットワークに同じシステムを設定するには、host 宣言ごとに異なる名前を使用します。そうでない場合には、DHCP デーモンの起動に失敗します。システムは、host 宣言の名前ではなく、hardware ethernet オプションで識別されます。

hardware ethernet 00:1A:6B:6A:2E:0B;

hardware ethernet オプションはシステムを識別します。アドレスを確認するには、ip link コマンドを実行します。

fixed-address 10.0.0.20;

fixed-address オプションは、hardware ethernet オプションで指定したシステムに有効な IP アドレスを割り当てます。このアドレスは、range オプションで指定した IP アドレスプール外である必要があります。

option ステートメントがセミコロンで終了しない場合、DHCP デーモンは起動に失敗し、以下のようなエラーが /var/log/messages に記録されます。

```
/etc/dhcp/dhcpd.conf line 20: semicolon expected.
dhcpd: }
dhcpd: ^
dhcpd: /etc/dhcp/dhcpd.conf line 38: unexpected end of file
dhcpd:
dhcpd: ^
dhcpd: Configuration file errors encountered -- exiting
```

複数のネットワークインターフェースを持つシステムの設定

以下の host 宣言では、複数のネットワークインターフェースを持つ 1 つのシステムを設定し、各インターフェースが同じ IP アドレスを受け取るようにします。両方のネットワークインターフェースが同じネットワークに同時に接続されている場合には、この設定は機能しません。

```
host interface0 {
 hardware ethernet 00:1a:6b:6a:2e:0b;
 fixed-address 10.0.0.18;
}
host interface1 {
```

```
hardware ethernet 00:1A:6B:6A:27:3A;
fixed-address 10.0.0.18;
}
```

以下の例では、interface0 は最初のネットワークインターフェースで、interface1 は 2 番目のインターフェースです。異なる hardware ethernet オプションは、各インターフェースを特定します。

このようなシステムが別のネットワークに接続されている場合は、host 宣言をさらに追加します。

- ホストが接続しているネットワークに有効な fixed-address を割り当てます。
- host 宣言の名前を一意にします。

host 宣言で指定した名前が一意でない場合は、DHCP デーモンは起動に失敗し、以下のようなエラーが /var/log/messages に記録されます。

```
dhcpd: /etc/dhcp/dhcpd.conf line 31: host interface0: already exists
dhcpd: }
dhcpd: ^
dhcpd: Configuration file errors encountered -- exiting
```

このエラーは、/etc/dhcp/dhcpd.conf に複数の host interface0 宣言が定義されているために生じました。

16.5. IPV6 の DHCP (DHCPV6)

ISC DHCP には、DHCPv6 サーバー、クライアント、リレーエージェント機能を使用する 4.x リリース以降、IPv6(DHCPv6)のサポートが含まれています。サーバー、クライアント、およびリレーエージェントは、IPv4 と IPv6 の両方をサポートします。ただし、クライアントとサーバーは、一度に 1 つのプロトコルのみを管理できます。デュアルサポートの場合、IPv4 と IPv6 に対して個別に起動する必要があります。

16.5.1. DHCPv6 サーバーの設定

DHCPv6 サーバー設定ファイルは、dhcp パッケージとともにインストールされ、/etc/dhcp/dhcpd6.conf にあります。

サーバー設定ファイルのサンプルは、Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;LinuxRed Hat Enterprise Linux 6nbsp;6(/usr/share/doc/dhcp- 4.1.1/dhcpd6.conf.sample)の

/usr/share/doc/dhcp-<version >/dhcpd6.conf.sample にあります。

シンプルな DHCPv6 サーバー設定ファイルは以下のようになります。

```
subnet6 2001:db8:0:1::/64 {
        range6 2001:db8:0:1::129 2001:db8:0:1::254;
        option dhcp6.name-servers fec0:0:0:1::1;
        option dhcp6.domain-search "domain.example";
}
```

その他の例は、dhcpd.conf(5) man ページを参照してください。

DHCPv6 サービスを起動するには、root でコマンド サービス dhcpd6 start を実行します。DHCPv6 サーバーを停止するには、コマンド サービスの dhcpdv6 stop を使用します。

DHCPv6 サービスの起動時に dhcpd デーモンにコマンドラインオプションを渡すには、/etc/sysconfig/dhcpd6 ファイルを使用します。このファイルは、/etc/sysconfig/dhcpd と同じ構造を使用します。

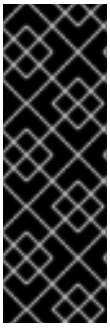
```
# cat /etc/sysconfig/dhcpd6
# Command line options here
DHCPDARGS=
```

DHCPDARGS オプションに追加した値は、DHCPv6 サービスに渡され、dhcpd デーモンに渡されます。詳細は、dhcpd-options(5) man ページの STANDARD DHCPV6 OPTIONS セクションを参照してください。その他の例は、Fedora Project wiki の 動的 IPv6 設定 を参照してください。

16.5.2. DHCPv6 クライアントの設定

DHCPv6 クライアントのデフォルト設定は、ほとんどの場合で正常に機能します。ただし、DHCP クライアントを手動で設定するには、/etc/dhcp/dhclient.conf ファイルを作成および変更します。クライアント設定ファイルの例は、/usr/share/doc/dhclient-4.1.1/dhclient6.conf.sample を参照してください。

プロトコルのタイミング、リース要件、要求、動的 DNS サポート、エイリアス、クライアント側の設定を上書き、追加、または追加するさまざまな値など、DHCPv6 クライアントオプションの高度な設定は、man ページの dhclient.conf(5)および dhcpd-options(5) man ページの STANDARD DHCPV6 OPTIONS セクションを参照してください。



重要

Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;LinuxRed Hat Enterprise Linux 6nbsp;6 では、DHCPv6 クライアントは NetworkManager のみで正しく処理されるため、通常は個別に実行することはできません。これは DHCPv4 とは異なり、DHCPv4 とは異なり、常にスタンドアロンのネットワーク設定プロトコルではなく、ルーター検出と併用する必要があります。

16.6. その他のリソース

*詳細 』*は、DHCP Handbook、Ralph Droms および Ted Lemon; か以下のリソースを*』 参照してください。*

16.6.1. インストールされているドキュメント

- *dhcpd man ページ - DHCP デーモンの動作が説明されています。*
- *man ページの dhcpd.conf - DHCP 設定ファイルの設定方法を検討し、いくつかの例が含まれています。*
- *dhcpd.leases man ページ - リースの永続的なデータベースを説明しています。*
- *man ページの dhcp-options - dhcpd.conf で DHCP オプションを宣言する構文には、いくつかの例が含まれています。*
- *dhcrelay の man ページ： DHCP リレーエージェントおよびその設定オプションを説明します。*
- */usr/share/doc/dhcp-<バージョン& gt;/: 現在の DHCP サービスのファイル、README ファイル、およびリリースノートが含まれています。*

第17章 DNS SERVERS: DNS サーバーの IP アドレス

DNS （ドメイン名システム）は、ネームサーバー とも呼ばれるネットワークシステムであり、ホスト名をそれぞれの IP アドレスに関連付けます。ユーザーにとっては、ネットワーク上のマシンを名前で参照できるという利点があります。システム管理者は、ネームサーバーを使用すると、名前ベースのクエリーに影響を与えることなく、ホストの IP アドレスを変更したり、これらのクエリーを処理するマシンを決定したりできます。

17.1. DNS の概要

DNS は通常、特定のドメインに対して権威のある 1 つ以上の集中型サーバーを使用して実装されます。クライアントホストがネームサーバーから情報を要求すると、通常ポート 53 に接続します。次にネームサーバーは要求された名前の解決を試みます。権威の回答がない場合や、以前のクエリーからキャッシュされた回答がない場合、root nameserver と呼ばれる他のネームサーバーをクエリーし、問題の名前で権威のあるネームサーバーを判断し、要求した名前を取得するためにクエリーを実行します。

17.1.1. ネームサーバーゾーン

BIND(Berkeley Internet Name Domain)などの DNS サーバーでは、すべての情報は リソースレコード (RR)と呼ばれる基本的なデータ要素に保存されます。リソースレコードは通常、ホストの 完全修飾ドメイン名 (FQDN)で、ツリーのような階層に分けられた複数のセクションに分かれています。この階層は、主要なトランク、プライマリーブランチ、セカンダリーブランチなどで構成されます。

> 例17.1 シンプルなリソースレコード
>
> ```
> bob.sales.example.com
> ```

階層の各レベルは、ピリオド(.)で分割されます。例17.1「シンプルなリソースレコード」 では、com で トップレベルのドメイン、そのサブドメイン の例 を定義し、example のサブドメインを売上 します。この場合、bob は sales.example.com ドメインの一部であるリソースレコードを特定します。左側(bob)以外では、各セクションは ゾーン と呼ばれ、特定の namespace を定義します。

ゾーンは、ゾーンを使用して権威ネームサーバーで定義さ れます。ゾーン には、各ゾーンのリソースレコードの定義が含まれます。ゾーンファイルは プライマリーネームサーバー（別名 マスターネームサーバー ）に保存され、ファイルに変更が加えられます。セカンダリーネームサーバー（ スレーブネームサーバーとも呼ばれる）は、プライマリーネームサーバー からゾーン定義を受け取るセカンダリーネームサーバーです。プライマリーネームサーバーとセカンダリーネームサーバーはいずれもゾーンに対して権威があり、クライアントと同じです。設定によっては、ネームサーバーは複数のゾーンに対してもプライマリーサーバーまたはセカンダリーサーバーとして機能します。

17.1.2. ネームサーバーの種別

ネームサーバーの設定には、2 つのタイプがあります。

権威

権威ネームサーバーは、ゾーンの一部のみのリソースレコードに応答します。このカテゴリーには、プライマリー（マスター）とセカンダリー（スレーブ）ネームサーバーの両方が含まれます。

再帰

再帰ネームサーバーは解決サービスを提供しますが、いずれのゾーンにも権威がありません。すべての解決への回答は一定期間はメモリにキャッシュされ、取得したリソースレコードで指定されます。

ネームサーバーは、同時に権威と再帰の両方を使用できますが、設定タイプを組み合わせることは推奨されません。権威サーバーが機能するには、これらが常にすべてのクライアントに利用可能となる必要があります。一方で、再帰的ルックアップは権威ある応答よりはるかに時間がかかるため、再帰的なサーバーは限られた数のクライアントにのみ利用可能とすべきです。それ以外の場合は、DDoS 攻撃 (分散型サービス拒否攻撃) の可能性が高まります。

17.1.3. ネームサーバーとしての BIND

BIND は一連の DNS 関連プログラムで構成されています。これには、named という名前のネームサーバー、rndc と呼ばれる管理ユーティリティー、および dig と呼ばれるデバッグツールが含まれます。Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux でサービスを実行する方法については、12章サービスおよびデーモン を参照してください。

17.2. BIND

本章では、Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux に含まれる DNS サーバーである BIND (Berkeley Internet Name Domain)について説明します。ここでは、その設定ファイルの構造にフォーカスし、ローカルとリモートの両方での管理方法を記述しています。

17.2.1. named サービスの設定

named サービスは起動時に、表17.1「named サービスの設定ファイル」に記載のファイルから設定を読み込みます。

表17.1 named サービスの設定ファイル

パス	説明
/etc/named.conf	主要設定ファイル。
/etc/named/	主要設定ファイル内に含まれている設定ファイル用の補助ディレクトリー。

設定ファイルは、中括弧を開いて閉じて、ネストされたオプションを持つステートメントのコレクションで構成されます。ファイルを編集しても構文エラーを行わないでください。編集しないと、named サービスは起動しません。一般的な /etc/named.conf ファイルは、以下のように整理されています。

```
statement-1 ["statement-1-name"] [statement-1-class] {
  option-1;
  option-2;
  option-N;
};
statement-2 ["statement-2-name"] [statement-2-class] {
  option-1;
  option-2;
  option-N;
};
statement-N ["statement-N-name"] [statement-N-class] {
  option-1;
  option-2;
  option-N;
};
```

CHROOT 環境での BIND の実行

bind-chroot パッケージをインストールすると、BIND サービスは /var/named/chroot 環境で実行されます。その場合、初期化スクリプトは mount --bind コマンドを使って上記の設定ファイルをマウントするので、この環境外で設定が管理できます。自動的にマウントされるため、/var/named/chroot ディレクトリーに何もコピーする必要はありません。これにより、chroot 環境で実行する場合は BIND 設定ファイルの特別な処理を行う必要がないため、メンテナンスが容易になります。BIND が chroot 環境で実行されていない場合には、すべて整理できます。

以下のディレクトリーは、/var/named/chroot ディレクトリーで空の場合、/var/named/chroot に自動的にマウントされます。/var/named/chroot にマウントする場合は、空のままにする必要があります。

- /var/named

- */etc/pki/dnssec-keys*
- */etc/named*
- */usr/lib64/bind または /usr/lib/bind (アーキテクチャーに依存)*

ターゲットファイルが /var/named/chroot に存在しない場合には、以下のファイルがマウントされます。

- */etc/named.conf*
- */etc/rndc.conf*
- */etc/rndc.key*
- */etc/named.rfc1912.zones*
- */etc/named.dnssec.keys*
- */etc/named.iscdlv.key*
- */etc/named.root.key*

17.2.1.1. 一般的なステートメントのタイプ

/etc/named.conf では、通常、以下のタイプのステートメントが使用されます。

acl

acl (Access Control List) (アクセス制御リスト) ステートメントにより、ホストのグループを定義できるようになるため、それらのホストはネームサーバーへのアクセスを許可/拒否できるようになります。以下の形式を取ります。

```
acl acl-name {
  match-element;
  ...
};
```

acl-name ステートメント名はアクセス制御リストの名前であり、match-element オプションは通常個別の IP アドレス（例： 10.0.1. 1）または CIDR(Classless Inter-Domain Routing)ネットワーク表記（例： 10.0.1 .0/24）です。定義済みのキーワードの一覧は、表17.2「事前定義されたアクセス制御リスト」を参照してください。

表17.2 事前定義されたアクセス制御リスト

キーワード	説明
any	すべての IP アドレスと一致します。
localhost	ローカルシステムが使用している IP アドレスと一致します。
localnets	ローカルシステムが接続している任意のネットワークの IP アドレスと一致します。
none	いずれの IP アドレスにも一致しません。

acl ステートメントは、特に オプション などの他のステートメントと併用できます。例17.2「acl をオプションと併用する」 ブラックリスト（ black-hats および red-hats ）の 2 つのアクセス制御リストを定義し、red-hats に通常のアクセスを付与する間にブラックリストに black-hats を追加します。

例17.2 acl をオプションと併用する

```
acl black-hats {
  10.0.2.0/24;
  192.168.0.0/24;
  1234:5678::9abc/24;
};
acl red-hats {
```

```
    10.0.1.0/24;
  };
  options {
    blackhole { black-hats; };
    allow-query { red-hats; };
    allow-query-cache { red-hats; };
  };
```

include

include ステートメントにより、ファイルを /etc/named.conf 内に含めることができるので、機密性のあるデータを制限のあるパーミッションで別のファイルに配置できます。以下の形式を取ります以下の形式を取ります。

```
include "file-name"
```

file-name ステートメント名はファイルへの絶対パスとなります。

例17.3 /etc/named.conf へのファイルの追加

```
include "/etc/named.rfc1912.zones";
```

options

options ステートメントでは、グローバルサーバー設定オプションや他のステートメントのデフォルトを設定できます。名前付き の作業ディレクトリーの場所、許可されたクエリーのタイプなどを指定できます。以下の形式を取ります。

```
options {
  option;
  ...
};
```

よく使用される option ディレクティブの一覧は、表17.3「一般的に使用されるオプション」を参照してください。

表17.3 一般的に使用されるオプション

オプション	説明

オプション	説明
allow-query	権限のあるリソースレコード用のネームサーバーにクエリーを許可されるホストを指定します。CIDR 表記では、アクセス制御リスト、IP アドレスのコレクション、またはネットワークを受け入れます。デフォルトではすべてのホストが許可されています。
allow-query-cache	再帰クエリーなど権限の必要ないデータ用のネームサーバーにクエリーを許可されるホストを指定します。デフォルトでは、**localhost** と **localnets** のみが許可されています。
blackhole	ネームサーバーへのクエリーを許可されないホストを指定します。このオプションは、特定のホストまたはネットワークにリクエストがあるサーバーをいっぱいにする場合に使用する必要があります。デフォルトのオプションは **none** です。
directory	**named** サービス用の作業ディレクトリーを指定します。デフォルトのオプションは **/var/named/** です。
dnssec-enable	DNSSEC 関連のリソースレコードを返すかどうかを指定します。デフォルトのオプションは **yes** です。
dnssec-validation	リソースレコードが DNSSEC を介して認証されていることを確認するかどうかを指定します。デフォルトのオプションは **yes** です。
forwarders	解決用に要求を転送するネームサーバーの有効な IP アドレス一覧を指定します。

オプション	説明
進む	forwarders ディレクティブの動作を指定します。以下のオプションを取ります。 • first: サーバーは、独自の名前の解決を試行する前に、forwarders ディレクティブに一覧表示されるネームサーバーをクエリーします。 • only: forwarders ディレクティブに一覧表示されるネームサーバーをクエリーできない場合、サーバーは独自の名前の解決を試行しません。
listen-on	クエリーをリッスンする IPv4 ネットワークインターフェースを指定します。ゲートウェイとしても機能する DNS サーバーでは、このオプションを使用して、1 つのネットワークからのみ発信されたクエリーに対応できます。デフォルトでは、すべての IPv4 インターフェースが使用されます。
listen-on-v6	クエリーをリッスンする IPv6 ネットワークインターフェースを指定します。ゲートウェイとしても機能する DNS サーバーでは、このオプションを使用して、1 つのネットワークからのみ発信されたクエリーに対応できます。デフォルトでは、すべての IPv6 インターフェースが使用されます。
max-cache-size	サーバー用キャッシュとして使用されるメモリーの最大容量を指定します。最大値に到達すると、その限度を超過しないようにサーバーは記録が早期期限切れになるようにします。複数表示を持つサーバーでは、この制限は各表示のキャッシュ毎に別々に適用されます。デフォルトのオプションは 32M です。

オプション	説明
notify	あるゾーンが更新された時にセカンダリーネームサーバーに通知するかどうかを指定します。以下のオプションを取ります。 • yes: サーバーはすべてのセカンダリーネームサーバーに通知します。 • no: サーバーはセカンダリーネームサーバーに通知しません。 • master-only: サーバーはゾーンに対してのみプライマリーサーバーに通知します。 • explicit: サーバーは、ゾーンステートメント内の also-notify 一覧で指定したセカンダリーサーバーのみを通知します。
pid-file	named サービスで作成されたプロセス ID ファイルの場所を指定します。
recursion	再帰的なサーバーとして動作するかどうかを指定します。デフォルトのオプションは yes です。
statistics-file	統計ファイルの代替の場所を指定します。デフォルトでは、/var/named/named.stats ファイルがデフォルトで使用されています。

選択したクライアントのみへの再帰サーバーの制限

DDoS(DDoS)攻撃を防止するには、allow-query-cache オプションを使用して、特定のクライアントサブセットの再帰 DNS サービスのみを制限することが推奨されます。

利用可能なオプションの詳細の一覧については、「インストールされているドキュメント」で参照されている『BIND 9 管理者リファレンスマニュアル』および named.conf man ページを参照してください。

例17.4 options ステートメントの使用

```
options {
  allow-query       { localhost; };
  listen-on port    53 { 127.0.0.1; };
  listen-on-v6 port 53 { ::1; };
  max-cache-size    256M;
  directory         "/var/named";
  statistics-file   "/var/named/data/named_stats.txt";

  recursion         yes;
  dnssec-enable     yes;
  dnssec-validation yes;
};
```

zone

zone ステートメントでは、設定ファイルやゾーン固有のオプションなど、ゾーンの特性を定義でき、グローバル options ステートメントを上書きするのに使用できます。以下の形式を取ります。

```
zone zone-name [zone-class] {
  option;
  ...
};
```

zone-name 属性はゾーン の名前で、zone-class はゾーンの オプション です。option は、表17.4「一般的に使用されるオプション」で説明されているように zone ステートメントオプションになります。

zone-name 属性は、/var/named/ ディレクトリーにある対応するゾーンファイル内で使用される $ORIGIN ディレクティブに割り当てられたデフォルト値であるため、特に重要になります。named デーモンはゾーンの名前を、ゾーンファイル内に一覧表示された非完全修飾型のドメイン名のいずれかに追記します。たとえば、zone ステートメントが example.com の名前空間を定義する場合は、zone-name として example.com を使用し、example.com ゾーンファイル内のホスト名の最後に配置されるようにします。

ゾーンファイルの詳細は、「ゾーンファイルの編集」 を参照してください。

表17.4 一般的に使用されるオプション

オプション	説明
allow-query	このゾーンに関する情報要求が出来るクライアントを指定します。このオプションはグローバル allow-query オプションを上書きします。デフォルトではすべてのクエリー要求が許可されます。
allow-transfer	ゾーン情報の転送要求を許可されるセカンダリーサーバーを指定します。デフォルトでは、すべての転送要求が許可されています。
allow-update	自身のゾーン内で動的な情報更新を許可されるホストを指定します。デフォルトオプションでは、すべての動的更新要求は拒否されます。 ホストがゾーンについての情報を更新可能とするには注意が必要です。サーバーが信頼できるネットワークにある場合を除き、このオプションに IP アドレスを設定しないでください。代わりに、「Transaction SIGnatures トランザクション署名 (TSIG)」 の説明に従って TSIG キーを使用します。
file	ゾーンの設定データを収納している named 作業ディレクトリー内のファイル名を指定します。
masters	権威ゾーン情報を要求する IP アドレスを指定します。このオプションは、ゾーンが type slave として定義されている場合にのみ使用されます。

オプション	説明
notify	あるゾーンが更新された時にセカンダリーネームサーバーに通知するかどうかを指定します。以下のオプションを取ります。 • yes: サーバーはすべてのセカンダリーネームサーバーに通知します。 • no: サーバーはセカンダリーネームサーバーに通知しません。 • master-only: サーバーはゾーンに対してのみプライマリーサーバーに通知します。 • explicit: サーバーは、ゾーンステートメント内の also-notify 一覧で指定したセカンダリーサーバーのみを通知します。

オプション	説明
type	ゾーンのタイプを指定します。以下のオプションを取ります。 • delegation-only: COM、NET、ORG などのインフラストラクチャーゾーンの委譲ステータスを強制します。明示的あるいは暗示的な委任のない受信回答は NXDOMAIN として扱われます。このオプションは、再帰的あるいはキャッシング実装で使用される TLD もしくは root のゾーンのファイルにのみ適用されます。 • forward: このゾーンに関する情報へのすべての要求を他のネームサーバーに転送します。 • hint: ゾーンが不明な場合にクエリーを解決するルートネームサーバーをポイントする特別な種類のゾーン。hint ゾーンでは、デフォルト以外の設定は必要ありません。 • master: このゾーンに対してネームサーバーを権威として指定します。ゾーンの設定ファイルがシステムに存在する場合は、ゾーンを master として設定する必要があります。 • slave: このゾーンに対してネームサーバーをスレーブサーバーとして指定します。マスターサーバーは masters ディレクティブで指定します。

プライマリーまたはセカンダリーのネームサーバーの /etc/named.conf ファイルに対するほとんどの変更には、zone ステートメントの追加、修正、または削除が含まれ、通常は zone ステートメントオプションの小さなサブセットのみが、ネームサーバーの効率的な機能のために必要となります。

例17.5「プライマリーネームサーバーのゾーンステートメント」では、ゾーンは

example.com として識別されており、タイプは master にセットされて、named サービスは /var/named/example.com.zone ファイルを読み込むように指示されています。これはまた、ゾーンの転送にセカンダリーネームサーバー (192.168.0.2) のみを許可します。

例17.5 プライマリーネームサーバーのゾーンステートメント

```
zone "example.com" IN {
  type master;
  file "example.com.zone";
  allow-transfer { 192.168.0.2; };
};
```

セカンダリーサーバーの zone ステートメントは、若干異なります。タイプは slave に設定され、masters ディレクティブはマスターサーバーの IP アドレスの 名前 を指定します。

例17.6「セカンダリーネームサーバーのゾーンステートメント」 では、example.com ゾーンに関する情報に対して、named サービスが IP アドレスにプライマリーサーバーをクエリーするように設定されます。受信した情報はその後、/var/named/slaves/example.com.zone ファイルに保存されます。すべてのスレーブゾーンを /var/named/slaves ディレクトリーに置く必要があります。そうでないと、ゾーンの送信に失敗します。

例17.6 セカンダリーネームサーバーのゾーンステートメント

```
zone "example.com" {
  type slave;
  file "slaves/example.com.zone";
  masters { 192.168.0.1; };
};
```

17.2.1.2. その他のステートメントタイプ

以下のタイプのステートメントは、通常、/etc/named.conf ではあまり使用されません。

controls

controls ステートメントにより各種設定が可能になり、named サービスを管理するための rndc コマンドの使用に必要なさまざまなセキュリティー要件を設定できるようになります。

rndc ユーティリティーとその使用方法についての詳細は、「rndc ユーティリティーの使用」を参照してください。

key

key ステートメントでは、名前で特定のキーを定義できます。キーは、安全な更新や、あるいは、rndc コマンドの使用など各種動作を認証するために使用されます。以下の 2 つのオプションが key と合わせて使用されます。

- *algorithm algorithm-name: 使用されるアルゴリズムのタイプ (たとえば、hmac-md5)。*
- *secret "key-value": 暗号化キー。*

rndc ユーティリティーとその使用方法についての詳細は、「rndc ユーティリティーの使用」を参照してください。

logging

logging ステートメントでは、channels と呼ばれる複数の種類のログを使用できます。ステートメント内で channel オプションを使用すると、独自のファイル名（ファイル）、サイズ制限（サイズ）、バージョン管理（バージョン）、重要度（重大度）を使用してログのカスタムタイプを作成できます。カスタマイズされたチャネルが定義されると、category オプションを使用して、チャネルを分類し、named サービスの再起動時にロギングを開始します。

デフォルトでは、named は標準メッセージを rsyslog デーモンに送信して、受信したデーモンはメッセージを /var/log/messages に配置します。それらの重要度レベルとして、default_syslog (情報ロギングメッセージを処理) と default_debug (特にデバッギングメッセージを処理) などがあります。default と呼ばれるデフォルトカテゴリは、組み込み型チャンネルを使用して特別な設定なしで通常のロギングを行います。

ロギングプロセスのカスタマイズは詳細なプロセスとなるため、本章の範囲外になります。カスタム BIND ログの作成に関する詳細は、「インストールされているドキュメント」 で参照されている 『BIND 9 Administrator Reference Manual』 を参照してください。

server

server ステートメントにより、named サービスがリモートのネームサーバーに対しての反応の仕方に影響する、特に通知とゾーン転送に関して影響するオプションを指定できるようになります。

transfer-format オプションは、各メッセージと共に送信されるリソースレコードの数を制御します。これには、1-answer (リソースレコード 1 つのみ)、または many-answer (複数リソースレコード) のいずれかになります。many-answers オプションはより効率的ですが、以前のバージョンの BIND ではサポートされていないことに注意してください。

ステートメントを使うと、安全な DNS (DNSSEC) に使用される各種パブリックキーを指定できるようになります。

trusted-keys ステートメントでは、セキュアな DNS(DNSSEC)に使用されるソートされた公開鍵を指定できます。このトピックの詳細については、「DNSSEC (DNS Security Extensions)」を参照してください。

match-clients

view ステートメントでは、ホストがネームサーバーをクエリーするネットワークに応じて、特別なビューを作成できます。これにより、他のホストが全く異なる情報を受け取る間、ゾーンに関する応答が 1 つの応答を受け取ることができます。また、信頼されないホスト以外のホストでは他のゾーンに対するクエリーしか実行できません。

view はその名前が一意になっていれば、複数のものを使用できます。match-clients オプションを使用すると、特定のビューに適用する IP アドレスを指定できます。options ステートメントがビュー内で使用されると、設定済みのグローバルオプションが上書きされます。最後に、ほとんどの view ステートメントには、match-clients リストに適用される複数の zone ステートメントが含まれます。

特定のクライアントの IP アドレスに一致する最初のステートメントが使用されるため、view ステートメントが一覧表示される順序が重要である点に注意してください。このトピックに関する詳細は、「複数表示」を参照してください。

17.2.1.3. コメントタグ

ステートメントのほかに、/etc/named.conf ファイルにはコメントも含まれています。コメントは named サービスには無視されますが、ユーザーに追加情報を提供する時に役に立ちます。以下に有効なコメントタグを示します。

//

// 文字の後のテキストはいずれもその行末までコメントとみなされます。以下に例を示します。

```
notify yes;  // notify all secondary nameservers
```

#

文字の後のテキストはいずれもその行末までコメントとみなされます。以下に例を示します。

```
notify yes;  # notify all secondary nameservers
```

/* and */

/* と */ によって囲まれたテキストのブロックはコメントとみなされます。以下に例を示します。

```
notify yes;  /* notify all secondary nameservers */
```

17.2.2. ゾーンファイルの編集

「ネームサーバーゾーン」 で説明しているように、インベントリーファイルには namespace についての情報が含まれます。デフォルトでは、/var/ named / にある名前付きの作業ディレクトリーに格納され、各ゾーンファイルの名前は zone ステートメントの file オプションに従って名前が付けられます。通常、問題のドメインに関連する方法で、example.com.zone などのゾーンデータを含むファイルを特定します。

表17.5 名前付きサービスゾーンファイル

パス	説明
/var/named/	named サービスの作業ディレクトリーです。ネームサーバーにはこのディレクトリーに書き込む許可がありません。
/var/named/slaves/	セカンダリーゾーンのディレクトリーです。このディレクトリーは named サービスによる書き込みが可能です。
/var/named/dynamic/	動的 DNS(DDNS)ゾーンや管理対象 DNSSEC キーなどの他のファイルのディレクトリー。このディレクトリーは named サービスによる書き込みが可能です。

パス	説明
/var/named/data/	様々な統計とデバッギングファイル用のディレクトリーです。このディレクトリーは named サービスによる書き込みが可能です。

ゾーンファイルはディレクティブとリソースの記録で構成されています。ディレクティブはネームサーバーに対してタスクを実行するか、または特別なセッティングをゾーンに適用するように指示し、リソースレコードはゾーンのパラメータを定義して識別子を個々のホストに割り当てます。ディレクティブはオプションですが、リソースレコードはゾーンにネームサービスを提供するために必須です。

ディレクティブとリソースレコードはすべて、個別の行で記入します。

17.2.2.1. 一般的なディレクティブ

ディレクティブはドル記号記号で始まり、その後にディレクティブの名前を付け、通常はファイルの一番上に表示されます。通常、ファイルの最上部に現れます。以下のディレクティブは一般的にゾーンファイルで使用されます。

$INCLUDE

$INCLUDE ディレクティブにより、それが出現する場所にもう 1 つのファイルを含めることができるため、他のゾーンセッティングは別個のゾーンファイルに保存できるようになります。

> **例17.7 $INCLUDE ディレクティブの使用**
>
> ```
> $INCLUDE /var/named/penguin.example.com
> ```

$ORIGIN

$ORIGIN ディレクティブを使うと、ホスト名だけの非完全修飾型の記録へドメイン名を追記できるようになります。デフォルトではゾーン名が使用されるので、/etc/named.conf 内でゾーンが指定されている場合は、このディレクティブの使用は不要です。

例17.8「$ORIGIN ディレクティブの使用」 では、末尾のピリオドで終了しないリソースレコードで使用される名前はすべて example.com が付けられます。

> **例17.8 $ORIGIN ディレクティブの使用**

```
$ORIGIN example.com.
```

$TTL

$TTL ディレクティブにより、ゾーン用のデフォルト TTL (Time to Live) 値をセットできるようになります。つまり、ゾーン記録が有効である時間の長さのセッティングです。各リソースレコードはそれ自身のTTL 値を含むことができるため、それがこのディレクティブを上書きします。

この値を増加させるとリモートのネームサーバーはより長い期間でゾーン情報をキャッシュ化できるようになり、ゾーンへのクエリー回数が減少し、リソースレコード変更の伝達に必要な時間を延長させることができます。

例17.9 $TTL ディレクティブの使用

```
$TTL 1D
```

17.2.2.2. 一般的なリソースレコード

以下のリソースレコードは一般的にゾーンファイル内で使用されます。

A

Address レコードは、名前に割り当てる IP アドレスを指定します。以下の形式を取ります。

```
hostname IN A IP-address
```

hostname の値ない場合、レコードは最後に指定された hostname を指します。

例17.10「A リソースレコードの使用」では、server1.example.com 用の要求は、10.0.1.3 または 10.0.1.5 を指しています。

例17.10 A リソースレコードの使用

```
server1  IN  A  10.0.1.3
         IN  A  10.0.1.5
```

CNAME

Canonical Name (別名) レコードはある名前を別の名前にマッピングします。このため、このタイプのレコードは、エイリアスレコードと呼ばれることもあります。以下の形式を取ります。

```
alias-name IN CNAME real-name
```

CNAME レコードは Web サーバー用の www のように、共通の命名基準を使用するサービスを指すために最も一般的に使用されます。しかし、それらの使用については複数の制限があります。

- CNAME レコードは他の CNAME レコードを指してはいけません。これは主に無限のループの可能性を避けるためです。
- CNAME レコードには、他のリソースレコードタイプ（A、NS、pid など）を含めないでください。ゾーンが署名されている場合、唯一の例外は DNSSEC 関連のレコード（RRSIG、NSEC など）です。
- ホストの完全修飾ドメイン名(FQDN)をポイントする他のリソースレコード（NS、pid、PTR）は CNAME レコードを参照できません。

例17.11「CNAME リソースレコードの使用」 では、A レコードはホスト名を IP アドレスにバインドしますが、CNAME レコードは一般的に使用される www ホスト名をその IP アドレスに指定します。

例17.11 CNAME リソースレコードの使用

```
server1  IN  A      10.0.1.5
www      IN  CNAME  server1
```

MX

Mail Exchange レコードは、このゾーンで制御されている特定のネームスペースに送信されるメールの行き先を指定します。以下の形式を取ります。

```
IN MX preference-value email-server-name
```

email-server-name は完全修飾型ドメイン名 (FQDN) です。preference-value によってネームスペースのメールサーバーの数値ランキングが可能になり、一部のメールシステムに他のシステムよりも優先度を与えます。最小の preference-value を持つ MX リソースレコードが他よりも優先されます。しかし複数メールサーバーが同じ値を持つ可能性があり、その場合はメールトラフィックをサーバー間で均等に分配することになります。

例17.12「リソースレコードの使用」では、example.com ドメイン宛のメール受信時には最初の mail.example.com メールサーバーが mail2.example.com メールサーバーよりも優先されます。

例17.12 リソースレコードの使用

```
example.com.  IN  MX  10  mail.example.com.
              IN  MX  20  mail2.example.com.
```

NS

Nameserver レコードはある特定のゾーン用に正当なネームサーバーを表明します。以下の形式を取ります。

```
IN NS nameserver-name
```

nameserver-name は完全修飾型ドメイン名 (FQDN) である必要があります。ドメインに対して 2 つのネームサーバーが正当だとして一覧表示されている時には、これらのネームサーバーがセカンダリーネームサーバーであるか、またはその 1 つがプライマリーサーバーであるかどうかは重要でありません。両方とも正当と考慮されます。

例17.13 NS リソースレコードの使用

```
IN  NS  dns1.example.com.
IN  NS  dns2.example.com.
```

PTR

Pointer レコードはネームスペースの別の部分を指します。以下の形式を取ります。

```
last-IP-digit IN PTR FQDN-of-system
```

last-IP-digit ディレクティブは、IP アドレスの最後の番号で、FQDN-of-system は完全修飾ドメイン名(FQDN)です。

PTR レコードは、主に IP アドレスを特定の名前に戻すため、逆引き名前解決に使用されます。使用中の PTR レコードの例は、「逆引き名前解決ゾーンファイル」を参照してください。

SOA

Start of Authority レコードはネームスペースについての信頼できる重要な情報をネームサーバーに表明します。ディレクティブの後に配置されていて、ゾーンファイルでは最初のリソースレコードです。以下の形式を取ります。

```
@  IN  SOA  primary-name-server hostmaster-email (
      serial-number
      time-to-refresh
      time-to-retry
      time-to-expire
      minimum-TTL )
```

ディレクティブは以下の通りです。

- @ シンボルは $ORIGIN ディレクティブ (または$ORIGIN ディレクティブがセットされていない場合は、ゾーン名) をこのSOA リソースレコードで定義されたネームスペースとして配置します。
- primary-name-server ディレクティブは、このドメインの正式なプライマリーネームサーバーのホスト名です。
- hostmaster-email ディレクティブは、ネームスペースに関して連絡する相手のメールです。
- serial-number ディレクティブは、named サービスがゾーンを再ロードする時間であることを示すためにゾーンファイルが変更される度に増加する数値です。
- time-to-refresh ディレクティブは、ゾーンに対して変更がなされたかどうかをプライマリーネームサーバーに尋ねるまで待機する時間の長さを決定するためにセカンダリーネームサーバーが使用する数値です。
- time-to-retry ディレクティブは、プライマリーネームサーバーが応答しない事態にリフレッシュ要求を出すまで待機する時間の長さを決定するためにセカンダリーネームサー

バーによって使用される数値です。time-to-expire ディレクティブ内で指定された時間が経過するまでに、プライマリーネームサーバーがリフレッシュ要求に応答しない場合は、セカンダリーサーバーはそのネームスペースに関する要求での権威としての応答を停止します。

- BIND 4 と 8 では、minimum-TTL ディレクティブは他のネームサーバーがゾーンの情報をキャッシュ化する時間の長さになります。BIND 9 では、これは否定的な回答がキャッシュ化される時間の長さを定義します。ネガティブな応答のキャッシュは最大 3 時間（つまり 3H）に設定できます。

BIND の設定時には、すべての時間は秒で指定されます。しかし、秒以外の時間単位を指定するのに短縮形を使用することができます。たとえば、分 (M)、時間 (H)、日 (D)、および週 (W) です。表17.6「秒表示とその他の時間単位」は、秒単位で、同等の時間 (秒単位) を別の形式で示しています。

表17.6 秒表示とその他の時間単位

秒	他の時間単位
60	1M
1800	30M
3600	1H
10800	3H
21600	6H
43200	12H
86400	1D
259200	3D
604800	1W
31536000	365D

例17.14 SOA リソースレコードの使用

```
@  IN  SOA  dns1.example.com.  hostmaster.example.com. (
        2001062501  ; serial
        21600       ; refresh after 6 hours
```

```
        3600       ; retry after 1 hour
        604800     ; expire after 1 week
        86400 )    ; minimum TTL of 1 day
```

17.2.2.3. コメントタグ

リソースレコードとディレクティブの他にも、ゾーンファイルもコメントを格納することができます。コメントは named サービスでは無視されますが、ユーザーに追加情報を提供する際に便利です。セミコロンの後の行末までのテキストはすべてコメントとみなされます。以下に例を示します。

```
604800  ; expire after 1 week
```

17.2.2.4. 使用例

下記の例は、ゾーンファイルの基本的使用法を示したものです。

17.2.2.4.1. 単純なゾーンファイル

例17.15「単純なゾーンファイル」では、標準のディレクティブとSOA 値の使用を提示しています。

例17.15 単純なゾーンファイル

```
$ORIGIN example.com.
$TTL 86400
@       IN  SOA  dns1.example.com.  hostmaster.example.com. (
          2001062501  ; serial
          21600       ; refresh after 6 hours
          3600        ; retry after 1 hour
          604800      ; expire after 1 week
          86400 )     ; minimum TTL of 1 day
;
;
        IN  NS     dns1.example.com.
        IN  NS     dns2.example.com.
dns1      IN  A      10.0.1.1
        IN  AAAA   aaaa:bbbb::1
dns2      IN  A      10.0.1.2
        IN  AAAA   aaaa:bbbb::2
;
;
@       IN  MX     10  mail.example.com.
        IN  MX     20  mail2.example.com.
mail      IN  A      10.0.1.5
        IN  AAAA   aaaa:bbbb::5
```

```
mail2    IN  A      10.0.1.6
         IN  AAAA   aaaa:bbbb::6
;
;
; This sample zone file illustrates sharing the same IP addresses
; for multiple services:
;
services  IN  A      10.0.1.10
          IN  AAAA   aaaa:bbbb::10
          IN  A      10.0.1.11
          IN  AAAA   aaaa:bbbb::11

ftp       IN  CNAME  services.example.com.
www       IN  CNAME  services.example.com.
;
;
```

この例では、権威ネームサーバーは dns1 .example.com および dns2.example.com として設定され、それぞれ A レコードを使用して 10. 0.1.1 および 10.0.1.2 IP アドレスに関連付けられます。

A レコードで設定されたメールサーバーは A レコードを介して mail および mail2 をポイントします。これらの名前はトレーリングピリオドで終了していないため、$ORIGIN ドメインがその後に配置されており、それらを mail.example.com および mail2.example.com に広げています。

www.example.com (WWW) などの標準の名前で利用可能なサービスは、CNAME レコードを使用して適切なサービスを指すようにしてあります。

このゾーンファイルは、以下のような zone ステートメントが /etc/named.conf ファイルに追加される場合に使用されます。

```
zone "example.com" IN {
  type master;
  file "example.com.zone";
  allow-update { none; };
};
```

17.2.2.4.2. 逆引き名前解決ゾーンファイル

逆引き名前解決のゾーンファイルは、特定の名前空間の IP アドレスを完全修飾ドメイン名(FQDN)に変換するために使用されます。標準のゾーンファイルと非常に似ていますが、PTR リソースレコードは、例17.16「逆引き名前解決ゾーンファイル」 に示されているように IP アドレスを完全修飾ドメイン名にリンクするために使用されます。

例17.16 逆引き名前解決ゾーンファイル

```
$ORIGIN 1.0.10.in-addr.arpa.
$TTL 86400
@  IN  SOA  dns1.example.com.  hostmaster.example.com. (
       2001062501  ; serial
       21600       ; refresh after 6 hours
       3600        ; retry after 1 hour
       604800      ; expire after 1 week
       86400 )     ; minimum TTL of 1 day
;
@  IN  NS   dns1.example.com.
;
1  IN  PTR  dns1.example.com.
2  IN  PTR  dns2.example.com.
;
5  IN  PTR  server1.example.com.
6  IN  PTR  server2.example.com.
;
3  IN  PTR  ftp.example.com.
4  IN  PTR  ftp.example.com.
```

この例では、IP アドレス 10.0.1.1 から 10.0.1.6 までは対応する完全修飾ドメイン名を参照します。

このゾーンファイルは、以下のような zone ステートメントが /etc/named.conf ファイルに追加される場合に使用されます。

```
zone "1.0.10.in-addr.arpa" IN {
  type master;
  file "example.com.rr.zone";
  allow-update { none; };
};
```

ゾーン名以外は、この例と標準の zone ステートメントにはほとんど違いがありません。逆引き名前解決ゾーンには、IP アドレスの最初の 3 つのブロックと、.in-addr.arpa が必要なことに注意してください。これにより、逆引き名前解決のゾーンファイルで使用される IP 番号の単一ブロックをゾーンに関連付けることができます。

17.2.3. rndc ユーティリティーの使用

rndc ユーティリティーは、ローカルとリモートマシンの両方から named サービスの管理を可能にするコマンドラインツールです。以下のような使用法になります。

```
rndc [option...] command [command-option]
```

17.2.3.1. ユーティリティーの設定

サービスへの不正アクセスを防ぐには、named が選択したポート（デフォルトでは 953 ）をリッスンするように設定し、サービスと rndc ユーティリティーの両方で同じキーを使用する必要があります。

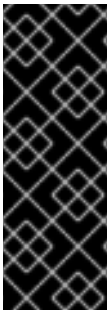
表17.7 関連ファイル

パス	説明
/etc/named.conf	named サービス用のデフォルト設定ファイル
/etc/rndc.conf	rndc ユーティリティー用のデフォルト設定ファイル
/etc/rndc.key	デフォルトキーの場所

rndc 設定は /etc/rndc.conf に配置されています。ファイルが存在しない場合は、ユーティリティーは、rndc-confgen -a コマンドを使用したインストールプロセス中に自動的に生成される、/etc/rndc.key にあるキーを使用します。

named サービスは、「その他のステートメントタイプ」 に記載されているように、/etc/named.conf 設定ファイルの control ステートメントを使用して設定されます。このステートメントが存在しない限り、ループバックアドレス(127.0.0.1)からの接続のみが許可され、/etc/rndc.key にあるキーが使用されます。

このトピックの詳細は、「その他のリソース」 に記載されている man ページと 『BIND 9 Administrator Reference Manual』 を参照してください。



適切なパーミッションの設定

特権のないユーザーがサービスに制御コマンドを送信しないようにするには、root のみが /etc/rndc.key ファイルを読み取れるようにします。

```
~]# chmod o-rwx /etc/rndc.key
```

17.2.3.2. サービスステータスの確認

named サービスの現在の状態をチェックするには、以下のコマンドを使用します。

```
~]# rndc status
version: 9.7.0-P2-RedHat-9.7.0-5.P2.el6
```

```
CPUs found: 1
worker threads: 1
number of zones: 16
debug level: 0
xfers running: 0
xfers deferred: 0
soa queries in progress: 0
query logging is OFF
recursive clients: 0/0/1000
tcp clients: 0/100
server is up and running
```

17.2.3.3. 設定とゾーンのリロード

設定ファイルとゾーンの両方をリロードするには、シェルプロンプトで以下を入力します。

```
~]# rndc reload
server reload successful
```

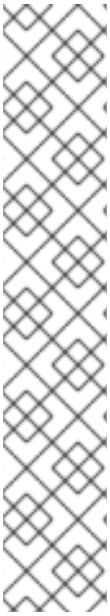
これがゾーンをリロードすると同時に以前にキャッシュ化した応答を維持するため、すべての保存済みの名前解決を消失することなくゾーンファイルを変更できます。

単独ゾーンをリロードするには、reload コマンドの後にその名前を指定します。例を示します。

```
~]# rndc reload localhost
zone reload up-to-date
```

最後に、設定ファイルと、新規に追加されたゾーンのみをリロードするには、以下を入力します。

```
~]# rndc reconfig
```



動的 DNS を使用したゾーンの変更

動的 DNS(DDNS)を使用するゾーンを手動で変更する場合は、最初に **freeze** コマンドを実行してください。

```
~]# rndc freeze localhost
```

完了したら、**thaw** コマンドを実行して **DDNS** を再度許可し、ゾーンを再読み込みします。

```
~]# rndc thaw localhost
The zone reload and thaw was successful.
```

17.2.3.4. ゾーンキーの更新

DNSSEC キーを更新してゾーンに署名するには、**sign** コマンドを使用します。以下に例を示します。

```
~]# rndc sign localhost
```

上記のコマンドでゾーンに署名するには、**zone** ステートメント内で **auto-dnssec** オプションを **maintain** に設定する必要があることに注意してください。例を示します。たとえば、以下のようになります。

```
zone "localhost" IN {
  type master;
  file "named.localhost";
  allow-update { none; };
  auto-dnssec maintain;
};
```

17.2.3.5. DNSSEC 検証の有効化

DNSSEC 検証を有効にするには、シェルプロンプトで以下を入力します。

```
~]# rndc validation on
```

同様に、このオプションを無効にするには、以下を入力します。

```
~]# rndc validation off
```

/etc/named.conf でこの オプション を設定する方法は、「一般的なステートメントのタイプ」 で説明されている options ステートメントを参照してください。

17.2.3.6. クエリーロギングの有効化

クエリーロギングを有効（または、無効な場合）にクエリーロギングするには、以下のコマンドを実行します。

```
~]# rndc querylog
```

現在の設定を確認するには、「サービスステータスの確認」 の説明に従って status コマンドを使用します。

17.2.4. dig ユーティリティーの使用

dig ユーティリティーは、DNS ルックアップを実行してネームサーバー設定をデバッグできるようにするコマンドラインツールです。これは通常、以下のように使用されます。

```
dig [@server] [option...] name type
```

一般的な タイプの一覧は、「一般的なリソースレコード」 を参照してください。

17.2.4.1. ネームサーバーのルックアップ

特定のドメイン用にネームサーバーをルックアップするには、以下の形式でコマンドを使用します。

```
dig name NS
```

例17.17「ネームサーバールックアップのサンプル」では、dig ユーティリティーが example.com 用のネームサーバーを表示するために使用されています。

例17.17 ネームサーバールックアップのサンプル

```
~]$ dig example.com NS

; <<>> DiG 9.7.1-P2-RedHat-9.7.1-2.P2.fc13 <<>> example.com NS
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 57883
```

```
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;example.com.                IN      NS

;; ANSWER SECTION:
example.com.          99374   IN      NS      a.iana-servers.net.
example.com.          99374   IN      NS      b.iana-servers.net.

;; Query time: 1 msec
;; SERVER: 10.34.255.7#53(10.34.255.7)
;; WHEN: Wed Aug 18 18:04:06 2010
;; MSG SIZE  rcvd: 77
```

17.2.4.2. IP アドレスのルックアップ

特定のドメインに割り当てられた IP アドレスを検索するには、以下の形式でコマンドを使用します。

```
dig name A
```

例17.18「IP アドレス検索のサンプル」 では、dig ユーティリティーを使用して example.com の IP アドレスを表示します。

例17.18 IP アドレス検索のサンプル

```
~]$ dig example.com A

; <<>> DiG 9.7.1-P2-RedHat-9.7.1-2.P2.fc13 <<>> example.com A
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 4849
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 0

;; QUESTION SECTION:
;example.com.                IN      A

;; ANSWER SECTION:
example.com.          155606  IN      A       192.0.32.10

;; AUTHORITY SECTION:
example.com.          99175   IN      NS      a.iana-servers.net.
example.com.          99175   IN      NS      b.iana-servers.net.

;; Query time: 1 msec
;; SERVER: 10.34.255.7#53(10.34.255.7)
;; WHEN: Wed Aug 18 18:07:25 2010
;; MSG SIZE  rcvd: 93
```

17.2.4.3. ホスト名の検索

特定の IP アドレスのホスト名を検索するには、以下の形式でコマンドを使用します。

```
dig -x address
```

例17.19「ホスト名の検索例」では、dig ユーティリティーを使用して 192.0.32.10 に割り当てられたホスト名を表示しています。

例17.19 ホスト名の検索例

```
~]$ dig -x 192.0.32.10

; <<>> DiG 9.7.1-P2-RedHat-9.7.1-2.P2.fc13 <<>> -x 192.0.32.10
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 29683
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 5, ADDITIONAL: 6

;; QUESTION SECTION:
;10.32.0.192.in-addr.arpa.      IN      PTR

;; ANSWER SECTION:
10.32.0.192.in-addr.arpa. 21600 IN      PTR     www.example.com.

;; AUTHORITY SECTION:
32.0.192.in-addr.arpa.  21600   IN      NS      b.iana-servers.org.
32.0.192.in-addr.arpa.  21600   IN      NS      c.iana-servers.net.
32.0.192.in-addr.arpa.  21600   IN      NS      d.iana-servers.net.
32.0.192.in-addr.arpa.  21600   IN      NS      ns.icann.org.
32.0.192.in-addr.arpa.  21600   IN      NS      a.iana-servers.net.

;; ADDITIONAL SECTION:
a.iana-servers.net.     13688   IN      A       192.0.34.43
b.iana-servers.org.     5844    IN      A       193.0.0.236
b.iana-servers.org.     5844    IN      AAAA    2001:610:240:2::c100:ec
c.iana-servers.net.     12173   IN      A       139.91.1.10
c.iana-servers.net.     12173   IN      AAAA    2001:648:2c30::1:10
ns.icann.org.           12884   IN      A       192.0.34.126

;; Query time: 156 msec
;; SERVER: 10.34.255.7#53(10.34.255.7)
;; WHEN: Wed Aug 18 18:25:15 2010
;; MSG SIZE  rcvd: 310
```

17.2.5. BIND の高度な機能

ほとんどの BIND 実装では、named のみを使用して名前解決サービスを提供したり、特定ドメイン用の権威として機能させます。ただし、BIND バージョン 9 には、よりセキュアで効率的な DNS サー

ビスを可能にする数多くの高度な機能があります。

この機能がサポートされていることの確認

DNSSEC、TSIG、IXFR (増分ゾーン転送) などの高度な機能の使用を試みる前に、特に BIND の古いバージョンや BIND 以外のサーバーを使用している場合は、その特定の機能がネットワーク環境内のすべてのネームサーバーでサポートされていることを確認してください。

上記のすべての機能の詳細は、「インストールされているドキュメント」 に記載されている『BIND 9 管理者リファレンスマニュアル』 を参照してください。

17.2.5.1. 複数表示

オプションとして、リクエストが発信されたネットワークに応じて異なる情報をクライアントに提供することができます。これは主に、ローカルネットワーク外のクライアントからの機密 DNS エントリーを拒否するために使用されます。一方、ローカルネットワーク内のクライアントからのクエリーを許可します。

複数表示を設定するには、view ステートメントを /etc/named.conf 設定ファイルに追加します。match-clients オプションを使用して IP アドレスまたはネットワーク全体に一致し、特別なオプションおよびゾーンデータを指定します。

17.2.5.2. IXFR (Incremental Zone Transfers 差分ゾーン転送)

Incremental Zone Transfers 差分ゾーン転送 (IXFR) により、セカンダリーネームサーバーはプライマリーネームサーバー上で修正されたゾーンの更新部分だけをダウンロードすることができます。標準の転送プロセスに比較すると、これが通知と更新のプロセスを格段に効率的にします。

IXFR は、動的な更新を使用してマスターゾーンレコードに変更を加える時にのみ使用可能なことに注意して下さい。ゾーンファイルを手動の編集で変更する場合は、Automatic Zone Transfer 自動ゾーン転送 (AXFR) が使用されます。

17.2.5.3. Transaction SIGnatures トランザクション署名 (TSIG)

トランザクション SIGnatures (TSIG)は、転送を許可する前に共有秘密鍵がプライマリーおよびセカンダリーネームサーバーの両方に存在することを確認します。これにより、攻撃者はゾーンを転送する際に IP アドレスにアクセスするだけでなく、秘密鍵を把握する必要があるため、標準の IP アドレスベースの転送方法が向上します。

バージョン 9 以降は、BIND は TKEY もサポートします。これはゾーン転送を認証するもう 1 つの共有秘密鍵メソッドです。

転送のセキュリティー保護

セキュアでないネットワーク上で通信する場合は、IP アドレスベースの認証のみに依存しないでください。

17.2.5.4. DNSSEC (DNS Security Extensions)

DNS(Domain Name System Security Extensions)は、DNS データの送信元認証、存在拒否、およびデータの整合性を提供します。特定のドメインがセキュアであるとマークされると、検証に失敗した各リソースレコードに SERFVAIL 応答が返されます。

DNSSEC 署名ドメインまたは DNSSEC 対応リゾルバーをデバッグするには、「dig ユーティリティーの使用」 の説明に従って dig ユーティリティーを使用できます。便利なオプションは +dnssec (DNSSEC OK ビット)、+cd (応答の検証に再帰的ネームサーバー)、および +bufsize=512 (パケットサイズを 512B に変更してファイアウォール経由で取得する。

17.2.5.5. インターネットプロトコルバージョン 6 (IPv6)

表17.3「一般的に使用されるオプション」で説明されているように、Internet Protocol version 6 (IPv6)は AAAA リソースレコードと listen-on-v6 ディレクティブを使用してサポートされます。

17.2.6. 回避すべき一般的な間違い

ネームサーバー設定時にユーザーが一般的な間違いを回避するためのアドバイス一覧を以下に示します。

セミコロンと弓形括弧の正しい使用

/etc/named.conf ファイル内のセミコロンの欠如や、不一致な弓形括弧は named サービスの開始を阻止してしまいます。

期間を正しく使用する

ゾーンファイル内では、ドメイン名の末尾のピリオドは完全修飾型ドメイン名を示します。省略した場合には、named サービスがゾーンの名前または $ORIGIN の値を追加して、完了します。

ゾーンファイルを編集する時のシリアル番号増加

シリアル番号が増加していない場合、プライマリーネームサーバーは正しくて新しい情報を持ちますが、セカンダリーネームサーバーは決して変更を通知されません。そのため、そのゾーンのデータをリフレッシュする試みをしません。

ファイヤーウォールの設定

ファイヤーウォールが、named サービスから他のネームサーバーへの接続をブロックしている場合は、ファイヤーウォールの設定変更が推奨されます。

固定 UDP ソースポートの使用を回避

DNS セキュリティーの最近の調査によると、DNS クエリーに固定の UDP ソースポートを使用することは、攻撃者がキャッシュポイズリング攻撃をより簡単に実行できるセキュリティー脆弱性があります。これを回避するには、ファイアウォールを、ランダムな UDP ソースポートからのクエリーを許可するように設定します。

17.2.7. その他のリソース

以下の資料は、BIND に関するその他のリソースを提供します。

17.2.7.1. インストールされているドキュメント

BIND は、多種多様なトピックを網羅した広範囲に及ぶインストール済みのドキュメントを特徴としています。各ドキュメントはその議題のディレクトリー内に配置されています。以下の各項目には、version の部分をシステム上にインストールしてある bind パッケージのバージョンに入れ替えてください。

/usr/share/doc/bind-version/

最新のドキュメンテーションを格納しているメインのディレクトリーです。

/usr/share/doc/bind-version/arm/

『BIND リソース要件の詳細、さまざまな種類のネームサーバーの設定方法、負荷分散の実行方法、その他の高度なトピックなど、HTML および SGML 形式で BIND 9 管理者リファレンスマニュアル』 を含むディレクトリー。BIND のほとんどの新しいユーザーでは、開始する最適な場所になります。

/usr/share/doc/bind-version/draft/

DNS サービスに関連する問題を確認し、それらに対応する方法を提案する、さまざまな技術ドキュメントを含むディレクトリー。

/usr/share/doc/bind-version/misc/

特定の高度な問題に対処するために設計されたディレクトリー。BIND バージョン 8 のユーザーは、BIND 9 に 移行する際に行う必要がある特定の変更について移行 ドキュメントを参照してください。options ファイルには、/etc/named.conf で使用される BIND 9 に実装されたすべてのオプションが一覧表示されます。

/usr/share/doc/bind-version/rfc/

BIND に関連するすべての RFC ドキュメントを提供するディレクトリー。

BIND に関連するさまざまなアプリケーションおよび設定ファイル用の man ページも多数あります。

man rndc

使用に関する全ドキュメントを含む rndc の man ページです。

man named

BIND nameserver デーモンの制御に使用できるさまざまな引数のドキュメントを含む 名前の man ページです。

man lwresd

軽量リゾルバーデーモンとその使用方法に関する完全なドキュメントを含む lwresd の man ページです。

man named.conf

この man ページには、named 設定ファイル内で利用可能なオプションの総合的一覧があります。

man rndc.conf

この man ページには、rndc 設定ファイル内で利用可能なオプションの総合的一覧があります。

17.2.7.2. 便利な Web サイト

http://www.isc.org/software/bind

BIND プロジェクトのホームページには、現在のリリースに関する情報と 『BIND 9 Administrator Reference Manual』 の PDF バージョンが含まれます。

17.2.7.3. 関連書籍

『DNS および BIND』 (Albitz and Cricket Liu; O'Reilly & Associates)

共通設定オプションと Esoteric BIND 設定オプションの両方を説明し、DNS サーバーのセキュリティーを保護するストラテジーを提供する一般的なリファレンスです。

『The Concise Guide to DNS and BIND』 by Nicolai Langfeldt; Que

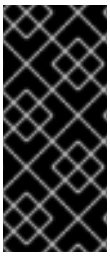
タスク指向の技術トピックで、複数のネットワークサービスと BIND 間の接続を確認します。

第18章 WEB サーバー

HTTP (Hypertext Transfer Protocol)サーバーまたは Web サーバー は、Web 経由でクライアントにコンテンツを提供するネットワークサービスです。これは通常 Web ページを指しますが、他のドキュメントも当てはまります。

Red Hat Enterprise Linux 6 で利用可能な Web サーバーは以下のとおりです。

- Apache HTTP サーバー
- nginx



重要

nginx Web サーバーは、Red Hat Enterprise Linux 6 の Software Collection としてのみ利用できます。nginx へのアクセスや Software Collections の使用などの詳細は、『Red Hat Software Collections リリースノート』を参照してください。

18.1. APACHE HTTP サーバー

本セクションでは、Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;Linux 6nbsp;6 に含まれる Apache HTTP Server 2.2 （ Apache Software Foundation により開発された堅牢な完全機能のオープンソース Web サーバー）を説明します。ここでは、httpd サービスの基本設定について説明し、サーバーモジュールの追加、仮想ホストの設定、セキュアな HTTP サーバーの設定など、高度なトピックを説明します。

Apache HTTP Server 2.2 とバージョン 2.0 には重要な違いがあり、Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux の以前のリリースからアップグレードする場合は、適切に httpd サービス設定を更新する必要があります。本セクションでは、新たに追加された機能の一部、重要な変更の概要、古い設定ファイルの更新について説明します。

18.1.1. 新機能

Apache HTTP Server バージョン 2.2 では、以下の機能強化が追加されました。

- mod_cache および mod_disk_cache のキャッシュモジュール（つまり mod_cache および mod_disk_cache ）が改善されました。

- プロキシー負荷分散、つまり mod_proxy_balancer モジュールのサポート。

- 32 ビットアーキテクチャーにおける大規模なファイルのサポートにより、Web サーバーは 2GB を超えるファイルを処理できます。

- 認証および承認サポートの新しい構造。以前のバージョンで提供される認証モジュールを置き換えます。

18.1.2. 主な変更点

バージョン 2.0 以降、デフォルトの httpd サービス設定にいくつかの変更が加えられました。

- mod_cern_meta および mod_asis はデフォルトでロードされなくなりました。

- 以下のモジュールは、デフォルトで mod_ext_filter によりロードされます。

18.1.3. 設定の更新

Apache HTTP Server バージョン 2.0 から設定ファイルを更新するには、以下の手順を行います。

1. モジュールは変更されている可能性があるため、すべてのモジュール名が正しいことを確認してください。名前が変更された各モジュールの LoadModule ディレクティブを調整します。

2. サードパーティーのモジュールは、読み込みを試みる前にすべて再コンパイルをします。これは一般的に認証と権限付与のモジュールに該当します。

3. mod_userdir モジュールを使用する場合は、ディレクトリー名（通常は public_html）を示す UserDir ディレクティブが指定されていることを確認してください。

4. Apache HTTP Secure Server を使用する場合は、Secure Sockets Layer (SSL) プロトコルの有効化に関する重要な情報について 「mod_ssl モジュールの有効化」 を参照してください。

以下のコマンドを使用すると、設定エラーを確認できます。

```
~]# service httpd configtest
Syntax OK
```

Apache HTTP Server 設定をバージョン 2.0 から 2.2 にアップグレードする方法は、を参照してください http://httpd.apache.org/docs/2.2/upgrading.html。

18.1.4. httpd サービスの実行

本セクションでは、Apache HTTP Server の起動、停止、再起動、および現在のステータスの確認を行う方法を説明します。httpd サービスを使用するには、httpd がインストールされていることを確認してください。これを行うには、次のコマンドを使用します。

```
~]# yum install httpd
```

ランレベルの概念と Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux でシステムサービスを管理する方法は、12章サービスおよびデーモン を参照してください。

18.1.4.1. サービスの起動

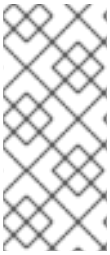
httpd サービスを実行する場合は、root で次のコマンドを実行します。

```
~]# service httpd start
Starting httpd:                                [  OK  ]
```

システムの起動時にサービスを自動的に起動するようにするには、以下のコマンドを使用します。

```
~]# chkconfig httpd on
```

これにより、ランレベル 2、3、4、および 5 に対してサービスが有効になります。または、「サービスの有効化および無効化」 の説明に従って Service Configuration ユーティリティーを使用できます。



セキュアなサーバーの使用

Apache HTTP サーバーをセキュアサーバーとして実行している場合は、暗号化したプライベートの SSL キーを使用すると、マシンが起動した後にパスワードが要求される可能性があります。

18.1.4.2. サービスの停止

実行中の httpd サービスを停止するには、root で次のコマンドを実行します。

```
~]# service httpd stop
Stopping httpd:                                [  OK  ]
```

システムの起動時にサービスが自動的に起動しないようにするには、以下を入力します。

```
~]# chkconfig httpd off
```

これにより、すべてのランレベルでサービスが無効になります。または、「サービスの有効化および無効化」 の説明に従って Service Configuration ユーティリティーを使用できます。

18.1.4.3. サービスの再起動

実行中の httpd サービスを再起動する方法は、3 通りあります。

1. サービスを完全に再起動するには、root で次のコマンドを実行します。

```
~]# service httpd restart
Stopping httpd:                                [  OK  ]
Starting httpd:                                [  OK  ]
```

これにより、稼働中の httpd サービスが停止し、すぐに再起動します。このコマンドは、PHP など、動的に読み込んだモジュールをインストールまたは削除した後に使用します。

2. 設定のリロードだけを行うには、root で以下を入力します。

```
~]# service httpd reload
```

これにより、実行中の **httpd** サービスが設定ファイルを再読み込みします。現在処理中のリクエストはすべて割り込みされるため、クライアントのブラウザーはエラーメッセージを表示したり、ページの一部をレンダリングしたりする可能性があります。

3.

アクティブなリクエストに影響を与えずに設定を再読み込みするには、**root** で次のコマンドを実行します。

```
~]# service httpd graceful
```

これにより、実行中の **httpd** サービスが設定ファイルを再読み込みします。現在処理されているリクエストでは、古い設定が使用されます。

または、「サービスの起動、再起動、停止」 の説明に従って **Service Configuration** ユーティリティーを使用できます。

18.1.4.4. サービスステータスの確認

httpd サービスが実行していることを確認するには、シェルプロンプトで以下を入力します。

```
~]# service httpd status
httpd (pid 19014) is running...
```

または、「Service 設定ユーティリティーの使用」 の説明に従って **Service Configuration** ユーティリティーを使用できます。

18.1.5. 設定ファイルの編集

httpd サービスが起動すると、デフォルトでは、表18.1「httpd サービスの設定ファイル」 に記載されている場所から設定が読み込まれます。

表18.1 httpd サービスの設定ファイル

パス	説明
/etc/httpd/conf/httpd.conf	主要設定ファイル。
/etc/httpd/conf.d/	主要設定ファイル内に含まれている設定ファイル用の補助ディレクトリー。

デフォルト設定はほとんどの状況に適していますが、重要な設定オプションをいくつか知っておくと役に立ちます。変更を有効にするには、web サーバーを再起動する必要があることに注意してください。httpd サービスを再起動する方法は、「サービスの再起動」 を参照してください。

設定に誤りがないことを確認するには、シェルプロンプトで以下のコマンドを実行します。

```
~]# service httpd configtest
Syntax OK
```

ファイルの変更に失敗してもすぐにやり直せるように、ファイルを編集する前にコピーを作成しておくことをお勧めします。

18.1.5.1. 一般的な httpd.conf ディレクティブ

以下のディレクティブは、一般的に /etc/httpd/conf/httpd.conf 設定ファイルで使用されます。

<Directory>

<Directory> ディレクティブを使用すると、特定のディレクティブを特定のディレクトリーにのみ適用できます。以下の形式を取ります。

```
<Directory directory>
  directive
  …
</Directory>
```

ディレクトリー は、ローカルファイルシステム内の既存のディレクトリーへの完全パスまたはワイルドカード式のいずれかになります。

このディレクティブは、ScriptAlias で指定されたディレクトリー外にあるサーバー側のスクリプトに追加の cgi-bin ディレクトリーを設定するために使用できます。この場合、ExecCGI ディレクティブおよび AddHandler ディレクティブを指定し、ターゲットディレクトリーのパーミッションを正しく設定する必要があります（つまり 0755）。

例18.1 <Directory> ディレクティブの使用

```
<Directory /var/www/html>
  Options Indexes FollowSymLinks
  AllowOverride None
  Order allow,deny
  Allow from all
</Directory>
```

-

<IfDefine>

IfDefine ディレクティブでは、特定のパラメーターがコマンドラインに提供されている場合に限り、特定のディレクティブを使用できます。以下の形式を取ります。

```
<IfDefine [!]parameter>
  directive
  ...
</IfDefine>
```

パラメーター は、-Dパラメーター コマンドラインオプション（例： httpd -DEnableHome）を使用してシェルプロンプトに指定できます。オプションの感嘆符（つまり !）が存在する場合、囲まれたディレクティブはパラメーターが指定されていない 場合にのみ使用されます。

例18.2 <IfDefine> ディレクティブの使用

```
<IfDefine EnableHome>
  UserDir public_html
</IfDefine>
```

<IfModule>

&lt ;IfModule> ディレクティブを使用すると、特定のモジュールがロードされる場合にのみ特定のディレクティブを使用できます。以下の形式を取ります。

```
<IfModule [!]module>
  directive
  ...
</IfModule>
```

モジュール は名前またはファイル名で識別できます。オプションの感嘆符（つまり !）が存在する場合は、モジュールがロードされていない 場合にのみ囲まれたディレクティブが使用されます。

例18.3 <IfModule> ディレクティブの使用

```
<IfModule mod_disk_cache.c>
  CacheEnable disk /
  CacheRoot /var/cache/mod_proxy
</IfModule>
```

-

<location>

&lt ;Location> ディレクティブを使用すると、特定のディレクティブを特定の URL のみに適用できます。以下の形式を取ります。

```
<Location url>
  directive
  …
</Location>
```

url は、DocumentRoot ディレクティブ（ /server-infoなど）で指定されたディレクトリーに対する相対パス、または http://example.com/server-info などの外部 URL のいずれかになります。

例18.4 <Location> ディレクティブの使用

```
<Location /server-info>
  SetHandler server-info
  Order deny,allow
  Deny from all
  Allow from .example.com
</Location>
```

<Proxy>

&lt ;Proxy> ディレクティブを使用すると、特定のディレクティブをプロキシーサーバーにのみ適用できます。以下の形式を取ります。

```
<Proxy pattern>
  directive
  …
</Proxy>
```

パターン は外部 URL またはワイルドカード式（ http://example.com/*など）にすることができます。

例18.5 <Proxy> ディレクティブの使用

```
<Proxy *>
  Order deny,allow
  Deny from all
  Allow from .example.com
</Proxy>
```

<VirtualHost>

<VirtualHost> ディレクティブを使用すると、特定のディレクティブを特定の仮想ホストにのみ適用できます。以下の形式を取ります。

```
<VirtualHost address[:port]…>
  directive
  …
</VirtualHost>
```

アドレス は、表18.2「利用可能な <VirtualHost> オプション」 に記載されているように、IP アドレス、完全修飾ドメイン名、または特別な形式になります。

表18.2 利用可能な <VirtualHost> オプション

オプション	説明
*	すべての IP アドレスを表します。
default	一致しない IP アドレスを表します。

例18.6 <VirtualHost> ディレクティブの使用

```
<VirtualHost *:80>
  ServerAdmin webmaster@penguin.example.com
  DocumentRoot /www/docs/penguin.example.com
  ServerName penguin.example.com
  ErrorLog logs/penguin.example.com-error_log
  CustomLog logs/penguin.example.com-access_log common
</VirtualHost>
```

AccessFileName

AccessFileName ディレクティブを使用すると、各ディレクトリーのアクセス制御情報をカスタマイズするために使用するファイルを指定できます。以下の形式を取ります。

```
AccessFileName filename…
```

ファイル 名 は、要求されたディレクトリーで検索するファイルの名前です。デフォルトでは、サーバーは .htaccess を検索します。

セキュリティー上の理由から、ディレクティブの後に Files タグを付け、.ht で始まるファイルが Web クライアントによってアクセスされないようにします。これには、.htaccess および .htpasswd ファイルが含まれます。

例18.7 AccessFileName ディレクティブの使用

```
AccessFileName .htaccess

<Files ~ "^\.ht">
  Order allow,deny
  Deny from all
  Satisfy All
</Files>
```

アクション

Action ディレクティブにより、特定のメディアタイプが要求されるときに実行される CGI スクリプトを指定できます。以下の形式を取ります。

```
Action content-type path
```

content-type は、text/html、image/png、application/pdf などの有効な MIME タイプである必要があります。パス は既存の CGI スクリプトを参照し、DocumentRoot ディレクティブで指定されたディレクトリーに対して相対的である必要があります（例： /cgi-bin/process-image.cgi）。

例18.8 Action ディレクティブの使用

```
Action image/png /cgi-bin/process-image.cgi
```

AddDescription

AddDescription ディレクティブを使用すると、指定のファイルについてのサーバー生成ディレクトリー一覧に表示する簡単な説明を指定できます。以下の形式を取ります。

```
AddDescription "description" filename…
```

この 説明 は、二重引用符（つまり "）で囲まれた短いテキストである必要があります。ファイル名 に は、完全なファイル名、ファイル拡張子、またはワイルドカード式を使用できます。

例18.9 AddDescription ディレクティブの使用

```
AddDescription "GZIP compressed tar archive" .tgz
```

AddEncoding

AddEncoding ディレクティブを使用すると、特定のファイル拡張子のエンコーディングタイプを指定できます。以下の形式を取ります。

```
AddEncoding encoding extension…
```

エンコーディング は、**x-compress**、**x-gzip** などの有効な **MIME** エンコーディングである必要があります。拡張機能 は大文字と小文字を区別するファイル拡張子で、従来は先頭のドット（**.gz**など）で記述されます。

このディレクティブは通常、ダウンロード時に特定のファイルタイプを圧縮解除するように **Web** ブラウザーに指示するために使用されます。

例18.10 AddEncoding ディレクティブの使用

```
AddEncoding x-gzip .gz .tgz
```

AddHandler

AddHandler ディレクティブを使用すると、特定のファイル拡張子を選択したハンドラーにマップできます。以下の形式を取ります。

```
AddHandler handler extension…
```

ハンドラー は、以前に定義したハンドラーの名前である必要があります。拡張機能 は大文字と小文字を区別するファイル拡張子で、従来は先頭のドット（**.cgi**など）で記述されます。

このディレクティブは通常、**.cgi** 拡張子を持つファイルを **CGI** スクリプトとして処理するために使用されます。さらに、これは一般的に、サーバー解析された **HTML** および **image-map** ファイルを処理するためにも使用されます。

例18.11 AddHandler オプションの使用

```
AddHandler cgi-script .cgi
```

AddIcon

AddIcon ディレクティブを使用すると、サーバー生成ディレクトリーの一覧の特定ファイルに表示されるアイコンを指定できます。以下の形式を取ります。

```
AddIcon path pattern…
```

パス は既存のアイコンファイルを参照し、**DocumentRoot** ディレクティブで指定されたディレクトリーに対して相対的である必要があります（例： **/icons/folder.png**）。**パターン** は、ファイル名、ファイル拡張子、ワイルドカード式、または以下の表に記載されている特別な形式になります。

表18.3 利用可能な AddIcon オプション

オプション	説明
^^DIRECTORY^^	ディレクトリーを表します。
^^BLANKICON^^	空の行を表します。

例18.12 AddIcon ディレクティブの使用

```
AddIcon /icons/text.png .txt README
```

AddIconByEncoding

AddIconByEncoding ディレクティブを使用すると、サーバー生成ディレクトリー一覧で特定のエンコーディングタイプに対して表示されるアイコンを指定できます。以下の形式を取ります。

```
AddIconByEncoding path encoding…
```

パス は既存のアイコンファイルを参照し、**DocumentRoot** ディレクティブで指定されたディレクトリーに対して相対的である必要があります（例： **/icons/compressed.png**）。**エンコーディング** は、**x-compress**、**x-gzip** などの有効な **MIME** エンコーディングである必要があります。

例18.13 AddIconByEncoding ディレクティブの使用

```
AddIconByEncoding /icons/compressed.png x-compress x-gzip
```

AddIconByType

AddIconByType ディレクティブを使用すると、サーバー生成ディレクトリー一覧の特定のメディアタイプについて表示されるアイコンを指定できます。以下の形式を取ります。

```
AddIconByType path content-type…
```

パス は既存のアイコンファイルを参照し、**DocumentRoot** ディレクティブで指定されたディレクトリーに対して相対的である必要があります（例： **/icons/text.png**）。**content-type** は、有効な **MIME** タイプ（例： **text/html** または **image/png**）か、**text/ ***、**image/*** などのワイルドカード式のいずれかである必要があります。

例18.14 AddIconByType ディレクティブの使用

```
AddIconByType /icons/video.png video/*
```

AddLanguage

AddLanguage ディレクティブを使用すると、ファイル拡張子を特定の言語に関連付けることができます。以下の形式を取ります。

```
AddLanguage language extension…
```

言語 は、**cs**、**en**、**fr** などの有効な **MIME** 言語である必要があります。拡張機能 は大文字と小文字を区別するファイル拡張子で、従来は先頭のドット（**.cs**など）で記述されます。

このディレクティブは、クライアントの言語設定に基づいて複数の言語でコンテンツを提供する **Web** サーバーに特に便利です。

例18.15 AddLanguage ディレクティブの使用

```
AddLanguage cs .cs .cz
```

AddType

AddType ディレクティブを使用すると、特定のファイル拡張子のメディアタイプを定義または上書きできます。以下の形式を取ります。

```
AddType content-type extension…
```

content-type は、text/html、image/png 等の有効な MIME タイプである必要があります。拡張機能 は大文字と小文字を区別するファイル拡張子で、従来は先頭のドット（.csなど）で記述されます。

例18.16 AddType ディレクティブの使用

```
AddType application/x-gzip .gz .tgz
```

Alias

Alias ディレクティブを使用すると、DocumentRoot ディレクティブで指定されたデフォルトディレクトリー外のファイルおよびディレクトリーを参照できます。以下の形式を取ります。

```
Alias url-path real-path
```

url-path は DocumentRoot ディレクティブで指定されたディレクトリーに対して相対的である必要があります（例： /images/）。リアルタイムパス は、ローカルファイルシステム内のファイルまたはディレクトリーへの完全パスです。

このディレクティブの後には、通常、ターゲットディレクトリーにアクセスするための追加のパーミッションを持つ Directory タグが続きます。デフォルトでは、/icons/ エイリアスが作成され、/var/www/icons/ からのアイコンがサーバー生成ディレクトリー一覧に表示されます。

例18.17 Alias ディレクティブの使用

```
Alias /icons/ /var/www/icons/

<Directory "/var/www/icons">
  Options Indexes MultiViews FollowSymLinks
  AllowOverride None
  Order allow,deny
  Allow from all
<Directory>
```

許可

Allow ディレクティブでは、指定のディレクトリーにアクセスするパーミッションがあるクライアントを指定できます。以下の形式を取ります。

> *Allow from client…*

クライアント は、ドメイン名、IP アドレス（フルおよびパーシャルの両方）、ネットワーク/ネットマスク のペア、またはすべてのクライアントの すべて になります。

> ***例18.18 Allow ディレクティブの使用***
>
> > *Allow from 192.168.1.0/255.255.255.0*

AllowOverride

AllowOverride ディレクティブを使用すると、.htaccess ファイルのディレクティブがデフォルト設定を上書きできます。以下の形式を取ります。

> ***AllowOverride** type…*

この タイプ は、表18.4「利用可能な AllowOverride オプション」 で説明されているように、利用可能なグルーピングオプションの 1 つである必要があります。

表18.4 利用可能な AllowOverride オプション

オプション	説明
すべて	.htaccess のすべてのディレクティブは、以前の設定を上書きすることができます。
なし	.htaccess のディレクティブは、以前の設定を上書きすることはできません。
AuthConfig	AuthName、AuthType、または Require などの承認ディレクティブを使用できます。
FileInfo	DefaultType、RequestHeader、または RewriteEngine、および Action ディレクティブなどのファイルタイプ、メタデータ、および mod_rewrite ディレクティブを使用できます。

オプション	説明
Indexes	AddDescription、AddIcon、または FancyIndexing などのディレクトリーインデックスディレクティブの使用を許可します。
制限	ホストアクセスディレクティブを許可します。つまり、Allow、Deny、および Order を使用できます。
options [=オプション,...]	Options ディレクティブを使用できます。さらに、このディレクティブを使用して設定できるオプションをカスタマイズするオプションのコンマ区切りリストを指定することもできます。

例18.19 AllowOverride ディレクティブの使用

```
AllowOverride FileInfo AuthConfig Limit
```

BrowserMatch

BrowserMatch ディレクティブを使用すると、クライアントの Web ブラウザータイプに基づいてサーバーの動作を変更できます。以下の形式を取ります。

```
BrowserMatch pattern variable…
```

この パターン は、User-Agent HTTP header フィールドに一致する正規表現です。変数 は、header フィールドがパターンと一致する場合に設定される環境変数です。

デフォルトでは、このディレクティブは既知の問題のある特定のブラウザーへの接続を拒否するために使用され、これらのアクションで問題が発生することがわかっているブラウザーの keepalives および HTTP ヘッダーのフラッシュを無効にします。

例18.20 BrowserMatch ディレクティブの使用

```
BrowserMatch "Mozilla/2" nokeepalive
```

CacheDefaultExpire

CacheDefaultExpire オプションでは、有効期限のないドキュメント、または最後の変更日がないドキュメントをキャッシュする期間を設定できます。以下の形式を取ります。

```
CacheDefaultExpire time
```

時間 は秒単位で指定します。デフォルトのオプションは 3600 （1 時間）です。

例18.21 CacheDefaultExpire ディレクティブの使用

```
CacheDefaultExpire 3600
```

CacheDisable

CacheDisable ディレクティブを使用すると、特定の URL のキャッシュを無効にできます。以下の形式を取ります。

```
CacheDisable path
```

パス は DocumentRoot ディレクティブで指定されたディレクトリーに対して相対的である必要があります（例： /files/）。

例18.22 CacheDisable ディレクティブの使用

```
CacheDisable /temporary
```

CacheEnable

CacheEnable ディレクティブを使用すると、特定の URL に使用するキャッシュタイプを指定できます。以下の形式を取ります。

```
CacheEnable type url
```

表18.5「利用可能なキャッシュタイプ」 の説明に従って、タイプ は有効なキャッシュタイプである必要があります。url には、DocumentRoot ディレクティブ（例： /images/）で指定されたディレクトリーに対する相対パス、プロトコル（ ftp:// など）、または http://example.com/ などの外部 URL を使用できます。

表18.5 利用可能なキャッシュタイプ

タイプ	説明
mem	メモリーベースのストレージマネージャー。
disk	ディスクベースのストレージマネージャー。
fd	ファイル記述子キャッシュ。

例18.23 CacheEnable ディレクティブの使用

```
CacheEnable disk /
```

CacheLastModifiedFactor

CacheLastModifiedFactor ディレクティブでは、有効期限が指定されていないドキュメントをキャッシュする時間をカスタマイズできますが、最終変更日に関する情報が提供されます。以下の形式を取ります。

```
CacheLastModifiedFactor number
```

この 数 は、ドキュメントの最終更新以降に渡された時間を掛けて乗算するために使用されます。デフォルトのオプションは **0.1** （10 分の 1）です。

例18.24 CacheLastModifiedFactor ディレクティブの使用

```
CacheLastModifiedFactor 0.1
```

CacheMaxExpire

CacheMaxExpire ディレクティブを使用すると、ドキュメントをキャッシュする最大時間を指定できます。以下の形式を取ります。

```
CacheMaxExpire time
```

時間 は秒単位で指定します。デフォルトのオプションは **86400** （1 日）です。

例18.25 CacheMaxExpire ディレクティブの使用

```
CacheMaxExpire 86400
```

CacheNegotiatedDocs

CacheNegotiatedDocs ディレクティブを使用すると、コンテンツに基づいてネゴシエートされたドキュメントのキャッシュを有効にできます。以下の形式を取ります。

```
CacheNegotiatedDocs option
```

この オプション は、表18.6「利用可能な CacheNegotiatedDocs オプション」 の説明に従って、有効なキーワードである必要があります。コンテンツネゴシエートされたドキュメントは時間の経過と共に変化するか、または要求側の入力により変更される可能性があるため、デフォルトのオプションは **Off** です。

表18.6 利用可能な CacheNegotiatedDocs オプション

オプション	説明
オン	コンテンツネゴシエートされたドキュメントのキャッシュを有効にします。
オフ	コンテンツネゴシエートされたドキュメントのキャッシュを無効にします。

例18.26 CacheNegotiatedDocs ディレクティブの使用

```
CacheNegotiatedDocs On
```

CacheRoot

CacheRoot ディレクティブを使用すると、キャッシュファイルを保存するディレクトリーを指定できます。以下の形式を取ります。

```
CacheRoot directory
```

ディレクトリー は、ローカルファイルシステム内の既存のディレクトリーへの完全パスである必要があります。デフォルトのオプションは **/var/cache/mod_proxy/** です。

例18.27 CacheRoot ディレクティブの使用

```
CacheRoot /var/cache/mod_proxy
```

CustomLog

CustomLog ディレクティブを使用すると、ログファイル名とログファイルの形式を指定できます。以下の形式を取ります。

```
CustomLog path format
```

パス はログファイルを参照し、ServerRoot ディレクティブ（デフォルトでは /etc/httpd/）で指定されたディレクトリーに対して相対的である必要があります。形式 は、明示的な形式の文字列、または LogFormat ディレクティブを使用して以前に定義したフォーマット名のいずれかである必要があります。

例18.28 CustomLog ディレクティブの使用

```
CustomLog logs/access_log combined
```

DefaultIcon

DefaultIcon ディレクティブでは、他のアイコンが関連付けられていない場合に、サーバー生成ディレクトリー一覧のファイルに表示されるアイコンを指定できます。以下の形式を取ります。

```
DefaultIcon path
```

パス は既存のアイコンファイルを参照し、DocumentRoot ディレクティブで指定されたディレクトリーに対して相対的である必要があります（例： /icons/unknown.png）。

例18.29 DefaultIcon ディレクティブの使用

```
DefaultIcon /icons/unknown.png
```

DefaultType

DefaultType ディレクティブでは、適切な MIME タイプをサーバーで判断できない場合に使用するメディアタイプを指定できます。以下の形式を取ります。

```
DefaultType content-type
```

content-type は、text/html、image/png、application/pdf など、有効な MIME タイプである必要があります。

例18.30 DefaultType ディレクティブの使用

```
DefaultType text/plain
```

拒否

Deny ディレクティブを使用すると、指定のディレクトリーへのアクセスを拒否しているクライアントを指定できます。以下の形式を取ります。

```
Deny from client…
```

クライアント は、ドメイン名、IP アドレス（フルおよびパーシャルの両方）、ネットワーク/ネットマスク のペア、またはすべてのクライアントの すべて になります。

例18.31 Deny ディレクティブの使用

```
Deny from 192.168.1.1
```

DirectoryIndex

DirectoryIndex ディレクティブでは、ディレクトリーが要求される際にクライアントに提供されるドキュメントを指定できます（つまり、URL が / 文字で終了する場合）。以下の形式を取ります。

```
DirectoryIndex filename…
```

ファイル 名 は、要求されたディレクトリーで検索するファイルの名前です。デフォルトでは、サーバーは index.html と index.html.var を検索します。

例18.32 DirectoryIndex ディレクティブの使用

```
DirectoryIndex index.html index.html.var
```

DocumentRoot

DocumentRoot ディレクティブを使用すると、コンテンツを提供するメインディレクトリーを指定できます。以下の形式を取ります。

```
DocumentRoot directory
```

ディレクトリー は、ローカルファイルシステム内の既存のディレクトリーへの完全パスである必要があります。デフォルトのオプションは **/var/www/html/** です。

例18.33 DocumentRoot ディレクティブの使用

```
DocumentRoot /var/www/html
```

ErrorDocument

ErrorDocument ディレクティブを使用すると、特定のエラーへの応答として表示するドキュメントまたはメッセージを指定できます。以下の形式を取ります。

```
ErrorDocument error-code action
```

error-code は **403 (Forbidden)**、**404 (Not Found)**、または **500 (Internal Server Error)**などの有効なコードである必要があります。アクション は、**URL**（ローカルおよび外部の両方）または二重引用符（つまり "）で囲まれたメッセージ文字列のいずれかになります。

例18.34 ErrorDocument ディレクティブの使用

```
ErrorDocument 403 "Access Denied"
ErrorDocument 404 /404-not_found.html
```

ErrorLog

ErrorLog ディレクティブを使用すると、サーバーエラーがログに記録されるファイルを指定できます。以下の形式を取ります。

```
ErrorLog path
```

パス はログファイルを参照し、**ServerRoot** ディレクティブ（デフォルトは **/etc/httpd/** ）で

指定されたディレクトリーへの相対パスのいずれかです。デフォルトのオプションは logs/error_log です。

> *例18.35 ErrorLog ディレクティブの使用*
>
> ```
> ErrorLog logs/error_log
> ```

ExtendedStatus

ExtendedStatus ディレクティブを使用すると、詳細なサーバーステータス情報を有効にできます。以下の形式を取ります。

```
ExtendedStatus option
```

この オプション は、表18.7「利用可能な ExtendedStatus オプション」 の説明に従って、有効なキーワードである必要があります。デフォルトのオプションは Off です。

表18.7 利用可能な ExtendedStatus オプション

オプション	説明
オン	詳細なサーバーステータスの生成を有効にします。
オフ	詳細なサーバーステータスの生成を無効にします。

> *例18.36 ExtendedStatus ディレクティブの使用*
>
> ```
> ExtendedStatus On
> ```

グループ

Group ディレクティブを使用すると、httpd サービスを実行するグループを指定できます。以下の形式を取ります。

```
Group group
```

この グループ は既存の UNIX グループである必要があります。デフォルトのオプションは apache です。

Group は < VirtualHost> 内で はサポートされなくなり、SuexecUserGroup ディレクティブに置き換えられました。

例18.37 Group ディレクティブの使用

```
Group apache
```

HeaderName

HeaderName ディレクティブを使用すると、サーバー生成ディレクトリーリストの最初に付加されるファイルを指定できます。以下の形式を取ります。

```
HeaderName filename
```

ファイル 名 は、要求されたディレクトリーで検索するファイルの名前です。デフォルトでは、サーバーは HEADER.html を検索します。

例18.38 HeaderName ディレクティブの使用

```
HeaderName HEADER.html
```

HostnameLookups

HostnameLookups ディレクティブを使用すると、IP アドレスの自動解決を有効にできます。以下の形式を取ります。

```
HostnameLookups option
```

この オプション は、表18.8「利用可能な HostnameLookups オプション」 の説明に従って、有効なキーワードである必要があります。サーバーのリソースを確保するため、デフォルトのオプションは Off です。

表18.8 利用可能な HostnameLookups オプション

オプション	説明

オプション	説明
オン	各接続の IP アドレスの解決を有効にして、ホスト名をログに記録できるようにします。ただし、これにより処理のオーバーヘッドが大きくなります。
double	二重の逆引き DNS ルックアップの実行を有効にします。上記のオプションと比較すると、処理のオーバーヘッドがさらに追加されます。
オフ	各接続の IP アドレスの解決を無効にします。

サーバーログファイルにホスト名が必要な場合は、DNS ルックアップを実行する多くのログアナライザーツールのいずれかを効率的に使用できます。

例18.39 HostnameLookups ディレクティブの使用

```
HostnameLookups Off
```

包含

Include ディレクティブを使用すると、他の設定ファイルを追加できます。以下の形式を取ります。

```
Include filename
```

ファイル 名 には、絶対パス、ServerRoot ディレクティブで指定されたディレクトリーに対する相対パス、またはワイルドカード式を使用できます。/etc/httpd/conf.d/ ディレクトリーのすべての設定ファイルは、デフォルトでロードされます。

例18.40 Include ディレクティブの使用

```
Include conf.d/*.conf
```

IndexIgnore

IndexIgnore ディレクティブを使用すると、サーバー生成ディレクトリーの一覧から省略されるファイル名の一覧を指定できます。以下の形式を取ります。

```
IndexIgnore filename…
```

filename オプションは、完全なファイル名またはワイルドカード式のいずれかです。

例18.41 IndexIgnore ディレクティブの使用

```
IndexIgnore .??* *~ *# HEADER* README* RCS CVS *,v *,t
```

IndexOptions

IndexOptions ディレクティブを使用すると、サーバー生成ディレクトリー一覧の動作をカスタマイズできます。以下の形式を取ります。

```
IndexOptions option…
```

この オプション は、表18.9「利用可能なディレクトリー一覧オプション」 の説明に従って、有効なキーワードである必要があります。デフォルトのオプションは Charset=UTF-8、FancyIndexing、HTMLTable、NameWidth=、および VersionSort です。*

表18.9 利用可能なディレクトリー一覧オプション

オプション	説明
charset=encoding	生成された Web ページの文字セットを指定します。エンコーディング は、UTF-8 や ISO-8859-2 などの有効な文字セットである必要があります。
Type=content-type	生成された Web ページのメディアタイプを指定します。content-type は、text/html や text/plain などの有効な MIME タイプである必要があります。
DescriptionWidth=value	description 列の幅を指定します。値 は、幅を自動的に調整する多数の文字またはアスタリスク(*)のいずれかになります。
FancyIndexing	特定のファイルのさまざまなアイコンや、列ヘッダーをクリックしてディレクトリー一覧を再分類できる高度な機能を有効にします。
FolderFirst	最初にディレクトリーの一覧表示を有効にします。常にディレクトリーにファイルを配置してください。

オプション	説明
HTMLTable	ディレクトリーの一覧への HTML テーブルの使用を有効にします。
IconsAreLinks	アイコンをリンクとしての使用を有効にします。
IconHeight=value	アイコンの高さを指定します。値 は複数のピクセルです。
IconWidth=value	アイコンの幅を指定します。値 は複数のピクセルです。
IgnoreCase	大文字と小文字を区別する方法でファイルとディレクトリーのソートを有効にします。
IgnoreClient	クライアントからのクエリー変数の受け入れを無効にします。
NameWidth=value	ファイル名列の幅を指定します。値 は、幅を自動的に調整する多数の文字またはアスタリスク(*)のいずれかになります。
ScanHTMLTitles	AddDescription ディレクティブでファイルが指定されていない場合に、説明（つまり title 要素）のファイルの解析を有効にします。
ShowForbidden	他のアクセスが制限されているファイルの一覧表示を有効にします。
SuppressColumnSorting	列ヘッダーをクリックして、ディレクトリー一覧の並べ替えを無効にします。
SuppressDescription	ファイルの説明のために領域の確保を無効にします。
SuppressHTMLPreamble	HeaderName ディレクティブで指定されたファイルが存在する場合に、標準の HTML プリアンブルの使用を無効にします。
SuppressIcon	ディレクトリー一覧でアイコンの使用を無効にします。
SuppressLastModified	ディレクトリー一覧の最後の変更フィールドの日付の表示を無効にします。
SuppressRules	ディレクトリー一覧での水平行の使用を無効にします。

オプション	説明
SuppressSize	ディレクトリー一覧のファイルサイズフィールドの表示を無効にします。
TrackModified	HTTP ヘッダーの Last-Modified および ETag の値の返信を有効にします。
VersionSort	期待した方法でバージョン番号が含まれるファイルのソートを有効にします。
XHTML	デフォルトの HTML 3.2 の代わりに XHTML 1.0 の使用を有効にします。

例18.42 IndexOptions ディレクティブの使用

```
IndexOptions FancyIndexing VersionSort NameWidth=* HTMLTable Charset=UTF-8
```

KeepAlive

KeepAlive ディレクティブを使用すると、永続的な接続を有効にできます。以下の形式を取ります。

```
KeepAlive option
```

この オプション は、表18.10「利用可能な KeepAlive オプション」 の説明に従って、有効なキーワードである必要があります。デフォルトのオプションは Off です。

表18.10 利用可能な KeepAlive オプション

オプション	説明
オン	永続的な接続を有効にします。この場合、サーバーは接続ごとに複数のリクエストを受け入れます。
オフ	キープアライブ接続を無効にします。

永続的な接続が有効になっていると、ビジーなサーバーで子プロセスの数がすぐに増大し、最終的に最大限度に達するため、サーバーが大幅に低下することに注意してください。リスクを軽減

するには、KeepAliveTimeout をより小さい値に設定し、/var/log/httpd/logs/error_log ログファイルを注意して監視することを推奨します。

例18.43 KeepAlive ディレクティブの使用

```
KeepAlive Off
```

KeepAliveTimeout

KeepAliveTimeout ディレクティブでは、接続を閉じる前に別の要求を待つ時間を指定できます。以下の形式を取ります。

```
KeepAliveTimeout time
```

時間 は秒単位で指定します。デフォルトのオプションは 15 です。

例18.44 KeepAliveTimeout ディレクティブの使用

```
KeepAliveTimeout 15
```

LanguagePriority

LanguagePriority ディレクティブでは、言語の優先順位をカスタマイズできます。以下の形式を取ります。

```
LanguagePriority language…
```

言語 は、cs、en、fr などの有効な MIME 言語である必要があります。

このディレクティブは、クライアントの言語設定に基づいて複数の言語でコンテンツを提供する Web サーバーに特に便利です。

例18.45 LanguagePriority ディレクティブの使用

```
LanguagePriority sk cs en
```

listen

Listen ディレクティブでは、リッスンする IP アドレスまたはポートを指定できます。以下の形式を取ります。

```
Listen [ip-address:]port [protocol]
```

ip-address はオプションで、指定しない限り、サーバーはすべての IP アドレスからの指定のポート で着信要求を受け入れます。プロトコル はポート番号から自動的に決定されるため、通常は省略可能です。デフォルトオプションでは、ポート 80 をリッスンします。

サーバーが 1024 未満のポートをリッスンするように設定されている場合は、スーパーユーザーだけが httpd サービスを起動することができることに注意してください。

例18.46 Listen ディレクティブの使用

```
Listen 80
```

LoadModule

LoadModule ディレクティブを使用すると、DSO(Dynamic Shared Object)モジュールを読み込むことができます。以下の形式を取ります。

```
LoadModule name path
```

名前 は、必要なモジュールの有効な識別子である必要があります。パス は既存のモジュールファイルを参照し、ライブラリーを配置するディレクトリーに対して相対的である必要があります（32 ビットの場合は /usr/lib/httpd/、デフォルトでは 64 ビットシステムの場合は /usr/lib/httpd/ ）。

Apache HTTP Server の DSO サポートの詳細は、「モジュールの使用」 を参照してください。

例18.47 LoadModule ディレクティブの使用

```
LoadModule php5_module modules/libphp5.so
```

LogFormat

LogFormat ディレクティブでは、ログファイルの形式を指定できます。以下の形式を取ります。

```
LogFormat format name
```

形式 は、表18.11「一般的な LogFormat オプション」 で説明されているオプションで構成される文字列です。名前 は CustomLog ディレクティブの形式文字列の代わりに使用できます。

表18.11 一般的な LogFormat オプション

オプション	説明
%b	応答のサイズ（バイト単位）を表します。
%h	リモートクライアントの IP アドレスまたはホスト名を表します。
%l	指定されている場合はリモートログ名を表します。そうでない場合は、代わりにハイフン（つまり -）が使用されます。
%r	ブラウザーまたはクライアントから送信される要求文字列の最初の行を表します。
%s	ステータスコードを表します。
%t	リクエストの日付と時間を表します。
%u	認証が必要な場合は、リモートユーザーを表します。そうでない場合は、代わりにハイフン（つまり -）が使用されます。
%{field}	HTTP ヘッダー フィールド の内容を表します。一般的なオプションには、%{Referer}（クライアントをサーバーに接続する Web ページの URL）および %{User-Agent}（リクエストを行う Web ブラウザーのタイプ）が含まれます。

例18.48 LogFormat ディレクティブの使用

```
LogFormat "%h %l %u %t \"%r\" %>s %b" common
```

LogLevel

LogLevel ディレクティブを使用すると、エラーログの詳細レベルをカスタマイズできます。

以下の形式を取ります。

```
LogLevel option
```

この オプション は、表18.12「利用可能な logLevel オプション」 の説明に従って、有効なキーワードである必要があります。デフォルトのオプションは warn です。

表18.12 利用可能な logLevel オプション

オプション	説明
emerg	サーバーが作業を実行できない場合、緊急の状況のみがログに記録されます。
alert	即時アクションが必要なときのすべての状況がログに記録される。
crit	重大な状態はすべてログに記録されます。
error	すべてのエラーメッセージがログに記録されます。
warn	すべての警告メッセージがログに記録されます。
notice	正常であっても、重要な状況はログに記録されます。
info	さまざまな情報メッセージがログに記録されます。
debug	さまざまなデバッグメッセージがログに記録されます。

例18.49 LogLevel ディレクティブの使用

```
LogLevel warn
```

MaxKeepAliveRequests

MaxKeepAliveRequests ディレクティブでは、永続的な接続の要求の最大数を指定できます。以下の形式を取ります。

```
MaxKeepAliveRequests number
```

数 値が大きいと、サーバーのパフォーマンスが向上することがあります。0 を使用すると、要

求数が無制限になることに注意してください。デフォルトオプションは 100 です。

例18.50 MaxKeepAliveRequests オプションの使用

```
MaxKeepAliveRequests 100
```

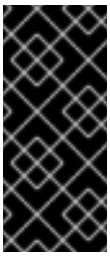
NameVirtualHost

NameVirtualHost ディレクティブを使用すると、名前ベースの仮想ホストの IP アドレスとポート番号を指定できます。以下の形式を取ります。

```
NameVirtualHost ip-address[:port]
```

ip-address は、すべてのインターフェースを表す完全な IP アドレス、またはアスタリスク(*) のいずれかになります。IPv6 アドレスは角括弧（ [および]）で囲む必要があることに注意してください。ポート はオプションです。

名前ベースの仮想ホストを使用すると、1 つの Apache HTTP Server が複数の IP アドレスを使用せずに異なるドメインに対応できます。



セキュアな HTTP 接続の使用

名前ベースの仮想ホストは、セキュアでない HTTP 接続でのみ 機能します。セキュアなサーバーで仮想ホストを使用する場合は、代わりに IP アドレスベースの仮想ホストを使用してください。

例18.51 NameVirtualHost ディレクティブの使用

```
NameVirtualHost *:80
```

オプション

Options ディレクティブでは、特定のディレクトリーで利用可能なサーバー機能を指定できます。以下の形式を取ります。

```
Options option…
```

この オプション は、表18.13「利用可能なサーバー機能」 の説明に従って、有効なキーワードである必要があります。

表18.13 利用可能なサーバー機能

オプション	説明
ExecCGI	CGI スクリプトの実行を有効にします。
FollowSymLinks	ディレクトリーの以下のシンボリックリンクを有効にします。
インクルード (Include)	サーバー側のインクルードを有効にします。
IncludesNOEXEC	サーバー側のインクルードを有効にしますが、コマンドの実行はできません。
Indexes	サーバー生成ディレクトリーの一覧を有効にします。
MultiViews	コンテンツネゴシエートされた 「複数ビュー」 を有効にします。
SymLinksIfOwnerMatch	リンクとターゲットファイルの両方に同じ所有者がある場合は、ディレクトリー内の以下のシンボリックリンクを有効にします。
すべて	複数のビュー 以外の上記のすべての機能を有効にします。
なし	上記のすべての機能を無効にします。

重要

SymLinksIfOwnerMatch オプションは、攻撃者が迂回できるためセキュリティー機能ではありません。

例18.52 Options ディレクティブの使用

```
Options Indexes FollowSymLinks
```

順序

Order ディレクティブでは、Allow ディレクティブおよび Deny ディレクティブを評価する順番を指定できます。以下の形式を取ります。

```
Order option
```

この オプション は、表18.14「利用可能な順序オプション」 の説明に従って、有効なキーワードである必要があります。デフォルトのオプションは allow,deny です。

表18.14 利用可能な順序オプション

オプション	説明
allow,deny	使用 できる ディレクティブは最初に評価されます。
deny,allow	deny ディレクティブが最初に評価されます。

例18.53 Order ディレクティブの使用

```
Order allow,deny
```

PidFile

PidFile ディレクティブでは、サーバーの プロセス ID (PID)を保存するファイルを指定できます。以下の形式を取ります。

```
PidFile path
```

パス は pid ファイルを参照します。これは、ServerRoot ディレクティブ（デフォルトでは /etc/httpd/ ）で指定されたディレクトリーへの相対または相対パスになります。デフォルトのオプションは run/httpd.pid です。

例18.54 PidFile ディレクティブの使用

```
PidFile run/httpd.pid
```

ProxyRequests

ProxyRequests ディレクティブを使用すると、転送プロキシー要求を有効にすることができます。以下の形式を取ります。

```
ProxyRequests option
```

この オプション は、表18.15「利用可能な ProxyRequests オプション」 の説明に従って、有効なキーワードである必要があります。デフォルトのオプションは Off です。

表18.15 利用可能な ProxyRequests オプション

オプション	説明
オン	転送プロキシー要求を有効にします。
オフ	転送プロキシー要求を無効にします。

例18.55 ProxyRequests ディレクティブの使用

```
ProxyRequests On
```

ReadmeName

ReadmeName ディレクティブを使用すると、サーバー生成ディレクトリー一覧の最後に追加するファイルを指定できます。以下の形式を取ります。

```
ReadmeName filename
```

ファイル 名 は、要求されたディレクトリーで検索するファイルの名前です。デフォルトでは、サーバーは README.html を検索します。

例18.56 ReadmeName ディレクティブの使用

```
ReadmeName README.html
```

リダイレクト

Redirect ディレクティブを使用すると、クライアントを別の URL にリダイレクトできます。以下の形式を取ります。

```
Redirect [status] path url
```

–

ステータス は任意で、表18.16「利用可能なステータスオプション」 で説明されているように有効なキーワードである必要があります。パス は古い場所を参照し、DocumentRoot ディレクティブで指定されたディレクトリーに対して相対的である必要があります（例： /docs）。URL は コンテンツの現在の場所を参照します（例： http://docs.example.com）。

表18.16 利用可能なステータスオプション

ステータス	説明
permanent	要求されたリソースが永続的に移動されたことを示します。301 (Moved Permanently)ステータスコードがクライアントに返されます。
temp	要求されたリソースが一時的に移動したことを示します。302 (Found)ステータスコードがクライアントに返されます。
seeother	要求されたリソースが置き換えられたことを示します。303 （その他を参照）のステータスコードがクライアントに返されます。
gone	要求されたリソースが永続的に削除されたことを示します。410 (Gone)ステータスはクライアントに返されます。

より高度なリダイレクト技術の場合は、Apache HTTP Server インストールに含まれる mod_rewrite モジュールを使用できます。

例18.57 Redirect ディレクティブの使用

```
Redirect permanent /docs http://docs.example.com
```

ScriptAlias

ScriptAlias ディレクティブを使用すると、CGI スクリプトの場所を指定できます。以下の形式を取ります。

```
ScriptAlias url-path real-path
```

url-path は DocumentRoot ディレクティブで指定されたディレクトリーに対して相対的である必要があります（例： /cgi-bin/）。リアルタイムパス は、ローカルファイルシステム内のファイルまたはディレクトリーへの完全パスです。

このディレクティブの後には、通常、ターゲットディレクトリーにアクセスするための追加のパーミッションを持つ Directory タグが続きます。デフォルトでは、/ var/www /cgi-bin/ にあるスクリプトにアクセスできるように、/cgi-bin/ alias が作成されます。

ScriptAlias ディレクティブは、CGI スクリプトが通常のテキストドキュメントとして表示されないようにするために使用されます。

例18.58 ScriptAlias ディレクティブの使用

```
ScriptAlias /cgi-bin/ /var/www/cgi-bin/

<Directory "/var/www/cgi-bin">
  AllowOverride None
  Options None
  Order allow,deny
  Allow from all
</Directory>
```

ServerAdmin

ServerAdmin ディレクティブでは、サーバー生成の Web ページに表示されるサーバー管理者のメールアドレスを指定できます。以下の形式を取ります。

```
ServerAdmin email
```

デフォルトのオプションは root@localhost です。

このディレクティブは、通常 webmaster@hostname に設定されます。hostname はサーバーのアドレスです。設定が完了すると、エイリアス webmaster は /etc/aliases の Web サーバーを担当するユーザー、および superuser として newaliases コマンドを実行します。

例18.59 ServerAdmin ディレクティブの使用

```
ServerAdmin webmaster@penguin.example.com
```

ServerName

ServerName ディレクティブを使用すると、Web サーバーのホスト名およびポート番号を指定できます。以下の形式を取ります。

```
ServerName hostname[:port]
```

ホスト名 は、サーバーの 完全修飾ドメイン名 (FQDN)である必要があります。ポート はオプションですが、指定する場合は Listen ディレクティブで指定された番号と一致する必要があります。

このディレクティブを使用する場合は、IP アドレスとサーバー名のペアが /etc/hosts ファイルに含まれていることを確認してください。

例18.60 ServerName ディレクティブの使用

```
ServerName penguin.example.com:80
```

ServerRoot

ServerRoot ディレクティブを使用すると、サーバーが機能するディレクトリーを指定できます。以下の形式を取ります。

```
ServerRoot directory
```

ディレクトリー は、ローカルファイルシステム内の既存のディレクトリーへの完全パスである必要があります。デフォルトのオプションは /etc/httpd/ です。

例18.61 ServerRoot ディレクティブの使用

```
ServerRoot /etc/httpd
```

ServerSignature

ServerSignature ディレクティブを使用すると、サーバー生成ドキュメントのサーバーに関する情報を表示できます。以下の形式を取ります。

```
ServerSignature option
```

この オプション は、表18.17「利用可能な ServerSignature オプション」 の説明に従って、有効なキーワードである必要があります。デフォルトのオプションは On です。

表18.17 利用可能な ServerSignature オプション

オプション	説明
オン	サーバー名とバージョンをサーバー生成ページに追加できるようにします。
オフ	サーバー名とバージョンをサーバー生成ページに追加するを無効にします。
Email	ServerAdmin ディレクティブで指定されたサーバー名、バージョン、およびメールアドレスをサーバー生成ページへ追加できるようにします。

例18.62 ServerSignature ディレクティブの使用

```
ServerSignature On
```

ServerTokens

ServerTokens ディレクティブを使用すると、Server 応答ヘッダーに含まれる情報をカスタマイズできます。以下の形式を取ります。

```
ServerTokens option
```

この オプション は、表18.18「利用可能な ServerTokens オプション」 の説明に従って、有効なキーワードである必要があります。デフォルトのオプションは OS です。

表18.18 利用可能な ServerTokens オプション

オプション	説明
Prod	製品名(Apache)のみが含まれます。
メジャー	製品名とサーバーのメジャーバージョンが含まれます（例： 2）。
マイナー	製品名とサーバーのマイナーバージョンが含まれます（例： 2.2）。
Min	製品名と最小バージョンのサーバー（例： 2.2.15）が含まれます。

オプション	説明
OS	製品名、サーバーの最小バージョン、および実行中のオペレーティングシステムのタイプ（Red Hatなど）が含まれます。
Full	上記のすべての情報を、読み込んだモジュールのリストとともに追加します。

セキュリティー上の理由から、できるだけ少ないサーバーに関する情報を表示することが推奨されます。

例18.63 ServerTokens ディレクティブの使用

```
ServerTokens Prod
```

SuexecUserGroup

SuexecUserGroup ディレクティブを使用すると、CGI スクリプトが実行されるユーザーおよびグループを指定できます。以下の形式を取ります。

```
SuexecUserGroup user group
```

ユーザー は既存のユーザーで、グループは有効な UNIX グループ である必要があります。

セキュリティー上の理由から、CGI スクリプトは root 権限で実行する必要があります。< VirtualHost& gt; では、SuexecUserGroup は User ディレクティブおよび Group ディレクティブを置き換えます。

例18.64 SuexecUserGroup ディレクティブの使用

```
SuexecUserGroup apache apache
```

タイムアウト

Timeout ディレクティブでは、接続を閉じる前のイベントを待機する時間を指定できます。以下の形式を取ります。

```
Timeout time
```

時間 は秒単位で指定します。デフォルトのオプションは 60 です。

例18.65 Timeout ディレクティブの使用

```
Timeout 60
```

TypesConfig

TypesConfig を使用すると、MIME タイプ設定ファイルの場所を指定できます。以下の形式を取ります。

```
TypesConfig path
```

パス は既存の MIME タイプの設定ファイルを参照し、絶対または ServerRoot ディレクティブ（デフォルトでは /etc/httpd/ ）で指定されたディレクトリーへの相対のいずれかになります。デフォルトのオプションは /etc/mime.types です。

/etc/mime.types を編集する代わりに、Apache HTTP Server に MIME タイプマッピングを追加する方法として AddType ディレクティブを使用することが推奨されます。

例18.66 TypesConfig ディレクティブの使用

```
TypesConfig /etc/mime.types
```

UseCanonicalName

UseCanonicalName を使用すると、サーバー自体の参照方法を指定できます。以下の形式を取ります。

```
UseCanonicalName option
```

この オプション は、表18.19「利用可能な UseCanonicalName オプション」 の説明に従って、有効なキーワードである必要があります。デフォルトのオプションは Off です。

表18.19 利用可能な UseCanonicalName オプション

オプション	説明
オン	ServerName ディレクティブで指定された名前の使用を有効にします。
オフ	ServerName ディレクティブで指定された名前の使用を無効にします。代わりに、要求側のクライアントが提供するホスト名とポート番号が使用されます。
DNS	ServerName ディレクティブで指定された名前の使用を無効にします。代わりに逆引き DNS ルックアップによって決定されるホスト名が使用されます。

例18.67 UseCanonicalName ディレクティブの使用

```
UseCanonicalName Off
```

ユーザー

User ディレクティブでは、httpd サービスを実行するユーザーを指定できます。以下の形式を取ります。

```
User user
```

既存の UNIX ユーザー である必要があります。デフォルトのオプションは apache です。

セキュリティー上の理由から、httpd サービスは root 権限で実行しないでください。User は < VirtualHost> 内で はサポートされなくなり、SuexecUserGroup ディレクティブに置き換えられました。

例18.68 User ディレクティブの使用

```
User apache
```

UserDir

UserDir ディレクティブでは、ユーザーのホームディレクトリーからのコンテンツの提供を有効にできます。以下の形式を取ります。

UserDir option

この オプション は、ユーザーのホームディレクトリー（通常は public_html）を検索するディレクトリーの名前、または 表18.20「利用可能な UserDir オプション」 で説明されているように有効なキーワードになります。デフォルトのオプションは 無効になっ ています。

表18.20 利用可能な UserDir オプション

オプション	説明
enabled user...	指定の ユーザーのホームディレクトリーからコンテンツを提供できます。
disabled [user...]	ホームディレクトリーのコンテンツの提供を無効にします（すべてのユーザー、またはユーザースペースで区切られた一覧が指定された場合）は、指定の ユーザーにのみコンテンツを提供します。

適切なパーミッションの設定

Web サーバーがコンテンツにアクセスできるようにするには、関連するディレクトリーおよびファイルのパーミッションを正しく設定する必要があります。すべてのユーザーがホームディレクトリーにアクセスでき、UserDir ディレクティブで指定されたディレクトリーの内容にアクセスできることを確認します。以下に例を示します。

```
~]# chmod a+x /home/username/
~]# chmod a+rx /home/username/public_html/
```

このディレクトリーのすべてのファイルは適宜設定する必要があります。

例18.69 UserDir ディレクティブの使用

```
UserDir public_html
```

18.1.5.2. 一般的な ssl.conf ディレクティブ

Secure Sockets Layer (SSL)ディレクティブを使用すると、Apache HTTP Secure Server の動作をカスタマイズできます。多くの場合は、インストール時に適切に設定されます。設定が間違っているとセキュリティーの脆弱性につながる可能性があるため、これらの設定を変更する場合には注意が必要です。

以下のディレクティブは、/etc/httpd/conf.d/ssl.conf で一般的に使用されます。

SetEnvIf

SetEnvIf ディレクティブでは、受信接続のヘッダーに基づいて環境変数を設定できます。以下の形式を取ります。

```
SetEnvIf option pattern [!]variable[=value]…
```

この オプション は、表18.21「利用可能な SetEnvIf オプション」 で説明されているように、HTTP ヘッダーフィールド、以前に定義した環境変数名、または有効なキーワードのいずれかです。この パターン は正規表現です。変数 は、オプションがパターンと一致する際に設定される環境変数です。オプションの感嘆符（つまり !）が存在する場合、変数は設定されずに削除されます。

表18.21 利用可能な SetEnvIf オプション

オプション	説明
Remote_Host	クライアントのホスト名を参照します。
Remote_Addr	クライアントの IP アドレスを参照します。
Server_Addr	サーバーの IP アドレスを参照します。
Request_Method	リクエストメソッドを参照します（例： GET）。
Request_Protocol	プロトコル名とバージョン（ HTTP/1.1 など）を参照します。
Request_URI	要求されたリソースを参照します。

SetEnvIf ディレクティブは、HTTP keepalives を無効にし、クライアントのブラウザーからクローズ通知なしに SSL が接続を閉じるのを許可するために使用されます。これは、SSL 接続を確実にシャットダウンしていない特定の Web ブラウザーに必要です。

例18.70 SetEnvIf ディレクティブの使用

```
SetEnvIf User-Agent ".*MSIE.*" \
        nokeepalive ssl-unclean-shutdown \
        downgrade-1.0 force-response-1.0
```

/etc/httpd/conf.d/ssl.conf ファイルが存在する場合は、mod_ssl をインストールする必要があります。SSL サーバーのインストールおよび設定方法の詳細は、「SSL サーバーの設定」 を参照してください。

18.1.5.3. 一般的な複数プロセスモジュールのディレクティブ

Multi-Processing Module (MPM)ディレクティブを使用すると、特定の MPM 固有の server-pool の動作をカスタマイズできます。その特性はどの MPM が使用されるかによって異なるため、ディレクティブは IfModule に組み込まれています。デフォルトでは、server-pool は prefork と worker MPM の両方に対して定義されます。

以下の MPM ディレクティブは、/etc/httpd/conf/httpd.conf で一般的に使用されます。

MaxClients

MaxClients ディレクティブを使用すると、一度に処理する同時接続クライアントの最大数を指定できます。以下の形式を取ります。

```
MaxClients number
```

prefork MPM を使用する場合は、高い 数値 を指定するとサーバーのパフォーマンスが向上しますが、prefork MPM を使用する場合は 256 を超えることは推奨されません。

例18.71 MaxClients ディレクティブの使用

```
MaxClients 256
```

MaxRequestsPerChild

MaxRequestsPerChild ディレクティブでは、終了前に子プロセスが処理できるリクエストの最大数を指定できます。以下の形式を取ります。

```
MaxRequestsPerChild number
```

数値を 0 に設定すると、要求数が無制限になります。

MaxRequestsPerChild ディレクティブは、有効期限の長いプロセスがメモリーリークを引き起こさないようにするために使用されます。

例18.72 MaxRequestsPerChild ディレクティブの使用

```
MaxRequestsPerChild 4000
```

MaxSpareServers

MaxSpareServers ディレクティブでは、スペア子プロセスの最大数を指定できます。以下の形式を取ります。

```
MaxSpareServers number
```

このディレクティブは、prefork MPM のみによって使用されます。

例18.73 MaxSpareServers ディレクティブの使用

```
MaxSpareServers 20
```

MaxSpareThreads

MaxSpareThreads ディレクティブを使用すると、スペアサーバースレッドの最大数を指定できます。以下の形式を取ります。

```
MaxSpareThreads number
```

数値 は MinSpareThreads および ThreadsPerChild の合計値以上である必要があります。このディレクティブは、worker MPM だけによって使用されます。

例18.74 MaxSpareThreads ディレクティブの使用

```
MaxSpareThreads 75
```

MinSpareServers

MinSpareServers ディレクティブでは、予備の子プロセスの最小数を指定できます。以下の形式を取ります。

```
MinSpareServers number
```

数 値が大きいと、サーバーに大量の処理負荷が発生する可能性があることに注意してください。このディレクティブは、**prefork MPM** のみによって使用されます。

例18.75 MinSpareServers ディレクティブの使用

```
MinSpareServers 5
```

MinSpareThreads

MinSpareThreads ディレクティブを使用すると、スペアサーバースレッドの最小数を指定できます。以下の形式を取ります。

```
MinSpareThreads number
```

このディレクティブは、**worker MPM** だけによって使用されます。

例18.76 MinSpareThreads ディレクティブの使用

```
MinSpareThreads 75
```

StartServers

StartServers ディレクティブでは、サービスの起動時に作成する子プロセスの数を指定できます。以下の形式を取ります。

```
StartServers number
```

子プロセスは現在のトラフィックの負荷に応じて動的に作成され、終了されるため、通常はこの値を変更する必要はありません。

例18.77 StartServers ディレクティブの使用

```
StartServers 8
```

ThreadsPerChild

***ThreadsPerChild** ディレクティブを使用すると、子プロセスが作成できるスレッド数を指定できます。以下の形式を取ります。*

```
ThreadsPerChild number
```

*このディレクティブは、**worker MPM** だけによって使用されます。*

例18.78 ThreadsPerChild ディレクティブの使用

```
ThreadsPerChild 25
```

18.1.6. モジュールの使用

***httpd** サービスは、モジュールアプリケーションであるため、多数の **Dynamic Shared Objects (DSOs)**とともに配布されます。これは、必要に応じてランタイム時に、動的に読み込まれたり、アンロードしたりできます。デフォルトでは、これらのモジュールは **32** ビットおよび **64** ビットシステムの **/usr/lib64/httpd/modules** / の **/usr/lib/httpd/modules/** にあります。*

18.1.6.1. モジュールの読み込み

*特定の **DSO** モジュールを読み込むには、「一般的な **httpd.conf** ディレクティブ」 の説明に従って **LoadModule** ディレクティブを使用します。別のパッケージが提供するモジュールでは、多くの場合、**/etc/httpd/conf.d/** ディレクトリーに独自の設定ファイルがあることに注意してください。*

例18.79 mod_ssl DSO の読み込み

```
LoadModule ssl_module modules/mod_ssl.so
```

*操作が終了したら、**Web** サーバーを再起動して設定を再読み込みします。**httpd** サービスを再起動する方法は、「サービスの再起動」 を参照してください。*

18.1.6.2. モジュールの作成

*新しい **DSO** モジュールを作成する場合は、**httpd-devel** パッケージがインストールされていることを確認してください。これを行うには、**root** で次のコマンドを実行します。*

```
~]# yum install httpd-devel
```

このパッケージには、モジュールをコンパイルするために必要なインクルードファイル、ヘッダーファイル、および APache eXtenSion (apxs)ユーティリティーが含まれます。

作成したら、以下のコマンドでモジュールを構築できます。

```
~]# apxs -i -a -c module_name.c
```

ビルドが成功すると、Apache HTTP Server で配布されるその他のモジュールと同じ方法でモジュールを読み込むことができます。

18.1.7. 仮想ホストの設定

Apache HTTP Server に組み込まれている仮想ホストを使用すると、どの IP アドレス、ホスト名、またはポートが要求されているかに基づいて、サーバーが異なる情報を提供できます。

名前ベースの仮想ホストを作成するには、例として /etc/httpd/conf/httpd.conf で提供される仮想ホストコンテナーを見つけ、各行の先頭にあるハッシュ記号（つまり #）を削除し、例18.80「仮想ホストの設定例」 に記載の要件に応じてオプションをカスタマイズします。

例18.80 仮想ホストの設定例

```
NameVirtualHost *:80

<VirtualHost *:80>
    ServerAdmin webmaster@penguin.example.com
    DocumentRoot /www/docs/penguin.example.com
    ServerName penguin.example.com
    ServerAlias www.penguin.example.com
    ErrorLog logs/penguin.example.com-error_log
    CustomLog logs/penguin.example.com-access_log common
</VirtualHost>
```

ServerName は、マシンに割り当てられた有効な DNS 名を指定する必要があることに注意してください。&lt ;VirtualHost> コンテナーは高度なカスタマイズが可能で、メインサーバー設定で利用可能なディレクティブのほとんどを使用できます。このコンテナーで対応していない ディレクティブには、SuexecUserGroup に置き換えられた User および Group が含まれます。

ポート番号の変更

仮想ホストがデフォルト以外のポートをリッスンするように設定する場合は、それに応じて /etc/httpd/conf/httpd.conf ファイルのグローバル設定セクションの Listen ディレクティブを必ず更新してください。

新規に作成された仮想ホストをアクティブ化するには、Web サーバーを再起動する必要があります。httpd サービスを再起動する方法は、「サービスの再起動」 を参照してください。

18.1.8. SSL サーバーの設定

Secure Sockets Layer (SSL)は、サーバーとクライアントがセキュアに通信できるようにする暗号化プロトコルです。Transport Layer Security (TLS)と呼ばれる拡張および改善されたバージョンとともに、プライバシーとデータの整合性の両方が確保されます。Apache HTTP Server を mod_ssl と組み合わせて、OpenSSL ツールキットを使用して SSL/TLS サポートを提供するモジュールは、SSL サーバー と呼ばれます。Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux は、TLS 実装としての Mozilla NSS の使用もサポートします。Mozilla NSS のサポートは mod_nss モジュールにより提供されます。

傍受できる人であればだれでも読み取りや修正が可能な HTTP 接続とは異なり、HTTPS と呼ばれる SSL/TLS over HTTP の使用により、送信したコンテンツの検査や修正が阻止されます。本セクションでは、Apache HTTP Server 設定内でこのモジュールを有効にする基本的な方法と、秘密鍵と自己署名の証明書の生成プロセスを説明します。

18.1.8.1. 証明書およびセキュリティーの概要

通信の安全性は、鍵の使用により確保されます。従来の暗号または 対称暗号 では、トランザクションの両端に、相互に送信をデコードするのに使用する鍵と同じキーがあります。一方、公開暗号または 非対称暗号 では、秘密鍵（ 秘密 を保持する）と公開鍵（通常は 公開鍵 ）が共存します。公開鍵でエンコードされたデータは秘密鍵でしかデコードできませんが、秘密鍵でエンコードされたデータは公開鍵でしかデコードできません。

SSL を使用してセキュアな通信を提供するには、SSL サーバーは 認証局 (CA)が署名したデジタル証明書を使用する必要があります。証明書は、サーバーのホスト名、会社名、会社の場所など、サーバーのさまざまな属性と、CA の秘密鍵を使用して生成した署名を一覧表示します。この署名は、特定の認証局が証明書を署名し、証明書がいかなる方法でも変更されていないことを確認するものです。

Web ブラウザーが新しい SSL 接続を確立すると、Web サーバーが提供する証明書が確認されます。証明書に、信頼できる CA からの署名がない場合や、証明書に一覧表示されているホスト名が、接続の確立に使用されたホスト名と一致しない場合は、サーバーとの通信が拒否され、通常は、ユーザーに適切なエラーメッセージが表示されます。

デフォルトでは、ほとんどの Web ブラウザーは、幅広く使用されている一連の証明書局を信頼するように設定されています。このため、安全なサーバーを設定する際には、適切な CA を選択して、ターゲットユーザーが接続を信頼できる状態にする必要があります。適切な CA を選択しないとエラーメッセージが表示され、証明書を手動で指定することが必要になります。ユーザーが証明書エラーを上書きするようにすると、攻撃者が接続を傍受できるため、可能な場合は信頼できる CA を使用する必要があります。詳細は、表18.22「一般的な Web ブラウザーが使用する CA リストに関する情報」 を参照してください。

表18.22 一般的な Web ブラウザーが使用する CA リストに関する情報

Web ブラウザー	リンク
Mozilla Firefox	Mozilla root CA の一覧
Opera	Opera が使用するルート証明書に関する情報。
Internet Explorer	Microsoft Windows が使用するルート証明書に関する情報
Chromium	Chromium プロジェクトが使用するルート証明書に関する情報

SSL サーバーを設定する場合は、証明書要求と秘密鍵を生成し、証明書要求、会社の ID の証明、および支払いを認証局に送信する必要があります。CA が証明書の要求および ID を確認すると、サーバーで使用できる署名付きの証明書が送信されます。また、CA 署名を含まない自己署名証明書を作成することもできるため、テスト目的でのみ使用してください。

18.1.9. mod_ssl モジュールの有効化

mod_ssl を使用して SSL または HTTPS サーバーをセットアップする場合は、別のアプリケーションまたはモジュール（ mod_nss など）で同じポートを使用するよう設定することは できません。ポート 443 は、HTTPS のデフォルトポートです。

mod_ssl モジュールおよび OpenSSL ツールキットを使用して SSL サーバーを設定するには、mod_ssl および openssl パッケージをインストールします。root で以下のコマンドを入力します。

```
~]# yum install mod_ssl openssl
```

これにより、/etc/httpd/conf.d/ssl.conf に mod_ssl 設定ファイルが作成されます。このファイルは、デフォルトでメインの Apache HTTP Server 設定ファイルに含まれています。モジュールを読み込むには、「サービスの再起動」 の説明に従って httpd サービスを再起動します。

重要

『POODLE: SSLv3 脆弱性(CVE-2014-3566)で説明されている脆弱性』 のため、Red Hat は、SSL を無効にし、有効にしている場合は TLSv1 . 1 または TLSv1.2 のみを使用することを推奨します。後方互換性は、TLSv1.0 を使用して実現できます。Red Hat がサポートする多くの製品は SSLv2 プロトコルまたは SSLv3 プロトコルを使用できます。ただし、SSLv2 または SSLv3 の使用が強く推奨されます。

18.1.9.1. mod_ssl での SSL および TLS の有効化および無効化

SSL および TLS プロトコルの特定のバージョンを有効および無効にするには、設定ファイルの「## SSL Global Context」 セクションに SSLProtocol ディレクティブを追加し、他のすべての部分でそのディレクティブを削除するか、すべての 「VirtualHost」 セクションの 「# SSL Protocol support」 の下にあるデフォルトエントリーを編集することによりグローバルで設定します。ドメインごとの VirtualHost セクションで指定しない場合、設定はグローバルセクションから継承されます。プロトコルバージョンが確実に無効になるように、管理者は SSLProtocol を 「SSL Global Context」 セクションに のみ 指定するか、ドメインごとの すべて の VirtualHost セクションで指定する必要があります。

Red Hat Enterprise Linux 6.8 SSLv2 では、デフォルトで無効になっていることに注意してください。

手順18.1 SSLv2 および SSLv3 の無効化

すべての VirtualHost セクションで SSL バージョン 2 および SSL バージョン 3 を無効にするには (SSL バージョン 2 および SSL バージョン 3 以外をすべて有効にすることを意味します)、以下の手順を実行します。

1. *root で /etc/httpd/conf.d/ssl.conf ファイルを開き、SSLProtocol ディレクティブの すべて のインスタンスを検索します。デフォルトでは、設定ファイルに以下のような 1 つのセクションが含まれます。*

```
~]# vi /etc/httpd/conf.d/ssl.conf
#   SSL Protocol support:
# List the enable protocol levels with which clients will be able to
# connect.  Disable SSLv2 access by default:
SSLProtocol all -SSLv2
```

このセクションは、VirtualHost セクション内にあります。

2. *SSLProtocol 行を以下のように編集します。*

```
#   SSL Protocol support:
# List the enable protocol levels with which clients will be able to
```

```
# connect.  Disable SSLv2 access by default:
SSLProtocol all -SSLv2 -SSLv3
```

すべての VirtualHost セクションに対してこのアクションを繰り返します。ファイルを保存してから閉じます。

3.

すべての SSLProtocol ディレクティブが以下のように変更したことを確認します。

```
~]# grep SSLProtocol /etc/httpd/conf.d/ssl.conf
SSLProtocol all -SSLv2 -SSLv3
```

この手順は、デフォルトの VirtualHost セクションが複数の場合に特に重要になります。

4.

以下のように Apache デーモンを再起動します。

```
~]# service httpd restart
```

セッションが中断されることに注意してください。

手順18.2 TLS 1 以上を除くすべての SSL および TLS プロトコルの無効化

TLS バージョン 1 以上を除く SSL および TLS プロトコルバージョンをすべて無効にするには、以下の手順を実行します。

1.

root で /etc/httpd/conf.d/ssl.conf ファイルを開き、SSLProtocol ディレクティブの すべて のインスタンスを検索します。デフォルトでは、このファイルには以下のようなセクションが含まれます。

```
~]# vi /etc/httpd/conf.d/ssl.conf
#   SSL Protocol support:
# List the enable protocol levels with which clients will be able to
# connect.  Disable SSLv2 access by default:
SSLProtocol all -SSLv2
```

2.

SSLProtocol 行を以下のように編集します。

```
#   SSL Protocol support:
# List the enable protocol levels with which clients will be able to
# connect.  Disable SSLv2 access by default:
```

```
SSLProtocol -all +TLSv1 +TLSv1.2
```

ファイルを保存してから閉じます。

3. 以下のように変更を確認します。

```
~]# grep SSLProtocol /etc/httpd/conf.d/ssl.conf
SSLProtocol -all +TLSv1 +TLSv1.2
```

4. 以下のように Apache デーモンを再起動します。

```
~]# service httpd restart
```

セッションが中断されることに注意してください。

手順18.3 SSL および TLS プロトコルのステータスのテスト

有効または無効な SSL および TLS のバージョンを確認するには、openssl s_client -connect コマンドを使用します。このコマンドの形式は

```
openssl s_client -connect hostname:port -protocol
```

です。port は、テストするポートで、protocol はテストするプロトコルバージョンです。ローカルで稼働している SSL サーバーをテストするには、localhost をホスト名として使用します。たとえば、SSLv3 が有効であるかどうかを確認するためにセキュアな HTTPS 接続のデフォルトポートであるポート 443 をテストするには、以下のようにコマンドを発行します。

1.

```
~]# openssl s_client -connect localhost:443 -ssl3
CONNECTED(00000003)
139809943877536:error:14094410:SSL routines:SSL3_READ_BYTES:sslv3 alert
handshake failure:s3_pkt.c:1257:SSL alert number 40
139809943877536:error:1409E0E5:SSL routines:SSL3_WRITE_BYTES:ssl handshake
failure:s3_pkt.c:596:
output omitted
New, (NONE), Cipher is (NONE)
Secure Renegotiation IS NOT supported
Compression: NONE
Expansion: NONE
```

```
SSL-Session:
    Protocol  : SSLv3
output truncated
```

上記の出力はハンドシェイクが失敗し、暗号化がネゴシエートされなかったことを示しています。

2.

```
~]$ openssl s_client -connect localhost:443 -tls1_2
CONNECTED(00000003)
depth=0 C = --, ST = SomeState, L = SomeCity, O = SomeOrganization, OU =
SomeOrganizationalUnit, CN = localhost.localdomain, emailAddress =
root@localhost.localdomain
output omitted
New, TLSv1/SSLv3, Cipher is ECDHE-RSA-AES256-GCM-SHA384
Server public key is 2048 bit
Secure Renegotiation IS supported
Compression: NONE
Expansion: NONE
SSL-Session:
    Protocol  : TLSv1.2
output truncated
```

上記の出力は、ハンドシェイクが失敗しておらず、暗号化セットがネゴシエートされたことを示しています。

openssl s_client コマンドのオプションは、s_client(1) man ページに記載されています。

SSLv3 の脆弱性とそのテスト方法の詳細は、Red Hat ナレッジベースの記事 『POODLE: SSLv3 vulnerability(CVE-2014-3566』)を参照してください。

18.1.10. mod_nss Module の有効化

mod_nss を使用して HTTPS サーバーをセットアップする場合、mod_ssl はデフォルトでポート 443 を使用するため、HTTPS サーバーでは mod_ssl を同時に使用することが できません が、これはデフォルトの HTTPS ポートになります。パッケージが必要ない場合は削除することが推奨されます。

mod_ssl を削除するには、root で次のコマンドを実行します。

```
~]# yum remove mod_ssl
```

注記

他の目的で mod_ssl が必要な場合は、443 以外のポートを使用するように /etc/httpd/conf.d/ssl.conf ファイルを変更し、リッスンするポートが 443 に変更されたときに mod_ssl が mod_nss と競合しないようにします。

HTTPS が必要な特定の VirtualHost の場合、mod_nss と mod_ssl は一意のポートを使用している場合のみ同時に共存できます。このため、mod_nss はデフォルトで 8443 を使用しますが、HTTPS のデフォルトポートはポート 443 です。ポートは Listen ディレクティブと VirtualHost 名またはアドレスで指定されます。

NSS のすべては 「トークン」 に関連付けられます。ソフトウェアトークンは NSS データベースに存在しますが、証明書を含む物理トークンを使用することもできます。OpenSSL では、個別証明書と秘密鍵は PEM ファイルに保持されます。NSS では、これらはデータベースに格納されます。各証明書およびキーはトークンと関連付けられ、各トークンにはパスワードを設定して保護することもできます。このパスワードはオプションですが、パスワードが使用される場合、Apache HTTP サーバーはシステムの起動時にユーザーの介入なしでデータベースを開くためにパスワードのコピーを必要とします。

手順18.4 mod_nss の設定

1. *root で mod_nss をインストールします。*

```
~]# yum install mod_nss
```

これにより、/etc/httpd/conf.d/nss.conf に mod_nss 設定ファイルが作成されます。/etc/httpd/conf.d/ ディレクトリーは、デフォルトでメインの Apache HTTP Server 設定ファイルに含まれます。モジュールを読み込むには、「サービスの再起動」 の説明に従って httpd サービスを再起動します。

2. *root で /etc/httpd/conf.d/nss.conf ファイルを開き、Listen ディレクティブの すべて のインスタンスを検索します。*

Listen 8443 行を以下のように編集します。

```
Listen 443
```

ポート 443 は、HTTPS のデフォルトポートです。

3.

デフォルトの VirtualHost _default_:8443 行を以下のように編集します。

```
VirtualHost _default_:443
```

デフォルト以外の他のすべての仮想ホストセクション (存在する場合) を編集します。ファイルを保存してから閉じます。

4.

Mozilla NSS は、証明書を /etc/httpd/conf.d/nss.conf ファイルの NSSCertificateDatabase ディレクティブで示唆される サーバー証明書データベース に保存します。デフォルトでは、パスはインストール時に作成された NSS データベースである /etc/httpd/alias に設定されます。

デフォルトの NSS データベースを表示するには、以下のコマンドを実行します。

```
~]# certutil -L -d /etc/httpd/alias

Certificate Nickname                                         Trust Attributes
                                                             SSL,S/MIME,JAR/XPI

cacert                                                       CTu,Cu,Cu
Server-Cert                                                  u,u,u
alpha                                                        u,pu,u
```

上記のコマンド出力では、Server-Cert がデフォルトの NSSNickname です。-L オプションは、証明書データベースにすべての証明書を一覧表示したり、名前付き証明書に関する情報を表示します。-d オプションは、証明書およびキーデータベースファイルを含むデータベースディレクトリーを指定します。その他のコマンドラインオプションは、certutil(1) の man ページを参照してください。

5.

別のデータベースを使用するように mod_nss を設定するには、/etc/httpd/conf.d/nss.conf ファイルの NSSCertificateDatabase 行を編集します。デフォルトファイルの VirtualHost セクション内には以下の行が含まれます。

```
#   Server Certificate Database:
#   The NSS security database directory that holds the certificates and
#   keys. The database consists of 3 files: cert8.db, key3.db and secmod.db.
#   Provide the directory that these files exist.
NSSCertificateDatabase /etc/httpd/alias
```

上記のコマンド出力では、alias がデフォルトの NSS データベースディレクトリー /etc/httpd/alias/ です。

6.

デフォルトの NSS 証明書データベースにパスワードを適用するには、root で以下のコマンドを使用します。

```
~]# certutil -W -d /etc/httpd/alias
Enter Password or Pin for "NSS Certificate DB":
Enter a password which will be used to encrypt your keys.
The password should be at least 8 characters long,
and should contain at least one non-alphabetic character.

Enter new password:
Re-enter password:
Password changed successfully.
```

7.

HTTPS サーバーをデプロイする前に、認証局 (CA) により署名された証明書を使用して新しい証明書データベースを作成します。

例18.81 Mozilla NSS データベースへの証明書の追加

certutil コマンドは、CA 証明書を NSS データベースファイルに追加するために使用されます。

```
certutil -d /etc/httpd/nss-db-directory/ -A -n "CA_certificate" -t CT,, -a -i
certificate.pem
```

上記のコマンドは、certificate.pem という名前の PEM 形式のファイルに保存されている CA 証明書を追加します。-d オプションは、証明書およびキーデータベースファイルを含む NSS データベースディレクトリーを指定します。-n オプションは、証明書の名前(-t CT,,)を設定します。これは、証明書が TLS クライアントおよびサーバーで使用されるように信頼されることを意味します。-A オプションは、既存の証明書を証明書データベースに追加します。データベースが存在しない場合は、作成されます。-a オプションを使用すると、入力または出力に ASCII 形式を使用でき、-i オプションは certificate.pem 入力ファイルをコマンドに渡します。

その他のコマンドラインオプションは、certutil(1) の man ページを参照してください。

8.

秘密鍵を保護するために、NSS データベースはパスワードで保護する必要があります。

例18.82 Setting_a_Password_for_a_Mozilla_NSS_database

NSS データベースのパスワードを設定するには、以下のように certutil ツールを使用できます。

```
certutil -W -d /etc/httpd/nss-db-directory/
```

たとえば、デフォルトのデータベースの場合は、root で以下のコマンドを実行します。

```
~]# certutil -W -d /etc/httpd/alias
Enter Password or Pin for "NSS Certificate DB":
Enter a password which will be used to encrypt your keys.
The password should be at least 8 characters long,
and should contain at least one non-alphabetic character.

Enter new password:
Re-enter password:
Password changed successfully.
```

9.

以下のように NSSPassPhraseDialog ディレクティブで行を変更して、NSS 内部ソフトウェアトークンを使用するように mod_nss を設定します。

```
~]# vi /etc/httpd/conf.d/nss.conf
NSSPassPhraseDialog file:/etc/httpd/password.conf
```

これにより、システムの起動時にパスワードを手動で入力する必要がなくなります。ソフトウェアトークンは NSS データベースに存在しますが、証明書を含む物理トークンを使用することもできます。

10.

NSS データベースに含まれる SSL サーバー証明書が RSA 証明書である場合は、NSSNickname パラメーターのコメントを解除し、上記の手順 4 で示されたニックネームに一致するようにします。

```
~]# vi /etc/httpd/conf.d/nss.conf
NSSNickname Server-Cert
```

NSS データベースに含まれる SSL サーバー証明書が ECC 証明書である場合は、NSSECCNickname パラメーターのコメントを解除し、上記の手順 4 で示されたニックネームに一致するようにします。

```
~]# vi /etc/httpd/conf.d/nss.conf
NSSECCNickname Server-Cert
```

NSSCertificateDatabase パラメーターがコメント解除され、上記の手順 4 で示された、または手順 5 で設定された NSS データベースディレクトリーを参照するようにします。

```
~]# vi /etc/httpd/conf.d/nss.conf
NSSCertificateDatabase /etc/httpd/alias
```

/etc/httpd/alias を、使用する証明書データベースへのパスに置き換えます。

11.

root で /etc/httpd/password.conf ファイルを作成します。

```
~]# vi /etc/httpd/password.conf
```

```
internal:password
```

の形式の行を追加します。上記の手順 6 で NSS セキュリティーデータベースに適用したパスワードでパスワードを置き換えます。

12.

適切な所有権とパーミッションを /etc/httpd/password.conf ファイルに適用します。

```
~]# chgrp apache /etc/httpd/password.conf
~]# chmod 640 /etc/httpd/password.conf
~]# ls -l /etc/httpd/password.conf
-rw-r-----. 1 root apache 10 Dec  4 17:13 /etc/httpd/password.conf
```

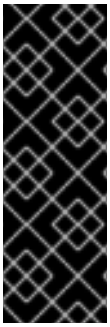
13.

NSS のソフトウェアトークンを使用するように mod_nss を設定するには、/etc/httpd/password.conf で /etc/httpd/conf.d/nss.conf を以下のように編集します。

```
~]# vi /etc/httpd/conf.d/nss.conf
```

14.

変更を反映するために、「サービスの再起動」 の説明通りに Apache サーバーを再起動します。



重要

『POODLE: SSLv3 脆弱性(CVE-2014-3566)で説明されている脆弱性』 のため、Red Hat は、SSL を無効にし、有効にしている場合は TLSv1 . 1 または TLSv1.2 のみを使用することを推奨します。後方互換性は、TLSv1.0 を使用して実現できます。Red Hat がサポートする多くの製品は SSLv2 プロトコルまたは SSLv3 プロトコルを使用できます。ただし、SSLv2 または SSLv3 の使用が強く推奨されます。

18.1.10.1. mod_nss での SSL および TLS の有効化および無効化

SSL および TLS プロトコルの特定のバージョンを有効および無効にするには、設定ファイルの 「## SSL Global Context」 セクションに NSSProtocol ディレクティブを追加し、他のすべての部分でそのディレクティブを削除するか、すべての 「VirtualHost」 セクションの 「# SSL Protocol」 の下にあるデフォルトエントリーを編集することによりグローバルで設定します。ドメインごとの VirtualHost セクションで指定しない場合、設定はグローバルセクションから継承されます。プロトコルバージョンが確実に無効になるように、管理者は 「SSL Global Context」 セクションに NSSProtocol のみ を指定するか、ドメインごとの すべて の VirtualHost セクションで指定する必要があります。

Red Hat Enterprise Linux 6.8 SSLv2 では、デフォルトで無効になっていることに注意してください。

手順18.5 mod_nss での TLS 1 以上を除くすべての SSL および TLS プロトコルの無効化

TLS バージョン 1 以上を除く SSL および TLS プロトコルバージョンをすべて無効にするには、以下の手順を実行します。

1. root で /etc/httpd/conf.d/nss.conf ファイルを開き、NSSProtocol ディレクティブの すべて のインスタンスを検索します。デフォルトでは、設定ファイルに以下のような 1 つのセクションが含まれます。

```
~]# vi /etc/httpd/conf.d/nss.conf
#   SSL Protocol:
output omitted
#   Since all protocol ranges are completely inclusive, and no protocol in the
#   middle of a range may be excluded, the entry "NSSProtocol SSLv3,TLSv1.1"
#   is identical to the entry "NSSProtocol SSLv3,TLSv1.0,TLSv1.1".
NSSProtocol TLSv1.0,TLSv1.1,TLSv1.2
```

このセクションは、VirtualHost セクション内にあります。

2.
NSSProtocol 行を以下のように編集します。

```
#   SSL Protocol:
NSSProtocol TLSv1.0,TLSv1.1,TLSv1.2
```

すべての VirtualHost セクションに対してこのアクションを繰り返します。

3.
Listen 8443 行を以下のように編集します。

```
Listen 443
```

4.
デフォルトの VirtualHost _default_:8443 行を以下のように編集します。

```
VirtualHost _default_:443
```

デフォルト以外の他のすべての仮想ホストセクション (存在する場合) を編集します。ファイルを保存してから閉じます。

5.
すべての NSSProtocol ディレクティブが以下のように変更したことを確認します。

```
~]# grep NSSProtocol /etc/httpd/conf.d/nss.conf
#   middle of a range may be excluded, the entry "NSSProtocol SSLv3,TLSv1.1"
#   is identical to the entry "NSSProtocol SSLv3,TLSv1.0,TLSv1.1".
NSSProtocol TLSv1.0,TLSv1.1,TLSv1.2
```

この手順は、VirtualHost セクションが複数の場合に特に重要になります。

6.
以下のように Apache デーモンを再起動します。

```
~]# service httpd restart
```

セッションが中断されることに注意してください。

手順18.6 mod_nss での SSL および TLS プロトコルのステータスのテスト

mod_nss で有効または無効な SSL および TLS のバージョンを確認するには、openssl s_client -connect コマンドを使用します。root で openssl パッケージをインストールします。

```
~]# yum install openssl
```

openssl s_client -connect コマンドの形式は

```
openssl s_client -connect hostname:port -protocol
```

になります。port は、テストするポートで、protocol はテストするプロトコルバージョンです。ローカルで稼働している SSL サーバーをテストするには、localhost をホスト名として使用します。たとえば、SSLv3 が有効であるかどうかを確認するためにセキュアな HTTPS 接続のデフォルトポートであるポート 443 をテストするには、以下のようにコマンドを発行します。

1.

```
~]# openssl s_client -connect localhost:443 -ssl3
CONNECTED(00000003)
3077773036:error:1408F10B:SSL routines:SSL3_GET_RECORD:wrong version
number:s3_pkt.c:337:
output omitted
New, (NONE), Cipher is (NONE)
Secure Renegotiation IS NOT supported
Compression: NONE
Expansion: NONE
SSL-Session:
    Protocol  : SSLv3
output truncated
```

上記の出力はハンドシェイクが失敗し、暗号化がネゴシエートされなかったことを示しています。

2.

```
~]$ openssl s_client -connect localhost:443 -tls1_2
CONNECTED(00000003)
depth=1 C = US, O = example.com, CN = Certificate Shack
output omitted
New, TLSv1/SSLv3, Cipher is AES256-SHA
Server public key is 1024 bit
Secure Renegotiation IS supported
Compression: NONE
Expansion: NONE
SSL-Session:
    Protocol  : TLSv1.2
    Cipher    : AES256-SHA
output truncated
```

上記の出力は、ハンドシェイクが失敗しておらず、暗号化セットがネゴシエートされたことを示しています。

openssl s_client コマンドのオプションは、s_client(1) man ページに記載されています。

SSLv3 の脆弱性とそのテスト方法の詳細は、Red Hat ナレッジベースの記事 『POODLE: SSLv3 vulnerability(CVE-2014-3566』)を参照してください。

18.1.11. 既存の鍵および証明書の使用

以前に作成した鍵と証明書がある場合は、新しいファイルを生成する代わりに、SSL サーバーを設定してこのファイルを使用できます。これが可能ではない状況は 2 つしかありません。

1. IP アドレスまたはドメイン名を変更している。

   証明書が、特定の IP アドレスとドメイン名のペアに対して発行されます。この値のいずれかが変更すると、証明書が無効になります。

2. VeriSign からの証明書があり、サーバーソフトウェアを変更している。

   Verivsftpd は、幅広く使用されている認証局で、特定のソフトウェア製品、IP アドレス、およびドメイン名の証明書を発行します。ソフトウェア製品を変更すると、証明書が無効になります。

上記のいずれの場合も、新しい証明書を入手する必要があります。このトピックの詳細については、「新しい鍵と証明書の生成」 を参照してください。

既存の鍵と証明書を使用する場合は、関連するファイルを /etc/pki/tls/private/ ディレクトリーと /etc/pki/tls/certs/ ディレクトリーに移動します。root で以下のコマンドを実行してこれを実行できます。

```
~]# mv key_file.key /etc/pki/tls/private/hostname.key
~]# mv certificate.crt /etc/pki/tls/certs/hostname.crt
```

次に、以下の行を /etc/httpd/conf.d/ssl.conf 設定ファイルに追加します。

```
SSLCertificateFile /etc/pki/tls/certs/hostname.crt
SSLCertificateKeyFile /etc/pki/tls/private/hostname.key
```

更新された設定を読み込むには、「サービスの再起動」 の説明に従って httpd サービスを再起動します。

例18.83 Red Hat Secure Web Server からの鍵と証明書の使用

```
~]# mv /etc/httpd/conf/httpsd.key /etc/pki/tls/private/penguin.example.com.key
~]# mv /etc/httpd/conf/httpsd.crt /etc/pki/tls/certs/penguin.example.com.crt
```

18.1.12. 新しい鍵と証明書の生成

新しい鍵と証明書のペアを生成するには、crypto-utils パッケージをシステムにインストールする必要があります。Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;LinuxRed Hat Enterprise Linux 6nbsp;6 では、genkey ユーティリティーで mod_ssl パッケージが必要になります。これらをインストールするには、root で次のコマンドを実行します。

```
~]# yum install crypto-utils mod_ssl
```

このパッケージは、SSL 証明書と秘密鍵を生成して管理するツールセットを提供し、鍵の生成プロセスをガイドする Red Hat Keypair Generation ユーティリティーである genkey が含まれます。

既存証明書の置き換え

有効な証明書を所有していて、それを新規のものに置き換える予定の場合は、異なるシリアル番号を指定します。これにより、クライアントのブラウザーがこの変更の通知を受けて予定どおりに新規の証明書に更新して、このページへのアクセスに失敗しないようにします。カスタムのシリアル番号で新しい証明書を作成するには、genkey の代わりに、以下のコマンドを使用します。

```
~]# openssl req -x509 -new -set_serial number -key hostname.key -out hostname.crt
```

以前に作成したキーを削除します。

システムに特定のホスト名用のキーファイルがすでにある場合は、genkey は起動を拒否します。このような場合には、root で以下のコマンドを使用して既存のファイルを削除します。

```
~]# rm /etc/pki/tls/private/hostname.key
```

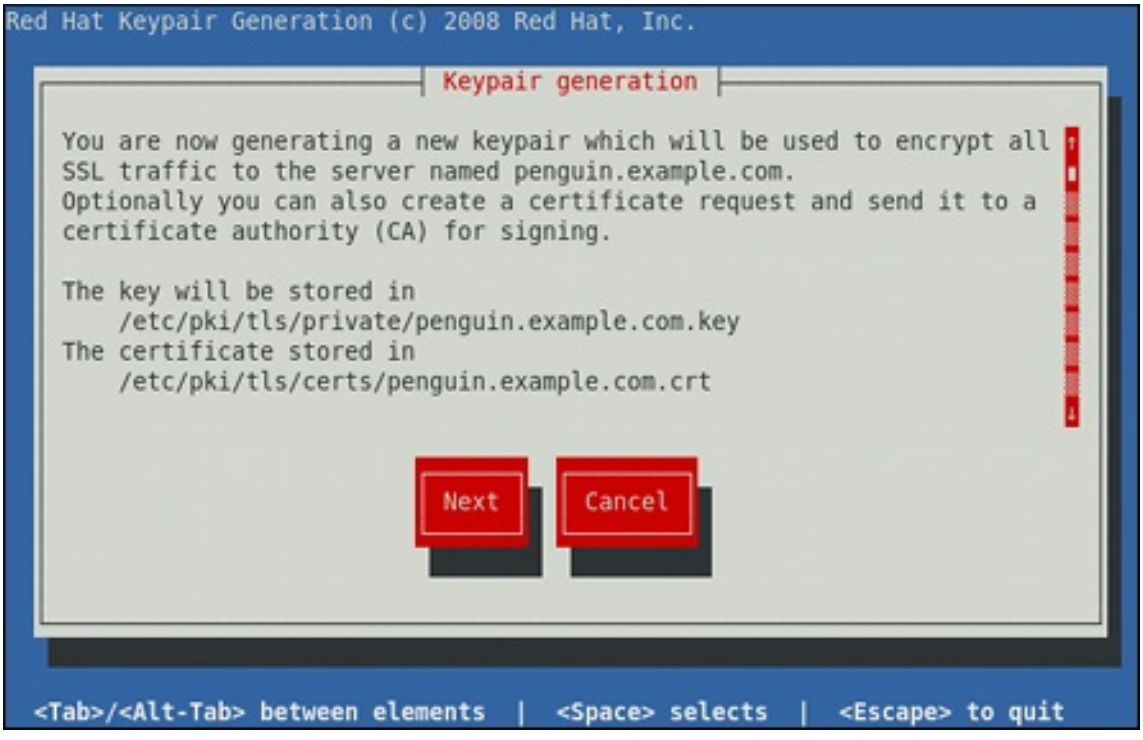
ユーティリティーを実行するには、root で genkey コマンドの後に、適切なホスト名（例：penguin.example.com）を付けて実行します。

```
~]# genkey hostname
```

鍵と証明書の生成を完了するには、以下の手順を行います。

1. 鍵と証明書を保存する場所を確認します。

図18.1 genkey ユーティリティーの実行



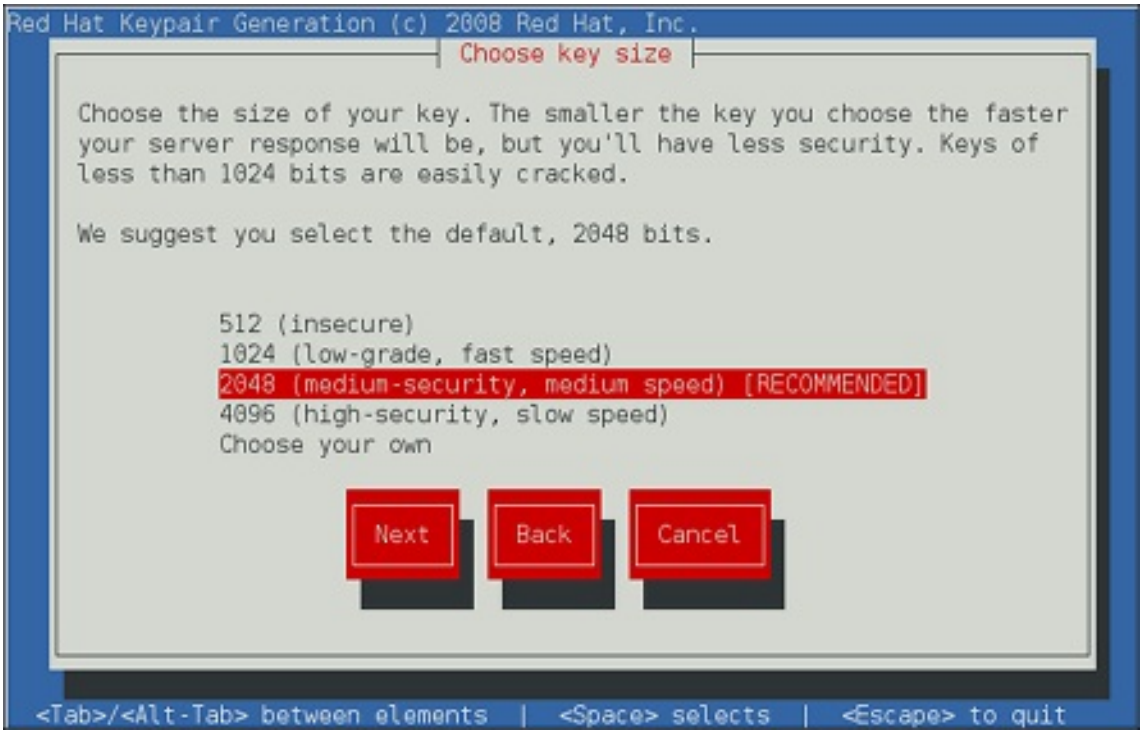
[D]

Tab キーを使用して Next (次へ) ボタンを選択してから、Enter キーを押すと次の画面を進みます。

2.

up と down の矢印キーを使用して、適切な鍵のサイズを選択します。鍵が大きくなればセキュリティーは向上しますが、サーバーの応答時間も長くなることに注意してください。NIST では、2048 bits の使用が推奨されています。『NIST Special Publication 800-131A』を参照してください。

図18.2 鍵のサイズ選択



[D]

終了したら、タブ キーを使用して 次へ ボタンを選択し、Enter キーを押すと、ランダムなビットの生成プロセスが開始します。選択した鍵のサイズによっては、時間がかかることもあります。

3.

証明書要求を認証局に送信するかどうかを決定します。

図18.3 証明書要求の生成

[D]

タブ キーを使用して Yes (はい) を選択し、証明書要求を組み立てるか、または No (いいえ) を選択して、自己署名の証明書を生成します。その後、Enter キーを押して、選択を確認します。

4.

Spacebar (スペースバー) キーを使用すると、秘密鍵の暗号化を有効にする ([*]) か、無効にする ([ ]) 選択ができます。

図18.4 秘密鍵の暗号化

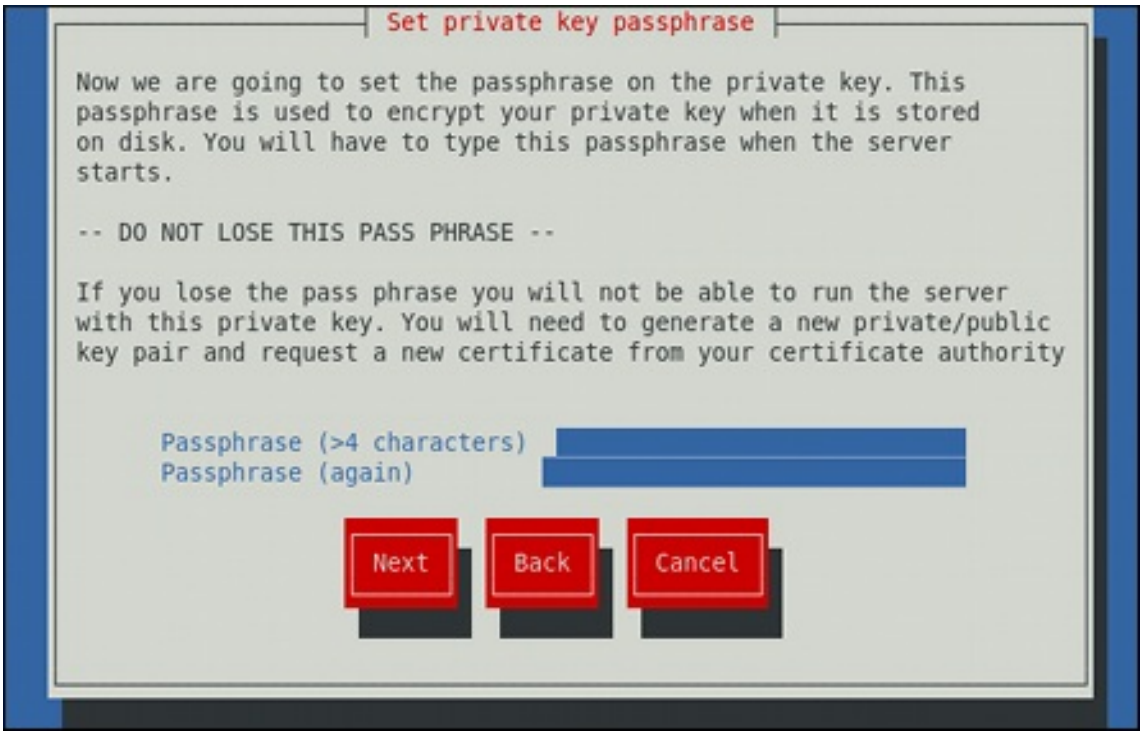
[D]

Tab キーを使用して Next (次へ) ボタンを選択してから、Enter キーを押すと次の画面を進みます。

5.

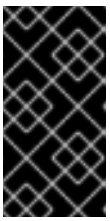
秘密鍵の暗号化を有効にしている場合は、適切なパスフレーズを入力します。セキュリティーの理由により入力時には文字が表示されませんが、最低でも 5 文字の長さが必要です。

図18.5 パスフレーズの入力



[D]

Tab キーを使用して Next (次へ) ボタンを選択してから、Enter キーを押すと次の画面を進みます。



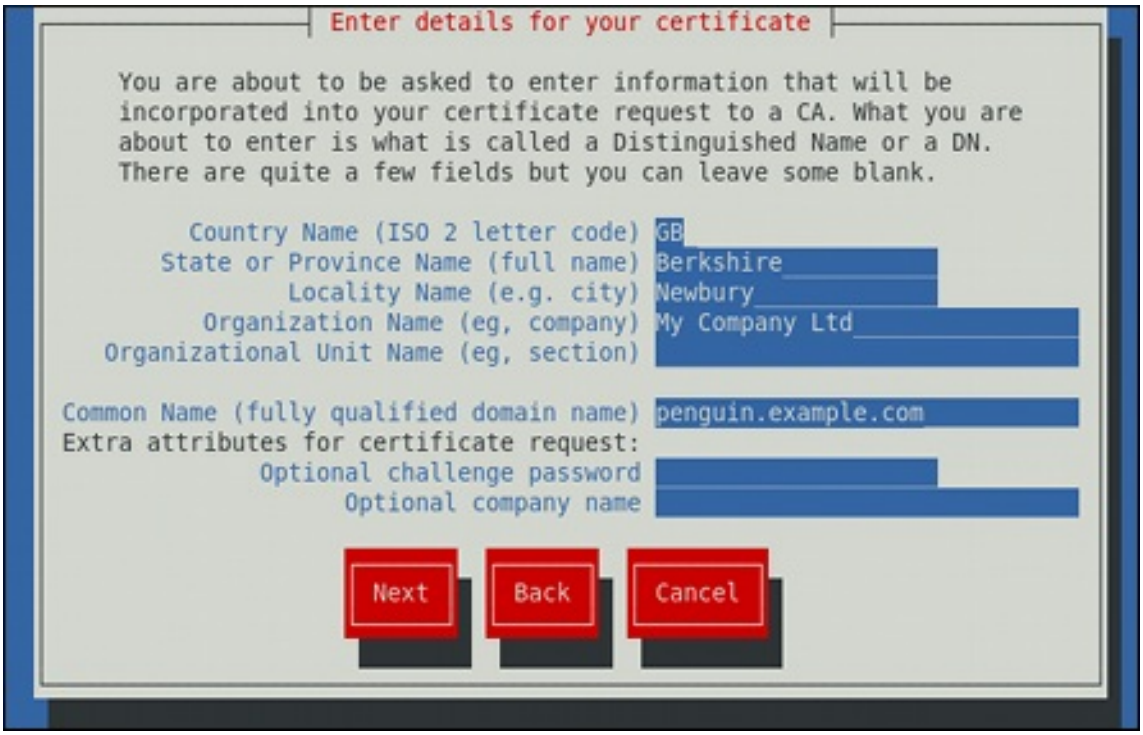
パスフレーズを忘れないようにしてください

サーバーを起動するには正しいパスフレーズの入力が必要です。それを紛失したり忘れたりした場合は、新しい鍵と証明書を生成する必要があります。

6.

証明書詳細のカスタマイズ

図18.6 証明書情報の指定



[D]

タブ キーを使用して Next ボタンを選択し、Enter を押すと鍵の生成が完了します。

7. *証明書要求の生成を有効にしていた場合は、それを認証局に送信するように求められます。*

図18.7 証明書要求を送信する方法の指示

[D]

Enter を押してシェルプロンプトに戻ります。

生成したら、鍵と証明書の場所を /etc/httpd/conf.d/ssl.conf 設定ファイルに追加します。

```
SSLCertificateFile /etc/pki/tls/certs/hostname.crt
SSLCertificateKeyFile /etc/pki/tls/private/hostname.key
```

最後に、「サービスの再起動」 の説明に従って httpd サービスを再起動して、更新された設定が読み込まれます。

18.1.13. コマンドラインを使用して HTTP 用および HTTPS 用にファイアウォールを設定

Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux は、デフォルトで HTTP および HTTPS トラフィックを許可しません。システムが Web サーバーとして機能するようにするには、必要に応じてポートとプロトコルを有効にします。HTTP のデフォルトポートは 80 で、HTTPS のデフォルトポートは 443 です。いずれの場合も、TCP がファイアウォールを通過することを許可されている必要があります。

コマンドラインで HTTP のポート 80 を有効にするには、root で以下のコマンドを発行します。

```
~]# lokkit --port=80:tcp --update
```

これにより、**--disabled** オプションで無効にされていない限り、ファイアウォールが再起動されることに注意してください。アクティブな接続は終了し、開始マシンでタイムアウトします。**lokkit --help** コマンドを使用して、組み込みのヘルプを表示します。

コマンドラインを使用して **HTTPS** のポート **443** を有効にするには、**root** で以下のコマンドを発行します。

```
~]# lokkit --port=443:tcp --update
```

これにより、**--disabled** オプションで無効にされていない限り、ファイアウォールが再起動されることに注意してください。アクティブな接続は終了し、開始マシンでタイムアウトします。サービスおよびそれらの関連付けられたポートの一覧は、**/etc/services** ファイルを参照してください。

管理ツールを使用して複数のインストールのために設定ファイルを準備する際には、ファイアウォール設定ファイルを直接編集すると便利です。設定ファイルの間違いがある場合は、予期せぬ結果が発生し、エラーが発生する可能性があり、ファイアウォール設定が適用されないことに注意してください。したがって、編集後に **/etc/sysconfig/system-config-firewall** ファイルの詳細を確認します。**/etc/sysconfig/system-config-firewall** の設定を適用するには、**root** で以下のコマンドを実行します。

```
~]# lokkit --update
```

たとえば、設定ファイルを編集して **HTTPS** がファイアウォールを通過できるようにするには、**root** ユーザーになり、以下の行を **/etc/sysconfig/system-config-firewall** に追加します。

```
--port=443:tcp
```

ファイアウォールが再読み込みされたり、システムが再起動されても、これらの変更は反映されないことに注意してください。**/etc/sysconfig/system-config-firewall** で変更を適用するには、**root** で以下のコマンドを実行します。

```
~]# lokkit --update
```

18.1.13.1. コマンドラインで着信 HTTPS および HTTPS のネットワークアクセスの確認

ファイアウォールが許可するものを確認するには、**root** で以下のコマンドを実行します。

```
~]# less /etc/sysconfig/system-config-firewall
```

```
# Configuration file for system-config-firewall

--enabled
--service=ssh
```

この例では、デフォルトのインストールでファイアウォールは有効になっていますが、HTTP と HTTPS は通過できません。

HTTP のデフォルトポートを有効にすると、以下の行が上記の行に加えて出力として表示されます。

```
--port=80:tcp
```

ファイアウォールが現在クライアントの受信 HTTP トラフィックを許可しているかどうかを確認するには、root で以下のコマンドを実行します。

```
~]# iptables -L -n | grep 'tcp.*80'
ACCEPT     tcp  --  0.0.0.0/0           0.0.0.0/0           state NEW tcp dpt:80
```

HTTPS のデフォルトポートを有効にすると、以下の行が上記の行に加えて出力として表示されます。

```
--port=443:tcp
```

ファイアウォールが現在クライアントの着信 HTTPS トラフィックを許可しているかどうかを確認するには、root で以下のコマンドを実行します。

```
~]# iptables -L -n | grep 'tcp.*443'
ACCEPT     tcp  --  0.0.0.0/0           0.0.0.0/0           state NEW tcp dpt:443
```

18.1.14. その他のリソース

Apache HTTP Server の詳細は、以下のリソースを参照してください。

インストールされているドキュメント

- *httpd(8): httpd サービスの man ページで、コマンドラインオプションの一覧が記載されます。*

- *genkey(1): crypto-utils パッケージが提供する genkey ユーティリティーの man ページです。*

インストール可能なドキュメント

- *http://localhost/manual/ - Apache HTTP Server の公式ドキュメントで、そのディレクティブと利用可能なモジュールの詳細を説明します。本書にアクセスするには、httpd-manual パッケージがインストールされ、Web サーバーが稼働している必要があることに注意してください。*

  ドキュメントにアクセスする前に、root で以下のコマンドを発行します。

```
~]# yum install httpd-manual
~]# service httpd graceful
```

オンラインドキュメント

- *http://httpd.apache.org/ - Apache HTTP Server の公式 Web サイトです。すべてのディレクティブおよびデフォルトモジュールの説明が記載されています。*

- *http://www.openssl.org/ - その他のドキュメント、よくある質問、メーリングリストへのリンクなどの役立つリソースを掲載した OpenSSL のホームページです。*

第19章 メールサーバー

Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux は、電子メールを提供し、アクセスするための高度なアプリケーションを多数提供します。本章では、現在使用されている最新の電子メールプロトコルと電子メールを送受信するプログラムについて説明します。

19.1. メールプロトコル

今日、電子メールはクライアント/サーバーのアーキテクチャーを使用して配信されています。電子メールのメッセージは、メールクライアントプログラムを使用して作成されます。次に、このプログラムがメッセージをサーバーに送信します。その後、サーバーはメッセージを受信者のメールサーバーに転送します。そこでメッセージは受信者の電子メールクライアントに渡されます。

このプロセスを有効にするために、各種の標準のネットワークプロトコルが異なるマシンによる (多くの場合、異なるオペレーティングシステムで、異なる電子メールプログラムを使用) 電子メールの送受信を可能にしています。

以下は、電子メールの転送に最も一般的に使用されているプロトコルです。

19.1.1. メール転送プロトコル

クライアントアプリケーションからサーバーへのメール配信、および送信元サーバーから宛先サーバーへのメール配信は Simple Mail Transfer Protocol (SMTP)によって処理されます。

19.1.1.1. SMTP

SMTP の第一の目的は、メールサーバー間における電子メールの転送です。ただし、これは、メールクライアントにも重要です。メールを送信するには、クライアントが送信メールサーバーにメッセージを送信し、配信先メールサーバーに接続します。このため、メールクライアントの設定時に SMTP サーバーを指定する必要があります。

Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux では、ユーザーはローカルマシンで SMTP サーバーを設定してメール配信を処理できます。ただし、送信メール用にリモート SMTP サーバーを設定することも可能です。

SMTP プロトコルに関して重要なのは認証が不要である点です。これにより、インターネット上の誰でも、個人や大規模なグループに対してでも電子メールを送信できます。これは、ジュークメールやスパム を可能にする SMTP のこの特徴です。リレー制限を課すと、インターネット上の任意のユー

ザーが、ご使用の SMTP サーバーを介してインターネット上の別のサーバーへ電子メールを送信することが制限されます。このような制限を課さないサーバーは、オープンリレー サーバーと呼ばれます。

Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux は、Postfix および Sendmail SMTP プログラムを提供します。

19.1.2. メールアクセスプロトコル

メールサーバーから電子メールを取得するために電子メールクライアントアプリケーションが使用する主要なプロトコルには、POP(Post Office Protocol)とIMAP(Internet Message Access Protocol)の 2 つがあります。

19.1.2.1. POP

Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux のデフォルトの POP サーバーは Dovecot で、dovecot パッケージで提供されます。

DOVECOT パッケージのインストール

Dovecot を使用するには、最初に root で以下を実行して dovecot パッケージがインストールされていることを確認します。

```
~]# yum install dovecot
```

yum を使用したパッケージのインストールは 「パッケージのインストール」 を参照してください。

POP サーバーを使用する場合、電子メールメッセージは電子メールクライアントアプリケーションによってダウンロードされます。デフォルトでは、ほとんどの POP メールクライアントは、電子メールサーバーのメッセージが正常に転送されるとそのメッセージを削除するように自動的に設定されますが、通常は変更できます。

POP は、電子メールのファイル添付を可能にするMIME（多目的インターネットメール拡張）などの 重要なインターネットメッセージング標準と完全に互換性があります。

POP は、電子メールを読むためのシステムが 1 つであるユーザーにとって最適に機能します。また、インターネットやメールサーバーを持つネットワークに常時接続していないユーザーにもうまく機能します。ネットワーク速度が遅いユーザーの場合は、POP はクライアントプログラムに対して、認

証を行った上で各メッセージのコンテンツ全体をダウンロードするよう要求します。このプロセスは、メッセージに大きなファイルが添付されている場合に長時間かかる場合があります。

標準 POP プロトコルの最新版は POP3 です。

ただし、あまり使用されていない POP プロトコルのバリアントにも様々な種類があります。

- APOP: MD5 認証を使用した POP3 です。暗号化されていないパスワードを送信するのではなく、エンコードされたユーザーパスワードのハッシュが電子メールクライアントからサーバーに送信されます。
- KPOP: Kerberos 認証を使用した POP3 です。
- RPOP: RPOP 認証を使用した POP3 です。これは、パスワードに似たユーザーごとの ID を使用し、POP 要求を認証します。ただし、この ID は暗号化されていないため、RPOP は標準の POP よりも安全ではありません。

セキュリティーを強化するには、クライアント認証とデータ転送セッションに Secure Socket Layer (SSL)暗号化を使用できます。これは、pop3s サービスまたは stunnel アプリケーションを使用して有効にできます。メール通信のセキュリティー保護に関する詳細は、「通信のセキュリティー保護」 を参照してください。

19.1.2.2. IMAP

Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux のデフォルトの IMAP サーバーは Dovecot で、dovecot パッケージで提供されます。Dovecot のインストール方法は 「POP」 を参照してください。

IMAP メールサーバーを使用する場合は電子メールメッセージはサーバーに残るため、ユーザーはメッセージの読み取りや削除が可能です。また、IMAP により、クライアントアプリケーションがサーバー上でメールディレクトリーの作成、名前変更、削除を行い電子メールを整理および保存することもできます。

IMAP は、複数のマシンを使用して電子メールにアクセスするユーザーに特に役立ちます。このプロトコルでは、メッセージが開封されるまでは、電子メールのヘッダー情報しかダウンロードされず帯域幅を節減できるため、低速な接続でメールサーバーに接続するユーザーにも便利です。ユーザーは、メッセージを表示またはダウンロードすることなく削除することも可能です。

便宜上、IMAP クライアントアプリケーションはメッセージのコピーをローカルでキャッシュすることが可能です。そのため、ユーザーは IMAP サーバーに直接接続していないときに、既読メッセージを閲覧できます。

IMAP は POP と同様に、電子メールのファイル添付を可能にする MIME などの重要なインターネットメッセージング標準と完全に互換性があります。

セキュリティーを強化するには、クライアント認証とデータ転送セッションに SSL 暗号化を使用できます。これは、imaps サービスまたは stunnel プログラムを使用して有効にできます。メール通信のセキュリティー保護に関する詳細は、「通信のセキュリティー保護」 を参照してください。

無償や商用の IMAP クライアントおよびサーバーは他にも提供されています。これらの多くは、IMAP プロトコルを拡張し、追加機能を提供します。

19.1.2.3. Dovecot

IMAP および POP3 プロトコルを実装する imap- login および pop 3-login プロセスは、dovecot パッケージに含まれるマスターの dovecot デーモンにより起動します。IMAP および POP の使用は、/etc/dovecot/dovecot.conf 設定ファイルで設定されます。デフォルトでは、dovecot は、SSL を使用するセキュアなバージョンとともに IMAP および POP3 を実行します。POP を使用するように dovecot を設定するには、以下の手順を実行します。

1. protocols 変数がコメント解除されていて（行頭のハッシュ記号(#)を削除）、pop3 引数を含むように /etc/dovecot/dovecot.conf 設定ファイルを編集します。以下に例を示します。

```
protocols = imap pop3 lmtp
```

   protocols 変数がコメントアウトされている場合、dovecot は上記のようにデフォルト値を使用します。

2. 以下のコマンドを実行して、現行セッションで変更を可能にします。

```
~]# service dovecot restart
```

3. この変更を次回の再起動後に有効にするには、以下のコマンドを実行します。

```
~]# chkconfig dovecot on
```

DOVECOT サービスが POP3 サーバーを起動する

dovecot は IMAP サーバーを起動したことしか報告せず、POP3 サーバーも起動することに注意してください。

SMTP とは異なり、IMAP と POP3 の両方では、接続クライアントがユーザー名とパスワードを使用して認証する必要があります。デフォルトでは、両方のプロトコルのパスワードは、暗号化されていないネットワーク上で渡されます。

dovecot で SSL を設定するには、以下を実行します。

- */etc/dovecot/conf.d/10-ssl.conf 設定を編集して、ssl_cipher_list 変数がコメント解除されていることを確認し、:!SSLv3 を追加します。*

```
ssl_cipher_list = ALL:!LOW:!SSLv2:!EXP:!aNULL:!SSLv3
```

これらの値により、dovecot は、安全でないことがわかっている SSL バージョン 2 および 3 を回避するようになります。これは、『POODLE: SSLv3 脆弱性(CVE-2014-3566)で説明されている脆弱性』 が原因です。詳細は、『Postfix および Dovecot の POODLE SSL 3.0 脆弱性(CVE-2014-3566)の解決』 方法を参照してください。

- */etc/pki/dovecot/dovecot-openssl.cnf 設定ファイルを必要に応じて編集します。ただし、標準的なインストールではこのファイルへの変更は必要ありません。*

- */etc/pki/dovecot/certs/dovecot.pem ファイルおよび /etc/pki/dovecot/private/dovecot.pem ファイルの名前変更、移動、または削除を行います。*

- */usr/libexec/dovecot/mkcert.sh スクリプトを実行して、dovecot の自己署名証明書を作成します。これらの証明書は、/etc/pki/dovecot/certs および /etc/pki/dovecot/private ディレクトリーにコピーされます。変更を実装するには、dovecot を再起動します。*

```
~]# service dovecot restart
```

dovecot の詳細は http://www.dovecot.org でオンラインで参照できます。

19.2. 電子メールプログラムの分類

一般的に、すべての電子メールアプリケーションは 3 つのタイプのうち 1 つ以上に分類されます。それぞれの分類は、電子メールメッセージの移動および管理のプロセスにおいてそれぞれ特定の役割を果たします。大半のユーザーはメッセージの送受信に使用する特定の電子メールプログラムだけを認識しますが、電子メールを正しい送信先に届けるにはすべての電子メールプログラムが重要になります。

19.2.1. メール転送エージェント (Mail Transport Agent)

MTA（ メール転送エージェント ）は、SMTP を使用してホスト間で電子メールメッセージを転送します。メッセージは目的の送信先に移動する時、様々な MTA に関わることがあります。

マシン間のメッセージ配信は簡単に見えるかもしれませんが、配信のためにある MTA がメッセージを受け入れることが可能か、または受け入れるべきかを判断する過程全体は非常に複雑です。さらに、スパムの問題により、特定の MTA の使用は通常 MTA の設定または MTA が置かれるネットワークのアクセス設定により制限されます。

最新の電子メールクライアントプログラムの多くは、電子メールの送信時に MTA として機能します。ただし、このアクションは、実際の MTA のロールと混同しないようにしてください。メールクライアントプログラムは、MTA などの電子メールを送信できる唯一の理由は、アプリケーションを実行するホストに独自の MTA がないためです。これは特に、UNIX ベースのオペレーティングシステム上の電子メールクライアントプログラムに適しています。ただし、これらのクライアントプログラムは、アウトバウンドメッセージを MTA にのみ送信し、メッセージを目的の受信者のメールサーバーに直接配信しません。

Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux は Postfix と Sendmail の 2 つの MTA を提供しているため、電子メールクライアントプログラムは MTA として動作する必要はありません。Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux には、Fetchmail と呼ばれる特別な目的の MTA も含まれています。

Postfix、Sendmail、Fetchmail の詳細は 「メール転送エージェント (MTA)」 を参照してください。

19.2.2. メール配信エージェント (MDA)

MDA（メール 配信エージェント ）は MTA によって呼び出され、適切なユーザーのメールボックスに受信メールをファイルします。多くの場合、MDA は実際には mail や Procmail などの LDA （ローカル配信エージェント）です。

電子メールクライアントアプリケーションが読み取り可能なポイントに配信されるメッセージを実際に処理するプログラムは、いずれも MDA と見なすことができます。このため、一部の MTA（Sendmail、Postfix など）は、ローカルユーザーのメールスプールファイルに新しい電子メールメッセージを追加するときに MDA の役割を埋めることができます。通常、MDA はシステム間での

メッセージの転送やユーザーインターフェースの提供は行いません。MDA は、ローカルマシン上でメッセージの配信と並べ替えを行い、電子メールクライアントアプリケーションがアクセスできるようにします。

19.2.3. メールユーザーエージェント

Mail User Agent (MUA)は、電子メールクライアントアプリケーションと同義語です。MUA は、最低でも電子メールメッセージの読み取りと作成を可能にするプログラムです。多くの MUA は、POP プロトコルまたは IMAP プロトコルを介してメッセージを取得したり、メッセージを格納するためのメールボックスを設定し、送信メッセージを MTA に送信することができます。

MUA は、Evolution のようなグラフィカルな場合と、ペンションなどの単純なテキストベースのインターフェースを持つ ことも できます。

19.3. メール転送エージェント (MTA)

Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux は、Postfix と Sendmail の 2 つの主要 MTA を提供します。Postfix はデフォルトの MTA として設定されますが、デフォルトの MTA を Sendmail に簡単に切り替えることができます。デフォルトの MTA を Sendmail に切り替えるには、Postfix をアンインストールするか、次のコマンドを使用して Sendmail に切り替えます。

```
~]# alternatives --config mta
```

以下の形式のコマンドを使用して、希望のサービスを有効または無効にすることもできます。

```
chkconfig service_name on | off
```

19.3.1. postfix

当初、IBM のセキュリティエキスパートであるプログラマーの Wietse Venema 氏によって開発された Postfix は、Sendmail 互換の MTA で、セキュア、高速、かつ容易に設定できるように設計されています。

セキュリティーを強化するために、Postfix はモジュラー設計を使用します。この場合、権限が限定された小さなプロセスは、マスター デーモンにより起動します。より小さく、権限の低いプロセスは、メール配信の様々な段階に関連する非常に特殊なタスクを実行してルートディレクトリーが変更された環境で稼働し、攻撃の影響を制限します。

Postfix がローカルコンピュータ以外のホストからのネットワーク接続を受け入れるよう設定するには、設定ファイルを多少変更するだけでできます。さらに、より複雑なニーズのために、Postfix は

様々な設定オプションだけでなくサードパーティのアドオンも提供するため、多用途でフル機能の MTA となっています。

Postfix の設定ファイルは人間に解読可能で、250 以上のディレクティブに対応しています。Sendmail とは異なり、変更を反映するためにマクロ処理は必要なく、また最も一般的に使用されるオプションの大部分は、多数のコメントが付いたファイルで説明されています。

19.3.1.1. Postfix のデフォルトインストール

Postfix 実行可能ファイルは /usr/sbin/postfix です。このデーモンは、メール配信の処理に必要なすべての関連プロセスを起動します。

Postfix は設定ファイルを /etc/postfix/ ディレクトリーに格納します。以下は、一般的に使用されるその他のファイルの一覧です。

- access: アクセス制御に使用します。このファイルは Postfix に接続可能なホストを指定します。
- main.cf: グローバル Postfix 設定ファイル設定オプションの大部分がこのファイルで指定されています。
- master.cf: メール配信を完了するために Postfix が様々なプロセスとやりとりを行う方法を指定します。
- transport: 電子メールアドレスをリレーホストにマッピングします。

aliases ファイルは /etc/ ディレクトリーにあります。このファイルは Postfix と Sendmail 間で共有されます。ユーザー ID エイリアスを記述するメールプロトコルが必要な設定可能な一覧です。

他のクライアント用のサーバーとしての POSTFIX の設定

デフォルトの /etc/postfix/main.cf ファイルでは、Postfix はローカルコンピューター以外のホストからのネットワーク接続を受け付けられないように設定されています。Postfix を他のクライアント用のサーバーとして設定する方法は 「Postfix の基本設定」 を参照してください。

/etc/ postfix ディレクトリーにある設定ファイルのオプションに変更を加えた後には、postfix サービスを再起動してこれらの変更を適用します。

```
~]# service postfix restart
```

19.3.1.2. Postfix の基本設定

デフォルトでは、Postfix はローカルホスト以外のホストからのネットワーク接続を受け付けません。ネットワーク上の他のホストに対するメール配信を有効にするには、root で以下のコマンドを実行します。

- vi などのテキストエディターで /etc/postfix/main.cf ファイルを編集します。
- mydomain 行のコメントを解除して、ハッシュ記号(#)を削除し、domain.tld を、example.com などのメールサーバーがサービスを提供しているドメインに置き換えます。
- myorigin = $mydomain 行のコメントを解除します。
- myhostname 行のコメントを解除し、host.domain.tld をマシンのホスト名に置き換えます。
- mydestination = $myhostname, localhost.$mydomain 行のコメントを解除します。
- mynetworks 行のコメントを解除して、168 .100.189.0/28 を、サーバーに接続可能なホストの有効なネットワーク設定に置き換えます。
- inet_interfaces = all 行のコメントを解除します。
- inet_interfaces = localhost 行をコメント化します。
- postfix サービスを再起動します。

これらの手順が完了したら、ホストは配信のため外部の電子メールを受け入れるようになります。

Postfix には様々な設定オプションがあります。Postfix の設定方法を学習する最適な方法の 1 つは、/etc/postfix/main.cf 設定ファイルのコメントを読むことです。Postfix 設定、SpamAssassin 統合、/etc/postfix/main.cf パラメーターの詳細などの補足情報は http://www.postfix.org/ で参照できます。

19.3.1.2.1. Postfix がトランスポート層セキュリティーを使用するように設定する

Transport Layer Security(TLS)を 使用するように postfix を設定する方法は、Red Hat ナレッジベースソリューション「『How to configure postfix with TLS?』」を参照してください。

重要

『Postfix および Dovecot における POODLE SSL 3 . 0 脆弱性(CVE-2014-3566)の解決で説明されている脆弱性』 のため、Red Hat では、SSL を無効にすることを推奨します。また、TLSv1. 1 または TLSv1.2 のみを使用することを推奨します。後方互換性は、TLSv1.0 を使用して実現できます。Red Hat がサポートする多くの製品は SSLv2 プロトコルまたは SSLv3 プロトコルを使用できます。ただし、SSLv2 または SSLv3 の使用が強く推奨されます。

19.3.1.3. LDAP での Postfix の使用

Postfix は LDAP ディレクトリーをさまざまなルックアップテーブルのソースとして利用できます（エイリアス、仮想、正規 など）。これにより LDAP は階層的なユーザー情報を保存でき、Postfix は LDAP クエリーの結果を必要な場合にのみ知らされます。この情報をローカルに保存しないことで、管理者は容易に管理することができます。

19.3.1.3.1. /etc/aliases ルックアップのサンプル

以下は、LDAP を使用して /etc/aliases ファイルを検索する基本的な例です。/etc/postfix/main.cf ファイルに以下の内容が含まれていることを確認してください。

```
alias_maps = hash:/etc/aliases, ldap:/etc/postfix/ldap-aliases.cf
```

/etc/postfix/ldap-aliases.cf ファイルがない場合は作成し、以下の内容を追加します。

```
server_host = ldap.example.com
search_base = dc=example, dc=com
```

ここで ldap.example.com、example、および com は、既存の利用可能な LDAP サーバーの仕様に置き換える必要があるパラメーターです。

/ETC/POSTFIX/LDAP-ALIASES.CF ファイル

/etc/postfix/ldap-aliases.cf ファイルは、LDAP SSL と STARTTLS を有効にするパラメーターなど、さまざまなパラメーターを指定できます。詳細は、ldap_table(5) man ページを参照してください。

LDAP の詳細は、「OpenLDAP」 を参照してください。

19.3.2. Sendmail

Sendmail の主な目的は、他の MTA と同様に、通常 SMTP プロトコルを使用して、ホスト間で電子メールを安全に転送することです。ただし、Sendmail は高度な設定が可能で、使用されるプロトコルを含め、電子メールの処理方法をほぼすべての方法で制御できます。多くのシステム管理者は、パワーとスケーラビリティーにより Sendmail を MTA として使用することを選択しています。

19.3.2.1. 用途と制約

認識すべき重要な点は、Sendmail ができないことではなく、Sendmail が何であるか、何ができるのかということです。複数の役割を果たすモノリシックなアプリケーションの時代には、Sendmail は組織内で電子メールサーバーを稼働するために必要な唯一のアプリケーションと思われるかもしれません。Sendmail は各ユーザーのディレクトリーにメールを送信し、ユーザー用にアウトバウンドメールを送信できるため、技術的にはこれが当てはまります。Sendmail はメールを各ユーザーのディレクトリーにスプールして、ユーザーに送信メールを配信できるからです。ユーザーは通常、POP または IMAP を使用してメッセージをローカルマシンにダウンロードする MUA を使用してメールと対話したい場合があります。ユーザーは通常、POP または IMAP を使用する MUA で電子メールとやりとりを行い、ローカルマシンにメッセージをダウンロードする方法を望みます。こうした他のアプリケーションを Sendmail と連動させることは可能ですが、実際、それらが存在する理由は異なり、独立して機能することができます。

Sendmail で設定すべき、また設定できるすべての用途の説明は、本セクションの対象外となります。Sendmail には文字どおり数百におよぶ様々なオプションやルールセットがあるため、Sendmail のあらゆる機能や問題修正方法に関する専門的な資料が多くあります。Sendmail リソースの一覧 「その他のリソース」 についてはを参照してください。

本セクションでは、Sendmail でデフォルトでインストールされたファイルを確認し、不要な電子メール（スパム）を停止する方法や LDAP(Lightweight Directory Access Protocol)で Sendmail を拡張する方法など、基本的な設定変更を確認します。

19.3.2.2. Sendmail のデフォルトのインストール

Sendmail を使用するには、root で以下を実行し、sendmail パッケージがシステムにインストールされていることを確認します。

```
~]# yum install sendmail
```

Sendmail を設定するには、root で以下を実行し、sendmail-cf パッケージがシステムにインストールされていることを確認します。

```
~]# yum install sendmail-cf
```

yum を使用したパッケージのインストールは 「パッケージのインストール」 を参照してください。

Sendmail を使用する前に、デフォルトの MTA が Postfix から切り替わっている必要があります。デフォルトの MTA の切り替え方法は、「メール転送エージェント (MTA)」 を参照してください。

Sendmail 実行可能ファイルは /usr/sbin/sendmail です。

Sendmail の長さと詳細な設定ファイルは /etc/mail/sendmail.cf です。sendmail.cf ファイルを直接編集しないでください。Sendmail に設定を変更するには、/etc/mail/sendmail.mc ファイルを編集し、元の /etc/mail/sendmail.cf ファイルをバックアップし、以下の代替を使用して新しい設定ファイルを生成します。

- /etc/mail/ に含まれる makefile を使用して、新しい /etc/mail/sendmail.cf 設定ファイルを作成します。

  ```
  ~]# make all -C /etc/mail/
  ```

  /etc/mail （db ファイル）内のその他の生成されたファイルはすべて、必要に応じて再生成されます。古い makemap コマンドは引き続き利用できます。make コマンドは、sendmail サービスを起動または再起動するたびに自動的に使用されます。

- m4 マクロプロセッサーを使用して、新しい /etc/mail/sendmail.cf を作成することもできます。m4 マクロプロセッサーはデフォルトでインストールされません。/etc/mail/sendmail.cf を作成する前に、root で m4 パッケージをインストールします。

```
~]# yum install m4
```

Sendmail の設定に関する詳細は 「Sendmail の一般的な設定変更」 を参照してください。

以下のような様々な Sendmail 設定ファイルが /etc/mail/ ディレクトリーにインストールされています。

- access: 電子メールの送信に Sendmail を使用できるシステムを指定します。
- domaintable: ドメイン名のマッピングを指定します。
- local-host-names: ホストのエイリアスを指定します。
- mailertable: 特定のドメインのルーティングを上書きする方法を指定します。
- virtusertable: ドメイン固有のエイリアス形式を指定し、1 台のマシンに複数の仮想ドメインをホストできるようにします。

アクセス、domaintable、mailertable、virtusertable などの /etc/mail/ の設定ファイルの中には、Sendmail が設定変更を使用できるようになる前に、実際にデータベースファイルに情報を保存する必要があります。データベースファイルの設定に変更を追加する場合は、root で以下のコマンドを実行します。

```
~]# makemap hash /etc/mail/<name> < /etc/mail/<name>
```

ここで、<name > は更新する設定ファイルの名前を表します。また、以下のコマンドを実行して変更を有効にするために sendmail サービスを再起動することもできます。

```
~]# service sendmail restart
```

たとえば、すべての電子メールを example.com ドメインに対応するには、以下 の行を virtusertable ファイルに追加します。

```
@example.com bob@other-example.com
```

変更を終了するには、virtusertable.db ファイルを更新する必要があります。

```
~]# makemap hash /etc/mail/virtusertable < /etc/mail/virtusertable
```

Sendmail は、新しい設定を含む更新された virtusertable.db ファイルを作成します。

19.3.2.3. Sendmail の一般的な設定変更

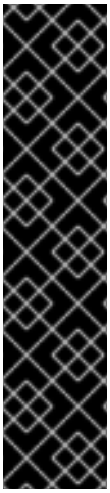
Sendmail 設定ファイルを変更する場合は、既存ファイルを編集せずに、新たに /etc/mail/sendmail.cf ファイルを生成するのが最適な方法です。

コンテンツを変更する前に SENDMAIL.CF ファイルをバックアップします。

sendmail.cf ファイルを置き換えたり変更したりする前に、バックアップコピーを作成してください。

希望する機能を Sendmail に追加するには、root で /etc/mail/sendmail.mc ファイルを編集します。編集が終了したら、sendmail サービスを再起動します。m4 パッケージがインストールされている場合は、m4 マクロプロセッサーが新しい sendmail.cf 設定ファイルを自動的に生成します。

```
~]# service sendmail restart
```



他のクライアント用のサーバーとしての SENDMAIL の設定

デフォルトの sendmail.cf ファイルでは、Sendmail はローカルコンピューター以外のホストからのネットワーク接続を受け入れることができません。Sendmail を他のクライアント用のサーバーとして設定するには、/etc/mail/sendmail.mc ファイルを編集して、DAEMON_OPTIONS ディレクティブの Addr= オプションで指定されているアドレスを 127.0.0.1 からアクティブなネットワークデバイスの IP アドレスに変更するか、行頭に dnl を付けて DAEMON_OPTIONS ディレクティブをすべてコメントアウトします。終了したら、サービスを再起動して /etc/mail/sendmail.cf を再生成します。

```
~]# service sendmail restart
```

Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux のデフォルト設定は、ほとんど

の SMTP専用サイトで機能します。ただし、U CP (UNIX-to-UNIX Copy Protocol)サイトでは動作しません。UUCP メール転送を使用している場合は、/etc/mail/sendmail.mc ファイルを再設定し、新しい /etc/mail/sendmail.cf ファイルを生成する必要があります。

/usr/share/sendmail-cf ディレクトリー下のディレクトリーにあるファイルを編集する前に、/usr/share/sendmail-cf/README ファイルを確認してください。このファイルは、/etc/mail/sendmail.cf ファイルの今後の設定に影響を及ぼす可能性があるためです。

19.3.2.4. マスカレーディング

一般的な Sendmail の設定の 1 つとして、1 台のマシンがネットワーク上の全マシンのメールのゲートウェイとして機能するように設定する方法があります。たとえば、ある企業が mail.example.com という名前のマシンですべての電子メールを処理して、すべての送信メールに一貫した返信アドレスを割り当てるとします。

このような状況では、Sendmail サーバーは、返信アドレスが user@example.com ではなく user@host.example.com になるように、企業のネットワーク上のマシン名をマスカレードする必要があります。

これを行うには、以下の行を /etc/mail/sendmail.mc に追加します。

```
FEATURE(always_add_domain)dnl
FEATURE(`masquerade_entire_domain')dnl
FEATURE(`masquerade_envelope')dnl
FEATURE(`allmasquerade')dnl
MASQUERADE_AS(`example.com.')dnl
MASQUERADE_DOMAIN(`example.com.')dnl
MASQUERADE_AS(example.com)dnl
```

m4 マクロプロセッサーを使用して新しい sendmail.cf ファイルを生成した後に、この設定により、ネットワーク内のメールがすべて example.com から送信されたかのように表示されます。

19.3.2.5. Spam の停止

電子メールのスパムは、通信を要求したことがないユーザーから受信した、不要な迷惑メールとして定義することができます。これは、破壊的でコストがかかる、広く蔓延したインターネット通信標準の悪用です。

Sendmail を使用すると、迷惑メールの送信に使用されている新たなスパム技術を比較的に簡単にブロックすることができます。さらに、数多くの一般的なスパム手法もデフォルトでブロックします。

sendmail で利用可能な主要なアンチスパム機能は ヘッダーチェック、リレー拒否 （バージョン 8.9）、アクセスデータベース、送信者情報の確認 です。

たとえば、リレーとも呼ばれる SMTP メッセージの転送は、Sendmail バージョン 8.9 以降デフォルトでは無効になっています。この変更前に、Sendmail はメールホスト(x.edu)に対して、ある当事者(y.com)からのメッセージを受け入れるよう指示し、それらを別の当事者(z.net)に送信していました。しかし、現在は任意のドメインがサーバーを介してメールをリレーするよう Sendmail を設定する必要があります。リレードメインを設定するには、/etc/mail/relay-domains ファイルを編集して Sendmail を再起動します。

```
~]# service sendmail restart
```

ただし、ユーザーはインターネット上のサーバーからスパムを送信することもできます。その場合は、/etc/mail/access ファイルで利用可能な Sendmail のアクセス制御機能を使用して、不要なホストからの接続を阻止することができます。以下の例は、このファイルを使用したブロックの方法と Sendmail サーバーへのアクセスを具体的に許可する方法を示しています。

```
badspammer.com ERROR:550 "Go away and do not spam us anymore" tux.badspammer.com OK
10.0 RELAY
```

この例では、basp ammer.com から送信 された電子メールはいずれも 550 RFC-821 準拠のエラーコードでブロックされ、メッセージは送り返されます。tux.badspammer.com サブドメインから送信される電子メールは受け入れられます。最後の行は、10.0..* ネットワークから送信された電子メールは、メールサーバーを介してリレーできることを示しています。

/etc/mail/access.db ファイルはデータベースであるため、makemap コマンドを使用して変更を更新します。これは、root で以下のコマンドを使用して行います。

```
~]# makemap hash /etc/mail/access < /etc/mail/access
```

メッセージヘッダー分析により、ヘッダーの内容に基づいてメールを拒否することができます。SMTP サーバーは、電子メールの取り組みに関する情報をメッセージヘッダーに保存します。メッセージがある MTA から別の MTA に移動すると、それぞれは他のすべての Received ヘッダーの上に Received ヘッダーに配置されます。この情報はスパムで変更可能である可能性があることに注意してください。

上記の例は、アクセスの許可や阻止に関する Sendmail が持つ機能のほんの一部です。詳細と例は、/usr/share/sendmail-cf/README ファイルを参照してください。

Sendmail は、メールの配信時に Procmail MDA を呼び出すため、SpamAssassin のようなスパムフィルタリングプログラムを使用して、ユーザーに対してスパムを識別してファイルに保存することも

可能です。SpamAssassin の詳細な使用方法は、「spam フィルター」 を参照してください。

19.3.2.6. LDAP での Sendmail の使用

LDAP の使用は、大規模なグループから特定のユーザーに関する特定の情報を検索する、非常に迅速かつ強力な方法です。たとえば、LDAP サーバーを使用すると、一般的な企業ディレクトリーから特定の電子メールアドレスをユーザーの名で検索できます。この種の実装では、LDAP はほとんど Sendmail から分離されています。LDAP は階層的なユーザー情報を保存し、Sendmail は事前にアドレスが入力された電子メールメッセージで LDAP クエリーの結果のみを知らています。

ただし、Sendmail は LDAP とのより優れた統合をサポートします。この場合、LDAP を使用して、中規模レベルの組織をサポートするさまざまなメールサーバーで、/etc/aliases や /etc/mail/virtusertables などのメンテナンスされたファイルを置き換えます。つまり、LDAP はメールルーティングレベルを Sendmail と、その別個の設定ファイルから、さまざまなアプリケーションで活用できる強力な LDAP クラスターに抽象化します。

Sendmail の現行版は LDAP に対応しています。LDAP を使用して Sendmail を拡張するには、最初に OpenLDAP などの LDAP サーバーを取得し、適切な設定を行います。次に、/etc/mail/sendmail.mc を編集して以下を追加します。

```
LDAPROUTE_DOMAIN('yourdomain.com')dnl
FEATURE('ldap_routing')dnl
```

詳細設定

これは LDAP を使用した非常に基本的な Sendmail の設定のみです。LDAP の実装によっては、これとは大幅に異なる可能性があります。特に、共通の LDAP サーバーを使用するようにいくつかの Sendmail マシンを設定する場合がそうです。

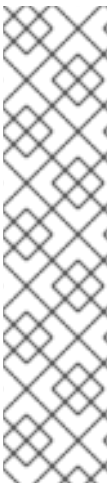
詳細な LDAP ルーティング設定の指示と例は、/usr/share/sendmail-cf/README を参照してください。

次に、m4 マクロプロセッサーを実行し、Sendmail を再起動して /etc/mail/sendmail.cf ファイルを再作成します。手順については 「Sendmail の一般的な設定変更」 を参照してください。

LDAP の詳細は、「OpenLDAP」 を参照してください。

19.3.3. Fetchmail

Fetchmail は、リモートサーバーから電子メールを取得してローカルの MTA に配信する MTA です。多くのユーザーは、リモートサーバー上にあるメッセージをダウンロードするプロセスと、MUA で電子メールを読み取り、整理するプロセスを別々にする機能性を評価してます。ダイアルアップユーザーのニーズを念頭に置いて設計された Fetchmail は、POP3 や IMAP などのプロトコルを使用して、メールスプールファイルに接続し、すべての電子メールメッセージを迅速にダウンロードします。必要に応じて、電子メールメッセージを SMTP サーバーに転送することもできます。



FETCHMAIL パッケージのインストール

Fetchmail を使用するには、最初に root で以下を実行し、fetchmail パッケージがシステムにインストールされていることを確認します。

```
~]# yum install fetchmail
```

yum を使用したパッケージのインストールは 「パッケージのインストール」 を参照してください。

Fetchmail は、各ユーザーのホームディレクトリー内の .fetchmailrc ファイルを使用して、各ユーザーに設定されています。これが存在しない場合は、ホームディレクトリーに .fetchmailrc ファイルを作成します。

Fetchmail は .fetchmailrc ファイル内の詳細設定を使用して、リモートサーバー上にある電子メールを確認し、ダウンロードします。その後、これをローカルマシンのポート 25 に配信し、ローカルの MTA を使用してメールを正しいユーザーのスプールファイルに配置します。Procmail が利用できる場合は起動して電子メールをフィルターし、MUA が読み込むことができるようにメールボックスに配置します。

19.3.3.1. Fetchmail の設定オプション

Fetchmail の実行時に、コマンドラインですべての必要なオプションを渡して、リモートサーバーの電子メールを確認することは可能ですが、.fetchmailrc ファイルを使用した方がはるかに簡単です。希望の設定オプションを .fetchmailrc ファイルに置き、それらのオプションが fetchmail コマンドを実行するたびに使用されるようにします。Fetchmail の実行時にオプションを上書きしたい場合は、コマンドラインでそのオプションを指定します。

ユーザーの .fetchmailrc ファイルには、3 つのクラスの設定オプションが含まれています。

- グローバルオプション: プログラムの動作を制御する、または電子メールを確認する全接続の設定を提供する指示を Fetchmail に指定します。

- **server options**: ポーリングされるサーバーに必要な情報を指定します。ホスト名をはじめ、確認するポートやタイムアウトになるまでの秒数など、特定の電子メールサーバーの設定などです。こうしたオプションは、該当するサーバーを使用する全ユーザーに影響を及ぼします。

- ユーザーオプション: 指定された電子メールサーバーを使用して、電子メールの認証や確認を行うにあたって必要なユーザー名、パスワードなどの情報を格納します。

グローバルオプションは .fetchmailrc ファイルの上部に表示され、その後に 1 つ以上のサーバーオプションが表示されます。各オプションは Fetchmail がチェックする異なるメールサーバーを指定します。ユーザーオプションは、そのメールサーバーをチェックする各ユーザーアカウントのサーバーオプションに従います。サーバーオプションと同様に、複数のユーザーオプションを指定することで特定のサーバーでの使用、同一サーバー上の複数の電子メールアカウントの確認を行うことができます。

サーバーオプションを .fetchmailrc ファイルで利用するには、サーバーの情報の先頭に poll または skip などの特別なオプションの動詞を使用します。poll アクションは、Fetchmail の実行時にこのサーバーオプションを使用して、指定されたユーザーオプションで電子メールを確認するよう Fetchmail に指示します。ただし、skip アクションの後にあるサーバーオプションは、Fetchmail が呼び出された時にサーバーのホスト名が指定されていない限り確認されません。skip オプションは、特に呼び出された時にスキップされたサーバーのみを確認し、現在稼働中の設定には影響を及ぼさないため、.fetchmailrc ファイルの設定をテストする際に役立ちます。

以下は、.fetchmailrc ファイルの例です。

```
set postmaster "user1"
set bouncemail

poll pop.domain.com proto pop3
   user 'user1' there with password 'secret' is user1 here

poll mail.domain2.com
   user 'user5' there with password 'secret2' is user1 here
   user 'user7' there with password 'secret3' is user1 here
```

この例では、グローバルオプションにより、最終手段としてユーザーに電子メールが送信されるように指定されており（postmaster オプション）、すべての電子メールエラーは送信者ではなく、ポストマスターに送信されます（bouncemail オプション）。set アクションは、この行にグローバルオプションが含まれていることを Fetchmail に指示します。その後、2 つのメールサーバーが指定されており、もう 1 つは POP3 を使用してチェックするように設定されます。2 つのユーザーは 2 番目のサーバーオプションを使用してチェックされますが、任意のユーザーで見つかったすべての電子メールは user1 の mail spool に送信されます。これにより、1 つの MUA 受信トレイに表示され、複数のサーバーで複数のクラスを確認できます。各ユーザーの固有の情報は、user アクションで開始します。

設定からのパスワードの省略

ユーザーはパスワードを .fetchmailrc ファイルに配置する必要はありません。with password '<password>' セクションを省略すると、Fetchmail は起動時にパスワードを要求します。

Fetchmail には、グローバルオプション、サーバーオプション、ローカルオプションが多数あります。これらの多くのオプションは、ほとんど使用されないか、非常に特殊な状況にのみ適用されます。fetchmail の man ページでは、各オプションの詳細が記載されていますが、最も一般的なものを以下の 3 セクションで説明します。

19.3.3.2. グローバルオプション

グローバルオプションは、set アクションの後に、それぞれ 1 行ずつ配置する必要があります。

- *daemon seconds : Fetchmail がバックグラウンドに残るデーモンモードを指定します。seconds を、Fetchmail がサーバーをポーリングするまでの待機時間の秒数に置き換えます。*

- *postmaster: 配信に問題が生じた場合にローカルユーザーがメールを送信するよう指定します。*

- *syslog: エラーとステータスメッセージのログファイルを指定します。デフォルトでは、これは /var/log/maillog です。*

19.3.3.3. サーバーオプション

サーバーオプションは、ポーリング または スキップ アクションの後に、.fetchmailrc 内の独自の行に配置する必要があります。

- *auth auth-type : auth-type を使用する認証のタイプに置き換えます。デフォルトでは、パスワード 認証が使用されますが、一部のプロトコルは、kerberos_v5、kerberos_v4、ssh など、他のタイプの認証をサポートします。any 認証タイプを使用すると、Fetchmail は、パスワードを必要としない方法を最初に試みます。次に、パスワードをマスクする方法を試みた後、最後にサーバーに暗号化されていないパスワードを送信して認証を試みます。*

- *interval number : 指定されたサーバーを、設定されたすべてのサーバーの電子メールをチェックするたびにポーリングします。このオプションは、通常のユーザーがほとんどメッ*

セージを受信しない電子メールサーバーに使用されます。

- port port-number : port-number をポート番号に置き換えます。この値は、指定されたプロトコルのデフォルトのポート番号を上書きします。

- proto protocol : protocol を、サーバー上のメッセージを確認する時に使用する pop3 や imap などのプロトコルに置き換えます。

- timeout seconds : seconds を、Fetchmail が接続の試行をやめてからサーバーが非アクティブになる秒数に置き換えます。この値を設定しないと、デフォルトの 300 秒が使用されます。

19.3.3.4. ユーザーオプション

ユーザーオプションは、サーバーオプションの下の各行に置かれる場合と、サーバーオプションと同じ行に置かれる場合があります。いずれの場合も、定義されるオプションは user オプション（以下で説明）に従う必要があります。

- fetchall: 既読メッセージを含め Fetchmail がキューにあるすべてのメッセージをダウンロードするように命令します。デフォルトでは、Fetchmail は新規メッセージのみをダウンロードするようになっています。

- fetchlimit number : number を、停止する前に取得するメッセージ数に置き換えます。

- flush: 新規メッセージを取得する前にキューにあるすべての既読メッセージを削除します。

- limit max-number-bytes : max-number-bytes を、Fetchmail で取得する時に許容されているメッセージの最大バイトサイズに置き換えます。このオプションでは低速のネットワークリンクが提供されるため、サイズが大きいメッセージのダウンロードに時間がかかりすぎる場合に有用です。

- password 'password' - password を、ユーザーのパスワードに置き換えます。

- preconnect "command" - command を、ユーザー宛のメッセージを取得する前に実行するコマンドに置き換えます。

- *postconnect " - command* を、ユーザー宛のメッセージを取得した後に実行するコマンドに置き換えます。

- *ssl: SSL* 暗号化を有効にします。この英語版が公開された時点で、デフォルトのアクションでは *SSL2*、*SSL3*、*SSL23*、*TLS1*、*TLS1.1*、および *TLS1.2* から最良のものを使用します。*SSL2* は廃止されたものと見なされ、*POODLE: SSLv3 脆弱性(CVE-2014-3566)のため*、*SSLv3* を使用しないでください。ただし、*TLS1* 以降の使用を強制できないため、接続するメールサーバーが *SSLv2* および *SSLv3* を使用し ない よう設定する必要があります。サーバーが *SSLv2* および *SSLv3* を使用し ない ように設定できない場合は、*stunnel* を使用します。

- *sslproto: SSL* プロトコルまたは *TLS* プロトコルを定義します。可能な値は *SSL2*、*SSL3*、*SSL23*、および *TLS1* です。*sslproto* が省略された場合、未設定の場合、または無効な値に設定された場合のデフォルト値は *SSL23* です。デフォルトのアクションは *SSLv3*、*TLSv1*、*TLS1.1*、および *TLS1.2* から最適なものを使用します。*SSL* または *TLS* の他の値を設定すると、他のすべてのプロトコルが無効になることに注意してください。*POODLE: SSLv3 脆弱性(CVE-2014-3566*)のため、このオプションを省略するか、*SSLv23* に設定し、対応するメールサーバーが *SSLv2* および *SSLv3* を使用し ない ように設定することが推奨されます。サーバーが *SSLv2* および *SSLv3* を使用し ない ように設定できない場合は、*stunnel* を使用します。

- *user ": username* を、*Fetchmail* がメッセージの取得に使用するユーザー名に置き換えます。このオプションは、他のすべてのユーザーオプションの前に付ける必要があります。

19.3.3.5. Fetchmail のコマンドオプション

fetchmail コマンドの実行時にコマンドライン上で使用される *Fetchmail* オプションの大半は、*.fetchmailrc* 設定オプションを反映します。この方法では、*Fetchmail* は設定ファイルの有無を問わず使用できます。これらのオプションは、*.fetchmailrc* ファイルに残しておいた方が簡単なため、ほとんどのユーザーがコマンドラインでは使用しません。

fetchmail コマンドは、特定の用途のオプションと併せて実行した方が望ましい場合もあります。コマンドラインで指定されるオプションはいずれも設定ファイルオプションを上書きするため、エラーが発生した場合は、コマンドオプションを使用して、エラーの原因になっている *.fetchmailrc* 設定を一時的に上書きすることが可能です。

19.3.3.6. 情報提供またはデバッグのオプション

fetchmail コマンドの後に使用されるオプションの一部は、重要な情報を提供する場合があります。

-

--configdump: .fetchmailrc および Fetchmail のデフォルトからの情報に基づいて可能なすべてのオプションを表示します。このオプションを使用すると、どのユーザーの電子メールも取得されません。

- *-s: Fetchmail をサイレントモードで実行し、fetchmail コマンドの後にエラー以外のメッセージが表示されないようにします。*

- *-v: Fetchmail を verbose モードで実行し、Fetchmail とリモートの電子メールサーバー間のすべての通信を表示します。*

- *-v: 詳細なバージョン情報の表示、グローバルオプションの一覧表示、電子メールプロトコルや認証方法など、各ユーザーと使用する設定の表示を行います。このオプションを使用すると、どのユーザーの電子メールも取得されません。*

19.3.3.7. 特殊なオプション

これらのオプションは、.fetchmailrc ファイルによく見られるデフォルト値を上書きする時に役立つ場合があります。

- *-a: Fetchmail は、新規または既読を問わず、すべてのメッセージをリモートの電子メールサーバーからダウンロードします。デフォルトでは、Fetchmail は新規メッセージのみをダウンロードします。*

- *-k: Fetchmail はメッセージをダウンロードした後、リモートの電子メールサーバー上にメッセージを残します。このオプションを使用すると、メッセージをダウンロード後に削除するデフォルトの動作は上書きされます。*

- *-l max-number-bytes : Fetchmail は特定のサイズを超えるメッセージはダウンロードせず、リモートの電子メールサーバー上に残します。*

- *--quit: Fetchmail デーモンのプロセスを終了します。*

その他のコマンドおよび .fetchmailrc オプションは、fetchmail の man ページを参照してください。

19.3.4. メール転送エージェント (MTA) の設定

メール 転送エージェント (MTA)は電子メールの送信に不可欠です。メールの読み取り および書き込みに使用するメールユーザーエージェント (MUA)（ Evolution, TEMPLATES、 Mutt ）などのメールユーザーエージェント(MUA)は、電子メールの読み取りと構成に使用されます。ユーザーが MUA から電子メールを送信すると、メッセージは MTA に渡されます。MTA は一連の MTA を通じて、メッセージが送信先に届くまで送信します。

ユーザーがシステムから電子メールを送信する予定でなくても、一部の自動化されたタスクまたはシステムプログラムは、/bin/mail コマンドを使用して、ログメッセージを含む電子メールをローカルシステムの root ユーザーに送信する場合があります。

Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;LinuxRed Hat Enterprise Linux 6nbsp;6 は、Postfix と Sendmail の 2 つの MTA を提供します。両方がインストールされている場合は、Postfix がデフォルトの MTA になります。

19.4. メール配信エージェント (MDA)

Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux には、Procmail と mail の 2 つの主要 MDA が含まれています。どちらのアプリケーションも LDA とみなされ、MTA のスプールファイルからユーザーのメールボックスにメールを移動します。ただし、Procmail の方が堅牢なフィルタリングシステムを提供します。

このセクションでは、Procmail についてのみ詳しく説明します。mail コマンドの詳細は、man ページの(man mail)を参照してください。

ローカルホストのメールスプールファイルに電子メールが置かれると、Procmail が配信とフィルタリングを行います。Procmail は強力な上、システムリソースの使用が低いため、幅広く利用されています。Procmail は、電子メールクライアントアプリケーションが読み取る電子メールを配信するという重要な役割を果たします。

Procmail は、様々な方法で呼び出すことができます。MTA が電子メールをメールスプールファイルの中に置くと常に Procmail が起動します。次に、Procmail は電子メールを MUA のためにフィルタリング、ファイル保存して、終了します。別の方法としては、メッセージを受信すると常に Procmail を実行するように MUA を設定し、メッセージが正しいメールボックスに移動するようにできます。デフォルトでは、/etc/procmailrc または ~/.procmailrc ファイル（別名 rc ファイル）がユーザーのホームディレクトリーにあると、MTA が新しいメッセージを受信するたびに Procmail が呼び出されます。

デフォルトでは、/etc/ ディレクトリーにシステム全体の rc ファイルが存在せず、ユーザーのホームディレクトリーに .procmailrc ファイルが存在しません。このため、Procmail を使用するには、各ユーザーが特定の環境変数とルールを使用して .procmailrc ファイルを構築する必要があります。

Procmail が電子メールメッセージに対応するかどうかは、そのメッセージが rc ファイルの特定の条件または レシピ と一致するかどうかによって異なります。あるメッセージが任意のレシピと適合する場合、電子メールは特定のファイルに置かれるか削除され、それ以外は処理されます。

Procmail が起動すると、電子メールメッセージを読み取り、ヘッダー情報から本文を切り離します。次に、Procmail は /etc/procmailrc ディレクトリー内の / etc/procmailrc ファイルと rc ファイルで、デフォルトのシステム全体の Procmail 環境変数とレシピを探します。その後 Procmail は、ユーザーのホームディレクトリー内で .procmailrc ファイルを検索します。多くのユーザーは、Procmail 用に追加の rc ファイルも作成します。これは、ホームディレクトリーの .procmailrc ファイル内で参照されます。

19.4.1. Procmail の設定

Procmail の設定ファイルには、重要な環境変数が含まれています。これらの変数は、並べ替えするメッセージ、およびどのレシピとも適合しないメッセージの処理を指定します。

これらの環境変数は通常 ~/.procmailrc ファイルの最初に表示されます。

```
env-variable="value"
```

この例では、env-variable が変数の名前で、value が変数を定義します。

ほとんどの Procmail ユーザーが使用していない環境変数が多くあります。また、重要な環境変数の多くがデフォルト値で定義されています。重要な環境変数の多くは、既にデフォルト値で定義されています。大抵の場合は、以下のような変数が使用されます。

- DEFAULT: どのレシピにも適合しないメッセージが配置された場合のデフォルトのメールボックスを設定します。

  デフォルトの DEFAULT 値は、$ORGMAIL と同じです。

- INCLUDERC: チェックするメッセージに対する多くのレシピを格納する追加の rc ファイルを指定します。これにより、Procmail レシピの一覧は、スパムのブロックや電子メール一覧の管理など異なるロールを満たす個別のファイルに分割されます。このファイルは、ユーザーの ~/.procmailrc ファイル内のコメント文字を使用して、オンやオフにすることができます。

たとえば、ユーザーの ~/.procmailrc ファイル内の行は以下のようになります。

```
MAILDIR=$HOME/Msgs
INCLUDERC=$MAILDIR/lists.rc
INCLUDERC=$MAILDIR/spam.rc
```

電子メールの一覧の Procmail フィルターをオフにしつつスパム制御を維持する場合は、最初の INCLUDERC 行をハッシュ記号(#)でコメントアウトします。現在のディレクトリーに相対的なパスが使用されることに注意してください。

- LOCKSLEEP: Procmail が特定のロックファイルの使用を試みるまでの時間を秒単位で設定します。デフォルトは 8 秒です。
- LOCKTIMEOUT: ロックファイルが最後に修正された後、Procmail がそれは古くて削除可能であるとみなすまでに経過する必要のある時間を秒単位で設定します。デフォルトは 1024 秒です。
- LOGFILE: Procmail の情報やエラーメッセージが書き込まれるファイルです。
- MAILDIR: Procmail 用の現在作業中のディレクトリーを設定します。設定されると、他の Procmail のパスはすべてこのディレクトリーに対する相対パスになります。
- ORGMAIL - 元のメールボックス、またはデフォルトまたはレシピで必要な場所にメッセージを配置できなかった場合に、メッセージを配置する別の場所を指定します。

  デフォルトでは、/var/spool/mail/$LOGNAME の値が使用されます。
- SUSPEND: スワップ領域など必要なリソースが利用できない場合に Procmail が一時停止する時間を秒単位で設定します。
- SWITCHRC: 追加の Procmail レシピが格納されている外部ファイルをユーザーが指定できるようにします。これは、INCLUDERC オプションとよく似ていますが、レシピのチェックが参照先の設定ファイルで実際に停止され、SWITCHRC- 指定のファイル上のレシピのみが使用される点が異なります。
-

VERBOSE: Procmail が詳細な情報をログ記録するようにします。このオプションはデバッグに役立ちます。

その他の重要な環境変数は、シェルからプルされます。たとえば、ログイン名、ログイン名、ホームディレクトリーの場所である HOME、デフォルトのシェルである SHELL などです。

すべての環境変数に関する包括的な説明やデフォルト値は、man ページの procmailrc を参照してください。

19.4.2. Procmail レシピ

多くの場合、新規ユーザーが Procmail の使用法を学習するにあたって最も難しいと感じるのは、レシピの構築です。この難易度は、レシピが適合する文字列の条件を指定するために使用される 正規表現 を使用してメッセージ照合を行うことが多くあります。ただ、正規表現の構築はそれほど難しくなく、読んで理解することも簡単です。その上、Procmail のレシピを書く方法は、正規表現にかかわらず一貫性があるため、例を使って学習すると簡単です。Procmail のレシピの例は、「レシピの例」 を参照してください。

Procmail レシピは以下の形式を使用します:

```
:0 [flags] [: lockfile-name ]
* [ condition_1_special-condition-character condition_1_regular_expression ]
* [ condition_2_special-condition-character condition-2_regular_expression ]
* [ condition_N_special-condition-character condition-N_regular_expression ]
    special-action-character
    action-to-perform
```

Procmail レシピの最初の 2 文字は、コロンとゼロです。ゼロの後に様々なフラグを追加して、Procmail がレシピを処理する方法を制御します。flags セクションの後ろにコロンを付けると、このメッセージに対してロックファイルが作成されることを示しています。ロックファイルが作成されると、lockfile-name を置き換えて名前を指定できます。

レシピには、メッセージと適合させる様々な条件を追加できます。条件がない場合は、すべてのメッセージがレシピと適合することになります。正規表現は、メッセージ照合を容易にするために、一部の条件で使用されます。複数の条件を使用する場合は、アクションが実行されるためにはすべてが適合しなければなりません。条件は、レシピの 1 行目に設定されているフラグに基づいてチェックされます。アスタリスク文字(*)の後にオプションの特殊文字を追加すると、さらに条件を制御できます。

action-to-perform 引数は、メッセージが条件の 1 つに適合する場合に実行するアクションを指定します。1 つのレシピに指定できるアクションは 1 つのみとなります。多くの場合、メールボックスの名前がここで使用され、適合するメッセージをファイルに誘導し、電子メールを効果的に並べ替えま

す。特別なアクションの文字は、アクションが指定される前に使用することもできます。詳細は、「特別な条件とアクション」 を参照してください。

19.4.2.1. 配信と非配信レシピ

レシピが特定のメッセージに適合する場合に使用されるアクションは、配信レシピまたは 非 配信レシピとみなされるかどうかを判断します。配信レシピには、ファイルへのメッセージの書き込み、別のプログラムへのメッセージ送信、別の電子メールアドレスへのメッセージ転送などのアクションが含まれています。非配信レシピは、ネストされたブロック などの他のアクションをカバーします。ネストされたブロックは、中括弧 { } で囲まれたアクションセットで、レシピの条件に適合するメッセージで実行されます。ネストされたブロックは、互いにネストさせることができるため、メッセージに対するアクションを特定して実行するにあたっての制御力が強化されます。

メッセージが配信レシピと適合すると、Procmail は指定されたアクションを実行し、その他のレシピとメッセージとの比較を停止します。非配信レシピと適合するメッセージの場合は、他のレシピに対する照合は継続されます。

19.4.2.2. フラグ

フラグは、レシピの条件をメッセージに照合する方法と、それを決定する上で不可欠です。egrep ユーティリティーは、条件の照合のために内部で使用されます。一般的に使用されるフラグは以下のとおりです。

- *a: A や a のフラグが付いていない以前のレシピもこのメッセージに適合する場合にのみ、このレシピが使用されることを指定します。*
- *a : A または a のフラグが付いた以前のレシピもこのメッセージに適合し、かつ 正常に完了した場合にのみこのレシピが使用されることを指定します。*
- *B: メッセージのボディーを解析し、適合する条件を検索します。*
- *b: ファイルへのメッセージの書き込みや転送など、結果として生じるアクションにボディーを使用します。これはデフォルトの動作です。*
- *c - 電子メールのカーボンコピーを生成します。必要なアクションをメッセージで実行し、メッセージのコピーは rc のファイル内で引き続き処理することができるため、レシピの配信に役立ちます。*
-

D: egrep の比較で大文字と小文字を区別します。デフォルトでは、照合プロセスでは大文字と小文字を区別していません。

- *E: A フラグと同様ですが、レシピ内の条件は、直前にある E フラグなしのレシピが適合しない場合のみに、メッセージと照合されます。これは else アクションと類似しています。*
- *e: 直前のレシピで指定されたアクションが失敗した場合のみ、レシピがメッセージに照合されます。*
- *f: フィルターとしてパイプを使用します。*
- *H: メッセージのヘッダーを解析し、適合する条件を検索します。これはデフォルトの動作です。*
- *h: 結果として生じるアクションでヘッダーを使用します。これはデフォルトの動作です。*
- *w: Procmail に対して、指定されたフィルターまたはプログラムが終了するのを待ち、メッセージがフィルターされたとみなす前に成功したかどうかを報告するよう指示します。*
- *w: 「プログラム障害」のメッセージが抑制されている点を除いては w と同じです。*

追加のフラグの詳細な一覧は、procmailrc man ページを参照してください。

19.4.2.3. ローカルロックファイルの指定

ロックファイルは、Procmail で複数のプロセスが 1 つのメッセージを同時に変更しないようにするために非常に役立ちます。ローカルロックファイルを指定するには、レシピの 1 行目の任意のフラグの後にコロン(:)を追加します。これにより、送信先のファイル名に基づいたローカルロックファイルと、LOCKEXT のグローバル環境変数で設定されたものすべてが作成されます。

別の方法としては、このレシピで使用するローカルロックファイルの名前をコロンの後に指定します。

19.4.2.4. 特別な条件とアクション

Procmail レシピの条件とアクションの前に使用される特殊文字により、解釈の仕方が変わります。

以下の文字は、レシピの条件の行頭でアスタリスク文字(*)の後に使用できます。

- !: 条件の行では、この文字により条件が反転し、条件がメッセージに一致しない場合にのみ、適合が発生します。
- & lt;: メッセージが、指定されているバイト数に収まっているかどうかを確認します。
- & gt;: メッセージが、指定されているバイト数を超えているかどうかを確認します。

以下の文字は、特別なアクションを実行するために使用されます。

- !: アクションの行では、この文字は、指定された電子メールアドレスにメッセージを転送するように Procmail に指示します。
- $: rc ファイルで以前に設定された変数を参照します。多くの場合は、さまざまなレシピによって参照される共通のメールボックスを設定するために使用されます。
- |: 指定したプログラムを開始し、メッセージを処理します。
- { and }: 適合するメッセージに適用する追加のレシピを格納するために使用されるネストされたブロックを構築します。

アクションの行頭に特殊文字を使用しない場合、Procmail はアクションの行がメッセージを書き込むためのメールボックスを指定していると仮定します。

19.4.2.5. レシピの例

Procmail は極めて柔軟性の高いプログラムですが、この柔軟性が原因で、新規ユーザーが Procmail のレシピを一から作成するのが難しい場合があります。

Procmail レシピの条件を構築するスキルを向上させる最適な方法は、正規表現をしっかり理解し、他の人が構築した多くの例を参照することから始まります。正規表現に関する詳細な説明は、本セクションでは扱いません。Procmail のレシピの構造と役立つ Procmail のサンプルレシピは、インターネット上の様々なところに掲載されています。正規表現の適切な使用と調整方法は、これらのレシピ例を参照してください。また、基本的な正規表現ルールの概要は、man ページの grep(1) を参照してください。

以下にあげる簡単な例は、Procmail のレシピの基本構造を記載しており、構造をさらに複雑にするための基盤を示しています。

以下の例に示すように、基本的なレシピには条件さえも含まれていません。

```
:0:
new-mail.spool
```

最初の行は、ローカルのロックファイルを作成することを指定しますが、名前を指定していません。そのため、Procmail は宛先ファイル名を使用して、LOCKEXT 環境変数に指定された値を追加します。条件が指定されていないため、すべてのメッセージがこのレシピと一致し、MAILDIR 環境変数で指定されたディレクトリー内にある new-mail.spool という単一の spool ファイルに配置されます。その後、MUA はこのファイルでメッセージを表示できます。

このような基本レシピは、rc ファイルの末尾に置かれ、メッセージをデフォルトの場所に送ります。

以下の例では、特定の電子メールアドレスからのメッセージを照合して、削除します。

```
:0
* ^From: spammer@domain.com
/dev/null
```

この例では、spammer@domain.com によって送信されたメッセージはすべて /dev/null デバイスに送信され、削除されます。

/DEV/NULL へのメッセージの送信

メッセージを /dev/null に送信して永久に削除してしまう前に、ルールが目的どおりに機能していることを確認してください。レシピが間違えて目的以外のメッセージを対象にすると、それらのメッセージは消えてしまい、ルールのトラブルシューティングが困難になります。

より優れた解決策は、レシピのアクションを特別なメールボックスにポイントさせることです。これは、誤検出を探すために時間から時刻にチェックできます。メッセージが間違って適合されることがなく満足できる状態になったら、そのメールボックスは削除して、メッセージを /dev/null に送信するよう指示します。

以下のレシピでは、特定のメーリングリストから送信された電子メールを取得して、特定のフォルダに配置します。

```
:0:
* ^(From|Cc|To).*tux-lug
tuxlug
```

tux-lug@domain.com メーリングリストから送信されたメッセージはすべて、MUA 用に自動的に tuxlug メールボックスに置かれます。From、Cc、または To 行にメーリングリストのメールアドレスがある場合は、この例の条件がメッセージに適合する点に注意してください。

さらに詳しい強力なレシピについては、「その他のリソース」 の Procmail に関する多くのオンライン資料を参照してください。

19.4.2.6. spam フィルター

Procmail は、新規の電子メールを受信すると Sendmail、Postfix、Fetchmail によって呼び出されるため、スパム対策の強力なツールとして使用できます。

これは、Procmail が SpamAssassin と併用された場合に特に有効です。これらの 2 つのアプリケーションを併用すると、スパムメールを迅速に特定して、並び替えまたは破棄できます。

SpamAssassin は、ヘッダー分析、テキスト分析、ブラックリスト、スパム追跡データベース、自己学習型 Bayesian スパム分析を使用して、迅速かつ正確にスパムの特定とタグ付けを行います。

SPAMASSASSIN パッケージのインストール

SpamAssassin を使用するには、root で以下を実行し、spamassassin パッケージがシステムにインストールされていることを確認します。

```
~]# yum install spamassassin
```

yum を使用したパッケージのインストールは 「パッケージのインストール」 を参照してください。

ローカルユーザーが SpamAssassin を使用する最も簡単な方法は、~/.procmailrc ファイルの最上部付近に以下の行を追加します。

```
INCLUDERC=/etc/mail/spamassassin/spamassassin-default.rc
```

/etc/mail/spamassassin/spamassassin-default.rc には、すべての受信メールに対して SpamAssassin を有効にする単純な Procmail ルールが含まれています。電子メールがスパムであると判断された場合には、ヘッダー内でタグ付けされ、タイトルの先頭には以下のようなパターンが追加されます。

```
*****SPAM*****
```

電子メールのメッセージ本文にも、スパム診断の理由となった要素の継続的な記録が先頭に追加されます。

スパムとしてタグ付けされた電子メールをファイル保存するには、以下と同様のルールを使用することができます。

```
:0 Hw * ^X-Spam-Status: Yes spam
```

このルールは、スパムとしてヘッダーにタグ付けされた電子メールをすべて、spam と呼ばれるメールボックスにファイルします。

SpamAssassin は Perl スクリプトであるため、ビジー状態のサーバーでバイナリー SpamAssassin デーモン(spamd)およびクライアントアプリケーション(spamc)を使用する必要がある

場合があります。ただし、SpamAssassin をこのように設定するには、ホストへの root アクセスが必要です。

spamd デーモンを起動するには、以下のコマンドを入力します。

```
~]# service spamassassin start
```

システムの起動時に SpamAssassin デーモンを起動するには、Services Configuration Tool (system-config-services)などの initscript ユーティリティーを使用して spamassassin サービスを有効にします。サービスの起動と停止に関する詳細は、12章サービスおよびデーモン を参照してください。

Procmail が Perl スクリプトではなく、SpamAssassin クライアントアプリケーションを使用するように設定するには、~/.procmailrc ファイルの最上部付近に以下の行を追加します。システム全体の設定の場合は、/etc/procmailrc に配置します。

```
INCLUDERC=/etc/mail/spamassassin/spamassassin-spamc.rc
```

19.5. メールユーザーエージェント

Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux は、Evolution のようなグラフィカル電子メールクライアントプログラムと、にわたようなテキストベースの電子メールプログラムなど、様々な電子メールプログラムを提供し ます。

本セクションでは、クライアントとサーバー間の通信のセキュリティー保護について重点的に説明していきます。

19.5.1. 通信のセキュリティー保護

Evolution や Mutt など、Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux に含まれる一般的な MUA は、SSL で暗号化された電子メールセッションを提供します。

暗号化されていないネットワークを行き来する他のサービスと同様に、ユーザー名、パスワード、メッセージ全体などの電子メールに関する重要な情報は、ネットワーク上のユーザーによって傍受、閲覧される可能性があります。また、標準の POP プロトコルおよび IMAP プロトコルは認証情報を暗号化せずに渡すため、ユーザー名とパスワードはネットワーク経由で渡される際に攻撃者がそれらを収集して、ユーザーアカウントにアクセスできる可能性があります。

19.5.1.1. セキュアな電子メールクライアント

リモートサーバー上の電子メールを確認するように設計されている Linux MUA のほとんどは、SSL 暗号化に対応しています。電子メールを取得する時に SSL を使用するためには、SSL は電子メールクライアントとサーバーの両方で有効である必要があります。

SSL はクライアント側で簡単に有効にできます。多くの場合、MUA の設定ウィンドウでボタンをクリックするか、MUA 設定ファイルのオプションを使用して実行できます。セキュアな IMAP および POP には、MUA がメッセージの認証およびダウンロードに使用する既知のポート番号（993 および 995 ）があります。

19.5.1.2. 電子メールクライアントの通信のセキュリティー保護

電子メールサーバー上の IMAP および POP ユーザーに SSL 暗号化を行うことは簡単です。

最初に SSL 証明書を作成します。これは、SSL 証明書の 認証局 (CA)に適用するか、自己署名証明書を作成するという 2 つの方法で実行できます。

自己署名証明書の使用の回避

自己署名証明書は、テスト目的のみで使用することをお勧めします。実稼働環境で使用するサーバーは、CA が付与する SSL 証明書を使用する必要があります。

IMAP または POP に自己署名 SSL 証明書を作成するには、/etc/pki/dovecot/ ディレクトリーに移動し、/etc/pki/dovecot/dovecot-openssl.cnf 設定ファイルの証明書パラメーターを編集し、root で以下のコマンドを入力します。

```
dovecot]# rm -f certs/dovecot.pem private/dovecot.pem
dovecot]# /usr/libexec/dovecot/mkcert.sh
```

完了したら、/etc/dovecot/conf.d/10-ssl.conf ファイルに以下の設定があることを確認します。

```
ssl_cert = </etc/pki/dovecot/certs/dovecot.pem
ssl_key = </etc/pki/dovecot/private/dovecot.pem
```

service dovecot restart コマンドを実行して、dovecot デーモンを再起動します。

または、stunnel コマンドを IMAP サービスまたは POP サービスへの標準的なセキュアでない接続で暗号化ラッパーとして使用することもできます。

stunnel ユーティリティーは、Red Hat Enterprise Linuxnbsp;Hat Enterprise Red Hat Enterprise Linuxnbsp;Linux に含まれる外部 OpenSSL ライブラリーを使用して強力な暗号化を提供し、ネットワーク接続を保護します。SSL 証明書を取得するためには、CA に申請することが推奨されますが、自己署名証明書を作成することも可能です。

stunnel のインストール方法と基本設定の作成方法については、『Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;Linux Red Hat Enterprise Linux 6nbsp;6 セキュリティーガイド』の「stunnel の使用」を参照してください。stunnel を IMAPS と POP3S のラッパーとして設定するには、以下の行を /etc/stunnel/stunnel.conf 設定ファイルに追加します。

```
[pop3s]
accept = 995
connect = 110

[imaps]
accept = 993
connect = 143
```

セキュリティーガイドでは、stunnel の起動および停止の方法を説明します。起動後は、IMAP または POP の電子メールクライアントを使用し、SSL 暗号化を使用して電子メールサーバーに接続できます。

19.6. その他のリソース

以下は、電子メールアプリケーションに関する補足のドキュメントの一覧です。

19.6.1. インストールされているドキュメント

- Sendmail の設定に関する情報は、sendmail パッケージおよび sendmail-cf パッケージに含まれています。
  - /usr/share/sendmail-cf/README: m4 マクロプロセッサーの情報、Sendmail のファイルの場所、対応するメーラー、強化機能へのアクセス方法などに関する情報が含まれています。

さらに、sendmail および aliases の man ページには、Sendmail の様々なオプションと Sendmail /etc/mail/aliases ファイルの適切な設定に関する役立つ情報が記載されています。

- */usr/share/doc/postfix-version-number/ : Postfix の設定方法に関する多くの情報が含まれています。version-number を Postfix のバージョン番号に置き換えてください。*

- */usr/share/doc/fetchmail-version-number : Fetchmail の機能の全一覧は、FEATURES ファイルおよび入門 FAQ ドキュメントに記載されています。version-number を Fetchmail のバージョン番号に置き換えてください。*

- */usr/share/doc/procmail-version-number/ : Procmail の概要を示す README ファイル、すべてのプログラム機能を調べる FEATURES ファイル、設定に関する多くのよくある質問への回答が含まれる FAQ ファイルが含まれています。version-number を、Procmail のバージョン番号に置き換えてください。*

  Procmail の仕組みや新しいレシピの作成方法を学習する場合は、以下にあげる Procmail の man ページが非常に役立ちます。

  - *Procmail: Procmail の仕組みと電子メールのフィルタリングに必要な手順を概説します。*
  - *procmailrc: レシピの構築に使用される rc のファイル形式を説明します。*
  - *procmailex: 実環境向けの役立つ Procmail のサンプルレシピを多数紹介します。*
  - *procmailsc: 特定のレシピとメッセージに適合するために Procmail で使用される加重スコアリング手法を説明します。*
  - */usr/share/doc/spamassassin-version-number/: SpamAssassin に関する多くの情報が含まれています。version-number を、spamassassin パッケージのバージョン番号に置き換えます。*

19.6.2. オンラインドキュメント

- *How to configure postfix with TLS?: postfix が TLS を使用するように設定することに関する Red Hat ナレッジベースの記事です。*

Red Hat ナレッジベースの記事「How to Configure a System to Manage Multiple Virtual Mailboxes Using Postfix and Dovecot」では、Postfix を Mail Transporting Agent(MTA)および Dovecot を IMAP サーバーとして使用する 1 つの実ユーザーアカウントで複数の仮想ユーザーを管理する方法を説明します。

- http://www.sendmail.org/: Sendmail の機能、ドキュメント、設定例の詳細を説明します。
- http://www.sendmail.com/: Sendmail に関連する認識、生産性、記事が含まれています。これには、利用可能な多くのオプションの幅広いビューが含まれます。
- http://www.postfix.org/: Postfix プロジェクトのホームページには、Postfix に関する豊富な情報が記載されています。メーリングリストは、特に情報検索に役立ちます。
- http://www.fetchmail.info/fetchmail-FAQ.html: Fetchmail に関する詳細な FAQ です。
- http://www.procmail.org/: Procmail のホームページで、Procmail 専用の各種メーリングリストへのリンクと、様々な FAQ ドキュメントへのリンクが含まれています。
- http://www.spamassassin.org/: SpamAssassin プロジェクトの公式サイトです。

19.6.3. 関連書籍

- 『Sendmail Milters: A Guide for Fighting Spam』 by Bryan Costales and Marcia Flynt; Addison-Wesley - メールフィルターのカスタマイズに役立つ優れた Sendmail ガイドです。
- 『Sendmail』 （Bryan Costales、Eric Allman etal.; O'Reilly & Associates） - Delivermail と Sendmail のオリジナルの作成者のサポートを受けた優れた Sendmail リファレンスです。
- 『Remove the Spam: Email Processing and Filtering』 by Geoff Mulligan; Addison-Wesley Publishing Company: Sendmail や Procmail 等の確立されたツールを使ってスパム問題を管理する電子メール管理者が使用するさまざまな方法を学習するボリューム。
- 『Internet Email Protocols: A Developer's Guide』 by Kevin Johnson; Addison-Wesley Publishing Company - 主要な電子メールプロトコルとそれが提供するセキュリティー

に関する非常に詳細なレビューを提供します。

- *『Managing IMAP』 by Dianna Mullet and Kevin Mullet; O'Reilly & Associates - IMAP サーバーの設定に必要な手順について詳しく説明します。*

第20章 ディレクトリーサーバー

20.1. OPENLDAP

LDAP (Lightweight Directory Access Protocol) は、ネットワーク上で一元的に保存された情報にアクセスするために使用されるオープンプロトコルのセットです。これは、ディレクトリー共有の X.500 標準に基づいていますが、それほど複雑ではなく、リソースを大量に消費します。このため、LDAP は 「X.500 Lite」 と呼ばれることもあります。

X.500 と同様に、LDAP はディレクトリーを使用して階層的な方法で情報を編成します。これらのディレクトリーは、名前、アドレス、電話番号などのさまざまな情報を保存し、Network Information Service (NIS) と同様に使用でき、ユーザーが LDAP 対応のネットワーク上のマシンからアカウントにアクセスできるようにすることもできます。

LDAP は通常、一元管理されたユーザーおよびグループ、ユーザー認証、またはシステム設定に使用されます。また、ユーザーは仮想電話ディレクトリーとしても提供でき、ユーザーは他のユーザーの連絡先情報に簡単にアクセスすることができます。さらに、ユーザーが世界中の他のLDAPサーバーを参照できるようにするため、情報のアドホックなグローバルリポジトリを提供できます。ただし、大学、政府機関、民間企業などの個々の組織で最も頻繁に使用されます。

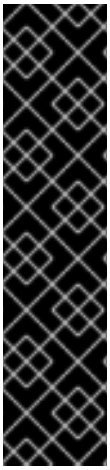
本セクションでは、LDAPv2 プロトコルおよび LDAPv3 プロトコルのオープンソース実装である OpenLDAP 2.4 のインストールおよび設定を説明します。

20.1.1. LDAP の概要

クライアントサーバーアーキテクチャーを使用すると、LDAP は、ネットワークからアクセスできる中央情報ディレクトリーを作成する信頼できる手段を提供します。クライアントがこのディレクトリー内で情報の修正を試みると、サーバーは、ユーザーに変更を行うパーミッションを検証し、要求された時にエントリーを追加または更新します。通信が保護されるようにするには、Transport Layer Security (TLS) 暗号プロトコルを使用して、攻撃者が送信を傍受しないようにすることができます。

重要

Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;LinuxRed Hat Enterprise Linux 6nbsp;6 の OpenLDAP スイートは OpenSSL を使用しなくなりました。代わりに、Network Security Services (NSS)の Mozilla 実装を使用します。OpenLDAP は、引き続き既存の証明書、鍵、およびその他の TLS 設定と連携します。Mozilla 証明書およびキーデータベースを使用する 『ように設定する方法は、「Mozilla NSS で TLS/SSL を使用する方法」を』 参照してください。



重要

Resolution for POODLE SSLv3.0 vulnerability (CVE-2014-3566) for components that do not allow SSLv3 to be disabled via configuration settings(設定から SSLv3 を無効にできないコンポーネントで POODLE SSLv3.0 脆弱性 (CVE-2014-3566) を解決する方法) に説明されている脆弱性により、Red Hat はセキュリティー保護のために SSLv3 に依存しないことを推奨しています。OpenLDAP は、SSLv3 を効果的に無効にできるようにする設定パラメーターを提供しないシステムコンポーネントの 1 つです。リスクを軽減するには、stunnel コマンドを使用してセキュアなトンネルを提供し、SSLv3 の使用から stunnel を無効にすることが推奨されます。stunnel の使用方法は、『Red Hat Enterprise Linux 6 セキュリティーガイド』を参照してください。

LDAPサーバーは、いくつかのデータベースシステムをサポートしているため、管理者は、提供する予定の情報の種類に最適なソリューションを柔軟に選択できます。明確に定義されたクライアントのアプリケーションプログラミングインターフェース (API) により、LDAP サーバーと通信できるアプリケーションの数は多数であり、数量と品質の両方で増加します。

20.1.1.1. LDAP の用語

以下は、本章で使用される LDAP 固有の用語の一覧です。

entry

LDAP ディレクトリー内の単一のユニット。各エントリーは、固有の 識別名 (DN) で識別されます。

attribute

エントリーに直接関連付けられた情報。たとえば、組織が LDAP エントリーとして表されている場合、この組織に関連付けられている属性にはアドレス、ファx 番号などが含まれます。同様に、個人の電話番号やメールアドレスなどの一般的な属性のエントリーとして、ユーザーを表示することもできます。

属性は、単一の値、または順序付けられていないスペースで区切られた値のリストのいずれかを持つことができます。特定の属性は任意ですが、その他は必須です。必要な属性は objectClass クラス定義を使用して指定し、/etc/openldap/slapd.d/cn=config/cn=schema/ ディレクトリーにあるスキーマファイルで確認できます。

属性とそれに対応する値のアサーションは、RDN (Relative Distinguished Name) とも呼ばれます。グローバルで一意となる識別名とは異なり、相対識別名はエントリーごとに一意のみになります。

LDIF

LDAP データ交換形式 (LDIF) は LDAP エントリーのプレーンテキスト表現です。以下の形式を取ります。

```
[id] dn: distinguished_name
attribute_type: attribute_value
attribute_type: attribute_value
...
```

任意の id は、エントリーの編集に使用されるアプリケーションによって決定される数値です。各エントリーには、対応するスキーマファイルにすべて定義されている限り、必要が数の attribute_type と attribute_value のペアを含めることができます。空白行は、エントリーの最後を示します。

20.1.1.2. OpenLDAP の機能

OpenLDAP スイートは、以下の重要な機能を提供します。

- **LDAPv3 サポート**: *LDAP バージョン 2 以降のプロトコルの変更の多くは、LDAP よりセキュアにするように設計されています。また、これには、Simple Authentication and Security Layer(SASL)、Transport Layer Security(TLS)プロトコルのサポートが含まれます。*

- **LDAP Over IPC**: *プロセス間の通信 (IPC) を使用すると、ネットワーク上で通信する必要がなくなります。*

- **IPv6 サポート**: *OpenLDAP は、インターネットプロトコルの次世代である IPv6 (Internet Protocol version 6) に準拠しています。*

- **LDIFv1 サポート**: *OpenLDAP は LDIF バージョン 1 に完全に準拠しています。*

- **更新された C API**: *現在の C API は、プログラマーが LDAP ディレクトリーサーバーに接続し、使用する方法を向上させます。*

- **強化されたスタンドアロン LDAP サーバー**: *これには、更新されたアクセス制御システム、スレッドプール、より良いツールなどが含まれています。*

20.1.1.3. OpenLDAP サーバーの設定

Red Hat Enterprise Linuxnbsp;Hat Enterprise Red Hat Enterprise Linuxnbsp;Linux に LDAP サーバーを設定する一般的な手順は以下のとおりです。

1. OpenLDAP スイートをインストールします。必要なパッケージの詳細は、「OpenLDAP スイートのインストール」 を参照してください。

2. 「OpenLDAP サーバーの設定」 の説明に従って設定をカスタマイズします。

3. 「OpenLDAP サーバーの実行」 の説明に従って slapd サービスを起動します。

4. ldapadd ユーティリティーを使用して、エントリーを LDAP ディレクトリーに追加します。

5. ldapsearch ユーティリティーを使用して、slapd サービスが情報が正しくアクセスされていることを確認します。

20.1.2. OpenLDAP スイートのインストール

OpenLDAP ライブラリーおよびツールのスイートは、以下のパッケージで提供されます。

表20.1 OpenLDAP パッケージの一覧

パッケージ	説明
openldap	OpenLDAP サーバーとクライアントアプリケーションの実行に必要なライブラリーを含むパッケージ。
openldap-clients	LDAP サーバーのディレクトリーを表示および変更するコマンドラインユーティリティーを含むパッケージ。
openldap-servers	LDAP サーバーを設定し、実行するサービスとユーティリティーの両方を含むパッケージ。これには、スタンドアロン LDAP デーモン slapd が含まれます。
compat-openldap	OpenLDAP 互換性ライブラリーを含むパッケージ。

パッケージ	説明

また、以下のパッケージは、一般的に LDAP サーバーで使用されます。

表20.2 一般的にインストールされている追加 LDAP パッケージの一覧

パッケージ	説明
sssd	SSSD(System Security Services Daemon) を含むパッケージ。リモートディレクトリーおよび認証メカニズムへのアクセスを管理するデーモンセットです。システムおよびプラグ可能な認証モジュール(PAM)に Name Service Switch(NSS)インターフェースおよびプラグ可能な認証モジュール(PAM)インターフェースを提供して、複数の異なるアカウントソースに接続します。
mod_authz_ldap	mod_authz_ldap （Apache HTTP Server の LDAP 認証モジュール）が含まれるパッケージ。このモジュールは、サブジェクトとクライアント SSL 証明書の発行者に識別名の短縮形式を使用して、LDAP ディレクトリー内のユーザーの識別名を決定します。また、そのユーザーの LDAP ディレクトリーエントリーの属性に基づいてユーザーを承認し、アセットのユーザーおよびグループの権限に基づいてアセットへのアクセスを判定し、期限切れのパスワードを持つユーザーのアクセスを拒否することもできます。mod_authz_ldap モジュールを使用する場合は mod_ssl モジュールが必要になることに注意してください。

これらのパッケージをインストールするには、以下の形式で yum コマンドを使用します。

```
yum install package
```

たとえば、基本的な LDAP サーバーインストールを実行するには、シェルプロンプトで以下を入力します。

```
~]# yum install openldap openldap-clients openldap-servers
```

このコマンドを実行するには、スーパーユーザーの権限 (つまり root としてログイン) が必要であることに注意してください。Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux に新しいパッケージをインストールする方法の詳細は、「パッケージのインストール」 を参照してください。

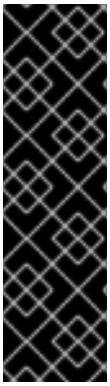
20.1.2.1. OpenLDAP サーバーユーティリティーの概要

管理タスクを実行するには、openldap-servers パッケージにより、slapd サービスとともに次のユーティリティーがインストールされます。

表20.3 OpenLDAP サーバーユーティリティーの一覧

コマンド	説明
slapacl	属性の一覧へのアクセスを確認できます。
slapadd	LDIF ファイルから LDAP ディレクトリーにエントリーを追加できます。
slapauth	認証および承認権限のIDのリストを確認できます。
slapcat	デフォルト形式の LDAP ディレクトリーからエントリーを取得し、LDIF ファイルに保存できます。
slapdn	利用可能なスキーマ構文に基づいて、識別名(DN)の一覧を確認できます。
slapindex	現在の内容に基づいて slapd ディレクトリーを再インデックス化できます。設定ファイルのインデックスオプションを変更する場合に、このユーティリティーを実行します。
slappasswd	ldapmodify ユーティリティーまたは slapd 設定ファイルで使用する暗号化されたユーザーパスワードを作成できます。
slapschema	対応するスキーマでデータベースのコンプライアンスを確認できます。
slaptest	LDAP サーバー設定を確認できるようにします。

これらのユーティリティーとその使用方法の詳細な説明は、「インストールされているドキュメント」 と呼ばれる対応する man ページを参照してください。



ファイルに正しい所有者があることを確認します。

slapadd を実行できるのは root のみですが、slapd サービスは ldap ユーザーとして実行します。このため、ディレクトリーサーバーは slapadd により作成されたファイルを変更できません。この問題を修正するには、slapd ユーティリティーを実行した後に、シェルプロンプトで以下を入力します。

```
~]# chown -R ldap:ldap /var/lib/ldap
```

これらのユーティリティーを使用する前に SLAPD を停止する

データの整合性を保持するには、slapadd、slapcat、または slapindex を使用する前に slapd サービスを停止します。これを行うには、シェルプロンプトで以下を実行できます。

```
~]# service slapd stop
Stopping slapd:                                    [  OK  ]
```

slapd サービスの現在の状態の開始、停止、再起動、および確認の方法は、「OpenLDAP サーバーの実行」 を参照してください。

20.1.2.2. OpenLDAP クライアントユーティリティーの概要

openldap-clients パッケージは、LDAP ディレクトリーのエントリーの追加、変更、および削除に使用できる以下のユーティリティーをインストールします。

表20.4 OpenLDAP クライアントユーティリティーの一覧

コマンド	説明
ldapadd	エントリーは、ファイルまたは標準入力から LDAP ディレクトリーに追加できます。ldapmodify -a へのシンボリックリンクです。
ldapcompare	指定属性を LDAP ディレクトリーエントリーと比較できます。
ldapdelete	LDAP ディレクトリーからエントリーを削除できます。
ldapexop	拡張 LDAP 操作を実行できます。

コマンド	説明
ldapmodify	LDAP ディレクトリー（ファイルまたは標準入力のいずれか）のエントリーを変更できます。
ldapmodrdn	LDAP ディレクトリーエントリーの RDN 値を変更できます。
ldappasswd	LDAP ユーザーのパスワードを設定または変更できるようにします。
ldapsearch	LDAP ディレクトリーエントリーを検索できます。
ldapurl	LDAP URL の組み立てまたは分解を可能にします。
ldapwhoami	LDAP サーバーで whoami 操作を実行できます。

ldapsearch の例外により、各ユーティリティーは、LDAP ディレクトリー内で変更する各エントリーに対してコマンドを入力するのではなく、変更を含むファイルを参照することで簡単に使用できます。このようなファイルの形式は、各ユーティリティーの man ページで説明されています。

20.1.2.3. 共通 LDAP クライアントアプリケーションの概要

サーバー上でディレクトリーを作成して変更できる各種のグラフィカル LDAP クライアントがありますが、Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux にはこれらは含まれません。読み取り専用モードでディレクトリーにアクセスできる一般的なアプリケーションには、Mozilla Thunderbird、Evolution、Ekiga が含まれます。

20.1.3. OpenLDAP サーバーの設定

デフォルトでは、OpenLDAP は設定を /etc/openldap/ ディレクトリーに保存します。表 20.5「OpenLDAP 設定ファイルとディレクトリーの一覧」 このディレクトリー内の最も重要なファイルおよびディレクトリーを強調表示します。

表20.5 OpenLDAP 設定ファイルとディレクトリーの一覧

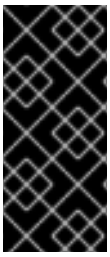
パス	説明
/etc/openldap/ldap.conf	OpenLDAP ライブラリーを使用するクライアントアプリケーションの設定ファイルこれには ldapadd、ldapsearch、Evolution などが含まれます。
/etc/openldap/slapd.d/	slapd 設定が含まれるディレクトリー。

パス	説明

Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;LinuxRed Hat Enterprise Linux 6nbsp;6 では、slapd サービスは /etc/openldap/slapd.d/ ディレクトリーにある設定データベースを使用し、このディレクトリーが存在しない場合は、古い /etc/openldap/slapd.conf 設定ファイルのみを読み取ります。以前のインストールの既存の slapd.conf ファイルがある場合は、次にこのパッケージを更新する際に openldap-servers パッケージが新しい形式に変換するか、root で次のコマンドを実行します。

```
~]# slaptest -f /etc/openldap/slapd.conf -F /etc/openldap/slapd.d/
```

slapd 設定は、階層的なディレクトリー構造で整理された LDIF エントリーで構成されます。これらのエントリーを編集する方法として、「OpenLDAP サーバーユーティリティーの概要」 に記載されているサーバーユーティリティーを使用することが推奨されます。



LDIF ファイルを直接編集しないでください。

LDIF ファイルのエラーは、slapd サービスを起動できない場合があります。このため、/etc/openldap/slapd.d/ ディレクトリーの LDIF ファイルを直接編集しないことが強く推奨されます。

20.1.3.1. グローバル設定の変更

LDAP サーバーのグローバル設定オプションは、/etc/openldap/slapd.d/cn=config.ldif ファイルに保存されます。一般的には、以下のディレクティブが使用されます。

olcAllows

olcAllows ディレクティブを使用すると、有効にする機能を指定できます。以下の形式を取ります。

```
olcAllows: feature
```

表20.6「利用可能なolcAllows オプション」 に記載されている、スペースで区切られた機能のリストを受け入れます。デフォルトオプションは bind_v2 です。

表20.6 利用可能なolcAllows オプション

オプション	説明
bind_v2	LDAP バージョン 2 バインド要求の受け入れを有効にします。
bind_anon_cred	識別名(DN) が空でない場合は匿名バインドを有効にします。
bind_anon_dn	識別名(DN) が空でない 場合は匿名バインドを有効にします。
update_anon	匿名更新操作の処理を有効にします。
proxy_authz_anon	匿名プロキシーの承認制御の処理を有効にします。

例20.1 olcAllows ディレクティブの使用

```
olcAllows: bind_v2 update_anon
```

olcConnMaxPending

olcConnMaxPending ディレクティブを使用すると、匿名セッションの保留中の要求の最大数を指定できます。以下の形式を取ります。

```
olcConnMaxPending: number
```

デフォルトオプションは 100 です。

例20.2 olcConnMaxPending ディレクティブの使用

```
olcConnMaxPending: 100
```

olcConnMaxPendingAuth

olcConnMaxPendingAuth ディレクティブを使用すると、認証されたセッションの保留中のリクエストの最大数を指定できます。以下の形式を取ります。

```
olcConnMaxPendingAuth: number
```

デフォルトオプションは 1000 です。

例20.3 olcConnMaxPendingAuth ディレクティブの使用

```
olcConnMaxPendingAuth: 1000
```

olcDisallows

olcDisallows ディレクティブを使用すると、無効にする機能を指定できます。以下の形式を取ります。

```
olcDisallows: feature
```

表20.7「利用可能な olcDisallows オプション」 に記載されている、スペースで区切られた機能のリストを受け入れます。デフォルトでは、機能は無効になりません。

表20.7 利用可能な olcDisallows オプション

オプション	説明
bind_anon	匿名バインド要求の受け入れを無効にします。
bind_simple	簡単なバインド認証メカニズムを無効にします。
tls_2_anon	STARTTLS コマンドを受け取ると、匿名セッションの強制を無効にします。
tls_authc	認証時に STARTTLS コマンドを許可しません。

例20.4 olcDisallows ディレクティブの使用

```
olcDisallows: bind_anon
```

olcIdleTimeout

olcIdleTimeout ディレクティブを使用すると、アイドル状態の接続を閉じる前に待機する秒数を指定できます。以下の形式を取ります。

```
olcIdleTimeout: number
```

このオプションは、デフォルトでは無効になっています (つまり 0 に設定されます)。

例20.5 olcIdleTimeout ディレクティブの使用

```
olcIdleTimeout: 180
```

olcLogFile

olcLogFile ディレクティブを使用すると、ログメッセージを書き込むファイルを指定できます。以下の形式を取ります。

```
olcLogFile: file_name
```

ログメッセージはデフォルトで標準エラーに書き込まれます。

例20.6 olcLogFile ディレクティブの使用

```
olcLogFile: /var/log/slapd.log
```

olcReferral

olcReferral オプションでは、サーバーがこれを処理できない場合に、要求を処理するサーバーの URL を指定できます。以下の形式を取ります。

```
olcReferral: URL
```

このオプションはデフォルトで無効になっています。

例20.7 olcReferral ディレクティブの使用

```
olcReferral: ldap://root.openldap.org
```

olcWriteTimeout

olcWriteTimeout オプションでは、未処理の書き込み要求との接続を閉じる前に待機する秒数を指定できます。以下の形式を取ります。

olcWriteTimeout

このオプションは、デフォルトでは無効になっています (つまり 0 に設定されます)。

例20.8 olcWriteTimeout ディレクティブの使用

```
olcWriteTimeout: 180
```

20.1.3.2. データベース固有の設定の変更

デフォルトでは、OpenLDAP サーバーは Berkeley DB(BDB)をデータベースバックエンドとして使用します。このデータベースの設定は、/etc/openldap/slapd.d/cn=config/olcDatabase={2}bdb.ldif ファイルに保存されます。以下のディレクティブは、データベース固有の設定で一般的に使用されます。

olcReadOnly

olcReadOnly ディレクティブを使用すると、データベースを読み取り専用モードで使用できます。以下の形式を取ります。

```
olcReadOnly: boolean
```

TRUE (読み取り専用モードを有効) または FALSE (データベースの変更を有効) のいずれかを受け入れます。デフォルトのオプションは FALSE です。

例20.9 olcReadOnly ディレクティブの使用

```
olcReadOnly: TRUE
```

olcRootDN

olcRootDN ディレクティブを使用すると、LDAP ディレクトリー上の操作に設定されたアクセス制御または管理制限パラメーターが無制限のユーザーを指定できます。以下の形式を取ります。

```
olcRootDN: distinguished_name
```

識別名 (DN) を受け入れます。デフォルトのオプションは cn=Manager,dc=my-domain,dc=com です。

例20.10 olcRootDN ディレクティブの使用

```
olcRootDN: cn=root,dc=example,dc=com
```

olcRootPW

olcRootPW ディレクティブを使用すると、olcRootDN ディレクティブを使用して指定されるユーザーのパスワードを設定できます。以下の形式を取ります。

```
olcRootPW: password
```

プレーンテキストの文字列またはハッシュのいずれかを指定できます。ハッシュを生成するには、シェルプロンプトで以下を入力します。

```
~]$ slappaswd
New password:
Re-enter new password:
{SSHA}WczWsyPEnMchFf1GRTweq2q7XJcvmSxD
```

例20.11 olcRootPW ディレクティブの使用

```
olcRootPW: {SSHA}WczWsyPEnMchFf1GRTweq2q7XJcvmSxD
```

olcSuffix

olcSuffix ディレクティブでは、情報を提供するドメインを指定できます。以下の形式を取ります。

```
olcSuffix: domain_name
```

完全修飾ドメイン名 (FQDN) を受け入れます。デフォルトのオプションは dc=my-domain,dc=com です。

例20.12 olcSuffix ディレクティブの使用

```
olcSuffix: dc=example,dc=com
```

20.1.3.3. スキーマの拡張

OpenLDAP 2.3 以降、/etc/openldap/slapd.d/cn=config/cn=schema/ ディレクトリーには、/etc/openldap/schema/ に以前あった LDAP 定義も含まれます。OpenLDAPで使用されるスキーマを拡張して、デフォルトのスキーマファイルをガイドとして使用して、追加の属性タイプとオブジェクトクラスをサポートすることができます。ただし、このタスクは本章の範囲外です。このトピックに関する詳細は を参照してください http://www.openldap.org/doc/admin/schema.html。

20.1.4. OpenLDAP サーバーの実行

本セクションでは、スタンドアロンの LDAP デーモン を起動、停止、再起動、および現在のステータスの確認を行う方法を説明します。システムサービスの管理全般に関する詳細情報は、12章サービスおよびデーモン を参照してください。

20.1.4.1. サービスの起動

slapd サービスを実行するには、シェルプロンプトで以下を入力します。

```
~]# service slapd start
Starting slapd:                                [  OK  ]
```

システムの起動時にサービスを自動的に起動するようにするには、以下のコマンドを使用します。

```
~]# chkconfig slapd on
```

「サービスの有効化および無効化」 で説明されているように、Service Configuration ユーティリティーを使用することもできます。

20.1.4.2. サービスの停止

実行中の slapd サービスを停止するには、シェルプロンプトで以下を入力します。

```
~]# service slapd stop
Stopping slapd:                                [  OK  ]
```

システムの起動時にサービスが自動的に起動しないようにするには、以下を入力します。

```
~]# chkconfig slapd off
```

または、「サービスの有効化および無効化」 の説明に従って Service Configuration ユーティリティーを使用できます。

20.1.4.3. サービスの再起動

実行中の slapd サービスを再起動するには、シェルプロンプトで以下を入力します。

```
~]# service slapd restart
Stopping slapd:                                [  OK  ]
Starting slapd:                                [  OK  ]
```

これにより、サービスが停止し、再起動します。以下のコマンドを使用して、設定を再読み込みします。

20.1.4.4. サービスステータスの確認

サービスが実行中かどうかを確認するには、シェルプロンプトで以下を入力します。

```
~]# service slapd status
slapd (pid  3672) is running...
```

20.1.5. OpenLDAP を使用してシステムを認証するためのシステムの設定

OpenLDAP を使用してシステムを認証するように設定するには、適切なパッケージが LDAP サーバーとクライアントマシンの両方にインストールされていることを確認してください。サーバーの設定方法は、「OpenLDAP スイートのインストール」 および 「OpenLDAP サーバーの設定」 の手順に従います。クライアントで、シェルプロンプトで以下を入力します。

```
~]# yum install openldap openldap-clients sssd
```

13章認証の設定 では、認証に LDAP を使用するようにアプリケーションを設定する方法に関する詳細な手順を説明します。

20.1.5.1. 以前の認証情報の LDAP 形式への移行

migrationtools パッケージは、認証情報を LDAP 形式に移行するのに役立つシェルおよび Perl スクリプトのセットを提供します。このパッケージをインストールするには、シェルプロンプトで以下を

入力します。

```
~]# yum install migrationtools
```

これにより、スクリプトが /usr/share/migrationtools/ ディレクトリーにインストールされます。インストールが完了したら、/usr/share/migrationtools/migrate_common.ph ファイルを編集し、以下の行を変更して正しいドメインを反映させます。

```
# Default DNS domain
$DEFAULT_MAIL_DOMAIN = "example.com";

# Default base
$DEFAULT_BASE = "dc=example,dc=com";
```

または、コマンドラインで直接環境変数を指定することもできます。たとえば、デフォルトのベースを dc=example,dc=com に設定して migrate_all_online.sh スクリプトを実行するには、以下を入力します。

```
~]# export DEFAULT_BASE="dc=example,dc=com" \
/usr/share/migrationtools/migrate_all_online.sh
```

ユーザーデータベースを移行するために実行するスクリプトを決定するには、表20.8「一般的に使用される LDAP 移行スクリプト」 を参照してください。

表20.8 一般的に使用される LDAP 移行スクリプト

既存のネームサービス	LDAP が実行しているか?	使用するスクリプト
/etc フラットファイル	はい	migrate_all_online.sh
/etc フラットファイル	いいえ	migrate_all_offline.sh
NetInfo	はい	migrate_all_netinfo_online.sh
NetInfo	いいえ	migrate_all_netinfo_offline.sh
NIS (YP)	はい	migrate_all_nis_online.sh
NIS (YP)	いいえ	migrate_all_nis_offline.sh

このスクリプトの使用方法は、/usr/share/doc/migrationtools-version/ ディレクトリーの

README ファイルおよび migration-tools.txt ファイルを参照してください。

20.1.6. その他のリソース

以下のリソースは、Lightweight Directory Access Protocol に関する追加情報を提供します。システムで LDAP を設定する前に、『OpenLDAP Software 管理者ガイド』 など、これらのリソースを確認することを強く推奨します。

20.1.6.1. インストールされているドキュメント

以下のドキュメントは、openldap-servers パッケージでインストールされます。

/usr/share/doc/openldap-servers-version/guide.html

『OpenLDAP ソフトウェアチャンネルのガイド の』 コピー。

/usr/share/doc/openldap-servers-version/README.schema

インストールされたスキーマファイルの説明が含まれる README ファイル。

また、パッケージ openldap、openldap-servers、および openldap-clientsでインストールされる man ページも多数あります。

クライアントアプリケーション

- man ldapadd: LDAP ディレクトリーにエントリーを追加する方法を説明します。
- man ldapdelete: LDAP ディレクトリー内のエントリーを削除する方法を説明します。
- man ldapmodify: LDAP ディレクトリー内のエントリーを変更する方法を説明しています。
- man ldapsearch: LDAP ディレクトリー内のエントリーの検索方法を説明します。
-

man ldappasswd: LDAP ユーザーのパスワードを設定または変更する方法を説明します。

- *man ldapcompare: ldapcompare ツールの使用方法について説明しています。*
- *man ldapwhoami: ldapwhoami ツールの使用方法について説明しています。*
- *man ldapmodrdn: エントリーの RDN を変更する方法を説明しています。*

サーバーアプリケーション

- *man slapd: LDAP サーバーのコマンドラインオプションを説明しています。*

管理アプリケーション

- *man slapadd: slapd データベースにエントリーを追加するために使用されるコマンドラインオプションを説明しています。*
- *man slapcat: slapd データベースから LDIF ファイルを生成するために使用されるコマンドラインオプションを説明しています。*
- *man slapindex: slapd データベースの内容に基づいてインデックスを再生成するために使用されるコマンドラインオプションを説明しています。*
- *man slappasswd: LDAP ディレクトリーのユーザーパスワードを生成するのに使用されるコマンドラインオプションを説明しています。*

設定ファイル

- *man ldap.conf: LDAP クライアントの設定ファイルで利用可能な形式とオプションを説明しています。*
- *man slapd-config: 設定ディレクトリー内で利用可能な形式とオプションを説明して*

います。

20.1.6.2. 便利な Web サイト

http://www.openldap.org/doc/admin24/

『OpenLDAP ソフトウェアチャンネルガイドの』 現行バージョン

20.1.6.3. 関連書籍

『OpenLDAP by Example』 by John Terpstra and Benjamin Coles; Prentice Hall.

OpenLDAP デプロイメントにおける実用的な演習のコレクション。

Mark Wilcox(Wrox Press, Inc)による 『LDAP の実装』。

システム管理者とソフトウェア開発者の両方の観点からの LDAP について扱います。

Tim Howes et al.; Macmillan Technical Publishing による 『LDAP Directory サービスの理解およびデプロイ』

LDAP 設計原則、および実稼働環境におけるデプロイメントについて説明します。

第21章 ファイルとプリントサーバー

21.1. SAMBA

Samba は、Linux 向けプログラムの標準的なオープンソース Windows 相互運用性スイートです。サーバーメッセージブロック (SMB)プロトコルを実装します。このプロトコルの最新版は、一般的なインターネットファイルシステム (CIFS)プロトコルとしても知られています。Microsoft Windows®、Linux、UNIX、およびその他のオペレーティングシステムのネットワークを有効にし、Windows ベースのファイルおよびプリンター共有へのアクセスを可能にします。Samba の SMB を使用すると、Windows クライアントへの Windows サーバーとして表示できます。

SAMBA パッケージのインストール

Samba を使用するには、最初に root で以下のコマンドを実行して、samba パッケージがシステムにインストールされていることを確認します。

```
~]# yum install samba
```

yum を使用したパッケージのインストールは 「パッケージのインストール」 を参照してください。

21.1.1. Samba の概要

Samba は、Linux Server と Warehouse を Active Directory(AD)環境にシームレスに統合する重要なコンポーネントです。ドメインコントローラー(NT4-style)または通常のドメインメンバー（AD または NT4-style）として機能できます。

Samba の機能：

- Linux、UNIX、Windows クライアントにディレクトリーツリーとプリンターを提供する
- ネットワークブラウズの支援（NetBIOS を使用）
- Windows ドメインログインの認証
- Windows Internet Name Service (WINS)ネームサーバー解決の提供
-

Windows の NT®スタイルの プライマリードメインコントローラー (PDC)として機能します。

- *Samba ベースの P DC のバックアップドメインコントローラー (BDC)として機能します。*
- *Active Directory ドメインメンバーサーバーとして機能*
- *Join a Windows NT/2000/2003/2008 PDC*

Samba が実行できる内容：

- *Windows PDC の BDC として機能します（その逆も同様です）。*
- *Active Directory ドメインコントローラーとして動作する*

21.1.2. Samba デーモンと関連サービス

Samba は、3 つのデーモン（smbd、nmbd、および winbindd）で構成されています。3 つのサービス（smb、nb、および winbind）は、デーモンの開始、停止、およびその他のサービス関連の機能を制御します。これらのサービスは、異なる init スクリプトとして機能します。各デーモンは、以下に詳細と、どの特定のサービスがこれを制御するかを示しています。

smbd

smbd サーバーデーモンは、Windows クライアントにファイル共有および印刷サービスを提供します。また、SMB プロトコルを介してユーザー認証、リソースロック、およびデータ共有を行います。サーバーが SMB トラフィックをリッスンするデフォルトのポートは TCP ポート 139 および 445 です。

smbd デーモンは、smb サービスにより制御されます。

nmbd

nmbd サーバーデーモンは、Windows ベースのシステムで SMB/CIFS が生成するなど、NetBIOS ネームサービス要求を理解し、応答します。このシステムには、Windows 95/98/ME、Windows NT、Windows 2000、Windows XP、および LanManager クライアントが含まれます。また、Windows Network Neighborhood ビューを構成する参照プロトコルにも参加します。サーバーが NMB トラフィックをリッスンするデフォルトのポートは UDP ポート 137 です。

nmbd デーモンは nmb サービスによって制御されます。

winbindd

winbind サービスは、Windows NT、Vagrant、Windows Server 2008、または Windows Server 2012 を実行しているサーバーから受け取ったユーザーおよびグループの情報を解決します。これにより、UNIX プラットフォームで Windows ユーザーおよびグループの情報を理解できるようになります。これは、Microsoft RPC 呼び出し、Pluggable Authentication Modules (PAM)、および Name Service Switch (NSS)を使用して実現されます。これにより、Windows NT ドメインおよび Active Directory ユーザーを表示し、UNIX マシンで UNIX ユーザーとして操作できます。Samba ディストリビューションにバンドルされていますが、winbind サービスは smb サービスとは別に制御されます。

winbind デーモンは winbind サービスによって制御され、動作のために smb サービスを起動する必要はありません。Samba が Active Directory メンバーである場合にも winbind を使用し、Samba ドメインコントローラーでも使用できます（ネスト化されたグループとドメイン間の信頼を実装するため）。winbind は Windows NT ベースのサーバーへの接続に使用されるクライアント側のサービスであるため、winbind の詳細な説明は本章の対象外となります。

認証に winbind を設定する方法は、「Winbind 認証の設定」 を参照してください。

SAMBA に含まれるユーティリティーの一覧の取得

Samba ディストリビューションに含まれるユーティリティーの一覧については、「Samba ディストリビューションプログラム」 を参照してください。

21.1.3. Samba 共有への接続

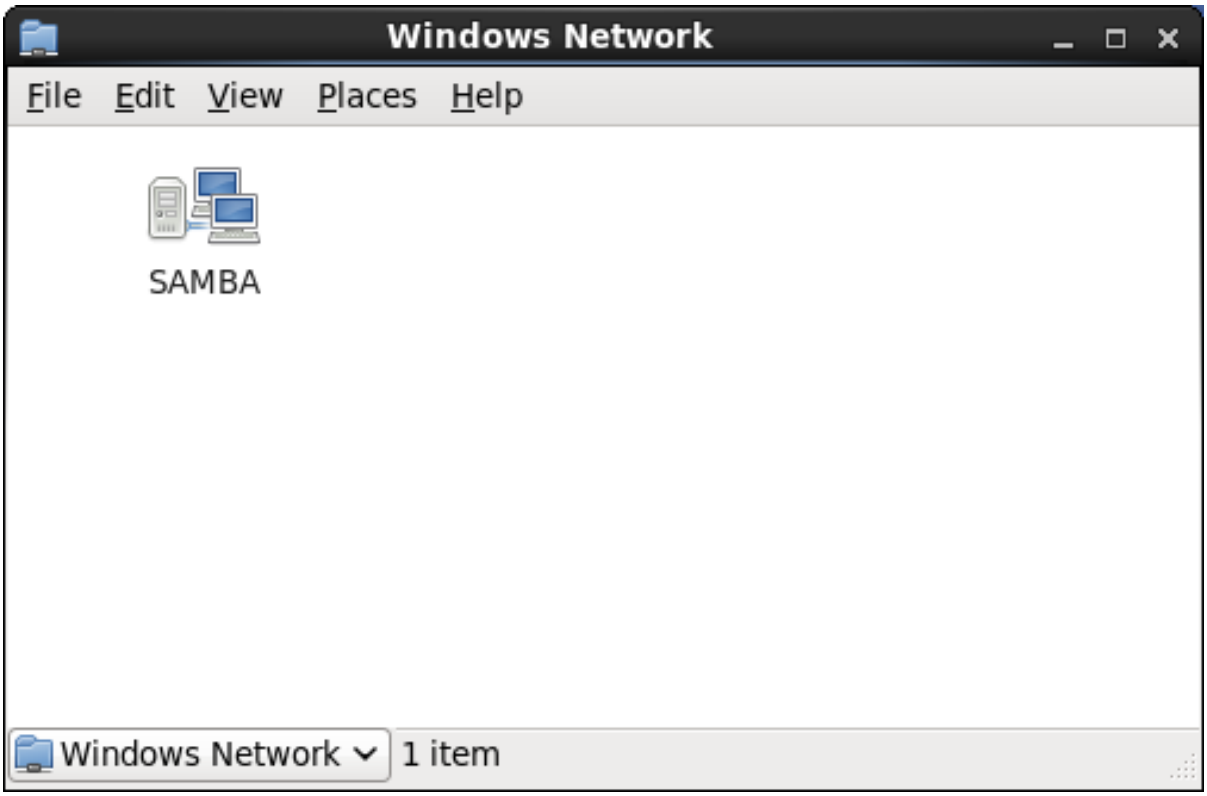
nautilus またはコマンドラインのいずれ か を使用して、利用可能な Samba 共有に接続できます。

手順21.1 nautilus を使用した Samba 共有への接続

1. ネットワーク上の Samba ワークグループおよびドメインの一覧を表示するには、GNOME パネルから Places → Network を選択し、必要なネットワークを選択します。または、Nautilus の File → Open Location バーに smb: と入力します。

   図21.1「nautilus の SMB Workgroups」 に示されているように、ネットワーク上の利用可能な SMB ワークグループまたはドメインごとにアイコンが表示されます。

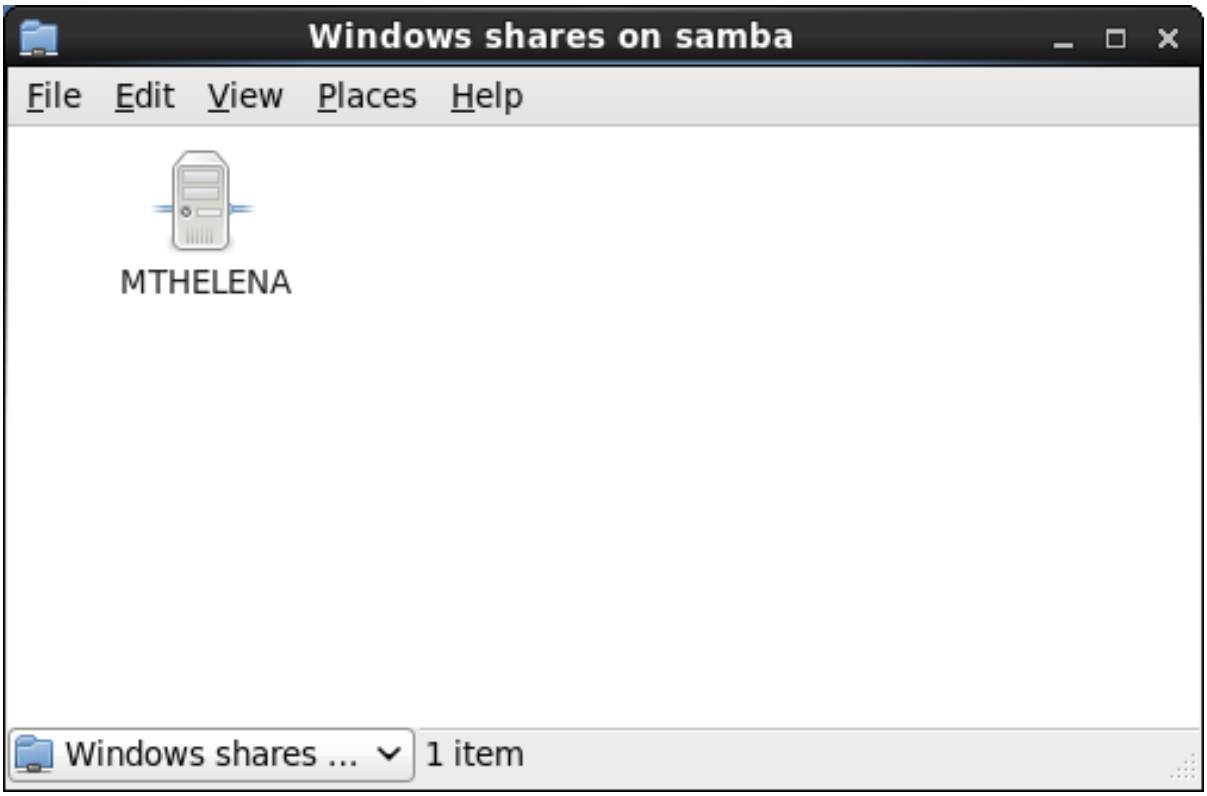
図21.1 nautilus の SMB Workgroups



[D]

2. *ワークグループまたはドメインアイコンのいずれかをダブルクリックして、ワークグループまたはドメイン内のコンピューターの一覧を表示します。*

図21.2 nautilus の SMB マシン



[D]

3.

図21.2「nautilus の SMB マシン」 に表示されているように、ワークグループ内の各マシンにはアイコンが存在します。アイコンをダブルクリックして、マシン上で Samba 共有を表示します。ユーザー名とパスワードの組み合わせが必要な場合は、そのユーザー名とパスワードの入力が求められます。

または、次の構文を使用して、Nautilus の Location: bar に Samba サーバーおよび sharename を指定することもできます（ servername と sharename を適切な値に置き換えます）。

```
smb://servername/sharename
```

手順21.2 コマンドラインを使用した Samba 共有への接続

1.

Samba サーバーのネットワークをクエリーするには、findsmb コマンドを使用します。見つかった各サーバーに対して、IP アドレス、NetBIOS 名、ワークグループ名、オペレーティングシステム、および SMB サーバーバージョンが表示されます。

```
findsmb
```

2.

シェルプロンプトから Samba 共有に接続するには、以下のコマンドを入力します。

```
smbclient //hostname/sharename -U username
```

hostname を、接続する Samba サーバーのホスト名または IP アドレスに置き換え、sharename は参照する共有ディレクトリーの名前に、username は、システムの Samba ユーザー名に置き換えます。正しいパスワードを入力するか、ユーザーにパスワードが必要ない場合は Enter を押します。

smb:\> プロンプトが表示された場合には、正常にログインしています。ログインしたら、コマンドのリストとして help と入力します。ホームディレクトリーの内容を参照する場合は、sharename をユーザー名に置き換えます。-U スイッチを使用しない場合、現行ユーザーのユーザー名が Samba サーバーに渡されます。

3. smbclient を終了するには、exit プロンプトで smb:\> と入力します。

21.1.3.1. 共有のマウント

Samba 共有をディレクトリーにマウントして、ディレクトリー内のファイルをローカルファイルシステムの一部であるかのように処理できるように、Samba 共有をマウントすると便利な場合があります。

Samba 共有をディレクトリーにマウントするには、それをマウントするディレクトリーを作成し（存在しない場合）、root で以下のコマンドを実行します。

```
mount -t cifs //servername/sharename /mnt/point/ -o
username=username,password=password
```

このコマンドは、/mnt/point/ の servername から sharename をマウントします。

samba 共有のマウントの詳細は、mount.cifs(8) man ページを参照してください。

CIFS-UTILS パッケージのインストール

mount.cifs ユーティリティーは別の RPM（Samba に依存しない）です。mount.cifs を使用するには、最初に root で以下のコマンドを実行して、cifs-utils パッケージがインストールされていることを確認します。

```
~]# yum install cifs-utils
```

yum を使用したパッケージのインストールは 「パッケージのインストール」 を参照してください。

cifs-utils パッケージには、kerberized CIFS マウントを実行するためにカーネルが呼び出される cifs.upcall バイナリーが含まれることに注意してください。cifs.upcall の詳細は、cifs.upcall(8) の man ページを参照してください。

プレーンテキストのパスワードを必要とする CIFS サーバー

一部の CIFS サーバーでは、認証にプレーンテキストのパスワードが必要です。プレーンテキストのパスワード認証のサポートは、root で以下のコマンドを使用して有効にできます。

```
~]# echo 0x37 > /proc/fs/cifs/SecurityFlags
```

警告： この操作は、パスワード暗号化を削除してパスワードを公開できます。

21.1.4. Samba サーバーの設定

デフォルトの設定ファイル(/etc/samba/smb.conf)を使用すると、ユーザーはホームディレクトリーを Samba 共有として表示できます。また、システムに設定されたすべてのプリンターを Samba 共有プリンターとして共有します。システムにプリンターを割り当てて、ネットワーク上の Windows マシンからプリンターを印刷できます。

21.1.4.1. グラフィカル設定

グラフィカルインターフェースを使用して Samba を設定するには、利用可能な Samba グラフィカルユーザーインターフェースのいずれかを使用します。利用可能な GUI の一覧

は、http://www.samba.org/samba/GUI/ を参照してください。

21.1.4.2. コマンドライン設定

Samba は、/etc/samba/smb.conf を設定ファイルとして使用します。この設定ファイルを変更すると、root で次のコマンドを実行して Samba デーモンを再起動するまで変更は反映されません。

```
~]# service smb restart
```

Windows のワークグループと Samba サーバーの簡単な説明を指定するには、/etc/samba/smb.conf ファイルで次の行を編集します。

```
workgroup = WORKGROUPNAME
server string = BRIEF COMMENT ABOUT SERVER
```

WORKGROUPNAME は、このマシンが属する Windows ワークグループ名に置き換えます。BRIEF COMMENT ABOUT SERVER はオプションで、Samba システムに関する Windows コメントとして使用されます。

Linux システムに Samba 共有ディレクトリーを作成するには、/etc/samba/smb.conf ファイルに以下のセクションを追加します（必要に応じて変更を反映するように修正します）。

例21.1 Samba サーバーの設定例

```
[sharename]
comment = Insert a comment here
path = /home/share/
valid users = tfox carole
writable = yes
create mask = 0765
```

上記の例では、Samba クライアントから、Samba サーバーの /home/share/ ディレクトリーに対する tfox および carole ユーザーが読み書きできるようになります。

21.1.4.3. 暗号化パスワード

暗号化パスワードは、より安全に使用できるため、デフォルトで有効になっています。暗号化されたパスワードでユーザーを作成するには、smbpasswd ユーティリティーを使用します。

```
smbpasswd -a username
```

21.1.5. Samba の起動および停止

Samba サーバーを起動するには、root で次のコマンドを実行します。

```
~]# service smb start
```

ドメインメンバーサーバーの設定

ドメインメンバーサーバーを設定するには、smb サービスを起動する 前 に net join コマンドを使用してドメインまたは Active Directory に参加する必要があります。また、smbd の前に winbind を実行することが推奨されます。

サーバーを停止するには、root で次のコマンドを実行します。

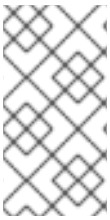
```
~]# service smb stop
```

restart オプションは、Samba を停止して起動する簡単な方法です。これは、Samba の設定ファイルを編集した後に設定変更を行う最も信頼性の高い方法です。restart オプションは、最初に実行していない場合でもデーモンを起動することに注意してください。

サーバーを再起動するには、root で次のコマンドを実行します。

```
~]# service smb restart
```

condrestart (conditional restart)オプションは、現在実行している条件で smb のみを停止し、開始します。このオプションは、デーモンが実行されていない場合はデーモンを起動しないため、スクリプトに便利です。



設定への変更の適用

/etc/samba/smb.conf ファイルを変更すると、Samba は数分後に自動的に再読み込みされます。手動での 再起動 または リロード の発行は効果的です。

サーバーを条件付きで再起動するには、root で以下のコマンドを入力します。

```
~]# service smb condrestart
```

-

/etc/samba/smb.conf ファイルの手動によるリロードは、smb サービスが自動再読み込みに失敗した場合に役に立ちます。サービスを再起動せずに Samba サーバー設定ファイルが再読み込みされるようにするには、root で以下のコマンドを入力します。

```
~]# service smb reload
```

デフォルトでは、smb サービスは、システムの起動時に自動的に起動し ません。Samba が起動時に開始するように設定するには、/sbin/chkconfig、/usr/sbin/ntsysv、または Services Configuration Tool プログラムなどの initscript ユーティリティーを使用します。これらのツールの詳細は、12章サービスおよびデーモン を参照してください。

21.1.6. Samba サーバータイプおよび smb.conf ファイル

Samba の設定が簡単です。Samba へのすべての変更は、/etc/samba/smb.conf 設定ファイルで行います。デフォルトの smb.conf ファイルは十分に文書化されていますが、LDAP、Active Directory、多くのドメインコントローラー実装などの複雑なトピックには対応しません。

以下のセクションでは、Samba サーバーを設定するさまざまな方法を説明します。設定を成功させるには、ニーズと /etc/samba/smb.conf ファイルに必要な変更に注意してください。

21.1.6.1. スタンドアロンサーバー

スタンドアロンサーバーは、ワークグループサーバーまたはワークグループ環境のメンバーになります。スタンドアロンサーバーはドメインコントローラーではなく、ドメインに参加しません。以下の例には、ユーザーレベルのセキュリティー設定が複数含まれています。セキュリティーモードの詳細は、「Samba セキュリティーモード」 を参照してください。

Anonymous Read-Only

以下の /etc/samba/smb.conf ファイルは、匿名の読み取り専用ファイル共有の実装に必要な設定例を示しています。匿名アクセスの設定には、map to guest = Bad user および guest account = nobody の 2 つのディレクティブが使用されます。

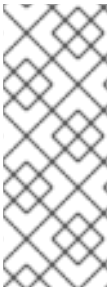
例21.2 Anonymous Read-Only Samba サーバーの設定例

```
[global]
workgroup = DOCS
netbios name = DOCS_SRV
security = user
guest account = nobody # default value
map to guest = Bad user

[data]
```

```
comment = Documentation Samba Server
path = /export
read only = yes
guest ok = yes
```

Anonymous Read/Write

以下の /etc/samba/smb.conf ファイルは、匿名の読み取り/書き込みファイル共有の実装に必要な設定例を示しています。匿名の読み取り/書き込みファイル共有を有効にするには、read only ディレクティブを no に設定します。force user ディレクティブおよび force group ディレクティブも追加され、共有で指定した新たに配置されたファイルの所有権を強制します。



匿名の読み取り/書き込みサーバーは使用しないでください。

匿名の読み取り/書き込みサーバーの使用は可能ですが、推奨されません。ユーザーに関係なく、共有スペースに置かれたファイルはすべて、/etc/samba/smb.conf ファイルの汎用ユーザー(force user)およびグループ(force group)で指定されたユーザー/グループの組み合わせが割り当てられます。

例21.3 Anonymous Read/Write Samba サーバーの設定例

```
[global]
workgroup = DOCS
security = user
guest account = nobody # default value
map to guest = Bad user

[data]
comment = Data
path = /export
guest ok = yes
writeable = yes
force user = user
force group = group
```

Anonymous Print Server

以下の /etc/samba/smb.conf ファイルは、匿名プリントサーバーの実装に必要な設定例を示しています。上記のように browseable を no に設定しても、Windows Network Neighborhood にプリンターは表示されません。参照を外すことはできませんが、プリンターを明示的に設定することはできません。NetBIOS を使用して DOCS_SRV に接続すると、クライアントが DOCS ワークグループに含まれる場合に、クライアントがプリンターにアクセスできます。また、use client driver ディレクティブは yes に設定されているため、クライアントに正しいローカルプリンタードライバーがインストールされていることを前提とします。この場合、Samba サーバーはプリンタードライバーをクライアントと共有する責任はありません。

例21.4 Anonymous Print Samba サーバーの設定例

```
[global]
workgroup = DOCS
netbios name = DOCS_SRV
security = user
map to guest = Bad user
printing = cups

[printers]
comment = All Printers
path = /var/spool/samba
guest ok = yes
printable = yes
use client driver = yes
browseable = yes
```

読み取り/書き込みファイルおよびプリントサーバーの保護

以下の /etc/samba/smb.conf ファイルは、セキュアな読み取り/書き込みファイルおよびプリントサーバーの実装に必要な設定例を示しています。security ディレクティブを user に設定すると、Samba がクライアント接続を認証するように強制します。[homes] 共有には [public] 共有が行われるため、force user または force group ディレクティブがないことに注意してください。[homes] 共有は、[public] の force user および force group ではなく、作成されたすべてのファイルに認証されたユーザーの詳細を使用します。

例21.5 セキュアな読み取り/書き込みファイルの設定例および Samba サーバーの印刷

```
[global]
workgroup = DOCS
netbios name = DOCS_SRV
security = user
printcap name = cups
disable spools = yes
show add printer wizard = no
printing = cups

[homes]
comment = Home Directories
valid users = %S
read only = no
browseable = no

[public]
comment = Data
path = /export
force user = docsbot
force group = users
guest ok = yes

[printers]
comment = All Printers
path = /var/spool/samba
printer admin = john, ed, @admins
create mask = 0600
```

```
guest ok = yes
printable = yes
use client driver = yes
browseable = yes
```

21.1.6.2. ドメインメンバーサーバー

スタンドアロンサーバーと同様に、ドメインメンバーはドメインコントローラー（Windows または Samba のいずれか）にログインし、ドメインのセキュリティールールの対象となります。ドメインメンバーサーバーの例として、Samba を実行する部門サーバーがあり、プライマリードメインコントローラー(PDC)上のマシンアカウントがあります。すべての部署のクライアントは、引き続き PDC で認証され、デスクトッププロファイルとすべてのネットワークポリシーファイルが含まれます。相違点は、部署サーバーがプリンターとネットワーク共有を制御できることです。

Active Directory ドメインメンバーサーバー

Active Directory ドメインメンバーサーバーを実装するには、以下の手順に従います。

手順21.3 Active Directory ドメインへのメンバーサーバーの追加

1. Active Directory ドメインに追加するメンバーサーバーに /etc/samba/smb.conf 設定ファイルを作成します。以下の行を設定ファイルに追加します。

```
[global]
realm = EXAMPLE.COM
security = ADS
encrypt passwords = yes
# Optional. Use only if Samba cannot determine the Kerberos server automatically.
password server = kerberos.example.com
```

上記の設定では、Samba はローカルで実行するサービスのユーザーを認証しますが、Active Directory のクライアントでもあります。kerberos の realm パラメーターがすべての上限（例： realm = EXAMPLE.COM）に表示されていることを確認します。Windows 2000/2003/2008 では Active Directory 認証に Kerberos が必要であるため、realm ディレクティブが必要です。Active Directory と Kerberos が異なるサーバーで実行している場合は、区別に役立つ password server ディレクティブが必要です。

2. メンバーサーバーで Kerberos を設定します。以下の内容で /etc/krb5.conf 設定ファイルを作成します。

```
[logging]
 default = FILE:/var/log/krb5libs.log

[libdefaults]
 default_realm = AD.EXAMPLE.COM
```

```
dns_lookup_realm = true
dns_lookup_kdc = true
ticket_lifetime = 24h
renew_lifetime = 7d
rdns = false
forwardable = false

[realms]
# Define only if DNS lookups are not working
# AD.EXAMPLE.COM = {
#  kdc = server.ad.example.com
#  admin_server = server.ad.example.com
#  master_kdc = server.ad.example.com
# }

[domain_realm]
# Define only if DNS lookups are not working
# .ad.example.com = AD.EXAMPLE.COM
# ad.example.com = AD.EXAMPLE.COM
```

DNS ルックアップが機能しない場合は、[realms] セクションおよび [domain_realm] セクションのコメントを解除します。

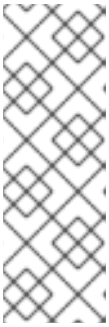
Kerberos および /etc/krb5.conf ファイルの詳細は、Red Hat Enterprise Linux 6nbsp;Hat Enterprise Red Hat Enterprise Linux 6nbsp;Linux Red Hat Enterprise Linux 6nbsp;6 『Managing Single Sign-On および Smart Cards』 の 『Kerberos の使用』 セクションを参照してください。

3.

Active Directory サーバーに参加するには、メンバーサーバーで root として以下のコマンドを入力します。

```
~]# net ads join -U administrator%password
```

net コマンドは、NT LAN Manager(NTLM)プロトコルを使用して 管理者 として認証し、マシンアカウントを作成します。次に、net はマシンアカウントの認証情報を使用して Kerberos で認証します。



セキュリティーオプション

security = ads ではなく security = user が使用されるため、smbpasswd などのローカルパスワードバックエンドは必要ありません。security = ads をサポートしない古いクライアントは、security = domain が設定されているかのように認証されます。この変更は機能に影響を与えず、ドメインに提供していないローカルユーザーを許可します。

Windows NT4 ベースドメインメンバー Server

以下の /etc/samba/smb.conf ファイルは、Windows NT4- ベースのドメインメンバーサーバーの実装に必要な設定例を示しています。NT4 ベースのドメインのメンバーサーバーになるのは、Active Directory への接続に似ています。主な違いは、NT4 ベースのドメインは認証方法で Kerberos を使用しないため、/etc/samba/smb.conf ファイルを簡素化します。この場合、Samba メンバーサーバーは NT4 ベースのドメインサーバーへパススルーします。

例21.6 Samba Windows NT4 ベースのドメインメンバーサーバーの設定例

```
[global]
workgroup = DOCS
netbios name = DOCS_SRV
security = domain

[homes]
comment = Home Directories
valid users = %S
read only = no
browseable = no

[public]
comment = Data
path = /export
force user = docsbot
force group = users
guest ok = yes
```

Samba をドメインメンバーサーバーとして設定することは、多くの状況で役に立ちます。Samba サーバーが、ファイルやプリンター共有以外の使用が可能であった場合もあります。ドメイン環境で使用するために Linux のみのアプリケーションが必要なインスタンスで Samba をドメインメンバーサーバーにすると便利です。管理者は、Windows ベースでなくても、ドメイン内のすべてのマシンを追跡します。Windows ベースのサーバーハードウェアが非推奨になると、/etc/samba/smb.conf ファイルを変更してサーバーを Samba ベースの PDC に変換する非常に簡単です。Windows NT ベースのサーバーが Windows 2000/2003/2008 にアップグレードされた場合、/etc/samba/smb.conf ファイルは、必要に応じてインフラストラクチャーの変更を Active Directory に組み込むのを簡単に変更できます。

SAMBA を起動する前にドメインに参加するようにしてください。

/etc/samba/smb.conf ファイルを設定したら、root で以下のコマンドを入力して Samba を起動する前 にドメインに参加します。

```
~]# net rpc join -U administrator%password
```

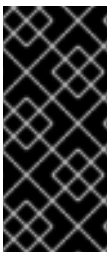
ドメインサーバーのホスト名を指定する -S オプションは、net rpc join コマンドに指定する必要はありません。Samba は、明示的に指定する代わりに、/etc/samba/smb.conf ファイルの workgroup

ディレクティブで指定されたホスト名を使用します。

21.1.6.3. ドメインコントローラー

Windows NT のドメインコントローラーは、Linux 環境の Network Information Service(NIS)サーバーと機能的に類似しています。ドメインコントローラーと NIS サーバーには、ユーザーおよびグループの情報データベースと関連サービスの両方をホストします。ドメインコントローラーは、主にドメインリソースにアクセスするユーザーの認証を含む、セキュリティーに使用されます。ユーザーおよびグループデータベースの整合性を維持するサービスは、Security Account Manager (SAM)と呼ばれます。SAM データベースは Windows と Linux の Samba ベースのシステムによって異なる方法で保存されるため、SAM レプリケーションは実現できず、PDC/BDC 環境ではプラットフォームを組み合わせることはできません。

Samba 環境では、PDC とゼロ以上の BDC のみを使用できます。



混合 SAMBA/WINDOWS ドメインコントローラー環境

Samba は、複数の Samba/Windows ドメインコントローラー環境に置くことはできません（Samba は Windows PDC の BDC にすることはできず、その逆も同様です）。または、Samba PDC と BDC を共存させることもでき ます。

tdbsamを使用したプライマリードメインコントローラー(PDC)

Samba PDC の最も簡単な実装および最も一般的な実装は、新しいデフォルトの tdbsam パスワードデータベースバックエンドを使用します。古い smbpasswd バックエンドを置き換えると、tdbsam には、「Samba アカウント情報データベース」 で説明されている多くの改良点があります。passdb backend ディレクティブは、PDC に使用するバックエンドを制御します。

以下の /etc/samba/smb.conf ファイルは、tdbsam パスワードデータベースバックエンドの実装に必要な設定例を示しています。

例21.7 tdbsamを使用した プライマリードメインコントローラー(PDC)の設定例

```
[global]
workgroup = DOCS
netbios name = DOCS_SRV
passdb backend = tdbsam
security = user
add user script = /usr/sbin/useradd -m "%u"
delete user script = /usr/sbin/userdel -r "%u"
add group script = /usr/sbin/groupadd "%g"
delete group script = /usr/sbin/groupdel "%g"
add user to group script = /usr/sbin/usermod -G "%g" "%u"
add machine script = /usr/sbin/useradd -s /bin/false -d /dev/null  -g machines "%u"
# The following specifies the default logon script
```

```
# Per user logon scripts can be specified in the user
# account using pdbedit logon script = logon.bat
# This sets the default profile path.
# Set per user paths with pdbedit
logon drive = H:
domain logons = yes
os level = 35
preferred master = yes
domain master = yes

[homes]
 comment = Home Directories
 valid users = %S
 read only = no

[netlogon]
 comment = Network Logon Service
 path = /var/lib/samba/netlogon/scripts
 browseable = no
 read only = no
# For profiles to work, create a user directory under the
# path shown.
# mkdir -p /var/lib/samba/profiles/john

[Profiles]
 comment = Roaming Profile Share
 path = /var/lib/samba/profiles
 read only = no
 browseable = no
 guest ok = yes
 profile acls = yes
# Other resource shares ... ...
```

tdbsam を使用する機能 PDC システムを提供するには、以下の手順に従います。

1. 例21.7「tdbsamを使用した プライマリードメインコントローラー(PDC)の設定例」 に従って、smb.conf 設定ファイルを調整します。

2. root ユーザーを Samba パスワードデータベースに追加します。root ユーザーの新しい Samba パスワードを提供するように求められます。

```
~]# smbpasswd -a root
New SMB password:
```

3. smb サービスを起動します。

```
~]# service smb start
```

4. すべてのプロファイル、ユーザー、および **netlogon** ディレクトリーが作成されていることを確認します。

5. ユーザーをメンバーにできるグループを追加します。

```
~]# groupadd -f users
~]# groupadd -f nobody
~]# groupadd -f ntadmins
```

6. **UNIX** グループをそれぞれの **Windows** グループに関連付けます。

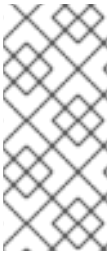
```
~]# net groupmap add ntgroup="Domain Users" unixgroup=users
~]# net groupmap add ntgroup="Domain Guests" unixgroup=nobody
~]# net groupmap add ntgroup="Domain Admins" unixgroup=ntadmins
```

7. ユーザーまたはグループへのアクセス権限を付与します。たとえば、**Samba** ドメインコントローラーのドメインにクライアントマシンを追加するパーミッションをドメイン管理グループのメンバーに付与するには、以下のコマンドを実行します。

```
~]# net rpc rights grant 'DOCS\Domain Admins' SetMachineAccountPrivilege -S PDC -U root
```

Windows システムは、**Domain Users** などのドメイングループにマッピングされたプライマリーグループを使用することが推奨されます。

Windows グループとユーザーは同じ名前空間を使用するため、**UNIX** のような同じ名前のグループとユーザーが存在することはありません。



TDBSAM 認証バックエンドの制限

複数のドメインコントローラーが必要な場合や、**250** ユーザーが複数ある場合は、**tdbsam** 認証バックエンドを使用し **ない** でください。このような場合には、**LDAP** が推奨されます。

Active Directory を使用するプライマリードメインコントローラー**(PDC)**

Samba を **Active Directory** のメンバーであることは可能ですが、**Samba** が **Active Directory** ドメインコントローラーとして動作することはできません。

21.1.7. Samba セキュリティーモード

Samba には、共有レベル と ユーザー レベルの 2 つのタイプのみがあります。これらは、セキュリティーレベルと全く知られています。共有レベルのセキュリティーは非推奨となり、Red Hat は代わりにユーザーレベルのセキュリティーを使用することを推奨します。ユーザーレベルのセキュリティーは、3 つの異なる方法で実装できます。セキュリティーレベルを実装するさまざまな方法は、セキュリティーモードと呼ばれ ます。

21.1.7.1. ユーザーレベルのセキュリティー

ユーザーレベルのセキュリティーはデフォルトであり、Samba に推奨される設定です。security = user ディレクティブが /etc/samba/smb.conf ファイルに一覧表示されていない場合でも、Samba によって使用されます。サーバーがクライアントのユーザー名とパスワードを受け入れると、クライアントはインスタンスごとにパスワードを指定せずに複数の共有をマウントできます。Samba はセッションベースのユーザー名およびパスワード要求を受け入れることもできます。クライアントは、ログオンごとに一意の UID を使用して複数の認証コンテキストを維持します。

/etc/samba/smb.conf ファイルでは、ユーザーレベルのセキュリティーを設定する security = user ディレクティブは次のとおりです。

```
[GLOBAL]
...
security = user
...
```

Samba ゲスト共有

上記のように、共有レベルのセキュリティーモードは非推奨となり、使用しないことを強く推奨します。security = share パラメーターを使用せずに Samba ゲスト共有を設定するには、以下の手順に従います。

手順21.4 Samba ゲスト共有の設定

1. この例では、/etc/samba/smbusers にユーザー名マップファイルを作成し、以下の行を追加します。

   ```
   nobody = guest
   ```

2. /etc/samba/smb.conf ファイルの main セクションに、以下のディレクティブを追加します。また、valid users ディレクティブも使用しないでください。

```
[GLOBAL]
...
security = user
map to guest = Bad User
username map = /etc/samba/smbusers
...
```

username map ディレクティブは、前のステップで指定したユーザー名マップファイルへのパスを提供します。

3. 以下のディレクティブを、/ect/samba/smb.conf ファイルの share セクションに追加します。valid users ディレクティブは使用しないでください。

```
[SHARE]
...
guest ok = yes
...
```

以下のセクションでは、ユーザーレベルのセキュリティーのその他の実装について説明します。

ドメインセキュリティーモード（ユーザーレベルのセキュリティー）

ドメインセキュリティーモードでは、Samba サーバーにマシンアカウント（ドメインセキュリティー信頼アカウント）があり、すべての認証要求がドメインコントローラーに渡されます。/etc/samba/smb.conf ファイルの以下のディレクティブを使用して、Samba サーバーがドメインメンバーサーバーに確立されます。

```
[GLOBAL]
...
security = domain
workgroup = MARKETING
...
```

Active Directory セキュリティーモード（ユーザーレベルのセキュリティー）

Active Directory 環境をお持ちの場合は、ドメインをネイティブの Active Directory メンバーとして参加させることができます。セキュリティーポリシーが NT 互換認証プロトコルの使用を制限する場合でも、Samba サーバーは Kerberos を使用して ADS に参加できます。Active Directory メンバーモードの Samba は、Kerberos チケットを許可できます。

/etc/samba/smb.conf ファイルで、以下のディレクティブで Samba に Active Directory メンバーサーバーが作成されます。

```
[GLOBAL]
...
```

```
security = ADS
realm = EXAMPLE.COM
password server = kerberos.example.com
...
```

21.1.7.2. 共有レベルのセキュリティー

共有レベルのセキュリティーでは、サーバーはクライアントからの明示的なユーザー名のないパスワードのみを受け入れます。サーバーは、ユーザー名とは関係なく、共有ごとにパスワードが必要です。Microsoft Windows クライアントが共有レベルのセキュリティーサーバーとの互換性の問題があることを示す最近のレポートがあります。このモードは非推奨になっており、Red Hat は共有レベルのセキュリティーの使用を強くお勧めします。手順21.4「Samba ゲスト共有の設定」 ディレクティブを使用する代わりに、security = share に記載の手順に従います。

21.1.8. Samba アカウント情報データベース

以下は、Samba で使用できる異なるバックエンドの一覧です。ここに記載の他のバックエンドも利用できる場合があります。

プレインテキスト

プレーンテキストのバックエンドは、/etc/passwd タイプのバックエンドにはありません。プレーンテキストのバックエンドでは、クライアントとサーバーはすべて、クライアントとサーバー間で暗号化されずに送信されます。この方法は安全ではないため、どのような手段でも使用することは推奨されません。プレーンテキストのパスワードで Samba サーバーに接続する別の Windows クライアントは、このような認証方法に対応できません。

smbpasswd

smbpasswd バックエンドは、MS Windows Lan アカウントおよび NT アカウント、暗号化されたパスワード情報が含まれるプレーンテキストの ASCII テキストレイアウトを使用します。smbpasswd バックエンドには、Windows NT/2000/2003 SAM 拡張コントロールのストレージがありません。smbpasswd バックエンドは、NT ベースのグループの RID などの Windows 情報をスケーリングしたり、保持したりしないため、推奨されません。tdbsam バックエンドはこれらの問題を解決し、小規模なデータベース（250 ユーザー）で使用する問題を解決しますが、エンタープライズレベルのソリューションではありません。

ldapsam_compat

ldapsam_compat バックエンドを使用すると、Samba のアップグレードバージョンと引き続き OpenLDAP サポートを使用できます。

tdbsam

デフォルトの tdbsam パスワードバックエンドは、ローカルサーバーのデータベースバックエンド、組み込みデータベースレプリケーションを必要としないサーバー、および LDAP のスケーラビリティーや複雑さを必要としないサーバーを提供します。tdbsam バックエンドには、smbpasswd データベース情報と、以前に除外された SAM 情報が含まれます。拡張 SAM データを含めることで、Samba は Windows NT/2000/2003/2008 ベースのシステムと同じアカウントおよびシステムアクセス制御を実装できます。

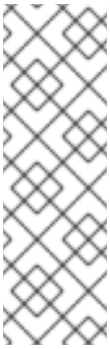
最大 250 ユーザーには、tdbsam バックエンドが推奨されます。大規模な組織では、スケーラビリティーやネットワークインフラストラクチャーの潜在的な問題により、Active Directory または LDAP 統合が必要になります。

ldapsam

ldapsam バックエンドは、Samba に最適な分散アカウントのインストール方法を提供します。LDAP は、Red Hat Directory Server や OpenLDAP Server などの任意の数のサーバーにデータベースを複製する機能が原因で最適です。LDAP データベースは軽量でスケーラブルなため、大規模な企業から推奨されます。ディレクトリーサーバーのインストールと設定については、本章の対象外となります。Red Hat Directory Server の詳細は、『Red Hat Directory Server 『9.0 デプロイメントガイド』』 を参照してください。LDAP の詳細は、「OpenLDAP」 を参照してください。

以前のバージョンの Samba から 3.0 にアップグレードする場合は、OpenLDAP スキーマファイル（/usr/share/doc/samba-バージョン/LDAP/samba.schema）および Red Hat Directory Server スキーマファイル(/usr/share/doc/samba-version/LDAP/samba-schema-FDS.ldif)が変更されたことに注意してください。これらのファイルには、ldapsam バックエンドが適切に機能するために必要な 属性構文の 定義 と objectclass 定義 が含まれます。

そのため、Samba サーバーに ldapsam バックエンドを使用している場合は、このスキーマファイルのいずれかを含めるように slapd を設定する必要があります。これを行う方法については、「スキーマの拡張」 を参照してください。



OPENLDAP-SERVERS パッケージがインストールされていることを確認します。

ldapsam バックエンドを使用する場合は、openldap-servers パッケージがインストールされている必要があります。パッケージがインストールされていることを確認するには、root で以下の コマンドを実行します。

```
~]# yum install openldap-servers
```

21.1.9. Samba Network Browsing

ネットワーク参照に より、Windows サーバーおよび Samba サーバーが Windows Network Neighborhood に表示されます。Network Neighborhood 内でアイコンはサーバーとして表され、開いている場合は利用可能なサーバーの共有とプリンターが表示されます。

ネットワーク参照機能には、TCP/IP を介した NetBIOS が必要です。NetBIOS ベースのネットワークはブロードキャスト(UDP)メッセージングを使用して、参照リスト管理を行います。TCP/IP ホスト名の解決の主な方法として NetBIOS および WINS を使用しないと、静的ファイル(/etc/hosts)や DNS などの他のメソッドを使用する必要があります。

ドメインマスターブラウザーは、ワークグループとサブネット間で参照ができるように、すべてのサブネットのローカルのマスターブラウザーから参照一覧を照合します。また、ドメインマスターブラウザーは、独自のサブネットのローカルマスターブラウザーを使用することが推奨されます。

21.1.9.1. ドメインの参照

デフォルトでは、ドメインの Windows サーバーの PDC は、そのドメインのドメインマスターブラウザーでもあります。このような状況では、Samba サーバーをドメインマスターサーバーとして設定することはできません。

Windows サーバーの PDC を含まないサブネットの場合、Samba サーバーをローカルのマスターブラウザーとして実装できます。ドメインコントローラー環境内のローカルのマスターブラウザー（または参照なし）に /etc/samba/smb.conf ファイルを設定することは、ワークグループの設定と同じです（「Samba サーバーの設定」を参照）。

21.1.9.2. WINS（Windows インターネットネームサーバー）

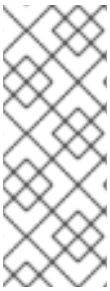
Samba サーバーまたは Windows NT サーバーのいずれかが WINS サーバーとして機能します。WINS サーバーを NetBIOS を有効にして使用する場合は、UDP ユニキャストをルーティングでき、ネットワーク全体で名前解決が可能になります。WINS サーバーを使用しないと、UDP ブロードキャストはローカルサブネットに制限されるため、他のサブネット、ワークグループ、またはドメインにはルーティングできません。WINS レプリケーションが必要な場合は、Samba は現在 WINS レプリケーションをサポートしないため、Samba をプライマリー WINS サーバーとして使用しないでください。

NT/2000/2003/2008 サーバーと Samba 環境では、Microsoft WINS 機能を使用することが推奨されます。Samba のみの環境では、WINS には Samba サーバーを 1 つだけ 使用することが推奨されます。

以下は、Samba サーバーが WINS サーバーとして機能する /etc/samba/smb.conf ファイルの例です。

例21.8 WINS サーバーの設定例

```
[global]
wins support = yes
```



WINS の使用

すべてのサーバー（Samba を含む）は、NetBIOS 名を解決するために WINS サーバーに接続する必要があります。WINS がないと、参照はローカルサブネットでのみ実行されます。さらに、ドメイン全体のリストが取得された方法であっても、WINS なしでクライアントに対してホストを解決できません。

21.1.10. Samba と CUPS 印刷サポート

Samba により、クライアントマシンは Samba サーバーに接続されているプリンターを共有できます。また、Samba により、クライアントマシンは Linux に組み込まれたドキュメントを Windows プリンター共有に送信できます。Red Hat Enterprise Linuxnbsp;Hat Enterprise Red Hat Enterprise Linuxnbsp;Linux で動作する他の印刷システムもありますが、Samba と密接に統合するため、CUPS(Common UNIX Print System)が推奨される印刷システムです。

21.1.10.1. 簡易 smb.conf の設定

以下の例は、CUPS サポートの基本的な /etc/samba/smb.conf 設定を示しています。

例21.9 CUPS サポートを使用した Samba の設定例

```
[global]
load printers = yes
printing = cups
printcap name = cups
[printers]
comment = All Printers
path = /var/spool/samba
browseable = no
guest ok = yes
writable = no
printable = yes
printer admin = @ntadmins
[print$]
comment = Printer Drivers Share
path = /var/lib/samba/drivers
write list = ed, john
printer admin = ed, john
```

その他の印刷設定も可能です。機密ドキュメントを出力するセキュリティーおよびプライバシーを追加して、ユーザーは公開パスに独自の印刷スプールを配置できます。ジョブが失敗すると、他のユーザーはファイルにアクセスできません。

print$ ディレクティブには、ローカルで利用できない場合にクライアントがアクセスするプリンタードライバーが含まれます。print$ ディレクティブは任意で、組織によっては必要でないことがあります。

Samba サーバーがドメインまたはワークグループに適切に設定されている場合、browseable を yes に設定すると、Windows Network Neighborhood でプリンターを表示できます。

21.1.11. Samba ディストリビューションプログラム

findsmb

```
findsmb <subnet_broadcast_address>
```

findsmb プログラムは、特定のサブネットの SMB対応システムに関する情報を報告する Perl スクリプトです。サブネットが指定されていない場合には、ローカルサブネットが使用されます。表示される項目には、IP アドレス、NetBIOS 名、ワークグループまたはドメイン名、オペレーティングシステム、およびバージョンが含まれます。findsmb コマンドは、以下の形式で使用されます。

以下の例は、システムで有効なユーザーとして findsmb を実行する出力を示しています。

```
~]$ findsmb
IP ADDR       NETBIOS NAME  WORKGROUP/OS/VERSION
------------------------------------------------------------------
10.1.59.25    VERVE         [MYGROUP] [Unix] [Samba 3.0.0-15]
10.1.59.26    STATION22     [MYGROUP] [Unix] [Samba 3.0.2-7.FC1]
10.1.56.45    TREK         +[WORKGROUP] [Windows 5.0] [Windows 2000 LAN Manager]
10.1.57.94    PIXEL         [MYGROUP] [Unix] [Samba 3.0.0-15]
10.1.57.137   MOBILE001     [WORKGROUP] [Windows 5.0] [Windows 2000 LAN Manager]
10.1.57.141   JAWS         +[KWIKIMART] [Unix] [Samba 2.2.7a-security-rollup-fix]
10.1.56.159   FRED         +[MYGROUP] [Unix] [Samba 3.0.0-14.3E]
10.1.59.192   LEGION        *[MYGROUP] [Unix] [Samba 2.2.7-security-rollup-fix]
10.1.56.205   NANCYN        +[MYGROUP] [Unix] [Samba 2.2.7a-security-rollup-fix]
```

net

```
net <protocol> <function> <misc_options> <target_options>
```

net ユーティリティーは、Windows および MS-DOS に使用される net ユーティリティーと似ています。最初の引数は、コマンドの実行時に使用するプロトコルを指定するために使用されます。protocol オプションは、サーバー接続のタイプを指定するために ads、rap、または rpc です。

Active Directory は ads を使用し、Win9x/NT3 は rap を使用し、Windows NT4/2000/2003/2008 は rpc を使用します。プロトコルを省略すると、net は自動的に判断を試行します。

以下の例は、wakko という名前のホストで利用可能な共有の一覧を表示します。

```
~]$ net -l share -S wakko
Password:
Enumerating shared resources (exports) on remote server:
Share name   Type     Description
----------   ----     -----------
data         Disk     Wakko data share
tmp          Disk     Wakko tmp share
IPC$         IPC      IPC Service (Samba Server)
ADMIN$       IPC      IPC Service (Samba Server)
```

以下の例は、wakko という名前のホストの Samba ユーザーの一覧を表示します。

```
~]$ net -l user -S wakko
root password:
User name            Comment
-----------------------------
andriusb             Documentation
joe                  Marketing
lisa                 Sales
```

nmblookup

```
nmblookup <options> <netbios_name>
```

nmblookup プログラムは、NetBIOS 名を IP アドレスに解決します。プログラムは、ターゲットマシンが応答するまでローカルサブネットのクエリーをブロードキャストします。

以下の例では、NetBIOS 名 trek の IP アドレスを表示します。

```
~]$ nmblookup trek
querying trek on 10.1.59.255
10.1.56.45 trek<00>
```

pdbedit

```
pdbedit <options>
```

pdbedit プログラムは、SAM データベースにあるアカウントを管理します。すべてのバックエンドには、smbpasswd、LDAP、および tdb データベースライブラリーが含まれます。

以下は、ユーザーの追加、削除、および一覧表示の例です。

```
~]$ pdbedit -a kristin
new password:
retype new password:
Unix username:        kristin
NT username:
Account Flags:        [U          ]
User SID:             S-1-5-21-1210235352-3804200048-1474496110-2012
Primary Group SID:    S-1-5-21-1210235352-3804200048-1474496110-2077
Full Name: Home Directory:        \\wakko\kristin
HomeDir Drive:
Logon Script:
Profile Path:         \\wakko\kristin\profile
Domain:               WAKKO
Account desc:
Workstations: Munged
dial:
Logon time:           0
Logoff time:          Mon, 18 Jan 2038 22:14:07 GMT
Kickoff time:         Mon, 18 Jan 2038 22:14:07 GMT
Password last set:    Thu, 29 Jan 2004 08:29:28
GMT Password can change:  Thu, 29 Jan 2004 08:29:28 GMT
Password must change: Mon, 18 Jan 2038 22:14:07 GMT
~]$ pdbedit -v -L kristin
Unix username:        kristin
NT username:
Account Flags:        [U          ]
User SID:             S-1-5-21-1210235352-3804200048-1474496110-2012
Primary Group SID:    S-1-5-21-1210235352-3804200048-1474496110-2077
Full Name:
Home Directory:       \\wakko\kristin
HomeDir Drive:
Logon Script:
Profile Path:         \\wakko\kristin\profile
Domain:               WAKKO
Account desc:
Workstations: Munged
dial:
Logon time:           0
Logoff time:          Mon, 18 Jan 2038 22:14:07 GMT
Kickoff time:         Mon, 18 Jan 2038 22:14:07 GMT
Password last set:    Thu, 29 Jan 2004 08:29:28 GMT
Password can change:  Thu, 29 Jan 2004 08:29:28 GMT
Password must change: Mon, 18 Jan 2038 22:14:07 GMT
~]$ pdbedit -L
andriusb:505:
joe:503:
lisa:504:
kristin:506:
```

```
~]$ pdbedit -x joe
~]$ pdbedit -L
andriusb:505: lisa:504: kristin:506:
```

rpcclient

```
rpcclient <server> <options>
```

rpcclient プログラムは、システム管理用の Windows 管理グラフィカルユーザーインターフェース (GUI)へのアクセスを提供する Microsoft RPC を使用して管理コマンドを実行します。これは、Microsoft RPC の完全な複雑さを理解する上級ユーザーによりよく使用されます。

smbcacls

```
smbcacls <//server/share> <filename> <options>
```

smbcacls プログラムは、Samba サーバーまたは Windows サーバーで共有されるファイルおよびディレクトリーの Windows ACL を変更します。

smbclient

```
smbclient <//server/share> <password> <options>
```

smbclient プログラムは、ftp ユーティリティーと同様の機能を提供する汎用 UNIX クライアントです。

smbcontrol

```
smbcontrol -i <options>
```

```
smbcontrol <options> <destination> <messagetype> <parameters>
```

smbcontrol プログラムは、smbd デーモン、nmbd デーモン、または winbindd デーモンの実行に制御メッセージを送信します。smbcontrol -i を実行すると、空の行または 'q' が入力されるまでコマンドを対話的に実行します。

smbpasswd

```
smbpasswd <options> <username> <password>
```

smbpasswd プログラムは、暗号化されたパスワードを管理します。このプログラムは、スーパーユーザーが実行して、ユーザーのパスワードを変更し、通常のユーザーが独自の Samba パスワードを変更することもできます。

smbspool

```
smbspool <job> <user> <title> <copies> <options> <filename>
```

smbspool プログラムは、Samba への CUPS 互換印刷インターフェースです。CUPS プリンターで使用するように設計されていますが、smbspool は CUPS 以外のプリンターでも機能します。

smbstatus

```
smbstatus <options>
```

smbstatus プログラムは、Samba サーバーへの現在の接続のステータスを表示します。

smbtar

```
smbtar <options>
```

smbtar プログラムは、Windows ベースの共有ファイルおよびディレクトリーのバックアップおよび復元をローカルテープアーカイブに対して実行します。tar ユーティリティーと同様に、これら 2 つのユーティリティーは互換性がありません。

testparm

```
testparm <options> <filename> <hostname IP_address>
```

testparm プログラムは、/etc/samba/smb.conf ファイルの構文をチェックします。smb.conf ファイルがデフォルトの場所(/etc/samba/smb.conf)にある場合は、場所を指定する必要はありません。testparm プログラムにホスト名および IP アドレスを指定すると、hosts.allow ファイルおよび host.deny ファイルが正しく設定されていることを確認します。testparm プログラムは、テスト後に smb.conf ファイルとサーバーのロール（stand-alone、domain など）の概要も表示します。これは、コメントを除外し、経験のある管理者が読み取る情報を簡潔に提示する時に、デバッグを行う場合に便利です。以下に例を示します。

```
~]$ testparm
Load smb config files from /etc/samba/smb.conf
Processing section "[homes]"
Processing section "[printers]"
Processing section "[tmp]"
Processing section "[html]"
Loaded services file OK.
Server role: ROLE_STANDALONE
Press enter to see a dump of your service definitions
<enter>
# Global parameters
[global]
 workgroup = MYGROUP
 server string = Samba Server
```

```
 security = SHARE
 log file = /var/log/samba/%m.log
 max log size = 50
 socket options = TCP_NODELAY SO_RCVBUF=8192 SO_SNDBUF=8192
 dns proxy = no
[homes]
 comment = Home Directories
 read only = no
 browseable = no
[printers]
 comment = All Printers
 path = /var/spool/samba
 printable = yes
 browseable = no
[tmp]
 comment = Wakko tmp
 path = /tmp
 guest only = yes
[html]
 comment = Wakko www
 path = /var/www/html
 force user = andriusb
 force group = users
 read only = no
 guest only = yes
```

wbinfo

```
wbinfo <options>
```

wbinfo プログラムは、winbindd デーモンからの情報を表示します。wbinfo を機能させるには、winbindd デーモンが実行している必要があります。

21.1.12. その他のリソース

以下のセクションでは、Samba をより詳細に調べる方法を説明します。

インストールされているドキュメント

- **/usr/share/doc/samba-<version-number& gt;/: Samba ディストリビューションに含まれる追加ファイルすべて。これには、ヘルパースクリプト、設定ファイルのサンプル、およびドキュメントが含まれます。**

- **特定の Samba 機能の詳細は、以下の man ページを参照してください。**

  - **smb.conf(5)**

- *samba(7)*
- *smbd(8)*
- *nmbd(8)*
- *winbindd(8)*

関連書籍

- 『Official Samba-3 HOWTO-Collection』 *by John H. Terpstra and Jelmer R. Vernooij; Prentice Hall - Samba 開発チームが発行した公式の Samba-3 ドキュメントこれは、ステップバイステップのガイドよりも多くのリファレンスガイドです。*
- 『Samba-3 by John H. Terpstra; Prentice Hall: これは、OpenLDAP、DNS、DHCP、および print 設定ファイルの詳細例』 *を説明する Samba 開発チームが発行する別の公式リリースです。これには、実際の実装に役立つステップバイステップの関連情報があります。*
- 『Samba、2nd Edition（Jan Ts による 2nd Edition、Robert Eckstein、および David Collier-Brown）の使用』。*O'Reilly - O'Revice to novice to novice to novice which includes comprehensive reference material.*

便利な Web サイト

- *http://www.samba.org/: Samba ディストリビューションのホームページと、Samba 開発チームが作成したすべての公式ドキュメント多くのリソースは HTML および PDF 形式で利用できますが、その他は購入でのみ利用できます。これらのリンクの多くは Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux 固有のものではなく、一部の概念が適用される場合があります。*
- *http://samba.org/samba/archives.html : Samba コミュニティーの有効なメール一覧。一覧アクティビティーのレベルが高くなるため、ダイジェストモードを有効にすることが推奨されます。*
- *Samba newsgroups - NNTP プロトコルを使用する Samba スレッドの newsgroups（www.gmane.org など）も利用可能です。これは、メーリングリストのメールを受信する代わりになります。*

21.2. FTP

FTP(File Transfer Protocol)は、現在インターネット上で見られる、最も古く、一般的に使用されているプロトコルです。この目的は、ユーザーがリモートホストに直接ログインしなくても、もしくはリモートシステムの使用法についての知識がなくとも、ネットワーク上のコンピューターホスト間で確実にファイルを転送することです。これにより、ユーザーは、標準の簡単なコマンドセットを使用してリモートシステム上のファイルにアクセスすることができます。

本セクションでは、FTP プロトコルの基本と、Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux に同梱されているプライマリー FTP サーバーである vsftpd について概説します。

21.2.1. ファイル転送プロトコル (FTP)

FTP は、クライアント/サーバーアーキテクチャーを使用し、TCP ネットワークプロトコルを使用してファイルを転送します。FTP は古いプロトコルであることから、暗号化されていないユーザー名とパスワード認証を使用します。このため、安全でないプロトコルとみなされており、絶対的に必要でない限り、使用するべきではありません。ただし、FTP はインターネット上で非常に有益であるため、共有ファイルの公開で必要となる場合がよくあります。このため、システム管理者は、FTP の固有の特性を認識する必要があります。

本セクションでは、vsftpd を設定して TLS によるセキュアな接続を確立する方法と、SELinux を用いて FTP サーバーのセキュリティーを保護する方法を説明します。FTP の代用となるのは、OpenSSH スイートからの sftp です。OpenSSH の設定方法および SSH プロトコル全般に関する情報は、14章OpenSSH を参照してください。

インターネット上で使用されているほとんどのプロトコルとは異なり、FTP が正しく機能するためには複数のネットワークポートを必要とします。FTP クライアントアプリケーションが FTP サーバーへの接続を開始すると、コマンド ポート と呼ばれるポート 21 をサーバー上で開きます。このポートは、すべてのコマンドをサーバーに発行するために使用されます。サーバーから要求されたデータは、データポート を介してクライアントに返されます。データ接続のポート番号、およびデータ接続を初期化する方法は、クライアントが active モードまたは passive モードでデータを要求するかどうかによって異なります。

これらのモードの定義は以下のとおりです。

アクティブモード

アクティブモードは、FTP プロトコルでクライアントへのデータ転送に使用される独自の方法です。FTP クライアントがアクティブモードのデータ転送を開始すると、サーバーは、サーバーのポート 20 から、クライアントで指定された IP アドレスとランダムな非特権ポート（ 1024以上）への接続を開きます。この配置は、クライアントマシンが 1024 を超えるポートを介して接続を受け入れることができる必要があることを意味します。インターネットのようなセキュリティー保護されていないネットワークが増加するにともない、ファイアウォールを使用したクライアントマシンの

保護が普及しています。このようなクライアント側のファイアウォールは、アクティブモードの FTP サーバーから着信する接続を拒否する場合が多いため、パッシブモードが実現されました。

パッシブモード

パッシブモードはアクティブモードと同様に、FTP クライアントアプリケーションによって開始されます。サーバーからデータを要求する際に、FTP クライアントはパッシブモードでデータにアクセスしたいことを示し、サーバーはサーバー上の IP アドレスとランダムな非特権ポート（1024以上）を提供します。クライアントは、サーバー上のそのポートに接続して要求した情報をダウンロードします。

パッシブモードは、クライアント側のファイアウォールによるデータ接続障害の問題を解決しますが、サーバー側のファイアウォール管理を複雑化させてしまう場合があります。FTP サーバー上の特権のないポートの範囲を制限することにより、サーバー上で開いているポートの数を減らすことができます。またこの方法により、サーバーを対象としたファイアウォールのルール設定の手順が簡略化されます。パッシブポートの制限の詳細は、「ネットワークオプション」 を参照してください。

21.2.2. vsftpd サーバー

vsftpd(Very Secure FTP Daemon)は、高速で安定性があり、安全性が重要なように、最大で重要なものから設計されています。vsftpd は、多数の接続を効率的かつ安全に処理できるため、Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux とともに配布される唯一のスタンドアロン FTP サーバーです。

vsftpd で使用されるセキュリティーモデルには、以下に挙げる 3 つの主要な側面があります。

- 特権プロセスと非特権プロセスの確固たる分離: 別個のプロセスが異なるタスクを処理します。各プロセスは、そのタスクに必要な最低限の権限で稼働します。
- 高い権限を必要とするタスクを、必要最小限の権限を伴うプロセスで処理: libcap ライブラリー内にある互換性を利用して、通常は完全な root 権限を必要とするタスクを、権限が低いプロセスでより安全に実行できます。
- ほとんどのプロセスを chroot jail で実行 する - 可能な場合は常に、プロセスはルートディレクトリーを共有ディレクトリーに変更します。このディレクトリーは、chroot jail と見なされます。たとえば、/var/ftp/ ディレクトリーが主要な共有ディレクトリーである場合、vsftpd は /var/ftp/ を新しい root ディレクトリー(/)に再割り当てします。これにより、新たな root

ディレクトリー下に格納されていないディレクトリーに対する、ハッカーの潜在的な悪質行為を行うことができなくなります。

これらのセキュリティープラクティスを使用すると、vsftpd によるリクエスト対応方法に以下のような影響があります。

- 親プロセスは、必要最小限の権限で稼働: 親プロセスは、リスクレベルを最低限に抑えるために必要とされる権限のレベルを動的に算出します。子プロセスは、FTP クライアントとの直接的な対話を処理し、できるだけ権限なしに近い状態で稼働します。

- 高い権限を必要とする操作はすべて、小さな親プロセスによって処理され ます。Apache HTTP Server と同様に、vsftpd は権限のない子プロセスを起動し、着信接続を処理します。これにより、権限のある親プロセスを最小限に抑えられ、比較的少ないタスクを処理することになります。

- 親プロセスは、権限のない子プロセスからのリクエストはどれも信頼しない: 子プロセスとの通信はソケット上で受信し、子プロセスからの情報の有効性は動作を実施する前に確認されます。

- FTP クライアントとのインタラクションの大半は、chroot jail 内の権限のない子プロセスによって処理： これらの子プロセスには権限がなく、共有ディレクトリーへのアクセスしかないため、プロセスがクラッシュした際 に攻撃者がアクセスできるのは共有ファイルのみです。

21.2.2.1. vsftpd の起動と停止

vsftpd RPM は、/etc/rc.d/init.d/vsftpd スクリプトをインストールします。これは、service コマンドを使用してアクセスできます。

サーバーを起動するには、root で以下を入力します。

```
~]# service vsftpd start
```

サーバーを以下のように停止するには、以下を実行します。

```
~]# service vsftpd stop
```

restart オプションは、vsftpd を停止して起動する簡単な方法です。これは、vsftpd の設定ファイルを編集した後に設定変更を行う最も効率的な方法です。

root で以下のコマンドを入力し、サーバーを再起動するには、以下を実行します。

```
~]# service vsftpd restart
```

condrestart（条件付き再起動）オプションは停止し、現在実行している場合にのみ vsftpd を起動します。このオプションは、デーモンが実行されていない場合はデーモンを起動しないため、スクリプトに便利です。try-restart オプションは同義語です。

root タイプでサーバーを条件的に再起動するには、以下を実行します。

```
~]# service vsftpd condrestart
```

デフォルトでは、vsftpd サービスはブート時に自動的に起動 されません。vsftpd サービスが起動時に開始するように設定するには、/sbin/chkconfig、/usr/sbin/ntsysv、または Services Configuration Tool プログラムなどの initscript ユーティリティーを使用します。これらのツールの詳細は、12章サービスおよびデーモン を参照してください。

21.2.2.2. vsftpd の複数コピーの起動

1 台のコンピューターを複数の FTP ドメインに使用する場合があります。これは、マルチホーミング と呼ばれる手法です。vsftpd を使用してマルチホーミングを行う方法の 1 つに、デーモンの複数コピーを実行し、各コピーに設定ファイルを与える方法があります。

これを行うには、最初に、関連するすべての IP アドレスをシステム上のネットワークデバイスまたはエイリアスネットワークデバイスに割り当てます。ネットワークデバイス、デバイスエイリアスの設定に関する詳細は、10章NetworkManager を参照してください。ネットワーク設定スクリプトの詳細は、11章Network Interfaces を参照してください。

次に、FTP ドメインの DNS サーバーが正しいマシンを参照するように設定する必要があります。BIND、Red Hat Enterprise Linuxnbsp;Hat Enterprise Red Hat Enterprise Linuxnbsp;Linux で使用されている DNS プロトコル実装、設定ファイルの詳細は、「BIND」 を参照してください。

vsftpd がさまざまな IP アドレスで要求に応答するには、デーモンの複数コピーが実行中である必要があります。これを可能にするには、FTP サーバーで必要な各インスタンスの個別の vsftpd 設定ファイルを作成し、それを /etc/vsftpd/ ディレクトリーに格納する必要があります。これらの設定ファイルは、（ /etc/vsftpd/vsftpd-site-2.confなどの）一意の名前を持ち、root ユーザーのみが読み取り、書き込み可能とする必要があることに注意してください。

IPv4 ネットワーク上でリッスンする各 FTP サーバーの設定ファイル内で、以下のディレクティブは一意のものである必要があります。

```
listen_address=N.N.N.N
```

N.N.N.N を、提供される FTP サイト用の一意 の IP アドレスに置き換えます。サイトが IPv6 を使用している場合は、代わりに listen_address6 ディレクティブを使用します。

複数の設定ファイルを /etc/vsftpd/ ディレクトリーに格納したら、vsftpd デーモンの設定済みインスタンスをすべて root として実行して起動できます。

```
~]# service vsftpd start
```

利用可能な他の サービス コマンドの説明は、「vsftpd の起動と停止」 を参照してください。

vsftpd デーモンの個別インスタンスは、以下のコマンドを使用して root シェルプロンプトから起動できます。

```
~]# vsftpd /etc/vsftpd/configuration-file
```

上記のコマンドで、configuration-file を、vsftpd-site-2.conf などの要求されたサーバーの設定ファイルの一意の名前に置き換えます。

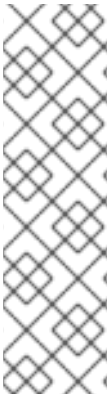
サーバーごとに変更するディレクティブには、以下のものがあります。

- anon_root
- local_root
- vsftpd_log_file
- xferlog_file

vsftpd デーモンの設定ファイルで使用できるディレクティブの詳細な一覧は、「vsftpd でインストールされたファイル」 を参照してください。

21.2.2.3. TLS を使用した vsftpd 接続の暗号化

ユーザー名、パスワード、およびデータを暗号化せずに送信する FTP の本質的にセキュアでない性質に対照するために、vsftpd デーモンが TLS プロトコルを使用して接続を認証し、すべての送信を暗号化するように設定できます。TLS をサポートする FTP クライアントは、TLS が有効になっている vsftpd と通信する必要があることに注意してください。



注記

SSL (Secure Sockets Layer)は、セキュリティープロトコルの古い実装の名前です。新しいバージョンは TLS (Transport Layer Security)と呼ばれます。SSL にはセキュリティーに関する深刻な脆弱性があるため、新しいバージョン(TLS)のみを使用してください。vsftpd サーバーに付随するドキュメントや vsftpd.conf ファイルで使用される設定ディレクティブでは、セキュリティー関連の項目を参照する際に SSL 名を使用しますが、TLS はサポートされており、ssl_enable ディレクティブが YES に設定されているときにデフォルトで使用されています。

vsftpd.conf ファイルの ssl_enable 設定ディレクティブを YES に設定して、TLS サポートを有効にします。ssl_enable オプションが有効になっていると自動的にアクティブになる他の TLS関連のディレクティブのデフォルト設定により、合理的に適切に設定された TLS のセットアップが提供されます。これには、すべての接続に TLS v1 プロトコルのみを使用する要件（安全でない SSL プロトコルバージョンはデフォルトで無効になるなど）や、匿名以外のすべてのログインでパスワードおよびデータ送信での TLS の使用を強制することなどです。

例21.10 TLS を使用するように vsftpd の設定

以下の例では、設定ディレクティブは vsftpd.conf ファイルでセキュリティープロトコルの古い SSL バージョンを明示的に無効にします。

```
ssl_enable=YES
ssl_tlsv1=YES
ssl_sslv2=NO
ssl_sslv3=NO
```

設定を変更したら、vsftpd サービスを再起動します。

```
~]# service vsftpd restart
```

vsftpd による TLS の使用を微調整するための他の TLS関連の設定ディレクティブについては、vsftpd.conf(5) man ページを参照してください。また、一般的に使用されるその他の vsftpd.conf 設定ディレクティブの説明は、「vsftpd 設定オプション」 を参照してください。

21.2.2.4. vsftpd 用の SELinux ポリシー

（他の ftpd プロセスとともに） vsftpd デーモンを管理する SELinux ポリシーは、強制アクセス制御を定義します。これはデフォルトでは、最低限必要なアクセスに基づいています。FTP デーモンが特定のファイルまたはディレクトリーにアクセスできるようにするには、それらに適切なラベルを割り当てる必要があります。

たとえば、ファイルを匿名で共有できるようにするには、public_content_t ラベルを共有するファイルおよびディレクトリーに割り当てる必要があります。これは、chcon コマンドを root として使用して実行できます。

```
~]# chcon -R -t public_content_t /path/to/directory
```

上記のコマンドでは、/path/to/directory を、ラベルを割り当てるディレクトリーへのパスに置き換えます。同様に、ファイルのアップロード用にディレクトリーを設定する場合は、その特定のディレクトリーに public_content_rw_t ラベルを割り当てる必要があります。さらに、SELinux のブール値オプション allow_ftpd_anon_write を 1 に設定する必要があります。以下のように、setsebool コマンドを root で実行します。

```
~]# setsebool -P allow_ftpd_anon_write=1
```

ローカルユーザーが FTP 経由でホームディレクトリーにアクセスできるようにするには、Red Hat Enterprise Linux 6nbsp;Hat Enterprise Red Hat Enterprise Linux 6nbsp;LinuxRed Hat Enterprise Linux 6nbsp;6 のデフォルト設定である FTP でホームディレクトリーにアクセスできるようにするには、ftp_home_dir のブール値オプションを 1 に設定する必要があります。vsftpd をスタンドアロンモードで実行できるようにするには（Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;LinuxRed Hat Enterprise Linux 6nbsp;6 でデフォルトで有効）、ftpd_is_daemon オプションも 1 に設定する必要があります。

FTP に関する SELinux ポリシーの設定方法の他の便利なラベルやブール値オプションの例は、man ページの ftpd_selinux(8) を参照してください。SELinux 全般に関する詳細情報は、Red Hat Enterprise Linux 6 Security-Enhanced Linux をご覧ください。

21.2.2.5. vsftpd でインストールされたファイル

vsftpd RPM は、デーモン(vsftpd)、その設定と関連ファイル、および FTP ディレクトリーをシステムにインストールします。以下は、vsftpd 設定に関連するファイルおよびディレクトリーの一覧で

す。

- /etc/pam.d/ vsftpd : vsftpd のPAM（プラグ可能な認証モジュール ）設定ファイルこのファイルは、FTP サーバーへのログインに必要な要件を指定します。PAM の詳細は、『Red Hat Enterprise Linux 6 Single Sign-On およびスマートカード』の「『PAM（プラグ可能な認証モジュール）の使用』」の章を参照してください。

- /etc/vsftpd/vsftpd.conf - vsftpd の設定ファイルこのファイルに含まれる重要なオプションのリストは、「vsftpd 設定オプション」 を参照してください。

- /etc/vsftpd/ftpusers: vsftpd へのログインが許可されていないユーザーの一覧です。デフォルトでは、この一覧には root、bin、および daemon ユーザーが含まれます。

- /etc/vsftpd/user_list: このファイルは、userlist_deny ディレクティブが YES （デフォルト）に設定されているか、または /etc/vsftpd/vsftpd.conf で NO に設定されているかに応じて、リストされているユーザーへのアクセスを拒否するか、許可するよう設定できます。/etc/vsftpd/user_list を使用してユーザーへのアクセス権限を付与する場合は、一覧にあるユーザー名が /etc/vsftpd/ftpusers に表示され ない 必要があります。

- /var/ftp/: vsftpd が提供するファイルが含まれるディレクトリー。また、匿名ユーザー用の /var/ftp/pub/ ディレクトリーも含まれています。どちらのディレクトリーも誰でも読み取り可能ですが、root ユーザーのみが書き込み可能です。

21.2.2.6. vsftpd 設定オプション

vsftpd は他の広く利用可能な FTP サーバーのカスタマイズレベルを提供しないかもしれませんが、ほとんどの管理者のニーズを満たすのに十分なオプションが提供されています。これは、機能的な制限設定およびプログラムによるエラーに制限されないということになります。

vsftpd のすべての設定は、設定ファイル /etc/vsftpd/vsftpd.conf により処理されます。各ディレクティブは、ファイル内の各行にあり、以下の形式に従います。

```
directive=value
```

各ディレクティブについて、directive を有効なディレクティブに置き換え、value を有効な値に置き換えます。

スペースは使用しないでください。

ディレクティブ、等号記号、およびディレクティブの 値 の間には空白を入れないでください。

コメント行はハッシュ記号(#)を付け、デーモンにより無視されます。

利用可能なディレクティブの全一覧は、vsftpd.conf の man ページを参照してください。vsftpd のセキュリティー保護に関する概要は『Red Hat Enterprise Linux 6 セキュリティーガイド』を参照してください。

以下は、/etc/vsftpd/vsftpd.conf 内の重要なディレクティブの一覧です。vsftpd の設定ファイル内で明示的に検出またはコメントアウトされていないすべてのディレクティブは、デフォルト値に設定されます。

21.2.2.6.1. デーモンオプション

以下は、vsftpd デーモンの全体的な動作を制御するディレクティブの一覧です。

- listen: 有効にすると、vsftpd はスタンドアロンモードで実行されます。つまり、デーモンは xinetd super-server ではなく独立して起動します。Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;LinuxRed Hat Enterprise Linux 6nbsp;6 は、この値を YES に設定します。vsftpd をスタンドアロンモードで実行できるようにするには、SELinux ftpd_is_daemon のブール値オプションを設定する必要があります。vsftpd のデフォルトの SELinux ポリシーとの対話に関する詳細は、「vsftpd 用の SELinux ポリシー」および ftpd_selinux(8) を参照してください。このディレクティブは、listen_ipv6 ディレクティブと併用できません。

  デフォルト値は NO です。Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;LinuxRed Hat Enterprise Linux 6nbsp;6 では、このオプションは設定ファイルで YES に設定されます。

- listen_ipv6: 有効にすると、vsftpd はスタンドアロンモードで実行されます。つまり、デーモンは xinetd super-server ではなく独立して起動します。このディレクティブを使用すると、IPv6 ソケットのみをリッスンします。vsftpd をスタンドアロンモードで実行できるようにするには、SELinux ftpd_is_daemon のブール値オプションを設定する必要があります。vsftpd のデフォルトの SELinux ポリシーとの対話に関する詳細は、「vsftpd 用の SELinux ポリシー」および ftpd_selinux(8) を参照してください。このディレクティブは listen ディレクティブと併用できません。

デフォルト値は NO です。

- session_support: 有効にすると、vsftpd は プラグ可能な認証モジュール (PAM)を介して各ユーザーのログインセッションを維持しようとします。詳細は、Red Hat Enterprise Linux 6 Single Sign-On および Smart Cards の 『PAM（プラグ可能な認証モジュール）の使用』 と、PAM の man ページを参照してください。セッションロギングが必要ない場合は、このオプションを無効にすると、プロセスを減らし、権限が低い vsftpd を実行することができます。

  デフォルト値は NO です。

21.2.2.6.2. ログインオプションおよびアクセス制御

以下は、ログイン動作とアクセス制御メカニズムを制御するディレクティブの一覧です。

- anonymous_enable: 有効にすると、匿名ユーザーがログインできるようになります。ユーザー名 anonymous および ftp が許可されます。

  デフォルト値は YES です。

  匿名ユーザーに影響するディレクティブの一覧は、「Anonymous User Options」 を参照してください。

- banned_email_file - deny_email_enable ディレクティブが YES に設定されている場合、このディレクティブはサーバーへのアクセスが許可されない匿名メールパスワードの一覧を含むファイルを指定します。

  デフォルト値は /etc/vsftpd/banned_emails です。

- banner_file: サーバーへの接続が確立されたときに表示されるテキストを含むファイルを指定します。このオプションは、ftpd_banner ディレクティブで指定されたテキストを上書きします。

  このディレクティブにはデフォルト値がありません。

- *cmds_allowed: サーバーが許可する FTP コマンドのカンマ区切りリストを指定します。その他のコマンドはすべて拒否されます。*

  このディレクティブにはデフォルト値がありません。

- *deny_email_enable: 有効にすると、/etc/vsftpd/banned_emails で指定された電子メールパスワードを使用する匿名ユーザーはサーバーへのアクセスを拒否します。このディレクティブで参照されるファイルの名前は、bann ed_email_file ディレクティブ を使用して指定できます。*

  デフォルト値は NO です。

- *ftpd_banner - 有効にすると、このディレクティブ内で指定された文字列は、サーバーへの接続が確立されたときに表示されます。このオプションは banner_file ディレクティブで上書きできます。*

  デフォルトでは、vsftpd は標準バナーを表示します。

- *local_enable: 有効にすると、ローカルユーザーはシステムにログインできます。このディレクティブが想定どおりに機能するには、SELinux ftp_home_dir のブール値オプションを設定する必要があります。vsftpd のデフォルトの SELinux ポリシーとの対話に関する詳細は、「vsftpd 用の SELinux ポリシー」 および ftpd_selinux(8) を参照してください。*

  デフォルト値は NO です。Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;LinuxRed Hat Enterprise Linux 6nbsp;6 では、このオプションは設定ファイルで YES に設定されます。

  ローカルユーザーに影響するディレクティブの一覧については、「local-User オプション」 を参照してください。

- *pam_service_name: vsftpd の PAM サービス名を指定します。*

  デフォルト値は ftp です。Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;LinuxRed Hat Enterprise Linux 6nbsp;6 では、このオプションは設定ファイルで vsftpd に設定されます。

- *tcp_wrappers: 有効にすると、TCP ラッパーを使用してサーバーへのアクセスを付与します。FTP サーバーが複数の IP アドレスに設定されている場合は、VSFTPD_LOAD_CONF 環境変数を使用して、クライアントが要求した IP アドレスに基づいて異なる設定ファイルを読み込むことができます。*

  デフォルト値は NO です。Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;LinuxRed Hat Enterprise Linux 6nbsp;6 では、このオプションは設定ファイルで YES に設定されます。

- *userlist_deny: userlist_enable ディレクティブと併用して NO に設定すると、ユーザー名が userlist_file ディレクティブで指定されたファイルに表示されない限り、すべてのローカルユーザーは拒否されます。クライアントがパスワードを要求する前にアクセスが拒否されるため、このディレクティブを NO に設定すると、ローカルユーザーがネットワーク経由で暗号化されていないパスワードを送信するのを防ぎます。*

  デフォルト値は YES です。

- *userlist_enable: 有効にすると、userlist_file ディレクティブで指定されたファイルに記載されているユーザーはアクセスが拒否されます。クライアントがパスワードを要求する前にアクセスが拒否されるため、ユーザーはネットワーク経由で暗号化されていないパスワードを送信することはできません。*

  デフォルト値は NO です。Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;LinuxRed Hat Enterprise Linux 6nbsp;6 では、このオプションは設定ファイルで YES に設定されます。

- *userlist_file: userlist_enable ディレクティブが有効な場合に vsftpd が参照するファイルを指定します。*

  デフォルト値は /etc/vsftpd/user_list です。これはインストール時に作成されます。

21.2.2.6.3. Anonymous User Options

以下は、サーバーへの匿名ユーザーアクセスを制御するディレクティブの一覧です。このオプションを使用するには、anonymous_enable ディレクティブを YES に設定する必要があります。

- *anon_mkdir_write_enable - write_enable ディレクティブとともに有効にすると、匿名ユーザーは書き込みパーミッションを持つ親ディレクトリーに新しいディレクトリーを作成で*

きます。

デフォルト値は NO です。

- anon_root: 匿名ユーザーがログインした後に vsftpd の変更を指定します。

  このディレクティブにはデフォルト値がありません。

- anon_upload_enable - write_enable ディレクティブとともに有効にすると、匿名ユーザーは書き込みパーミッションを持つ親ディレクトリー内のファイルをアップロードできます。

  デフォルト値は NO です。

- anon_world_readable_only: 有効にすると、匿名ユーザーは誰でも読み取り可能なファイルのみをダウンロードできます。

  デフォルト値は YES です。

- ftp_username - 匿名の FTP ユーザーに使用されるローカルユーザーアカウント（ /etc/passwdにリストされている）を指定します。ユーザーの /etc/passwd で指定したホームディレクトリーは、匿名の FTP ユーザーのルートディレクトリーです。

  デフォルト値は ftp です。

- no_anon_password: 有効にすると、匿名ユーザーはパスワードを要求されません。

  デフォルト値は NO です。

- secure_email_list_enable: 有効にすると、匿名ログインに対して指定された電子メールパスワード一覧のみが許可されます。これは、仮想ユーザーを使用せずに、限定されたセキュリティーをパブリックコンテンツに提供する便利な方法です。

/etc/vsftpd/email_passwords に提供されたパスワードが一覧表示されていない限り、匿名ログインは回避されます。ファイル形式は、1 行に 1 つのパスワードで、最後の空白はありません。

デフォルト値は NO です。

21.2.2.6.4. local-User オプション

以下は、ローカルユーザーがサーバーにアクセスする方法を特徴とするディレクティブの一覧です。このオプションを使用するには、local_enable ディレクティブを YES に設定する必要があります。ユーザーがホームディレクトリーにアクセスできるようにするには、SELinux ftp_home_dir のブール値オプションを設定する必要があります。vsftpd のデフォルトの SELinux ポリシーとの対話に関する詳細は、「vsftpd 用の SELinux ポリシー」 および ftpd_selinux(8) を参照してください。

- chmod_enable - 有効にすると、FTP コマンドの SITE CHMOD がローカルユーザーに許可されます。このコマンドにより、ユーザーはファイルのパーミッションを変更できます。

  デフォルト値は YES です。

- chroot_list_enable: 有効にすると、ログイン時に chroot_list_file ディレクティブに指定されたファイルに記載されているローカルユーザーが chroot jail に配置されます。

  chroot_local_user ディレクティブとともに有効にすると、chroot_list_file ディレクティブで指定されたファイルに記載されているローカルユーザーは、ログイン時に chroot jail に配置され ません。

  デフォルト値は NO です。

- chroot_list_file: chroot_list_enable ディレクティブが YES に設定されている場合に、参照されるローカルユーザーの一覧を含むファイルを指定します。

  デフォルト値は /etc/vsftpd/chroot_list です。

- chroot_local_user: 有効にすると、ログイン後にローカルユーザーがホームディレクトリーに変更します。

デフォルト値は NO です。

CHROOT_LOCAL_USER オプションの有効化を回避

chroot_local_user を有効にすると、特にアップロード権限を持つユーザーにとって、多くのセキュリティー問題が開きます。このため、推奨され ません。

- guest_enable: 有効にすると、匿名以外のユーザーはすべて、 guest _username ディレクティブで指定したユーザー guest でログインします。

  デフォルト値は NO です。

- guest_username: ゲスト ユーザーがマップされるユーザー名を指定します。

  デフォルト値は ftp です。

- local_root: ローカルユーザーがログインした後に vsftpd の変更を指定します。

  このディレクティブにはデフォルト値がありません。

- local_umask: ファイル作成の umask 値を指定します。デフォルト値は 8 進数形式（8 つの数字のシステム）です。これには「0」のプレフィックスが含まれます。それ以外の場合は、値は base-10 整数として処理されます。

  デフォルト値は 077 です。Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;LinuxRed Hat Enterprise Linux 6nbsp;6 では、このオプションは設定ファイルで 022 に設定されています。

- passwd_chroot_enable - chroot_local_user ディレクティブとともに有効にすると、 vsftpd は /etc/passwd 内の home-directory フィールドで /./ の発生に基づいてローカル

ユーザーを変更します。

デフォルト値は NO です。

- user_config_dir: これらのユーザーの特定の設定が含まれるローカルシステムユーザーの名前を持つ設定ファイルを含むディレクトリーへのパスを指定します。ユーザーの設定ファイルのディレクティブは、/etc/vsftpd/vsftpd.conf にあるディレクティブを上書きします。

  このディレクティブにはデフォルト値がありません。

21.2.2.6.5. ディレクトリーオプション

以下は、ディレクトリーに影響するディレクティブの一覧です。

- dirlist_enable: 有効にすると、ユーザーはディレクトリー一覧を表示できます。

  デフォルト値は YES です。

- dirmessage_enable: 有効にすると、ユーザーがメッセージファイルのあるディレクトリーに入るたびにメッセージが表示されます。このメッセージは、現在のディレクトリーにあります。このファイルの名前は message_file ディレクティブに指定されており、デフォルトでは .message です。

  デフォルト値は NO です。Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;LinuxRed Hat Enterprise Linux 6nbsp;6 では、このオプションは設定ファイルで YES に設定されます。

- force_dot_files: 有効にすると、. および .. ファイルを除き、ドット(.)で始まるファイルがディレクトリー一覧に表示されます。

  デフォルト値は NO です。

- hide_ids: これを有効にすると、各ファイルのユーザーおよびグループとして、すべてのディレクトリー一覧が ftp と表示されます。

デフォルト値は NO です。

- *message_file: dirmessage_enable ディレクティブの使用時にメッセージファイルの名前を指定します。*

  デフォルト値は .message です。

- *text_userdb_names: 有効にすると、UID エントリーおよび GID エントリーの代わりにテキストユーザー名およびグループ名が使用されます。このオプションを有効にすると、サーバーのパフォーマンスに悪影響を及ぼす可能性があります。*

  デフォルト値は NO です。

- *use_localtime: 有効にすると、ディレクトリーの一覧は、GMT ではなくコンピューターのローカルタイムを表示します。*

  デフォルト値は NO です。

21.2.2.6.6. ファイル転送オプション

以下は、ディレクトリーに影響するディレクティブの一覧です。

- *download_enable: 有効にすると、ファイルのダウンロードが許可されます。*

  デフォルト値は YES です。

- *chown_uploads: 有効にすると、匿名ユーザーによってアップロードされるすべてのファイルは、chown_username ディレクティブで指定されたユーザーが所有します。*

  デフォルト値は NO です。

- *chown_username: chown_uploads ディレクティブが有効な場合に匿名でアップロード*

されたファイルの所有権を指定します。

デフォルト値は root です。

- write_enable - 有効にすると、ファイルシステムが変更できる FTP コマンド（DELE、R NFR、STOR など）。

  デフォルト値は NO です。Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;LinuxRed Hat Enterprise Linux 6nbsp;6 では、このオプションは設定ファイルで YES に設定されます。

21.2.2.6.7. ロギングのオプション

以下は、vsftpd のロギング動作に影響するディレクティブの一覧です。

- dual_log_enable: xferlog_enable と併用すると、vsftpd は 2 つのファイルを同時に書き込みます。wu-ftpd- xferlog_file ディレクティブ（デフォルトでは/var/log/xferlog ）で指定された標準の vsftpd ログファイル （デフォルトでは/var/log/vsftpd.log ）です。

  デフォルト値は NO です。

- log_ftp_protocol: xferlog_enable とともに有効にされ、xferlog_std_format が NO に設定されていると、FTP コマンドおよび応答がすべてログに記録されます。このディレクティブはデバッグに役立ちます。

  デフォルト値は NO です。

- syslog_enable - xferlog_enable と併用すると、通常は vsftpd _log_file ディレクティブ（デフォルトでは/var/log/vsftpd.log ）で指定された標準の vsftpd ログファイルに書き込まれるすべてのロギングが FTPD ファシリティー下でシステムロガーに送信されます。

  デフォルト値は NO です。

- vsftpd_log_file: vsftpd ログファイルを指定します。このファイルを使用するには、xferlog_enable を有効にし、xferlog_std_format を NO に設定する

か、xferlog_std_format が YES に設定されている必要があります。syslog_enable が YES に設定されている場合には、このディレクティブで指定されたファイルの代わりにシステムログが使用されることに注意してください。

デフォルト値は /var/log/vsftpd.log です。

- *xferlog_enable: 有効にすると、vsftpd は接続（vsftpd 形式のみ）およびファイル転送情報を vsftpd_log_file ディレクティブ（デフォルトでは/var/log/vsftpd.log ）に指定したログファイルに記録します。xferlog_std_format が YES に設定されている場合、ファイル転送情報はログに記録されますが、接続はログに記録されず、代わりに xferlog_file （デフォルトでは/var/log/xferlog ）で指定されたログファイルが使用されます。dual_log_enable が YES に設定されている場合、ログファイルとログ形式の両方が使用されることに注意してください。*

  デフォルト値は NO です。Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;LinuxRed Hat Enterprise Linux 6nbsp;6 では、このオプションは設定ファイルで YES に設定されます。

- *xferlog_file - wu-ftpd- 互換のログファイルを指定します。このファイルを使用するには、xferlog_enable を有効にし、xferlog_std_format を YES に設定する必要があります。dual_log_enable が YES に設定されている場合にも使用されます。*

  デフォルト値は /var/log/xferlog です。

- *xferlog_std_format: xferlog_enable と併用すると、wu-ftpd- 互換の file-transfer ログのみが、xferlog_file ディレクティブ（デフォルトでは/var/log/xferlog ）に指定されたファイルに書き込まれます。このファイルではファイル転送のみが記録され、サーバーへの接続はログに記録されないことに注意してください。*

  デフォルト値は NO です。Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;LinuxRed Hat Enterprise Linux 6nbsp;6 では、このオプションは設定ファイルで YES に設定されます。

古いログファイル形式との互換性の維持

古い wu-ftpd FTP サーバーに書き込まれたログファイルとの互換性を維持するために、Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;Linux 6nbsp;6 では xferlog_std_format ディレクティブが YES に設定されます。ただし、この設定は、サーバーへの接続がログに記録されないことを意味します。vsftpd 形式の両方のログ接続と wu-ftpd- 互換のファイル転送ログを維持するには、nmy _log_enable を YES に設定します。wu-ftpd- 互換のファイル転送ログを維持することが重要でない場合は、xferlog_std_format を NO に設定し、ハッシュ記号（「#」）で行をコメント化するか、行を完全に削除します。

21.2.2.6.8. ネットワークオプション

以下は、vsftpd がネットワークと対話する方法を定義するディレクティブの一覧です。

- *accept_timeout: パッシブモードを使用して接続を確立するクライアントの時間を指定します。*

  デフォルト値は 60 です。

- *anon_max_rate: 匿名ユーザーのデータ転送速度を 1 秒あたりのバイト単位で指定します。*

  デフォルト値は 0 で、転送レートは制限しません。

- *connect_from_port_20: 有効にすると、vsftpd はアクティブモードのデータ転送中にサーバーでポート 20 を開くのに十分な権限で稼働します。このオプションを無効にすると、vsftpd は少ない権限で実行できますが、一部の FTP クライアントと互換性がなくなる可能性があります。*

  デフォルト値は NO です。Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;LinuxRed Hat Enterprise Linux 6nbsp;6 では、このオプションは設定ファイルで YES に設定されます。

- *connect_timeout: アクティブモードを使用するクライアントがデータ接続に応答する最大時間を指定します（秒単位）。*

デフォルト値は 60 です。

- *data_connection_timeout: データ転送が停止できる最大時間（秒単位）を指定します。トリガーされると、リモートクライアントへの接続が閉じられます。*

  デフォルト値は 300 です。

- *ftp_data_port: connect_from_port_20 が YES に設定されている場合に、アクティブなデータ接続に使用するポートを指定します。*

  デフォルト値は 20 です。

- *idle_session_timeout: リモートクライアントからのコマンド間の最大時間を指定します。トリガーされると、リモートクライアントへの接続が閉じられます。*

  デフォルト値は 300 です。

- *listen_address: vsftpd がネットワーク接続をリッスンする IP アドレスを指定します。*

  このディレクティブにはデフォルト値がありません。

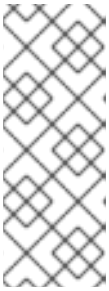
  

  VSFTPD の複数コピーの実行

  異なる IP アドレスを提供する vsftpd の複数のコピーを実行する場合は、vsftpd デーモンの各コピー用の設定ファイルに、このディレクティブにはそれぞれ異なる値を指定する必要があります。マルチホーム FTP サーバーの詳細は、「vsftpd の複数コピーの起動」 を参照してください。

- *listen_address6: listen_ipv6 が YES に設定されている場合に vsftpd がネットワーク接続をリッスンする IPv6 アドレスを指定します。*

  このディレクティブにはデフォルト値がありません。



VSFTPD の複数コピーの実行

異なる IP アドレスを提供する vsftpd の複数のコピーを実行する場合は、vsftpd デーモンの各コピー用の設定ファイルに、このディレクティブにはそれぞれ異なる値を指定する必要があります。マルチホーム FTP サーバーの詳細は、「vsftpd の複数コピーの起動」を参照してください。

- *listen_port: vsftpd がネットワーク接続をリッスンするポートを指定します。*

  デフォルト値は 21 です。

- *local_max_rate: サーバーにログインするローカルユーザーに対してデータを転送する最大レートを指定します（バイト毎秒単位）。*

  デフォルト値は 0 で、転送レートは制限しません。

- *max_clients: スタンドアロンモードで実行する際にサーバーに接続可能な同時クライアントの最大数を指定します。追加のクライアント接続により、エラーメッセージが生成されます。*

  デフォルト値は 0 で、接続に制限されません。

- *max_per_ip: 同じソース IP アドレスから接続できるクライアントの最大数を指定します。*

  デフォルト値は 50 です。値が 0 の場合は、制限をオフにします。

- *pasv_address: Network Address Translation (NAT)ファイアウォールの背後にあるサーバーのパブリック向け IP アドレスの IP アドレスを指定します。これにより、vsftpd が passive-mode 接続の正しいリターンアドレスを渡します。*

  このディレクティブにはデフォルト値がありません。

- *pasv_enable - 有効にすると、passive-mode 接続が許可されます。*

  デフォルト値は YES です。

- *pasv_max_port - パッシブモード接続用に FTP クライアントに送信される可能な最大ポートを指定します。この設定は、ファイアウォールルールの作成を容易にするために、ポート範囲を制限するために使用されます。*

  デフォルト値は 0 で、最も高い passive-port 範囲の制限はありません。値は 65535 を超えることはできません。

- *pasv_min_port - パッシブモード接続用に FTP クライアントに送信されるポートの下限を指定します。この設定は、ファイアウォールルールの作成を容易にするために、ポート範囲を制限するために使用されます。*

  デフォルト値は 0 で、最小の passive-port 範囲の制限はありません。値は 1024 未満にすることはできません。

- *pasv_promiscuous: 有効な場合、データ接続はチェックされず、それらが同じ IP アドレスから取得されていることを確認します。この設定は、特定のトンネリングのタイプにのみ役立ちます。*

  PASV_PROMISCUOUS オプションを有効にしない

  パッシブモードの接続がデータ転送を開始するコントロール接続と同じ IP アドレスから送信されることを検証する重要なセキュリティー機能を無効にするため、絶対に必要な場合を除き、このオプションを有効にしないでください。

  デフォルト値は NO です。

- *port_enable: 有効にすると、アクティブモードの接続が許可されます。*

  デフォルト値は YES です。

21.2.2.6.9. セキュリティーオプション

以下は、vsftpd のセキュリティー向上に使用できるディレクティブの一覧です。

- *isolate_network: 有効にすると、vsftpd は CLONE_NEWNET コンテナーフラグを使用して、権限のないプロトコルハンドラープロセスを分離して、任意に connect（） を呼び出しないようにし、代わりにソケットの特権プロセスを要求する必要があります（ port_promiscuous オプションを無効にする必要があります）。*

  デフォルト値は YES です。

- *isolate: 有効にすると、vsftpd は CLONE_NEWPID および CLONE_NEWIPC コンテナーフラグを使用してプロセスを IPC および PID 名前空間に分離し、それらのプロセスが互いに対話できないようにします。*

  デフォルト値は YES です。

- *ssl_enable: vsftpd の SSL のサポートを有効にします（TLSを含む）。SSL は認証と後続のデータ転送の両方に使用されます。ssl_enable が YES に設定されている場合に限り、その他の SSL関連のオプションが適用されます。*

  デフォルト値は NO です。

- *allow_anon_ssl: 匿名ユーザーがセキュアな SSL 接続を使用することを許可するかどうかを指定します。*

  デフォルト値は NO です。

- *require_cert: 有効な場合は、クライアント証明書を提示するためにすべての SSL クライアント接続が必要になります。*

デフォルト値は NO です。

21.2.3. その他のリソース

vsftpd 設定に関する詳しい情報は、以下の資料を参照してください。

21.2.3.1. インストールされているドキュメント

- /usr/share/doc/vsftpd-version-number/ ディレクトリー： TUNING ファイルには基本的なパフォーマンス調整のヒントが含まれ、SECURITY/ ディレクトリーには vsftpd が使用するセキュリティーモデルに関する情報が含まれます。

- vsftpd: 関連する man ページ： デーモンおよび設定ファイルに関する man ページは多数あります。以下は、重要な man ページの一部です。

  サーバーアプリケーション

  - vsftpd(8) - vsftpd で利用可能なコマンドラインオプションを説明しています。

  設定ファイル

  - vsftpd.conf(5) - vsftpd の設定ファイルで利用可能なオプションの詳細な一覧が含まれます。

  - hosts_access(5) - hosts. allow および hosts.deny の TCP ラッパーの設定ファイルで利用可能な形式とオプションを説明します。

  SELinux とのインタラクション

  - man ftpd_selinux: ftpd プロセスを管理する SELinux ポリシーと、SELinux ラベルの割り当て方法、およびブール値セットを説明します。

21.2.3.2. オンラインドキュメント

vsftpd および FTP 全般について

- *http://vsftpd.beasts.org/: vsftpd プロジェクトページは、最新のドキュメントやソフトウェアの作成者の連絡先に最適な場所です。*

- *http://slacksite.com/other/ftp.html: この Web サイトでは、アクティブモードと passive-mode FTP の相違点を簡単に説明します。*

Red Hat Enterprise Linux のドキュメンテーション

- *Red Hat Enterprise Linux 6 Security-Enhanced Linux - 『Security-Enhanced Linux』 for Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;LinuxRed Hat Enterprise Linux 6nbsp;6 では、SELinux の基本原則と、Apache HTTP Server、Postfix、PostgreSQL、OpenShift などのさまざまなサービスで SELinux を設定および使用する方法が詳細に説明されています。また、SELinux アクセスパーミッションを systemd が管理するシステムサービス用に設定する方法も説明しています。*

- *Red Hat Enterprise Linux 6 セキュリティーガイド: Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;Linux Red Hat Enterprise Linux 6nbsp;6 の 『セキュリティーガイド』 は、ユーザーおよび管理者がローカルおよびリモートの侵入、悪用、悪意のあるアクティビティーに対してワークステーションおよびサーバーを保護するプロセスとプラクティスを学習する際に役立ちます。また、重大なシステムサービスを保護する方法についても説明しています。*

関連する RFC ドキュメント

- *『RFC 0959:』 IETF からの FTP プロトコルの元の Request for Comments (RFC)*

- *『RFC 1123』: 短い FTP- 関連のセクションで、RFC 0959 を拡張、明確化します。*

- *『RFC 2228』: FTP セキュリティー拡張。vsftpd は、TLS 接続および SSL 接続のサポートに必要な小規模のサブセットを実装します。*

- *『RFC 2389』: FEAT および OPTS コマンドを指定します。*

- *『RFC 2428:』 IPv6 サポート。*

21.3. プリンターの設定

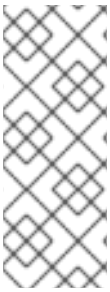
プリンター 設定 ツールは、プリンター設定、プリンター設定ファイルのメンテナンス、印刷スプールディレクトリー、印刷フィルター、プリンタークラスの管理を可能にします。

このツールは、Common Unix Printing System(CUPS)に基づいています。CUPS を使用した以前の Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux バージョンからシステムをアップグレードした場合、アップグレードプロセスは設定済みのプリンターを保持します。

重要

cupsd.conf の man ページには、CUPS サーバーの設定が記載されています。これには、SSL サポートを有効にするためのディレクティブが含まれています。ただし、CUPS では使用されるプロトコルバージョンのコントロールが許可されません。『設定から SSLv3 を無効にできないコンポーネントに対する POODLE SSLv3.0 脆弱性(CVE-2014-3566)の解決で説明されて』 いる脆弱性のため、Red Hat は、セキュリティーのためにこれに依存しないことを推奨します。stunnel を使用してセキュアなトンネルを提供し、SSLv3 を無効にすることが推奨されます。stunnel の使用方法は、『Red Hat Enterprise Linux 6 セキュリティーガイド』 を参照してください。

リモートシステムの 印刷設定 ツールへのアドホックのセキュアな接続は、「X11 転送」 の説明に従って SSH で X11 転送を使用します。



CUPS WEB アプリケーションまたはコマンドラインツールの使用

CUPS Web アプリケーションまたはコマンドラインから直接、同一および追加の動作をプリンターで実行できます。アプリケーションにアクセスするには、Web ブラウザーで http://localhost:631/ にアクセスします。CUPS マニュアルについては、Web サイトの Home タブにあるリンクを参照してください。

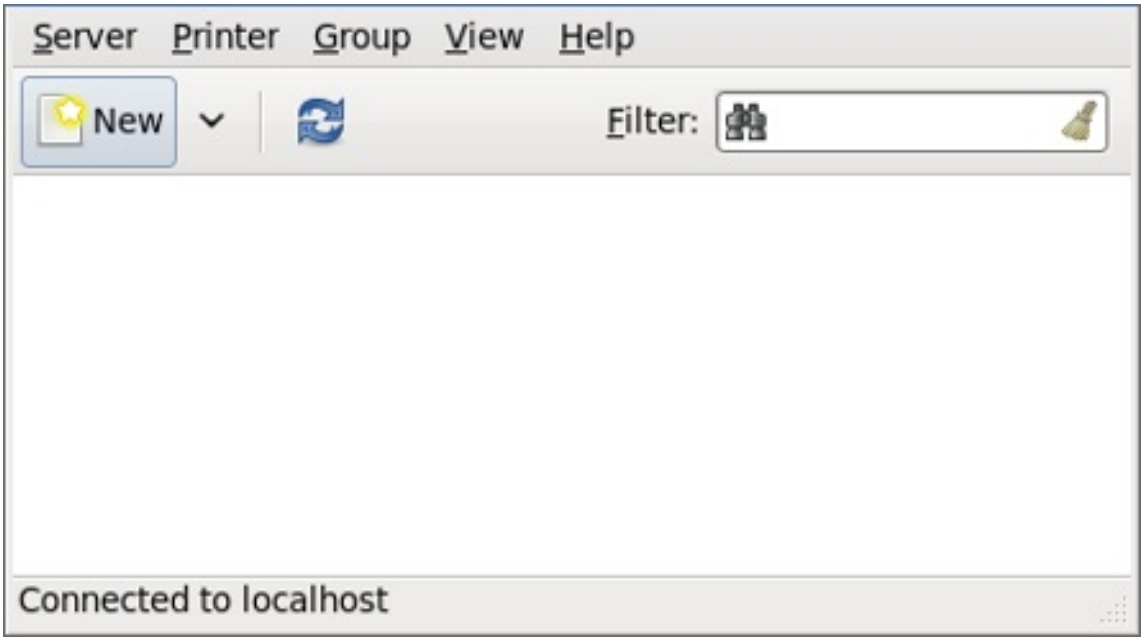
21.3.1. プリンター設定ツールの起動

Printer Configuration ツールを使用すると、既存のプリンターでさまざまな操作を実行したり、新しいプリンターを設定できます。ただし、CUPS を直接使用することもできます（CUPS にアクセスするには http://localhost:631/ を開きます）。

パネルで System → Administration → Printing をクリックするか、コマンドラインから system-config-printer コマンドを実行してツールを起動します。

図21.3「プリンター設定ウィンドウ」に示されている Printer Configuration ウインドウが表示されます。

図21.3 プリンター設定ウィンドウ



[D]

21.3.2. プリンター設定の開始

プリンターの設定プロセスは、プリンターキューのタイプにより異なります。

USB に接続されているローカルプリンターを設定すると、プリンターが検出され、自動的に追加されます。インストールするパッケージの確認と、root パスワードの入力が求められます。他のポートタイプに接続したローカルプリンターやネットワークプリンターの場合は、手動で設定する必要があります。

プリンターを手動で設定するには、以下の手順に従います。

1. プリンター設定ツールを起動します（「プリンター設定ツールの起動」を参照）。
2. Server → New → Printer に移動します。
3. Authenticate ダイアログボックスで、root ユーザーのパスワードを入力して確認します。

4.
プリンターの接続タイプを選択し、右側エリアでその詳細を記入します。

21.3.3. ローカルプリンターの追加

以下の手順に従って、シリアルポート以外に接続されたローカルプリンターを追加します。

1.
新規プリンターダイアログを 開きます（「プリンター設定の開始」を参照）。

2.
デバイスが自動的に表示されない場合は、左側の一覧でプリンターを接続するポートを選択します（Serial Port #1 や LPT #1など）。

3.
右側で、接続プロパティーを入力します。

Otherの場合

URI （例： file:/dev/lp0）

Serial Portの場合

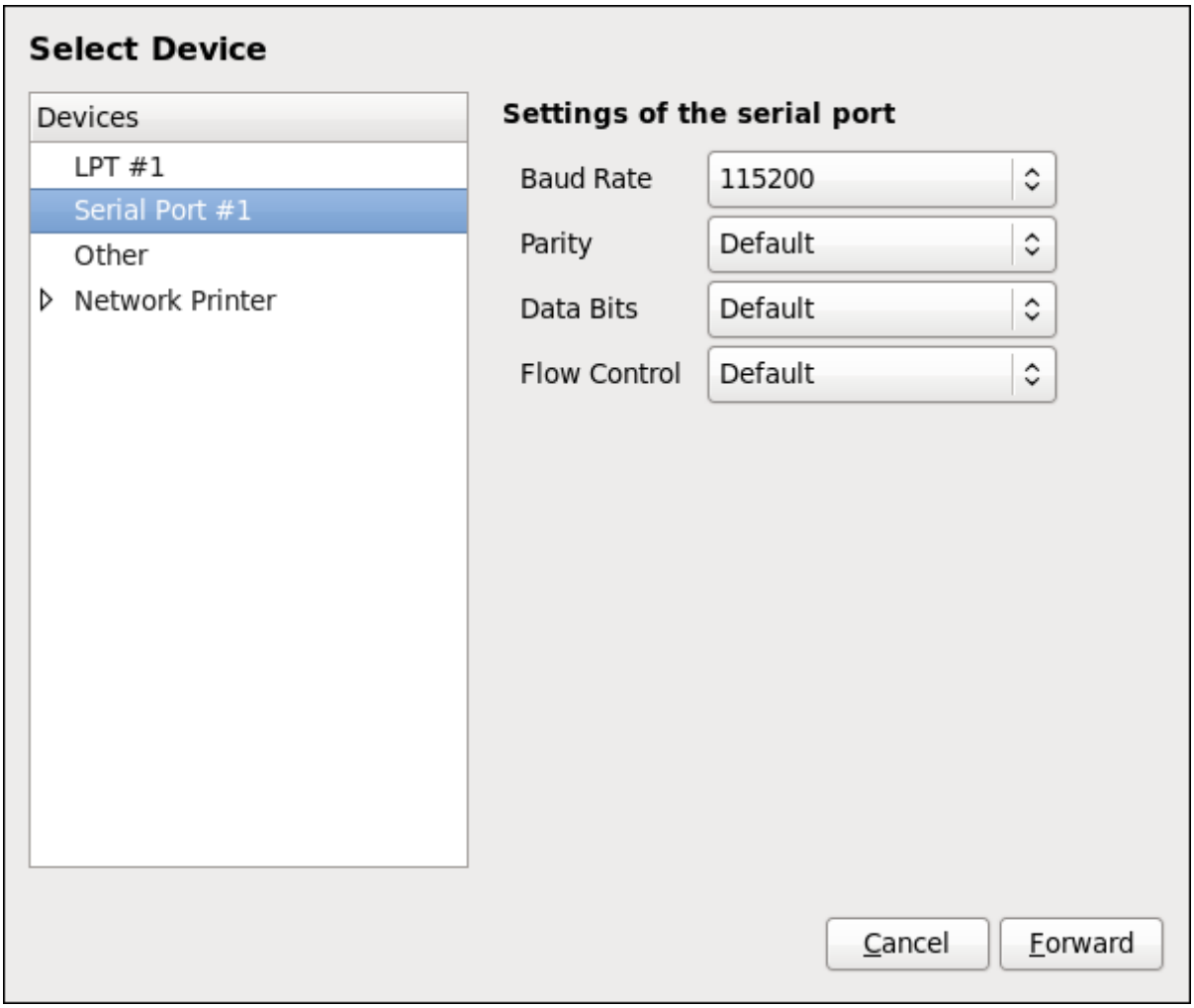
通信速度

パリティー

データビット

フロー制御

図21.4 ローカルプリンターの追加



[D]

4. *Forward をクリックします。*

5. *プリンターのモデルを選択します。詳しくは「プリンターモデルの選択と完了」 を参照してください。*

21.3.4. AppSocket/HP JetDirect プリンターの追加

以下の手順に従って AppSocket/HP JetDirect プリンターを追加します。

1. *新規プリンターダイアログを 開きます（「プリンター設定ツールの起動」を参照）。*

2. *左側の一覧で Network Printer → AppSocket/HP JetDirect の順に選択します。*

3.

右側で、接続設定を入力します。

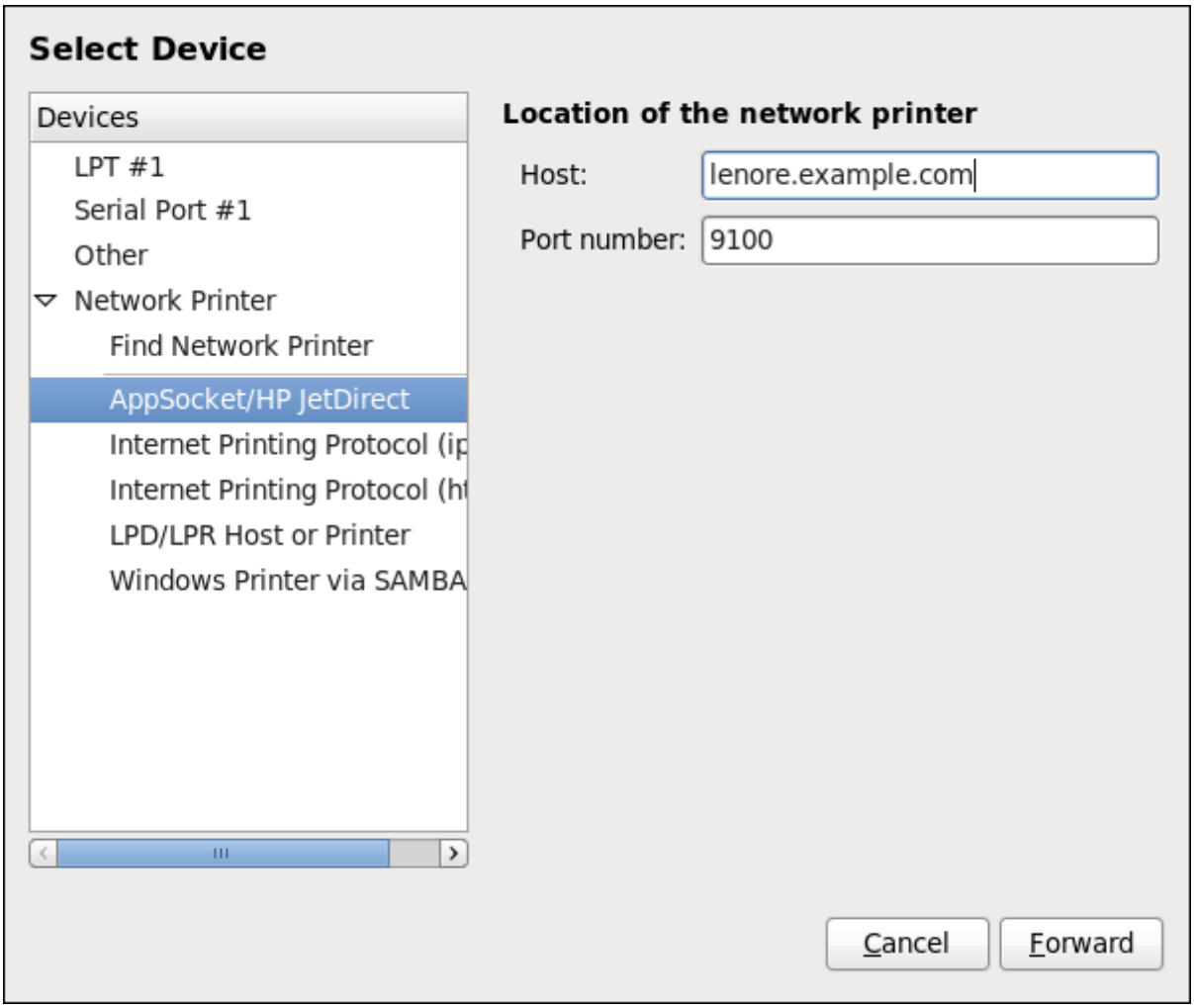
Hostname

プリンターホスト名または IP アドレス。

ポート番号

印刷ジョブをリッスンするプリンターポート (デフォルトは 9100)

図21.5 JetDirect プリンターの追加



[D]

4.

Forward をクリックします。

5.

プリンターのモデルを選択します。詳しくは 「プリンターモデルの選択と完了」 を参照

してください。

21.3.5. IPP プリンターの追加

IPP プリンターは同じ TCP/IP ネットワークにある別のシステムに接続されているプリンターです。このプリンターがアタッチされているシステムは CUPS を実行しているか、IPP を使用するように設定されている。

プリンターサーバーでファイアウォールが有効な場合は、ポート 631 で受信 TCP 接続を許可するようにファイアウォールを設定する必要があります。プロトコルを参照する CUPS により、クライアントマシンは共有 CUPS キューを自動的に検出することが可能です。これを有効にするには、クライアントマシンのファイアウォールをポート631 で受信 UDP パケットを許可するように設定する必要があります。

IPP プリンターを追加するには、以下の手順に従います。

1. 新規プリンターダイアログを 開きます（「プリンター設定の開始」を参照）。

2. 左側のデバイスの一覧で、Network Printer and Internet Printing Protocol(ipp) または Internet Printing Protocol(https) を選択します。

3. 右側で、接続設定を入力します。

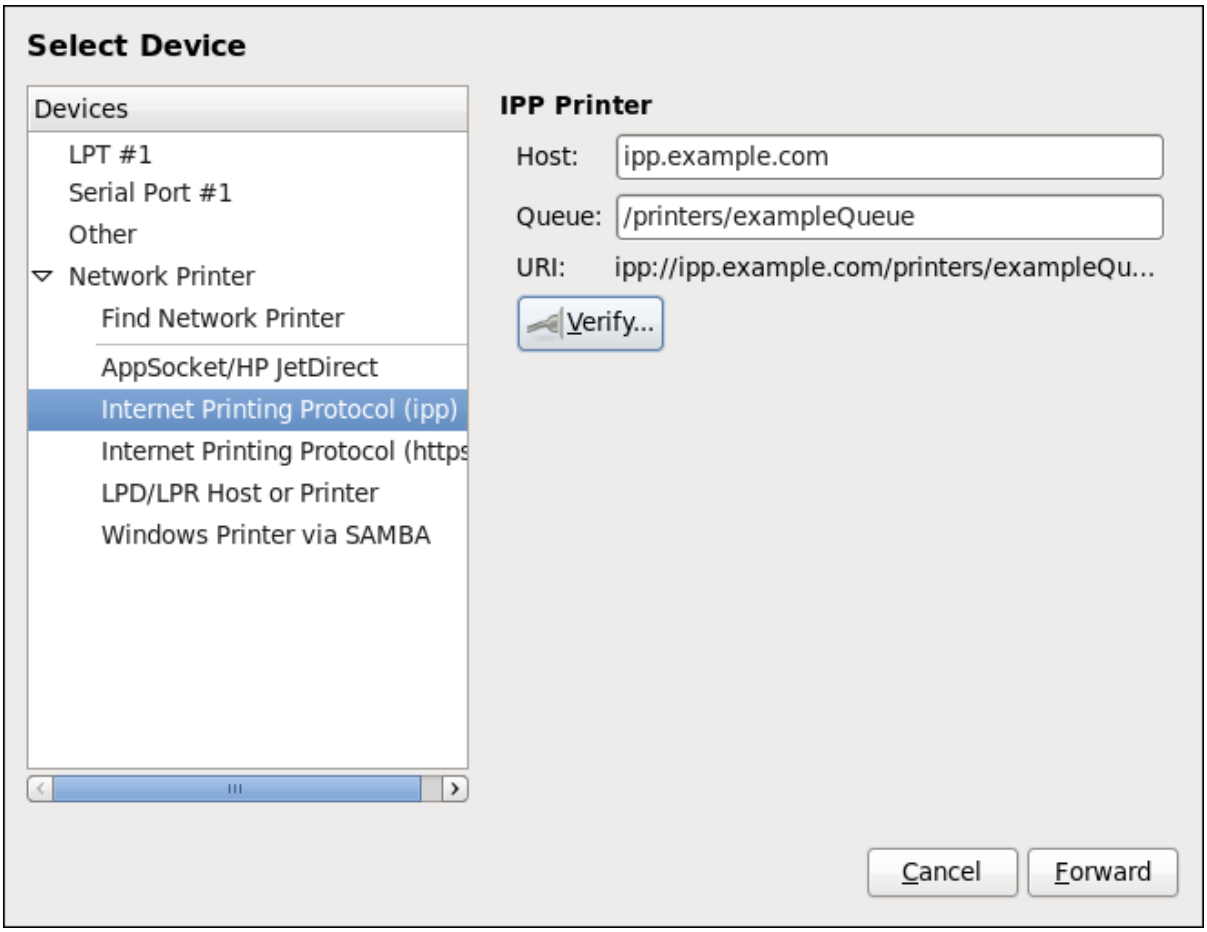
ホスト

IPP プリンターのホスト名。

Queue

新規のキューに与えるキューの名前です (このボックスを空白のままにすると、デバイスノードに基づいた名前が使用されます)。

図21.6 IPP プリンターの追加



[D]

4. *進む をクリックして続けます。*

5. *プリンターのモデルを選択します。詳しくは「プリンターモデルの選択と完了」 を参照してください。*

21.3.6. LPD/LPR Host or Printer の追加

以下の手順に従って LPD/LPR ホストまたはプリンターを追加します。

1. *新規プリンターダイアログを 開きます（「プリンター設定の開始」を参照）。*

2. *左側のデバイス一覧で、Network Printer → LPD/LPR Host or Printer の順に選択します。*

3.

右側で、接続設定を入力します。

Host

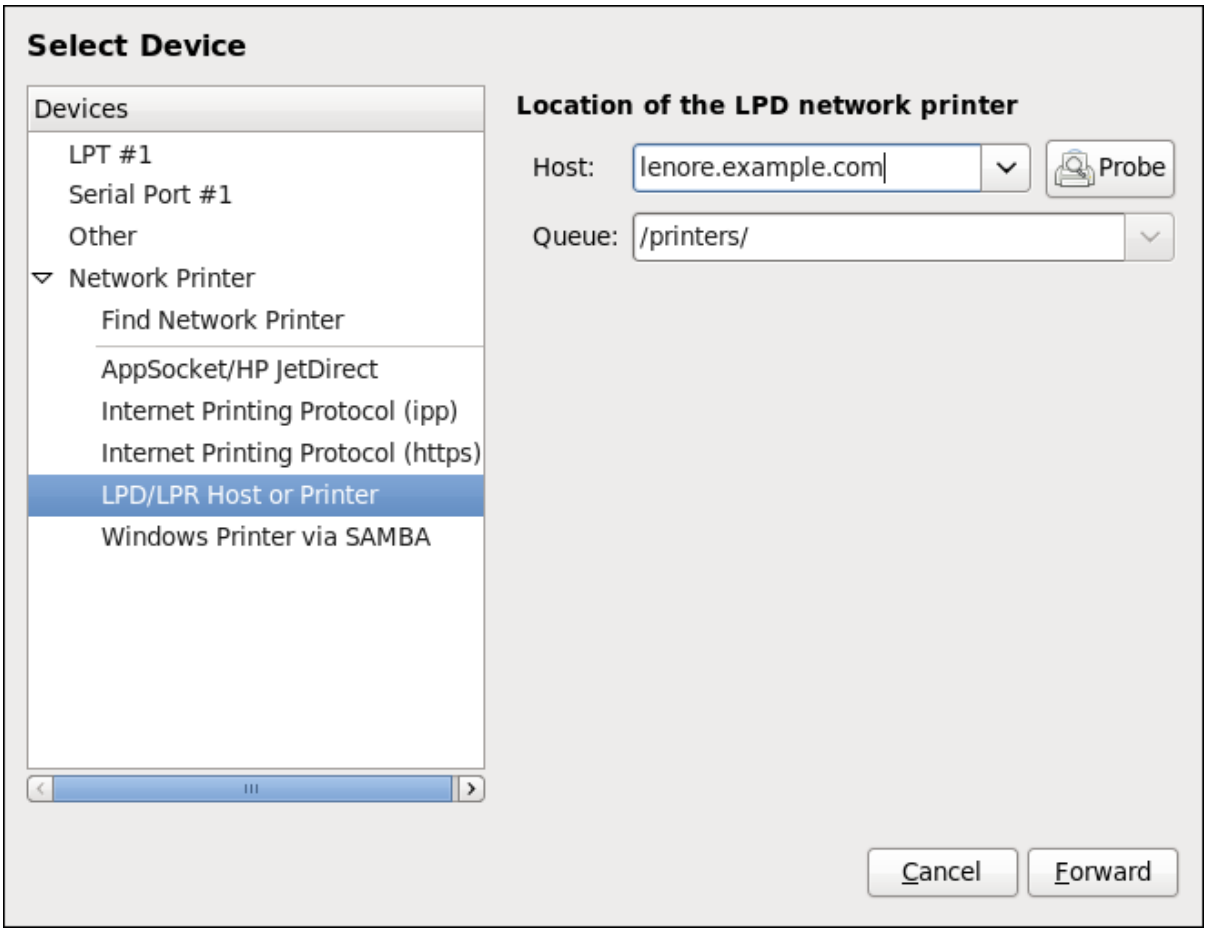
LPD/LPR プリンターまたはホストのホスト名

必要に応じて、Probe をクリックし、LSD ホスト上のキューを検索します。

Queue

新規のキューに与えるキューの名前です (このボックスを空白のままにすると、デバイスノードに基づいた名前が使用されます)。

図21.7 LPD/LPR プリンターの追加



[D]

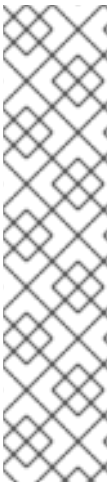
4.

進む をクリックして続けます。

5.
プリンターのモデルを選択します。詳しくは 「プリンターモデルの選択と完了」 を参照してください。

21.3.7. Samba (SMB) プリンターの追加

Samba プリンターを追加するには、以下の手順を実行します。



SAMBA-CLIENT パッケージのインストール

Samba プリンターを追加するには、samba-client パッケージがインストールされている必要があります。root で以下のコマンドを実行してこれを実行できます。

```
yum install samba-client
```

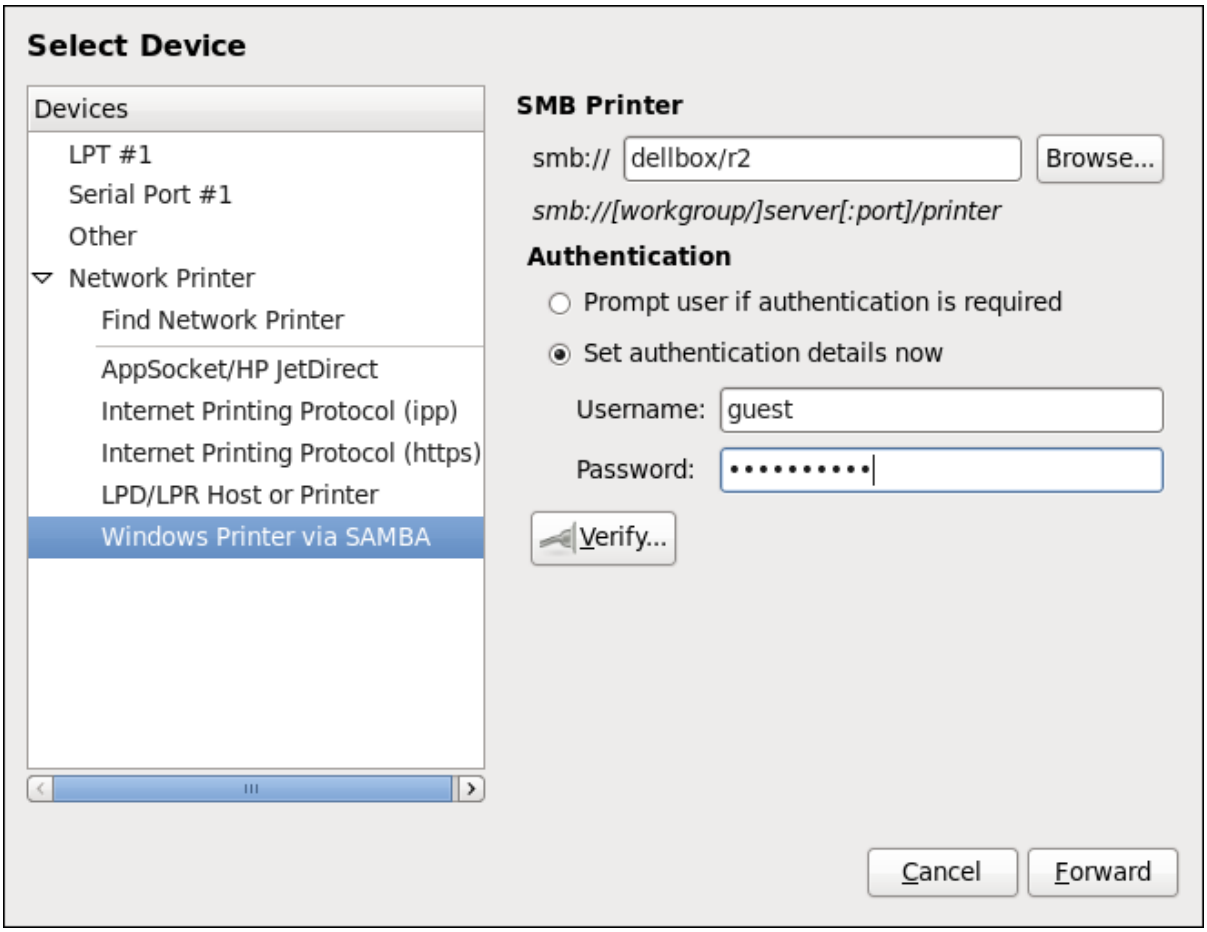
yum を使用したパッケージのインストールは 「パッケージのインストール」 を参照してください。

1.
新規プリンターダイアログを 開きます（「プリンター設定の開始」を参照）。

2.
左側の一覧で、Network Printer → Windows Printer via SAMBA の順に選択します。

3.
smb:// フィールドに SMB アドレスを入力します。computer name/printer share の形式を使用します。図21.8「SMB プリンターの追加」 では、コンピューター名 は dellbox で、プリンター共有 は r2 です。

図21.8 SMB プリンターの追加



[D]

4.

利用できるワークグループやドメインを確認するには、閲覧する (Browse) をクリックします。特定のホストのキューだけを表示させるには、ホスト名 (NetBios 名) を入力して、閲覧する をクリックします。

5.

以下のオプションのいずれかを選択します。

- *認証が必要な場合にはプロンプトユーザー （ユーザー名およびパスワードが必要）は、 ドキュメントを印刷する際にユーザーから収集されます。*

- *ここで認証の詳細 を設定する：認証情報を提供するため、後で認証情報は必要ありません。ユーザー 名 フィールドには、プリンターにアクセスするユーザー名を入力します。このユーザーは、SMB システムで存在している必要があり、ユーザーはプリンターへ*

のアクセス権限を持っている必要があります。デフォルトのユーザー名は通常、Windows サーバーでは guest、または Samba サーバーの場合は nobody です。

6.

ユーザー名 フィールドに指定したユーザーの パスワード （必要な場合）を 入力 します。

パスワードを選択する際には注意してください。

Samba プリンターのユーザー名とパスワードは、root および lpd(Linux Printing Daemon)が読み取り可能な暗号化されていないファイルとしてプリンターサーバーに保存されます。そのため、プリンターサーバーに root アクセスを持つ他のユーザーは、Samba プリンターへのアクセスに使用するユーザー名とパスワードを表示できます。

したがって、Samba プリンターにアクセスするためのユーザー名とパスワードを選択する場合は、ローカルの Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux システムへのアクセスに使用するパスワードとは異なるパスワードを選択することが推奨されます。

Samba プリンターサーバーで共有するファイルがある場合も、印刷キューで使用されるパスワードとは異なるパスワードを使用することが推奨されます。

7.

確認 (Verify) をクリックし、接続をテストします。確認が成功すると、ダイアログボックスが表示され、プリンター共有のアクセスを確認します。

8.

Forward をクリックします。

9.

プリンターのモデルを選択します。詳しくは 「プリンターモデルの選択と完了」 を参照してください。

21.3.8. プリンターモデルの選択と完了

適切なプリンターの接続タイプを選択すると、システムはドライバーを取得するよう試行します。プロセスが失敗した場合は、ドライバーリソースを手動で検索できます。

以下の手順に従い、プリンタードライバーを設定してインストールを完了します。

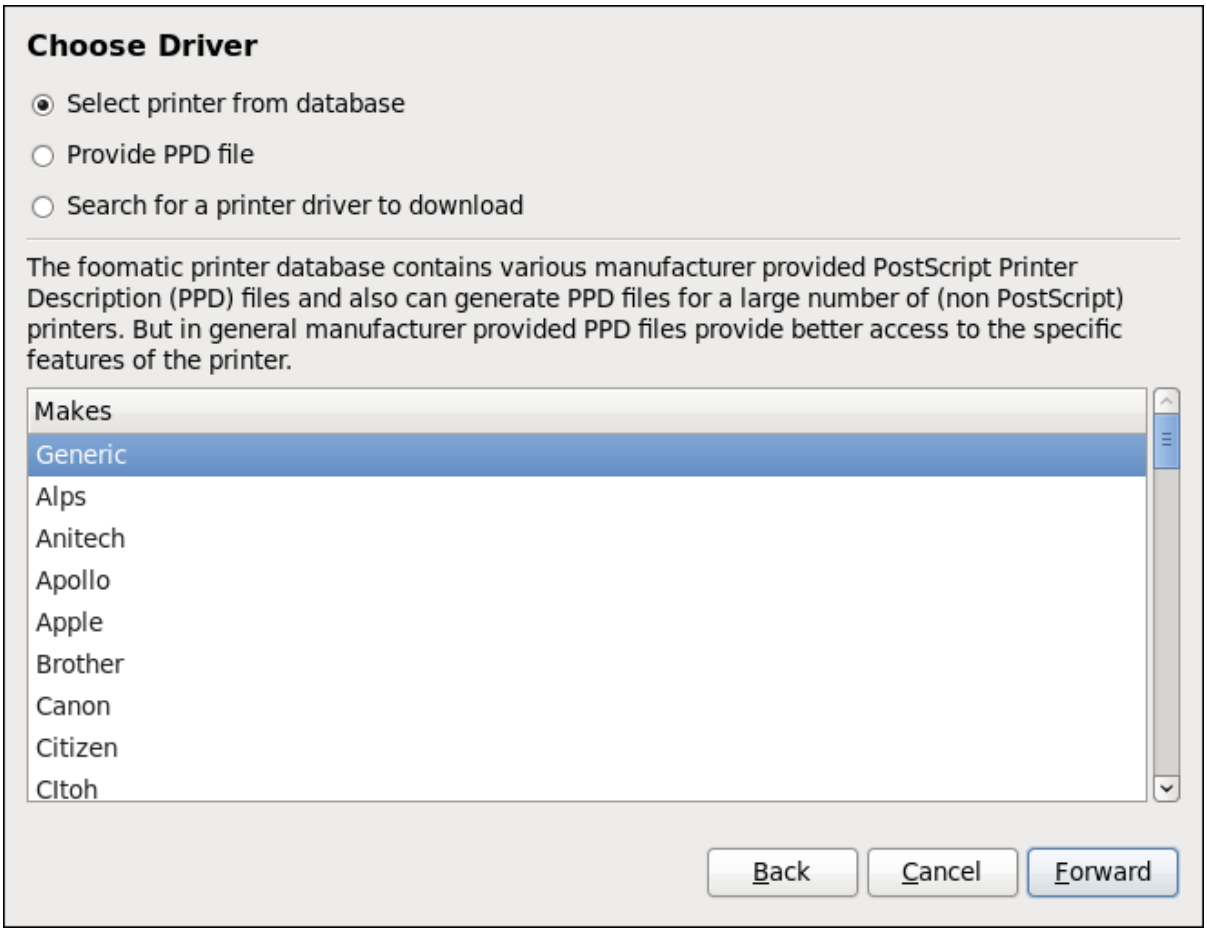
1.
ドライバーの自動検知が失敗すると、ウィンドウが表示されます。以下のオプションのいずれかを選択します。

- データベースからプリンターを選択(Select a Printer from database): システムは、Makes の一覧から選択したプリンターの製造元に基づいてドライバーを選択します。ご使用のプリンターのモデルが一覧にない場合は、Generic を選択します。

- PPD ファイルを提供(Provide PPD file)- システムは、提供されている PPD(PostScript Printer Description)ファイルを使用してインストールを行います。PPD ファイルは製造元が通常提供するプリンターに同梱されています。PPD ファイルが利用可能な場合は、このオプションを選択し、オプションの詳細の下にあるブラウザバーを使用して、PPD ファイルを選択できます。

- ダウンロードするプリンタードライバーを検索し ます。製造元とプリンターのモデルを Make and model フィールドに入力し、OpenPrinting.org で適切なパッケージを検索します。

図21.9 プリンターブランドの選択



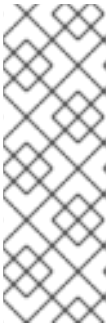
[D]

2. *上で選択した内容により、以下に表示される詳細は異なります。*

   - *データベースからプリンターを選択 オプションの場合は、プリンター ブランドが表示されます。*
   - *PPD ファイルを提供 オプションの場合は、PPD ファイル の場所を指定します。*
   - *ダウンロードするプリンタードライバーの検索オプションに使用するプリンター 製造元とモデル*

3. *進む をクリックして続けます。*

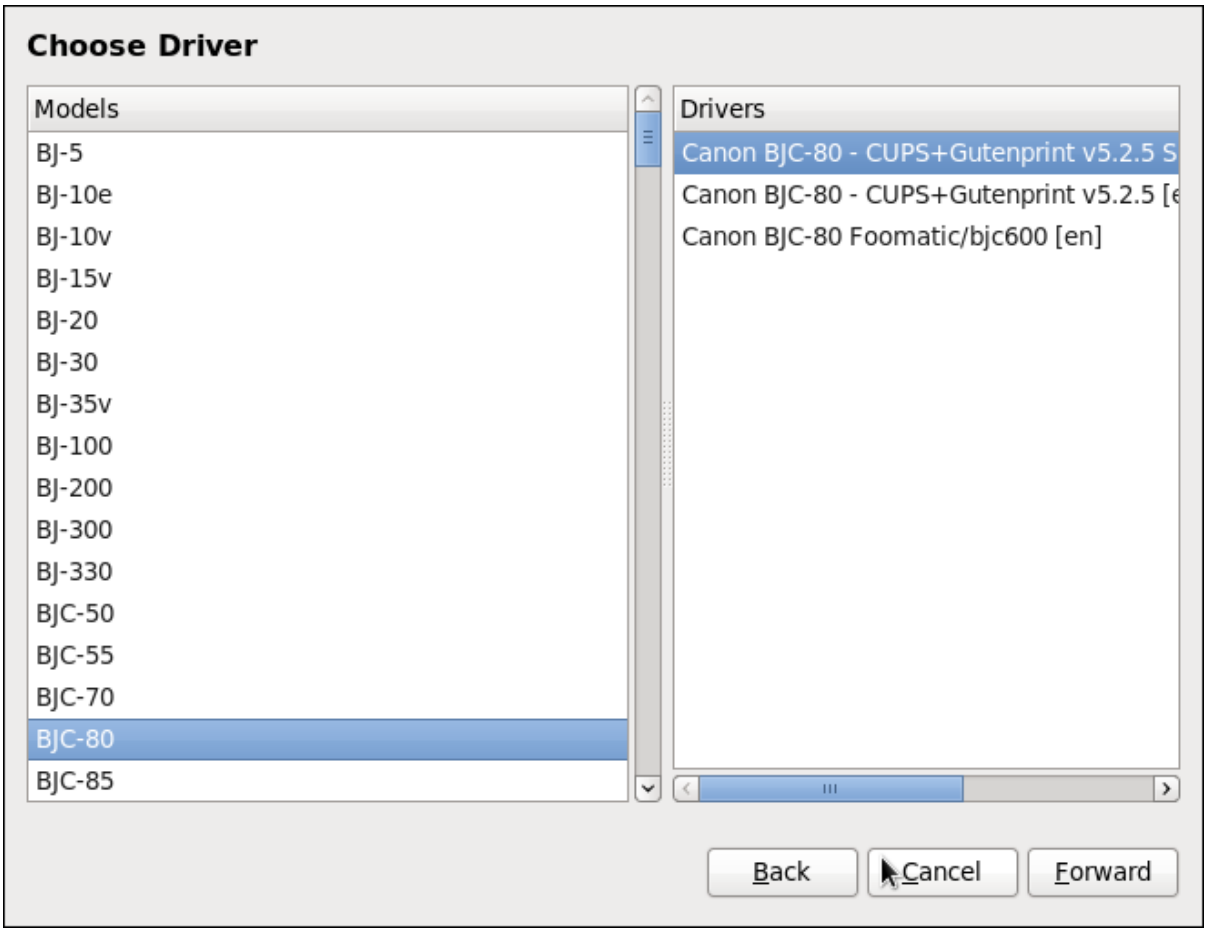
4. *選択したオプションが該当する場合は、図21.10「プリンターモデルの選択」 のようなウィンドウが表示されます。左側の Models 列で該当するモデルを選択します。*



プリンタードライバーの選択

右側で、推奨される印刷ドライバーが自動的に選択されています。ただし、別の利用可能なドライバーを選択することもできます。ただし、別の利用可能なドライバーを選ぶことも可能です。ローカルプリンターはコンピューターに直接接続されているため、プリンターに送信されるデータを処理するにはプリンタードライバーが必要です。

図21.10 プリンターモデルの選択



[D]

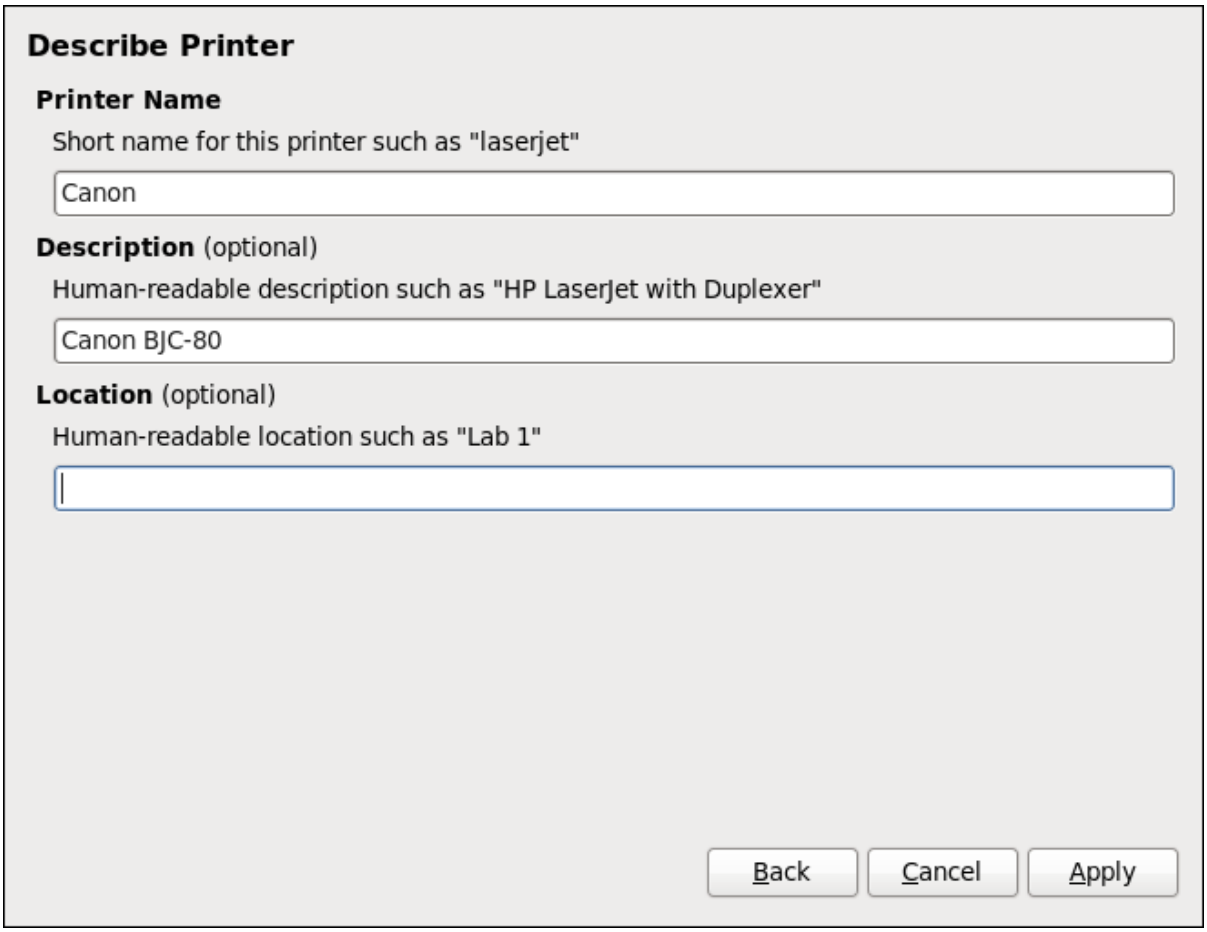
5.

Forward をクリックします。

6.

説明 プリンター の下の、プリンター名 フィールドに一意のプリンター名を 入力 します。プリンター名には、文字、数字、ダッシュ(-)、およびアンダースコア(_)を使用できます。スペースを含めること はできません。また、説明 フィールドおよび Location フィールドを使用して、さらにプリンター情報を追加することもできます。どちらもオプションで、スペースを入れることは可能です。

図21.11 プリンターの設定



[D]

7. *設定が正しければ、適用 (Apply) をクリックして、ご使用のプリンター設定を確認し、印刷キューを追加できます。戻る (Back) をクリックすると、プリンター設定を変更できます。*

8. *変更が適用されると、テストページの印刷を行うダイアログボックスが表示されます。Yes をクリックするとテストページが印刷されます。「テストページの印刷」 の記載通りに、テストページを後で印刷することもできます。*

21.3.9. テストページの印刷

プリンターを設定、またはプリンターの設定を変更した後は、テストページを印刷して、プリンターが適切に機能していることを確認します。

1. *印刷(Printing)ウィンドウでプリンターを右クリックし、Properties をクリックします。*

2. *プロパティーウィンドウで、左側の 設定 を クリックします。*

3. 表示されている Settings タブで、テストページの印刷(Print Test Page)ボタンをクリックします。

21.3.10. 既存プリンターの修正

既存のプリンターを削除するには、Printer Configuration ウィンドウでプリンターを選択し、Printer → Delete に移動します。プリンターの削除を確認します。別の方法として、Delete キーを押しても削除できます。

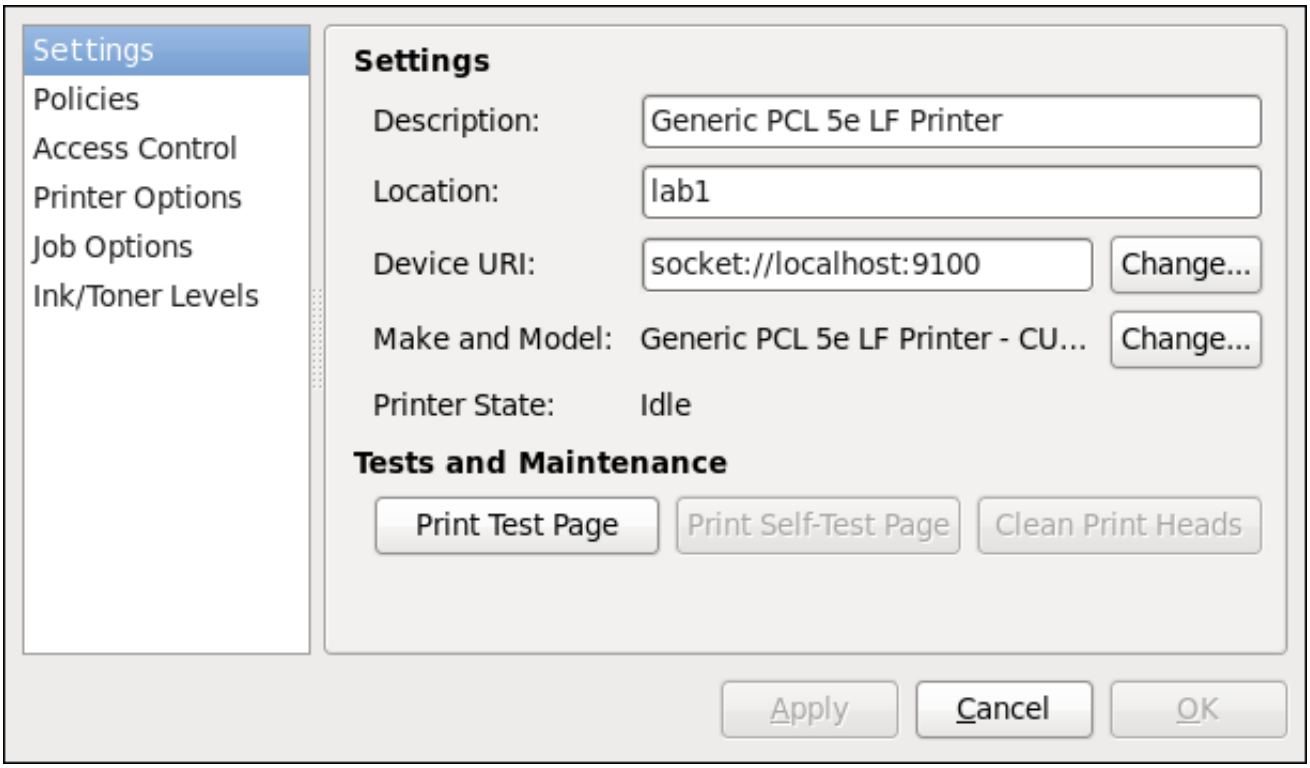
デフォルトのプリンターを設定するには、プリンター一覧でプリンターを右クリックし、コンテキストメニューで Set as Default ボタンをクリックします。

21.3.10.1. 設定のページ

プリンターのドライバー設定を変更するには、プリンター 一覧で該当する名前をダブルクリックして、左側の 設定 ラベルをクリックして 設定 ページを表示 し ます。

製造元やモデルなどのプリンター設定の変更、テストページの印刷、デバイスの場所 (URI) の変更など行うことができます。

図21.12 設定ページ



[D]

21.3.10.2. ポリシーページ

プリンターの状態や出力を変更するには、左側の ポリシー ボタンをクリックします。

プリンターの状態を選択したり、プリンターの エラーポリシー(Error Policy)を設定できます（エラーが発生した場合は、印刷ジョブを中止、再試行、停止できます）。

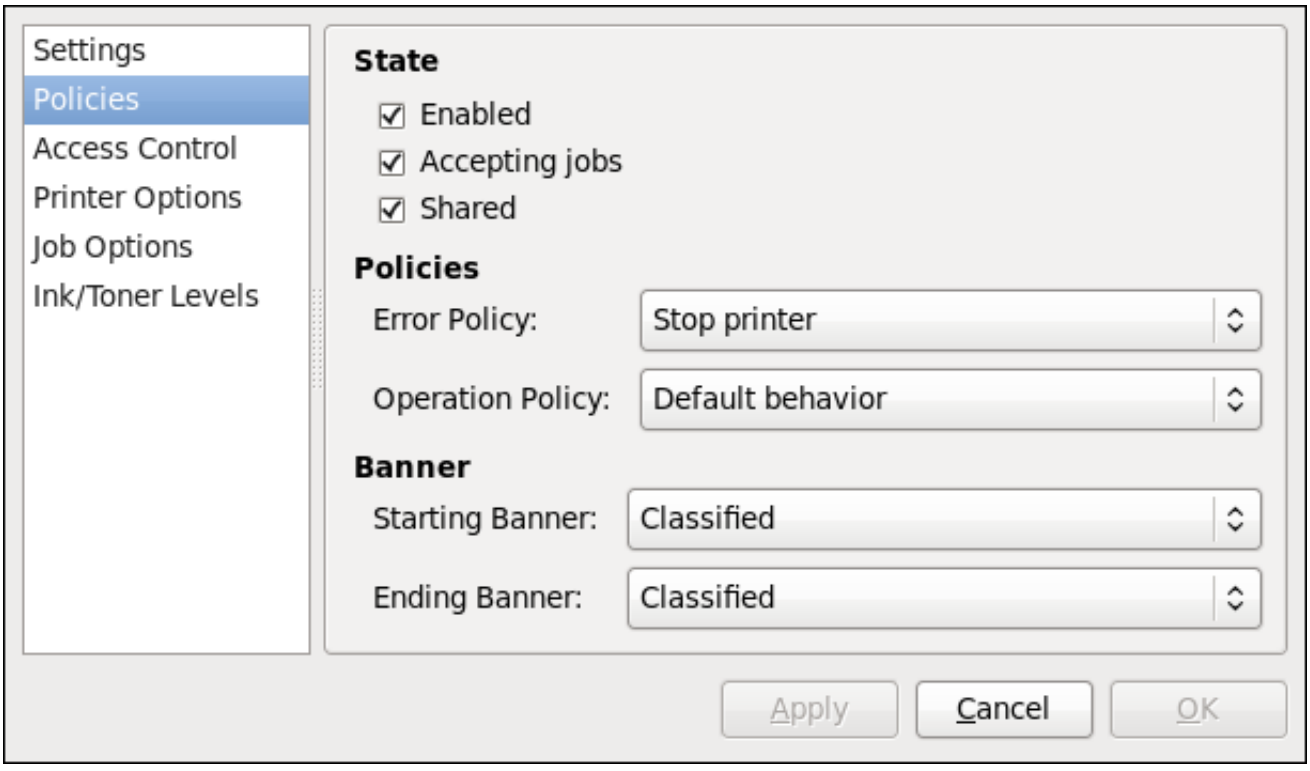
バナーページ （送信元プリンター、ジョブを 開始 したユーザー名、印刷中の文書のセキュリティー状態など、印刷ジョブの特徴を説明するページ）の作成も可能です。開始バナーまたは 終了バナーをクリックし、印刷ジョブの 性質に最適なオプションを選択します（ 機密ジョブなど）。

21.3.10.2.1. プリンターの共有

ポリシー ページでは、プリンターを共有としてマークできます。プリンターが共有されている場合は、ネットワーク上で公開されているユーザーはそれを使用できます。プリンターの共有機能を許可するには、Server → Settings に移動し、このシステムに接続されている共有プリンターを公開 を 選択します。

最後に、ファイアウォールで system-config-firewall のポート 631 への着信 TCP 接続（Network Printing Server(IPP)）を許可するようにしてください。

図21.13 ポリシーページ

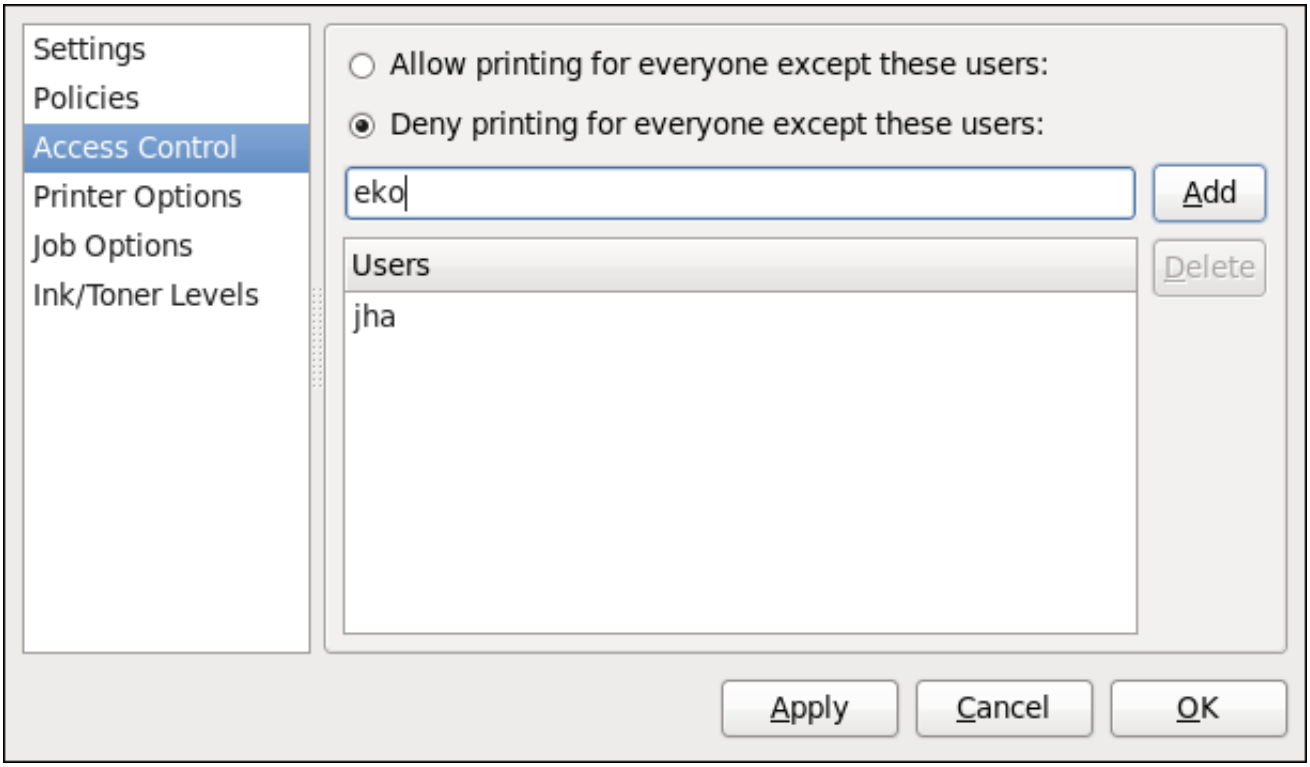


[D]

21.3.10.2.2. アクセス制御のページ

アクセス 制御 ページで設定したプリンターへ のユーザーレベルのアクセスを変更できます。左側の **Access Control** ラベルをクリックし、ページが表示されます。これらのユーザー以外のすべてのユーザーに対して印刷を許可 するか、またはこれらのユーザー以外の印刷を拒否する(**Deny printing for everyone except these users**)を選択し、以下のようにユーザーセットを定義します。テキストボックスにユーザー名を入力し、追加 ボタンをクリックしてユーザーセットにユーザーセットを追加します。

図21.14 アクセス制御のページ

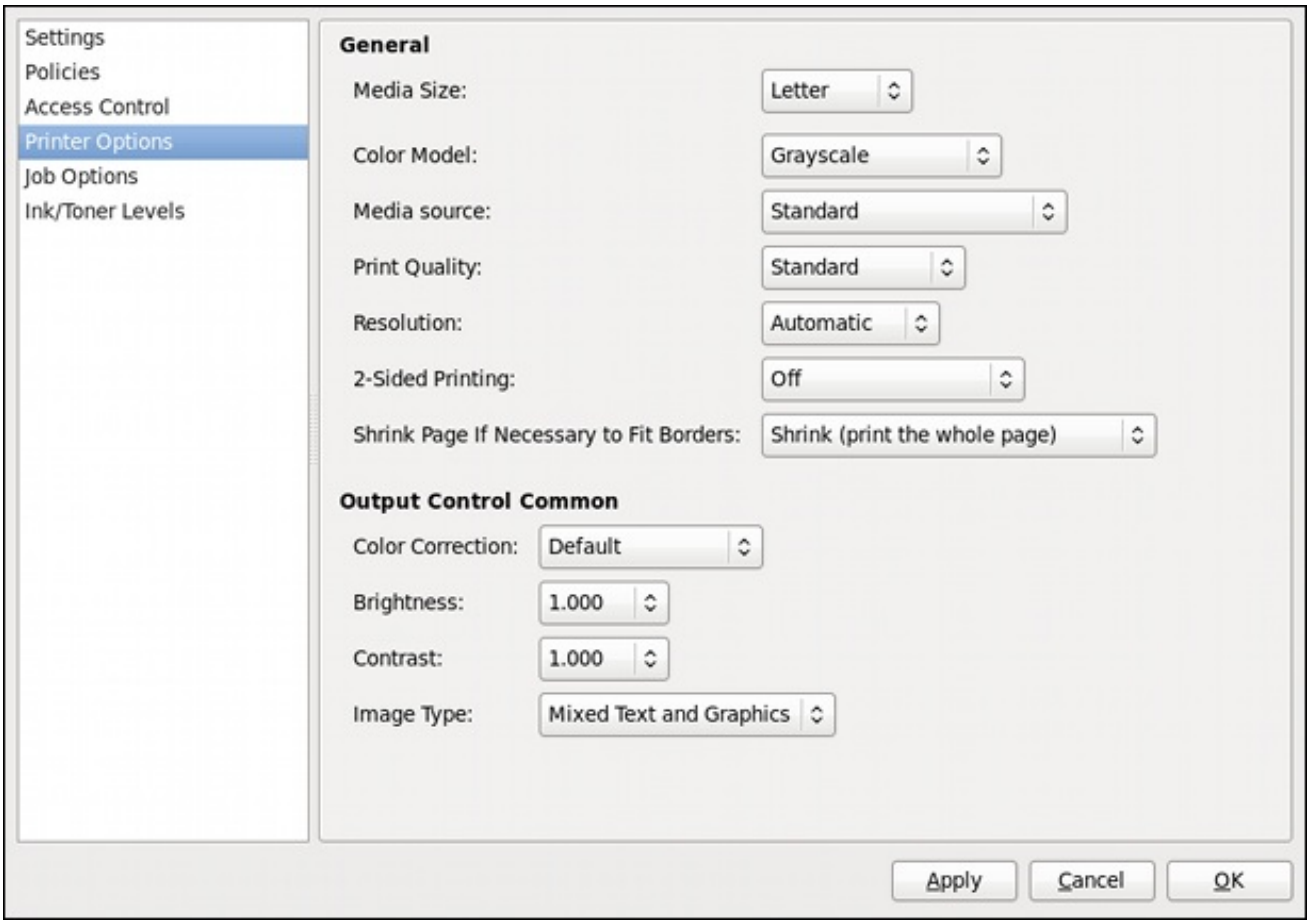


[D]

21.3.10.2.3. プリンターオプションのページ

プリンター オプション ページにはプリンターのメディアや出力の様々な設定オプションがあります。内容はプリンターごとに異なる場合があります。一般的な印刷の用紙、品質、サイズ設定が含まれます。

図21.15 プリンターオプションのページ

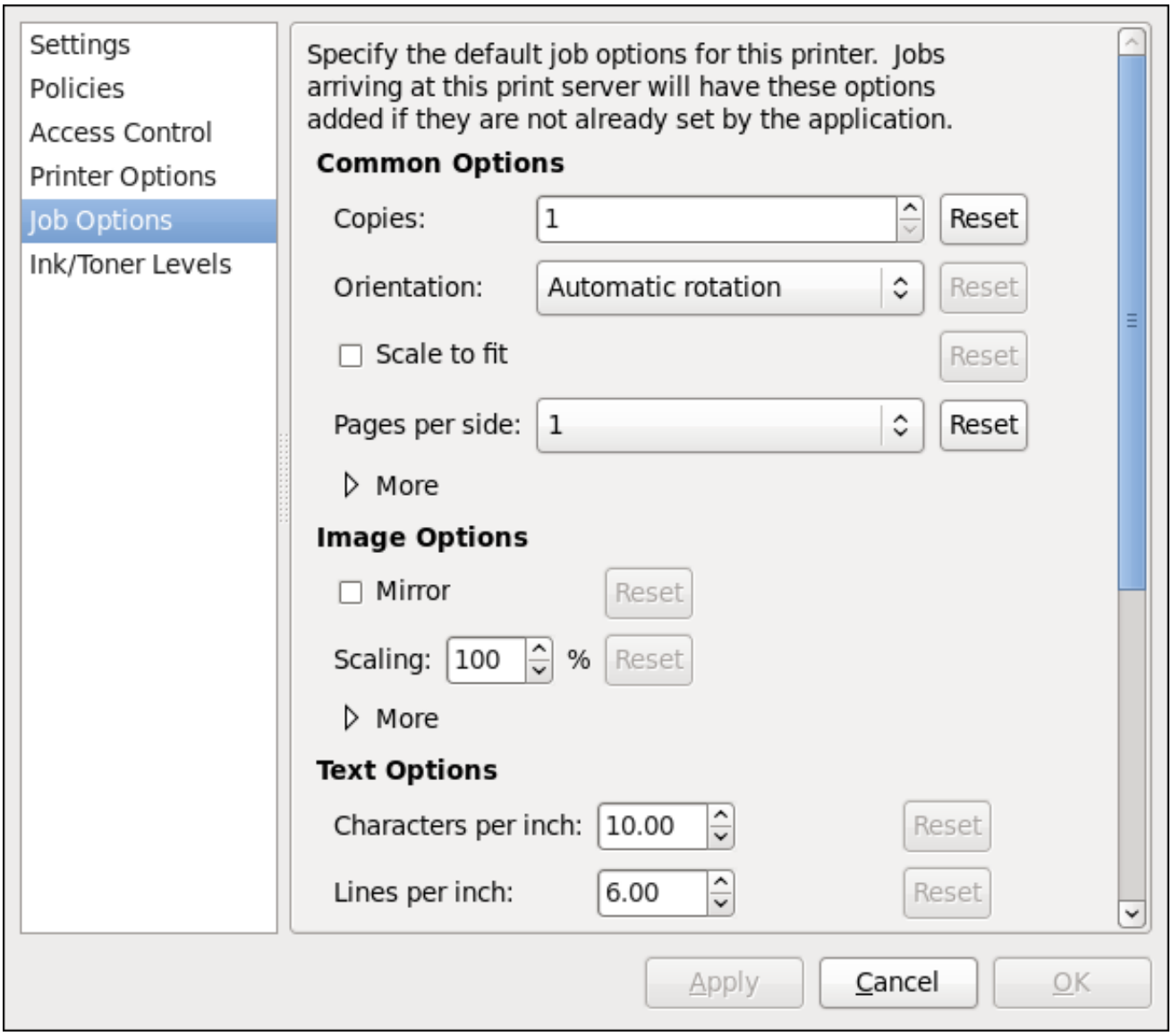


[D]

21.3.10.2.4. ジョブオプションページ

ジョブ オプション ページで、プリンタージョブのオプションの詳細を設定できます。左側の ジョブオプション ラベルをクリックし、ページを表示します。デフォルト設定を編集し、部数、印刷の向き、スライドごとのページ、拡大縮小 (印刷可能領域のサイズを拡大または縮小して、サイズが印刷領域を超えるものを印刷媒体である用紙に合うようにします)、テキストオプションなど、カスタムのジョブオプションを適用します。

図21.16 ジョブオプションページ

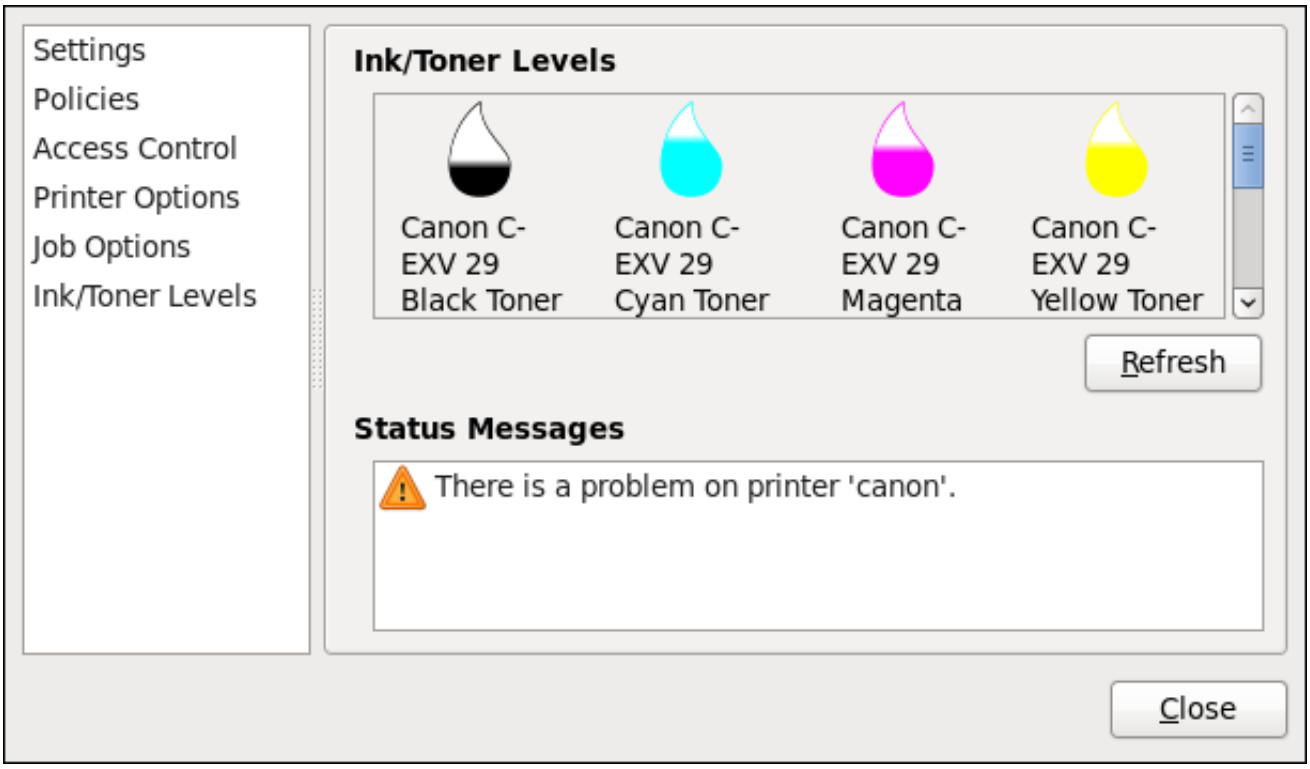


[D]

21.3.10.2.5. Ink/Toner Levels ページ

***Ink/Toner Levels** ページには、トナーの状態の詳細（ある場合）およびプリンターの状態のメッセージが表示されます。左側の **Ink/Toner Levels** ラベルをクリックし、ページが表示されます。*

図21.17 Ink/Toner Levels ページ



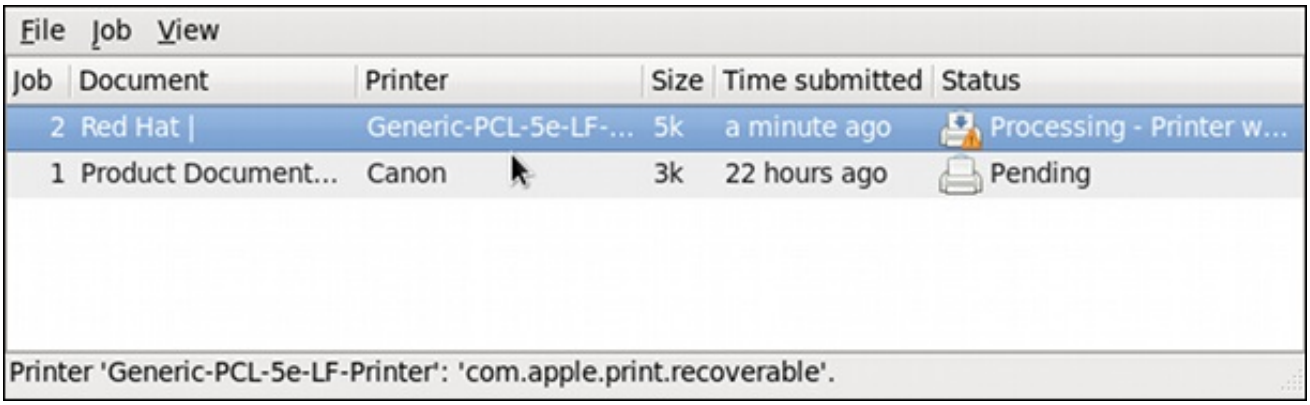
[D]

21.3.10.3. 印刷ジョブの管理

Emacs からのテキストファイルの印刷、GIMP からの画像の印刷など、プリンターデーモンに印刷ジョブを送信すると、印刷ジョブは印刷スプールキューに追加されます。印刷のスプールキューはプリンターに送られた印刷ジョブの一覧で、各印刷要求に関する情報 (印刷要求の状態、ジョブ番号など) を表示します。

印刷プロセス中に、Printer Status アイコンがパネルの Notification Area に表示されます。印刷ジョブのステータスを確認するには、Printer Status をクリックすると、図21.18「GNOME 印刷の状態」 のようなウィンドウが表示されます。

図21.18 GNOME 印刷の状態



[D]

印刷ジョブをキャンセル、保持、リリース、再印刷、認証するには、GNOME Print Status でジョブを選択し、ジョブ メニューでそれぞれのコマンドをクリックします。

シェルプロンプトから印刷スプールの印刷ジョブの一覧を表示するには、lpstat -o コマンドを入力します。最後の数行は以下のようになります。

例21.11 lpstat -o 出力の例

```
$ lpstat -o
Charlie-60          twaugh          1024   Tue 08 Feb 2011 16:42:11 GMT
Aaron-61            twaugh          1024   Tue 08 Feb 2011 16:42:44 GMT
Ben-62              root            1024   Tue 08 Feb 2011 16:45:42 GMT
```

印刷ジョブをキャンセルするには、コマンド lpstat -o コマンドで要求のジョブ番号を見つけ、cancel job number を使用します。たとえば、cancel 60 は 例21.11「lpstat -o 出力の例」 で印刷ジョブをキャンセルします。cancel コマンドでは、他のユーザーが開始された印刷ジョブは キャンセル できません。ただし、cancel -U root job_numberコマンドを実行して、このジョブ の削除を強制できます。このようなキャンセルを防ぐには、プリンターの操作ポリシーを Authenticated に変更し、root 認証を強制します。

シェルプロンプトから直接ファイルを印刷することもできます。たとえば、lp sample.txt コマンドはテキストファイル sample.txt を出力します。印刷フィルターはファイルのタイプを決定し、プリンターが理解できる形式に変換します。

21.3.11. その他のリソース

Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux での印刷に関する詳細は、以下の資料を参照してください。

21.3.11.1. インストールされているドキュメント

man lp

コマンドラインからのファイルの印刷を可能にする lp コマンドの man ページです。

man lpr

コマンドラインからのファイルの印刷を可能にする lpr コマンドの man ページです。

man cancel

印刷キューから印刷ジョブを削除するためのコマンドラインユーティリティーの man ページです。

man mpage

1 枚の用紙に複数ページを印刷するためのコマンドラインユーティリティーの man ページです。

man cupsd

CUPS プリンターデーモンの man ページです。

man cupsd.conf

CUPS プリンターデーモン設定ファイルの man ページです。

man classes.conf

CUPS のクラス設定ファイルの man ページです。

man lpstat

クラス、ジョブ、プリンターのステータス情報を表示する lpstat コマンドの man ページです。

21.3.11.2. 便利な Web サイト

http://www.linuxprinting.org/

『GNU/Linux 印刷には』、Linux での印刷に関する多くの情報が含まれています。

http://www.cups.org/

CUPS に関するドキュメント、FAQ、および newsgroups。

第22章 NTPD を使用した NTP 設定

22.1. NTP の概要

Network Time Protocol (NTP)を使用すると、ネットワークまたはインターネットで共通の参照と同期しているネットワーク化されたコンピューターシステム上のタイムクロックを維持できるように、正確な時間と日付情報を取り除くことができます。世界中の多くの標準機関には原子時計があり、これが参照先として利用可能になっている場合があります。Global Position システムを構成するサテライトには、複数の atomic クロックが含まれるため、時間のシグナルが非常に正確になる可能性があります。この信号は、軍事的な理由で意図的に弱められる場合があります。理想的な状況では、各サイトに独自の参照時計があるサーバーがあり、これがサイト全体のタイムサーバーとして機能します。低周波数のラジオ伝送またはグローバル位置システム (GPS) を介して、日時と日付を取得するデバイスが多数存在します。ただし多くの場合、インターネットに接続され、各地に分散する公開されたアクセス可能なタイムサーバーを使うことができます。これらの NTP サーバーは、「協定世界時」 (UTC)を提供します。これらのタイムサーバーに関する情報は、『www.pool.ntp.org』 を参照してください。

IT では、多くの理由で正確な時間の維持が重要です。たとえばネットワーキングでは、パケットとログのタイムスタンプが正確であることが必要になります。ログはサービスとセキュリティーの問題を調査するために使用されるため、異なるシステム上のタイムスタンプは同期クロックで行われるため、実際の値である必要があります。システムおよびネットワークがますます高速化するにつれ、これに対応してクロックの正確性と精度の必要性も高まっています。国によっては、正確な同期クロックを保持することが法律で定められているところもあります。詳細は、『www.ntp.org』 を参照してください。Linux システムでは、NTP はユーザー空間で実行しているデーモンにより実装されます。Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;LinuxRed Hat Enterprise Linux 6nbsp;6 のデフォルトの NTP デーモンは ntpd です。

ユーザースペースのデーモンは、カーネルで実行しているソフトウェアクロックであるシステムクロックを更新します。Linux は、「リアルタイム」 クロック (RTC) と呼ばれる一般的な埋め込みハードウェアクロックよりも、より優れた解像度を実現するために、ソフトウェアクロックをシステムクロックとして使用します。ハードウェアクロックに関する情報は、man ページの rtc (4) および hwclock(8)を参照してください。システムクロックは、さまざまなクロックソースを使用して時間を維持します。通常、Time Stamp Counter (TSC)が使用されます。TSC は、最後にリセットされた時点からのサイクル数を計測する CPU レジスターです。TSC は、それが最後にリセットされてからのサイクル数をカウントする CPU レジスターです。これは非常に高速で精度が高く、割り込みがありません。RTC が保持する時間は、気温の変化により、1 カ月あたり最大 5 分ごとに実際の時間から離れます。RTC が維持している時間は実際の時間と比べて、温度変化によりひと月で最大 5 分間の誤差を生じます。ntpd がシステムクロックを同期している場合には、カーネルは自動的に RTC を 11 分ごとに更新します。

22.2. NTP STRATA (階層)

NTP サーバーは、時間信号のソースとなる原子時計からの同期距離によって分類されます。サーバーは、上は 1 から下は 15 までの stratum (階層) に分類されていると考えられます。このため、特定

の層に言及する際は、stratum という言葉が使用されます。アトミッククロックはソースである Stratum 0 と呼ばれます。ただし、インターネットでは Stratum 0 パケットが送信されず、すべての stratum 0 アトミッククロックは stratum 1 と呼ばれるサーバーにアタッチされます。これらのサーバーは、Stratum 1としてマークされたパケットを送信します。stratum n とマークされたパケットで同期されるサーバーは、次の下位の stratum に属し、そのパケットを stratum n+1 とマークします。stratum n のマークがついているパケットで同期されるサーバーは、その次に下位の stratum に所属し、パケットを stratum n+1 とマークします。Stratum 16 という名称は、サーバーが現在信頼できるタイムソースと同期していないことを意味します。

デフォルトでは、NTP クライアントはそれよりも下位の stratum にあるシステムのサーバーとして機能することに注意してください。

以下は、NTP Strata（階層）のサマリーです。

Stratum 0:

原子時計と無線および GPS による信号送信

- *GPS (全地球測位システム)*
- *携帯電話システム*
- *低周波無線送信 WWVB (米国コロラド州)、JJY-40 および JJY-60 (日本)、DCF77 (ドイツ)、および MSF (英国)*

これらの信号は専用デバイスで受信可能で、通常は RS-232 で組織全体またはサイト全体のタイムサーバーとして使用されるシステムに接続されます。

Stratum 1:

電波時計、GPS 時計、または原子時計に接続しているコンピューター

Stratum 2

stratum 1 から読み取り、下位の strata に提供

Stratum 3:

stratum 2 から読み取り、下位の strata に提供

Stratum n+1:

stratum n から読み取り、下位の strata に提供

Stratum 15:

stratum 14 から読み取り、これが最下位の stratum になります。

このプロセスは有効な最下位の Stratum 15 まで続きます。Stratum 16 というラベルは、非同期状態を示します。

22.3. NTP の概要

Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux で使用される NTP のバージョンは、RFC 1305 Network Time Protocol(Version 3)Specification, Implementation and Analysis and『RFC 5905 Network Time Protocol Version』4: Protocol and Algorithms Specification で説明されています。

NTP を実装すると、10 秒以下の正確性が達成できます。インターネット上では、数十ミリ秒の正確性の維持は普通のことです。ローカルエリアネットワーク (LAN) 上では、1 ミリ秒の正確性は理想的な条件下では可能です。これは、クロックのドリフトを考慮して修正され、以前の、シンプルな時間プロトコルシステムで行われていないためです。64 ビットのタイムスタンプを使用することにより、233 ピコ秒の解像度が得られます。ここではタイムスタンプの最初の 32 ビットが秒に使われ、次の 32 ビットが 1 秒未満に使われます。

NTP は、1900 年 1 月 1 日の GMT 午前 0 分からの経過時間を秒数で表します。秒のカウントには 32 ビットが使用されます。これは、時間が 2036 で「ロールオーバー」されることを意味します。ただし、NTP はタイムスタンプ間の差異で機能するため、タイムプロトコルの他の実装と同じレベルの問題はありません。誤差が 68 年以内のハードウェアクロックが起動時に利用可能であれば、NTP は正確に現在の日時を解釈します。NTP4 仕様は、「Era Number」および「Era Offset」を提供します。このセットを使用すると、68 年を超える時間の長さを扱う場合にソフトウェアをより堅牢にすることができます。これは Unix 年の 2038 年問題と混同しないようにしてください。

NTP プロトコルは、正確性を高めるために追加情報を提供します。4 つのタイムスタンプを使うことで、往復時間とサーバー応答時間の計算が可能になります。NTP クライアントとしての役割でシステムが参照時間サーバーと同期するために、パケットは「元のタイムスタンプ」で送信されます。パケットが到達すると、タイムサーバーは「受信タイムスタンプ」を追加します。日時情報の要求を処理した後、パケットの返信前に「転送タイムスタンプ」が追加されます。返信パケットが NTP クライアントに届くと、「受信タイムスタンプ」が生成されます。クライアントはこれで往復時間が計算で

き、処理時間を差し引くことで実際の移動時間が導き出されます。送信時間と受信時間が同じだと仮定すると、NTP データを受信する際の 1 回の移動の遅延が計算されます。正式な NTP アルゴリズムは、ここで示されているものよりもはるかに複雑です。

時間情報を含むパケットは、受信後にただちに処理されるのではなく、最初に検証され、その後にいくつかの他の時間サンプルと一緒に処理されて、時間を予想します。その後、システムクロックを比較して時間オフセットを決定し、システムクロックの時間と ntpd が時間と決定する時間の違いを決定します。システムクロックは、使用されているカウンターの頻度を変更することで、このオフセットを減らすために、最大で 0.5ms のレートで調整されます。この方法を使用してクロックを 1 秒で調整するには、最低 2000 秒かかります。この方法でクロックを 1 秒調整するには、少なくとも 2000 秒かかります。クロックの時間オフセットが 128ms を超える場合（デフォルト設定）、ntpd はクロックの転送または後方を 「ステップ」 できます。システムののタイムオフセットが 1000 ミリ秒を超える場合は、ユーザーまたはインストールスクリプトで手動の調整を行う必要があります。2章日付と時刻の設定を参照してください。-g オプションを ntpd コマンドに指定すると（デフォルトで使用）、システム起動時のオフセットは修正されますが、通常の操作中に修正されるオフセットは最大 1000 秒までです。

時間を遅らせると、失敗したりエラーになるソフトウェアもいくつかあります。時間のステップ変更の影響を受けるシステムでは、しきい値を 128ms ではなく 600s に変更できます（ -g オプションとは関係しません）。-x オプションを使用してステップの制限を 0.128s から 600s に増やすと、クロック制御に異なる方法が使用される必要があるため、マイナス面があります。カーネルのクロック規範が無効になり、クロックの正確性にマイナスの影響が出る可能性があります。-x オプションは、/etc/sysconfig/ntpd 設定ファイルに追加できます。

22.4. 誤差ファイルの概要

誤差ファイルは、通常の周波数で稼働しているシステムクロックと、UTC と同期し続けるために必要な周波数との間の周波数オフセットを保存するために使われます。誤差ファイルに値がある場合は、システム起動時に読み取られ、クロックソースの修正に使われます。誤差ファイルを使用すると、安定的かつ正確な時間の達成に必要な時間が短縮されます。この値は、1 時間ごとに ntpd が計算して、誤差ファイルは置換されます。誤差ファイルは更新されるのではなく置換されるので、ntpd が書き込みパーミッションのあるディレクトリーに格納される必要があります。

22.5. UTC、タイムゾーン、および DST

NTP は完全に UTC（協定世界時）であるため、タイムゾーンと DST(DST)はシステムがローカルで適用します。/etc/localtime ファイルは、/usr/share/zoneinfo のコピー、または へのシンボリックリンクです。RTC は、/etc/adjtime の 3 行目に指定してあるとおり、ローカルタイムまたは UTC になります。これは、RTC クロックの設定方法を示す LOCAL または UTC のいずれかです。system-config-date グラフィカル設定ツールの System Clock Uses UTC チェックボックスを使用して、この設定を簡単に変更できます。そのツールの使用方法は、2章日付と時刻の設定 を参照してください。夏時間を変更する際には、各種の問題を避けるために RTC を UTC で実行することが推奨されます。

ntpd の操作の詳細は、ntpd(8)の man ページを参照してください。リソースセクションでは、役に立つ情報ソースを紹介しています。「その他のリソース」を参照してください。

22.6. NTP の認証オプション

NTPv4 は、公開非対称暗号をベースとしながら対称鍵暗号にも対応している Autokey Security Architecture のサポートが追加されました。Autokey セキュリティーアーキテクチャーについては、『RFC 5906 Network Time Protocol Version 4: Autokey Specification』で説明されています。ntpd の認証オプションとコマンドについては、ntp_auth(5) の man ページで説明しています。

ネットワーク上の攻撃者は、不正確な時間情報のある NTP パケットを送信することで、サービスを中断できます。NTP サーバーのパブリックプールを使用するシステムでは、/etc/ntp.conf のパブリック NTP サーバー一覧に 4 つ以上の NTP サーバーを記載することで、このリスクが軽減されます。1 つのタイムソースのみが危険にさらされるか、またはなりすましを受けた場合、ntpd はそのソースを無視します。リスク評価を実行し、不正確な時間がアプリケーションおよび組織に及ぼす影響を検討してください。内部のタイムソースがある場合は、NTP パケットが配布されるネットワークを保護する手段を検討してください。リスク評価を実行して、リスクを許容でき、アプリケーションへの影響が最小限であると判断した場合は、認証を使わないことを選択することもできます。

ブロードキャストおよびマルチキャストの両モードは、デフォルトで認証を必要とします。ネットワークが信頼できると判断した場合は、ntp.conf ファイルの disable auth ディレクティブを使用して認証を無効にできます。別の方法では、SHA1 または MD5 シンメトリックキーを使って認証を設定するか、Autokey スキームを使用して公開 (非対称) キー暗号法で認証を設定する必要があります。非対称暗号法の Autokey スキームは、ntp_auth(8)man ページで、キーの生成については ntp-keygen(8) で説明されています。対称鍵暗号を実装するには、キー オプションの説明を 「鍵を使った対称認証の設定」を参照してください。

22.7. 仮想マシン上での時間管理

仮想マシンは実際のハードウェアクロックにアクセスできず、仮想クロックの安定性はホストシステムの作業量に依存することから、十分な安定性がありません。このため、使用する仮想化アプリケーションが準仮想化クロックを提供する必要があります。KVM を搭載した Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux では、デフォルトのクロックソースは kvm-clock です。『仮想 『化ホスト設定および『ゲストインストールガイド』の「KVM ゲストタイミング管理』」の章を参照』してください。

22.8. うるう秒の概要

グリニッジ標準時(GMT)は、イレイジャー日の測定から導き出されました。これは、Earth のローテーションに依存します。原子時計が最初に作成された際に、より正確な時間の定義が可能になりました。1958 年に国際原子時 (TAI) がより正確で安定的な原子時計を基に導入されました。さらに正確な天文時である世界時 1 (UT1) も導入され、GMT に代わるものとなりました。実際、原子時計は地球の自転よりもはるかに安定しているので、この 2 つの時間の差異が広がり始めました。これが理由で、現実的な方法として UTC が導入されました。これは UT1 の 1 秒以内に維持されますが、簡単な調整を何度も行うことを避けるため、うるう秒 の概念を導入し、管理可能な方法で差異を調整する決定されました。UT1 と UTC の差異は、0.5 秒以上になるまで監視されます。1 秒進めるまたは遅らす調整が必要と

みなされた場合にのみ、これが実行されます。地理の自転速度が不安定であるため、調整の必要性は今後予測できません。調整を行うタイミングの決定は、「『International Earth Rotation and Reference Systems Service(IERS)』」により行われます。ただし、NTP は保留中のうるう秒についての情報を転送し、これらを自動的に適用するので、この発表が重要となるのは、Stratum 1 サーバーの管理者にのみ重要です。

22.9. NTPD 設定ファイルについて

ntpd デーモンは、システム起動時またはサービスの再起動時に設定ファイルを読み取ります。このファイルのデフォルトの場所は /etc/ntp.conf で、以下のコマンド入力して確認することができます。

```
~]$ less /etc/ntp.conf
```

設定コマンドについては、本章の後半で簡単に説明しています。また、ntp.conf(5) man ページで詳細に詳細を説明します。「NTP の設定」

ここでは、以下でデフォルトの設定ファイルを簡単に説明します。

driftfile エントリー

drift ファイルへのパスが指定されていると、Red Hat Enterprise Linuxnbsp;Hat Enterprise Red Hat Enterprise Linuxnbsp;Linux のデフォルトエントリーは以下のようになります。

```
driftfile /var/lib/ntp/drift
```

これを変更する場合は、ディレクトリーが ntpd で書き込み可能となっていることを確認してください。ファイルには、システムもしくはサービス起動時に毎回システムクロックの周波数を調整する値が含まれます。詳細情報は、誤差ファイルの概要 を参照してください。

アクセス制御エントリー

次の行は、デフォルトのアクセス制御制限を設定します。

```
restrict default kod nomodify notrap nopeer noquery
restrict -6 default kod nomodify notrap nopeer noquery
```

kod オプションは、不要なクエリーを減らすために「Kiss-o'-death」パケットを送信することを意味します。nomodify オプションは、設定に変更が加えられないようにします。notrap オプションは、ntpdc 制御メッセージプロトコルトラップを防ぎます。nopeer オプションは、ピア関連付けが形成されないようにします。noquery オプションは、ntpq および ntpdc クエリーの応答を防ぎますが、タイムクエリーは除外されます。-6 オプションは、IPv6 アドレスの前に必要になります。

127.0.0.0/8 範囲内のアドレスは、さまざまなプロセスまたはアプリケーションで必要になることがあります。上記の "restrict default" 行は明示的に許可されていないすべてのものへのアクセスを禁止するため、IPv4 および IPv6 の標準ループバックアドレスへのアクセスは以下の行で許可されます。

```
# the administrative functions.
restrict 127.0.0.1
restrict -6 ::1
```

別のアプリケーションで特に必要とされる場合は、アドレスはすぐ下に追加できます。-6 オプションは、IPv6 アドレスの前に必要になります。

ローカルネットワーク上のホストは、上記の "restrict default" 行のために許可されません。これを変更するには、たとえば 192.0.2.0/24 ネットワークからのホストが時間および統計情報のみをクエリーできるようにするには、以下の形式の行が必要になります。

```
restrict 192.0.2.0 mask 255.255.255.0 nomodify notrap nopeer
```

特定のホスト（例： 192.0.2.250/32 ）からの無制限のアクセスを許可するには、以下の形式の行が必要になります。

```
restrict 192.0.2.250
```

指定がない場合は、255.255.255.255 のマスクが適用されます。

制限コマンドは、ntp_acc(5) man ページで説明されています。

公開サーバーエントリー

デフォルトでは、Red Hat Enterprise 6.5 では ntp.conf ファイルに 4 つの公開サーバーエントリーが含まれます。

```
server 0.rhel.pool.ntp.org iburst
server 1.rhel.pool.ntp.org iburst
server 2.rhel.pool.ntp.org iburst
server 3.rhel.pool.ntp.org iburst
```

以前のマイナーリリースからアップグレードし、/etc/ntp.conf ファイルを変更した場合には、Red Hat Enterprise Linux 6.5 へのアップグレードで新しいファイル /etc/ntp.conf.rpmnew が作成され、既存の /etc/ntp.conf ファイルは変更されません。

ブロードキャストマルチキャストサーバーエントリー

デフォルトでは、ntp.conf ファイルにはコメントアウトされた例がいくつか含まれています。これらは見ればすぐにわかります。特定のコマンド 「NTP の設定」 の説明を参照してください。必要に応じて、例のすぐ下にコマンドを追加します。

注記

DHCP クライアントプログラムである dhclient が DHCP サーバーから NTP サーバーの一覧を受信したら、ntp.conf に追加され、サービスを再起動します。この機能を無効にするには、PEERNTP=no を /etc/sysconfig/network に追加します。

22.10. NTPD SYSCONFIG ファイルの概要

このファイルは、サービス起動時に ntpd init スクリプトが読み取ります。デフォルトのコンテンツは以下のとおりです。

```
# Drop root to id 'ntp:ntp' by default.
OPTIONS="-u ntp:ntp -p /var/run/ntpd.pid -g"
```

-g オプションを指定すると、ntpd は 1000s のオフセット制限を無視し、オフセットが 1000 秒よりも大きい場合でも、システムの起動時にのみ同期を試みます。このオプションを指定しないと、時間オフセットが 1000 秒を超えると、ntpd は終了します。また、-g オプションを使用しても、サービスが再起動し、オフセットが 1000 秒を超えると、システムの起動後に終了します。

-p オプションは pid file へのパスを設定し、-u はデーモンが root 権限をドロップするユーザーおよびグループを設定します。

22.11. NTP デーモンのインストールを確認する

ntpd がインストールされていることを確認するには、root で以下のコマンドを発行します。

```
~]# yum install ntp
```

NTP デーモンまたはサービス ntpd で実装されます。これは、ntp パッケージに含まれています。

22.12. NTP デーモン (NTPD) のインストール

ntpd をインストールするには、root で以下のコマンドを発行します。

```
~]# yum install ntp
```

デフォルトのインストールディレクトリーは /usr/sbin/ です。

22.13. NTP ステータスの確認

ntpd がシステム起動時に実行されるよう設定されているかどうかを確認するには、以下のコマンドを発行します。

```
~]$ chkconfig --list ntpd
ntpd            0:off 1:off 2:on 3:on 4:on 5:on 6:off
```

デフォルトでは、ntpd のインストール時に、システム起動ごとに起動するように設定されています。

ntpd が実行中かどうかを確認するには、以下のコマンドを実行します。

```
~]$ ntpq -p
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================

+clock.util.phx2 .CDMA.          1 u  111  128  377  175.495    3.076   2.250
*clock02.util.ph .CDMA.          1 u   69  128  377  175.357    7.641   3.671
 ms21.snowflakeh .STEP.         16 u    - 1024    0    0.000    0.000   0.000
 rs11.lvs.iif.hu .STEP.         16 u    - 1024    0    0.000    0.000   0.000
 2001:470:28:bde .STEP.         16 u    - 1024    0    0.000    0.000   0.000
```

このコマンドは、接続されたタイムサーバーを一覧表示し、最後にポーリングされた日時と応答の安定性を示す情報を表示します。列見出しは以下のようになります。

- リモートおよび refid: リモート NTP サーバーおよびその NTP サーバー
- st: stratum of server

- T: サーバーの種類（ローカル、ユニキャスト、マルチキャスト、またはブロードキャスト）

- poll: サーバーをクエリーする頻度（秒単位）

- when: 最後のポーリングからの経過時間（秒単位）

- Reach: 8 進数のビットマスク(left-shifted); 377 = 11111111 = all recent queries was successful; 257 = 10101111 = 4 most recent was successful, 5 and 7 failed

- 遅延： ネットワークラウンドトリップタイム（ミリ秒単位）

- オフセット： ローカルクロックとリモートクロックの違い（ミリ秒単位）

- ジッター： サーバーからの連続した時間値の差異（高ジッターは不安定なクロックによるもの、またはネットワークパフォーマンスが低下する可能性あり）

ntpd から簡単なステータスレポートを取得するには、以下のコマンドを発行します。

```
~]$ ntpstat
unsynchronised
  time server re-starting
   polling server every 64 s
```

```
~]$ ntpstat
synchronised to NTP server (10.5.26.10) at stratum 2
   time correct to within 52 ms
   polling server every 1024 s
```

22.14. 着信 NTP パケットを許可するファイアウォールの設定

NTP トラフィックはポート 123 上の UDP パケットで構成されており、NTP が機能するにはネットワークおよびホストベースのファイアウォール通過が許可されている必要があります。

22.14.1. グラフィカルツールを使用したファイアウォールの設定

NTP がファイアウォールを通過できるようにするには、グラフィカルツール system-config-firewall を使用して root で以下のコマンドを実行します。

```
~]# system-config-firewall
```

Firewall Configuration ウィンドウが開きます。左側のリストから Other Ports を選択します。追加 をクリックします。Port and Protocol ウィンドウが開きます。ポート番号のいずれかをクリックし、123 の入力を開始します。UDP をプロトコルとして使用して「ポート 123」 エントリーを選択します。OK をクリックします。Port and Protocol ウインドウを閉じます。Firewall Configuration ウィンドウで Apply をクリックし、変更を適用します。アクションを確認するようダイアログボックスがポップアップ表示され、Yes をクリックします。Yes をクリックすると、既存のセッションがすべて終了することに注意してください。

22.14.2. コマンドラインを使用したファイアウォールの設定

NTP がコマンドラインを使用してファイアウォールを通過できるようにするには、root で以下のコマンドを実行します。

```
~]# lokkit --port=123:udp --update
```

これにより、--disabled オプションで無効にされていない限り、ファイアウォールが再起動されることに注意してください。アクティブな接続は終了し、開始マシンでタイムアウトします。

管理ツールを使用して複数のインストールのために設定ファイルを準備する際には、ファイアウォール設定ファイルを直接編集すると便利です。設定ファイルの間違いがある場合は、予期しない影響が発生し、エラーが発生する可能性があり、ファイアウォール設定が適用されないことに注意してください。したがって、編集後に /etc/sysconfig/system-config-firewall ファイルの詳細を確認します。

NTP がファイアウォールを通過できるようにするには、設定ファイルを編集して root ユーザーになり、以下の行を /etc/sysconfig/system-config-firewall に追加します。

```
--port=123:udp
```

この変更は、ファイアウォールが再読み込みされるか、システムが再起動するまで反映されないことに注意してください。

22.14.2.1. コマンドラインを使用した着信 NTP のネットワークアクセスの確認

ファイアウォールがコマンドラインを使用してクライアントの着信 NTP トラフィックを許可するように設定されているかどうかを確認するには、root で以下のコマンドを実行します。

```
~]# less /etc/sysconfig/system-config-firewall
# Configuration file for system-config-firewall

--enabled
--service=ssh
```

この例では、デフォルトのインストールでファイアウォールは有効になっていますが、NTP は通過できません。有効にすると、以下の行が上記の行に加えて出力として表示されます。

```
--port=123:udp
```

ファイアウォールが現在クライアントの着信 NTP トラフィックを許可しているかどうかを確認するには、root で以下のコマンドを実行します。

```
~]# iptables -L -n | grep 'udp.*123'
ACCEPT     udp  --  0.0.0.0/0           0.0.0.0/0           state NEW udp dpt:123
```

22.15. NTPDATE サーバーの設定

ntpdate サービスの目的は、システムの起動時にクロックを設定することです。ntpdate が正確な時間を確保し、クロックでジャンプが生じないようにすることができます。ntpdate の使用と step-tickers の一覧は非推奨とみなされるため、Red Hat Enterprise Linux 6 はデフォルトで -g オプションを ntpd コマンドに指定しますが、ntpdate ではなく、この -g オプションが使用されます。ただし、-g オプションを指定すると、ntpd は 1000s のオフセット制限を無視し、時間を同期しようとします。他のプログラムやサービスの開始時に時間が正しいことを保証する訳ではありません。そのため、ntpd サービスは、ntpd が無効であるか、または正しい時間で開始する必要があるサービスがあり、クロックでジャンプが生じないようにする場合に便利です。

ntpdate サービスがシステム起動時に実行されるようになっていることを確認するには、以下のコマンドを発行します。

```
~]$ chkconfig --list ntpdate
ntpdate         0:off 1:off 2:on 3:on 4:on 5:on 6:off
```

システム起動時にサービスが実行するようにするには、root で以下のコマンドを発行します。

```
~]# chkconfig ntpdate on
```

ntpdate サーバーを設定するには、テキストエディターを root として実行し、/etc/ntp/step-tickers を編集して以下のように 1 つ以上のホスト名を含めます。

```
clock1.example.com
clock2.example.com
```

ntpdate は、システム起動時に日付情報を取得するためにこのファイルを 1 回使用するだけなので、記載されているサーバー数は重要ではありません。内部のタイムサーバーがある場合は、そのホスト名を 1 行目に使います。2 行目に追加のホストをバックアップとしておくのがよいでしょう。バックアップサーバーにどれを選ぶか、また 2 番目のホストを内部または外部とするかは、リスク評価によります。たとえば、1 番目のサーバーに影響する問題が 2 番目のサーバーにも影響する可能性はどの程度か。1 番目のサーバーにアクセスできなくなるネットワーク障害時に、より接続性が高いのは外部サーバーかそれとも内部サーバーか、といった点を考慮します。

ntpdate サービスには、システム起動時に使用される NTP サーバーの一覧が含まれる必要があるファイルがあります。誤ったチェッカー（正しいタイムソース）の可能性を減らすために、時間オフセット計算の質に影響を与えるために、最後の 4 つのサーバーを一覧表示することが推奨されます。「」ただし、一般にアクセス可能なタイムソースはほとんど正しくありません。

22.16. NTP の設定

NTP サービスのデフォルト設定を変更するには、root でテキストエディターを使用して /etc/ntp.conf ファイルを編集します。このファイルは ntpd とともにインストールされ、Red Hat プールからのタイムサーバーを使用するデフォルト設定になっています。ntp.conf(5) の man ページでは、アクセスおよびレート制限コマンドとは別に、設定ファイルで使用可能なコマンドオプションが説明されています。アクセスおよびレート制限コマンドは、ntp_acc(5) man ページで説明されています。

22.16.1. NTP サービスへのアクセス制御の設定

システムで実行している NTP サービスへのアクセスを制限または制御するには、ntp.conf ファイルの restrict コマンドを利用します。コメントアウトされた例は以下のとおりです。

```
# Hosts on local network are less restricted.
#restrict 192.168.1.0 mask 255.255.255.0 nomodify notrap
```

restrict コマンドは以下の形式になります。

```
restrict address mask option
```

ここでの address と mask は、制限を適用する IP アドレスを指定します。オプション は以下のいずれかになります。

- ignore: ntpq および ntpdc クエリーを含むすべてのパケットは無視されます。
- Ko D: 「Kiss-o'-death」 パケットが送信され、不要なクエリーが少なくなります。
- limited: パケットがレート制限のデフォルト値または discard コマンドで指定された値に違反する場合、タイムサービス要求に応答しません。ntpq および ntpdc クエリーは影響を受けません。discard コマンドおよびデフォルト値に関する詳細情報は、「NTP サービスへのレート制限アクセスの設定」 を参照してください。
- lowpriotrap: 一致するホストがトラップを低い優先度に設定します。
- nomodify: 設定に変更を加えられないようにします。
- noquery: ntpq および ntpdc クエリーに応答しないようにしますが、タイムクエリーは除外されます。
- nopeer: ピア関連付けが形成されないようにします。
- noserve: ntpq および ntpdc クエリー以外のパケットをすべて拒否します。

- *notrap: ntpdc 制御メッセージプロトコルトラップを防ぎます。*

- *notrust: 暗号法で認証されないパケットを拒否します。*

- *ntpport: 発信元ポートが標準の NTP UDP ポート 123 の場合、一致アルゴリズムが制限のみを適用するように修正します。*

- *version: 現在の NTP バージョンに一致しないパケットを拒否します。*

レート制限アクセスがクエリーに対してまったく応答しないように設定するには、各 restrict コマンドに limited オプションを指定する必要があります。ntpd が KoD パケットで応答する必要がある場合は、restrict コマンドに limited オプションと kod オプションの両方が必要です。

ntpq および ntpdc クエリーは増幅攻撃に使用できます（詳細は 『CVE-2013-5211』 を参照してください）。アクセスを公開しているシステムでは、restrict default コマンドから noquery オプションを削除しないでください。

22.16.2. NTP サービスへのレート制限アクセスの設定

システムで実行している NTP サービスへのレート制限アクセスを有効にするには、「NTP サービスへのアクセス制御の設定」 で説明されているように、limited オプションを restrict コマンドに追加します。デフォルトの discard パラメーターを使用したくない場合は、以下で説明するように discard コマンドも使用できます。

discard コマンドは以下の形式になります。

```
discard [average value] [minimum value] [monitor value]
```

- *average: 許可される最小限の平均パケット間隔を指定し、log_2 秒の引数を取ります。デフォルト値は 3 です（2^3 は 8 秒に相当）。*

- *minimum: 許可される最小限のパケット間隔を指定し、log_2 秒の引数を取ります。デフォ*

ルト値は 1 です（2^1 は 2 秒に相当します）。

- monitor: 許可されるレート制限を超えた場合のパケットの discard の確率を指定します。デフォルト値は、3000 秒です。このオプションは、1 秒あたり 1000 以上のリクエストを受信するサーバー用に用意されています。

discard コマンドの例を以下に示します。

```
discard average 4
```

```
discard average 4 minimum 2
```

22.16.3. ピアアドレスの追加

ピアのアドレス、つまり同一 stratum の NTP サービスを実行しているサーバーのアドレスを追加するには、ntp.conf ファイルの peer コマンドを利用します。

peer コマンドは以下の形式になります。

```
peer address
```

ここでの address は、IP ユニキャストアドレスまたは DNS の解決可能な名前になります。アドレスは、同一 stratum のメンバーに既知のシステムのものである必要があります。ピアは、相互に異なる時間ソースを少なくとも 1 つ持っている必要があります。ピアは通常、同一管理制御下にあるシステム群です。

22.16.4. サーバーアドレスの追加

サーバーのアドレス、つまり上位 stratum の NTP サービスを実行しているサーバーのアドレスを追加するには、ntp.conf ファイルの server コマンドを利用します。

server コマンドは以下の形式になります。

```
server address
```

ここでの address は、IP ユニキャストアドレスまたは DNS の解決可能な名前になります。パケッ

トの送信元となるリモートリファレンスサーバーまたはローカルリファレンスクロックのアドレスになります。

22.16.5. ブロードキャストまたはマルチキャストサーバーアドレスの追加

送信用のブロードキャストまたはマルチキャストアドレス、つまり NTP パケットをブロードキャストまたはマルチキャストするアドレスを追加するには、ntp.conf ファイルの broadcast コマンドを利用します。

ブロードキャストおよびマルチキャストの両モードは、デフォルトで認証を必要とします。「NTP の認証オプション」を参照してください。

broadcast コマンドは以下の形式になります。

```
broadcast address
```

ここでの address は、パケットの送信先となる IP ブロードキャストまたはマルチキャストアドレスになります。

このコマンドは、システムが NTP ブロードキャストサーバーとして機能するように設定します。使用するアドレスは、ブロードキャストかマルチキャストアドレスである必要があります。ブロードキャストアドレスは、IPv4 アドレス 255.255.255.255 を意味します。デフォルトでは、ルーターはブロードキャストメッセージを渡しません。マルチキャストアドレスは、IPv4 Class D アドレスまたは IPv6 アドレスになります。IANA には、IPv4 マルチキャストアドレス 224.0.1.1 と IPv6 アドレス FF05::101（サイトローカル）が NTP に割り当てられています。『RFC 2365 Administratively Scoped IP Multicast の説明にあるように、管理』者が範囲指定した IPv4 マルチキャストアドレスを使用することもできます。

22.16.6. Manycast クライアントアドレスの追加

manycast クライアントアドレスを追加するには、つまり NTP サーバーの検出に使用するマルチキャストアドレスを設定するには、ntp.conf ファイルの manycastclient コマンドを利用します。

manycastclient コマンドは以下の形式になります。

```
manycastclient address
```

ここでの address は IP マルチキャストアドレスで、ここからパケットが受信されます。クライアントはこのアドレスにリクエストを送信し、応答から最善のサーバーを選んで他を無視します。その

後、NTP 通信は発見された NTP サーバーが ntp.conf に記載されているかのようにユニキャスト関連付けを使用します。

このコマンドは、システムが NTP クライアントのように動作するように設定します。システムは、同時にクライアントとサーバーの両方になることができます。

22.16.7. ブロードキャストクライアントアドレスの追加

ブロードキャストクライアントのアドレスを追加するには、つまりブロードキャスト NTP パケット用にブロードキャストアドレスを監視するように設定するには、ntp.conf ファイルの broadcastclient コマンドを利用します。

broadcastclient コマンドは以下の形式になります。

```
broadcastclient
```

ブロードキャストメッセージを受信可能にします。デフォルトで認証を必要とします。「NTP の認証オプション」を参照してください。

このコマンドは、システムが NTP クライアントのように動作するように設定します。システムは、同時にクライアントとサーバーの両方になることができます。

22.16.8. Manycast サーバーアドレスの追加

manycast サーバーのアドレスを追加する、つまり NTP パケットをマルチキャストすることでクライアントがサーバーを発見できるようにするアドレスを設定するには、ntp.conf ファイルの manycastserver コマンドを利用します。

manycastserver コマンドは以下の形式になります。

```
manycastserver address
```

マルチキャストメッセージの送信ができるようになります。ここでの address は、マルチキャスト送信先のアドレスになります。これは認証と合わせて使用して、サービス中断を防ぎます。

このコマンドは、システムが NTP サーバーとして機能するように設定します。システムは、同時にクライアントとサーバーの両方になることができます。

22.16.9. マルチキャストクライアントアドレスの追加

マルチキャストクライアントのアドレスを追加するには、つまりマルチキャスト NTP パケット用にマルチキャストアドレスを監視するように設定するには、ntp.conf ファイルの multicastclient コマンドを利用します。

multicastclient コマンドは以下の形式になります。

```
multicastclient address
```

マルチキャストメッセージの受信ができるようになります。ここでの address は、サブスクライブするアドレスになります。これは認証と合わせて使用して、サービス中断を防ぎます。

このコマンドは、システムが NTP クライアントのように動作するように設定します。システムは、同時にクライアントとサーバーの両方になることができます。

22.16.10. Burst オプションの設定

パブリックサーバーに burst オプションを使用することは、不正使用とみなされます。公開 NTP サーバーでは、このオプションを使用しないでください。このオプションは、組織内のアプリケーションにのみ使用するようにしてください。

時間オフセットの統計情報の平均的な品質を向上させるには、サーバーコマンドの最後に以下のオプションを追加します。

```
burst
```

サーバーが反応している場合は、ポーリングの間隔ごとにシステムが通常の 1 パケットではなく、最大 8 パケットのバーストを送信します。server コマンドを使うと、時間オフセット計算の平均的な質が向上します。

22.16.11. iburst オプションの設定

初回同期にかかる時間を改善するには、サーバーコマンドの最後に以下のオプションを追加します。

iburst

サーバーに到達できない場合は、1 パケットではなく、8 パケットのバーストが送信されます。パケットの間隔は通常は 2 秒間隔ですが、1 番目と 2 番目のパケット間の間隔は、モデムまたは ISDN 呼び出しの完了に必要な追加の時間を許可できるように calldelay コマンドを使用して変更することができます。server コマンドと使用すると、初回の同期にかかる時間が短縮されます。Red Hat Enterprise Linux 6.5 では、これは設定ファイルのデフォルトオプションになります。

22.16.12. 鍵を使った対称認証の設定

鍵を使った対称認証を設定するには、サーバーまたはピアコマンドの最後に以下のオプションを追加します。

key number

ここでの number は、1 から 65534 までの数字になります。このオプションを使用すると、パケットで メッセージ認証コード (MAC)を使用できます。このオプションは、peer、server、broadcast、および manycastclient コマンドで使用します。

このオプションは、以下のように /etc/ntp.conf ファイルで使用できます。

```
server 192.168.1.1 key 10
broadcast 192.168.1.255 key 20
manycastclient 239.255.254.254 key 30
```

「NTP の認証オプション」 も参照してください。

22.16.13. ポーリング間隔の設定

デフォルトのポーリング間隔を変更するには、サーバーまたはピアコマンドの最後に以下のオプションを追加します。

minpoll value and maxpoll value

デフォルトのポーリング間隔を変更するオプション。ここでは、秒単位の間隔を 2 の 値 の累乗で計算します。つまり、間隔は $\log_2$ 秒で表されます。デフォルトの minpoll 値は 6 で、2^6 は 64 に相当します。maxpoll のデフォルト値は 10 で、1024 秒に相当します。使用できる値は、3 から 17 の範囲

で、それぞれ 8s から 36.4h に相当します。これらのオプションは、peer または server で使用します。maxpoll を短く設定すると、クロックの精度が向上する場合があります。

22.16.14. サーバー優先順位の設定

特定のサーバーが他の同様の統計情報のサーバーよりも優先されるよう指定するには、サーバーまたはピアコマンドの最後に以下のオプションを追加します。

prefer

他の同様の統計情報のサーバーに優先して、このサーバーが同期に使用されます。このオプションは、peer または server コマンドで使用します。

22.16.15. NTP パケットの Time-to-Live (有効期限) の設定

デフォルトで使用される特定の time-to-live (TTL)値を指定するには、サーバーまたはピアコマンドの最後に以下のオプションを追加します。

ttl value

ブロードキャストサーバーおよびマルチキャスト NTP サーバーが送信するパケットで使用される TTL の値を指定します。manycast クライアントによる 「リング拡張に使用する最大 Time-to」 -live 値を指定します。デフォルト値は 127 です。

22.16.16. 使用する NTP バージョンの設定

デフォルトで使用される特定バージョンの NTP を指定するには、サーバーまたはピアコマンドの最後に以下のオプションを追加します。

version value

作成された NTP パケット内の NTP セットのバージョンを指定します。値は 1 から 4 になります。デフォルト値は 4 です。

22.17. ハードウェアクロック更新の設定

ntpdate を実行した後に、リアルタイムクロック(RTC)とも呼ばれるハードウェアクロックを更新するようにシステムクロックを設定するには、以下の行を /etc/sysconfig/ntpdate に追加します。

-

```
SYNC_HWCLOCK=yes
```

システムクロックからハードウェアクロックを更新するには、root で以下のコマンドを発行します。

```
~]# hwclock --systohc
```

ntpd がシステムクロックを同期している場合には、カーネルは自動的に RTC を 11 分ごとに更新します。

22.18. クロックソースの設定

システムで使用可能なクロックソースを一覧表示するには、以下のコマンドを発行します。

```
~]$ cd /sys/devices/system/clocksource/clocksource0/
clocksource0]$ cat available_clocksource
kvm-clock tsc hpet acpi_pm
clocksource0]$ cat current_clocksource
kvm-clock
```

上記の例では、カーネルは kvm-clock を使用しています。これは仮想マシンなので、起動時にこのクロックソースが選択されています。

デフォルトのクロックソースを上書きするには、grub.conf に以下のような行を追加します。

```
clocksource=tsc
```

利用可能なクロックソースはアーキテクチャーに依存します。

22.19. その他のリソース

以下の情報ソースで NTP および ntpd に関する追加リソースが提供されています。

22.19.1. インストールされているドキュメント

- *ntpd(8) man ページ： ntpd の詳細な説明があり、コマンドラインオプションも含まれています。*

- *ntp.conf(5) man ページ： サーバーおよびピアとの関連付けの設定方法に関する情報が含まれています。*

- *ntpq(8) man ページ： NTP サーバーの監視およびクエリーを行う NTP クエリーユーティリティーを説明しています。*

- *ntpdc(8) man ページ： ntpd の状態をクエリーおよび変更する ntpd ユーティリティーを説明しています。*

- *ntp_auth(5) man ページ： ntpd の認証オプション、コマンド、および鍵管理を説明しています。*

- *ntp_keygen(8) man ページ： ntpd の公開および秘密鍵生成を説明しています。*

- *ntp_acc(5) man ページ： restrict コマンドを使ったアクセス制御オプションを説明しています。*

- *ntp_mon(5) man ページ： 統計情報収集に関する監視オプションを説明しています。*

- *ntp_clock(5) man ページ： 基準クロックを設定するコマンドを説明しています。*

- *ntp_misc(5) man ページ： その他のオプションが説明されています。*

22.19.2. 便利な Web サイト

http://doc.ntp.org/

NTP 資料のアーカイブ

http://www.eecis.udel.edu/~mills/ntp.html

Network Time Synchronization Research Project

http://www.eecis.udel.edu/~mills/ntp/html/manyopt.html

NTPv4 での Automatic Server Discovery に関する情報

第23章 PTP4L を使用した PTP の設定

23.1. PTP の概要

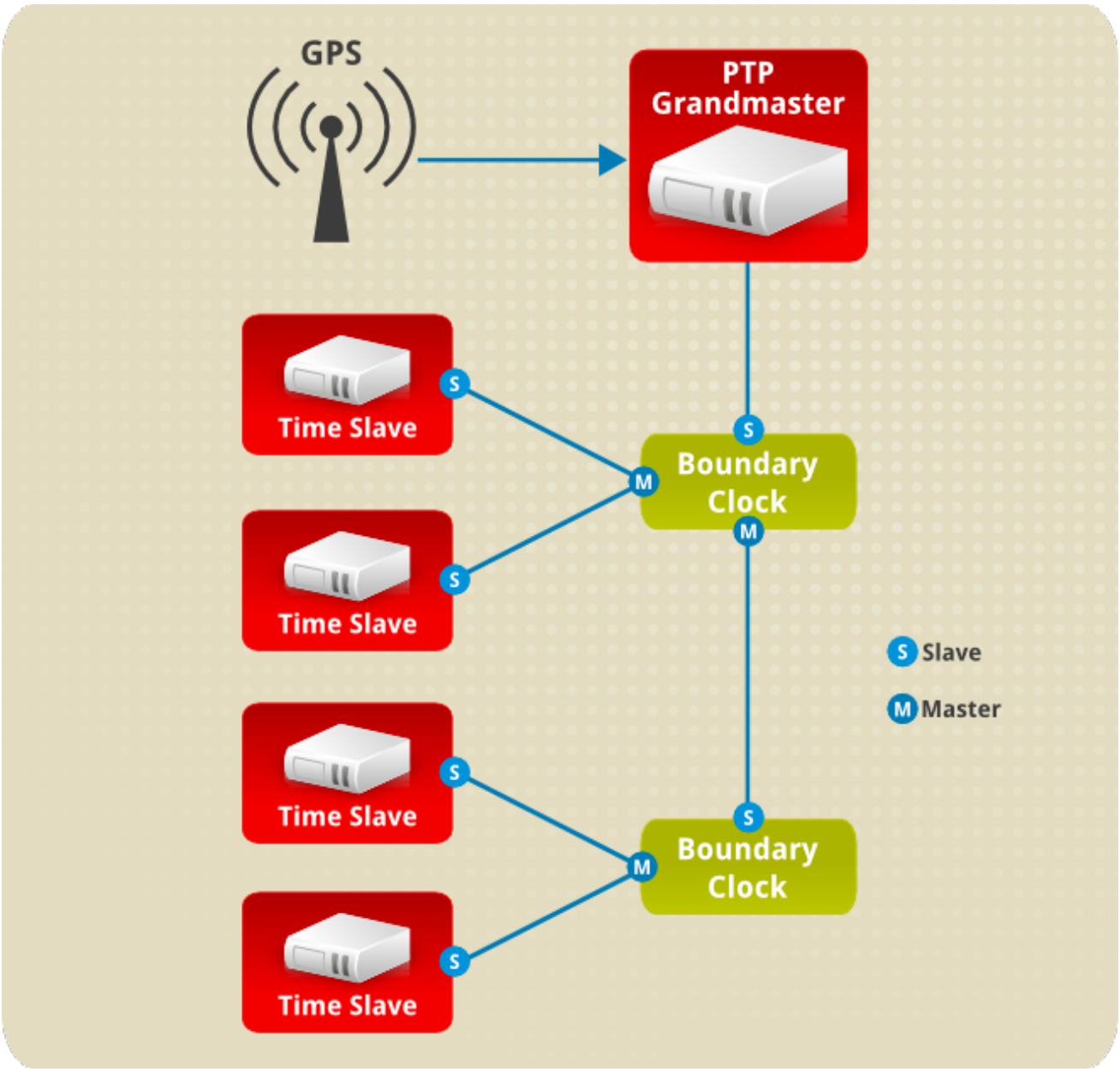
Precision Time Protocol (PTP)は、ネットワーク内でのクロックの同期に使用されるプロトコルです。ハードウェアサポートと合わせて使用すると PTP はマイクロ秒以下の正確性があり、これは NTP で得られる正確性が非常に優れています。PTP サポートは、カーネルとユーザースペースに分けられます。Red Hat Enterprise Linux 6 のカーネルに PTP クロックのサポートが含まれるようになりました。これは、ネットワークドライバーが提供しています。実際のプロトコル実装は linuxptp と呼ばれ、これは Linux の IEEE 標準 1588 に準拠した PTPv2 実装です。

linuxptp パッケージには、クロック同期用の ptp4l および phc2sys プログラムが含まれます。ptp4l プログラムは、PTP 境界クロックと通常のクロックを実装します。ハードウェアタイムスタンプでは、PTP ハードウェアクロックのマスタークロックとの同期に使用され、ソフトウェアタイムスタンプではシステムクロックのマスタークロックとの同期に使用されます。phc2sys プログラムは、システムクロックを ネットワークインターフェースカード (NIC)上の PTP ハードウェアクロックと同期するハードウェアタイムスタンプでのみ必要です。

23.1.1. PTP を理解する

PTP で同期するクロックは、マスター/スレーブ階層で組織されています。スレーブはマスターと同期し、このマスターは別のマスターのスレーブとなっています。この階層は、最適なマスタークロック (BMC)アルゴリズムにより自動的に作成され、更新されます。クロックにポートが 1 つしかない場合は、master または slave を指定できます。このようなクロックは 通常のクロック (OC)と呼ばれます。1 つのポートを持つクロックは 1 つのポート上にマスターとなり、別のポートでスレーブできます。このようなクロックは 境界 クロック(BC)と呼ばれます。トップレベルのマスターは、Global Positioning System (GPS)のタイムソースを使用して同期できる グランドマスタークロック と呼ばれます。GPS ベースの時間ソースを使うことで、高度の正確性を保って異なるネットワークが同期可能になります。

図23.1 PTP グランドマスター、境界、スレーブの各クロック



[D]

23.1.2. PTP の利点

PTP が Network Time Protocol (NTP)よりも対応している主な利点の 1 つは、さまざまな ネットワークインターフェースコントローラー (NIC)およびネットワークスイッチにあるハードウェアサポートです。この特化されたハードウェアにより、PTP はメッセージ転送の遅延を考慮し、時間同期の精度を大幅に高めます。ネットワーク内で PTP 以外に対応するハードウェアコンポーネントを使用することは可能ですが、その場合、変動が増えたり、遅れが非対称となったりして、同期が不正確になります。これにより、通信パスで使用される複数の非 PTP 対応コンポーネントが追加されます。可能な限りの精度を実現するには、PTP クロック間の全ネットワークコンポーネントが PTP ハードウェアを有効にすることが推奨されます。ネットワークハードウェアが PTP に対応していない大規模なネットワークでの時間同期は、NTP に適している場合があります。

ハードウェア PTP サポートでは、NIC には独自のオンボードクロックがあります。これは受信およ

び送信の PTP メッセージのタイムスタンプで使用されます。PTP マスターに同期しているオンボードクロックで、コンピューターのシステムクロックは NIC 上の PTP ハードウェアクロックに同期されます。ソフトウェア PTP サポートでは、システムクロックは PTP メッセージのタイムスタンプに使用され、PTP マスターに直接同期します。ソフトウェア PTP サポートではオペレーティングシステムによる追加の PTP パケット処理を必要としますが、ハードウェア PTP サポートでは、NIC が PTP パケットの送受信時のタイムスタンプができるので、より優れた正確性が得られます。

23.2. PTP の使用

PTP を使用するには、対象とするインターフェース用のカーネルネットワークドライバーがソフトウェアまたはハードウェアのタイムスタンプ機能をサポートしている必要があります。

23.2.1. ドライバーおよびハードウェアサポートの確認

ドライバーがハードウェアタイムスタンプをサポートしていることに加え、NIC も物理的なハードウェアでこの機能をサポートできる必要があります。特定のドライバーおよび NIC のタイムスタンプ機能を検証する最善の方法は、ethtool ユーティリティーを使用して以下のようにインターフェースをクエリーすることです。

```
~]# ethtool -T eth3
Time stamping parameters for eth3:
Capabilities:
        hardware-transmit     (SOF_TIMESTAMPING_TX_HARDWARE)
        software-transmit     (SOF_TIMESTAMPING_TX_SOFTWARE)
        hardware-receive      (SOF_TIMESTAMPING_RX_HARDWARE)
        software-receive      (SOF_TIMESTAMPING_RX_SOFTWARE)
        software-system-clock (SOF_TIMESTAMPING_SOFTWARE)
        hardware-raw-clock    (SOF_TIMESTAMPING_RAW_HARDWARE)
PTP Hardware Clock: 0
Hardware Transmit Timestamp Modes:
        off                   (HWTSTAMP_TX_OFF)
        on                    (HWTSTAMP_TX_ON)
Hardware Receive Filter Modes:
        none                  (HWTSTAMP_FILTER_NONE)
        all                   (HWTSTAMP_FILTER_ALL)
```

eth3 は、チェックするインターフェースに置き換えます。

ソフトウェアタイムスタンプのサポートについては、パラメーター一覧に以下を含めます。

- SOF_TIMESTAMPING_SOFTWARE

- *SOF_TIMESTAMPING_TX_SOFTWARE*

- *SOF_TIMESTAMPING_RX_SOFTWARE*

ハードウェアタイムスタンプのサポートについては、パラメーター一覧に以下を含めます。

- *SOF_TIMESTAMPING_RAW_HARDWARE*

- *SOF_TIMESTAMPING_TX_HARDWARE*

- *SOF_TIMESTAMPING_RX_HARDWARE*

23.2.2. PTP のインストール

Red Hat Enterprise Linux 6 のカーネルが PTP に対応するようになりました。ユーザー空間のサポートは、linuxptp パッケージのツールで提供されます。linuxptp をインストールするには、root で以下のコマンドを発行します。

```
~]# yum install linuxptp
```

これにより、ptp4l および phc2sys がインストールされます。

システムクロックの時間を設定するサービスを同時に 2 つ以上実行しないでください。NTP を使用して PTP 時間を提供する場合は、「NTP を使用した PTP 時間の実行」 を参照してください。

23.2.3. ptp4l の起動

ptp4l プログラムは、デフォルトでハードウェアのタイムスタンプを使用しようとします。ハードウェアタイムスタンプ対応のドライバーおよび NIC で ptp4l を使用するには、使用するネットワークインターフェースに -i オプションを指定する必要があります。root で以下のコマンドを入力します。

```
~]# ptp4l -i eth3 -m
```

eth3 は、設定するインターフェースに置き換えます。以下は、NIC 上の PTP クロックがマスターに同期された際の ptp4l からの出力例です。

```
~]# ptp4l -i eth3 -m
selected eth3 as PTP clock
port 1: INITIALIZING to LISTENING on INITIALIZE
port 0: INITIALIZING to LISTENING on INITIALIZE
port 1: new foreign master 00a069.fffe.0b552d-1
selected best master clock 00a069.fffe.0b552d
port 1: LISTENING to UNCALIBRATED on RS_SLAVE
master offset -23947 s0 freq +0 path delay       11350
master offset -28867 s0 freq +0 path delay       11236
master offset -32801 s0 freq +0 path delay       10841
master offset -37203 s1 freq +0 path delay       10583
master offset  -7275 s2 freq -30575 path delay   10583
port 1: UNCALIBRATED to SLAVE on MASTER_CLOCK_SELECTED
master offset  -4552 s2 freq -30035 path delay   10385
```

マスターオフセットの値は、マスターからナノ秒で測定されたオフセットです。s0、s1、s2 の各ストリングは、異なるクロックのサーボ状態を示しています。s0 はアンロックされています。s1 はクロックステップで、s2 です。servo がロック状態(s2)になると、p_offset_const オプションが設定ファイルの正の値（ptp4l(8)で説明）に設定されていない限り、クロックはステップになりません（徐々に調整されるのみ）。freq 値は、10億分の 1(ppb)単位のクロックの周波数調整です。パス遅延値は、マスターから送信される同期メッセージの遅延 (ナノ秒単位) です。Port 0 は、ローカル PTP 管理に使用される Unix ドメインソケットです。Port 1 は eth3 インターフェースです（上記の例に基づいています）。 INITIALIZING、LISTENING、UNCALIBRATED、および SLAVE は、INITIALIZE、RS_SLAVE、MASTER_CLOCK_SELECTED イベントで変化する可能性のあるポート状態です。最後の状態変更メッセージでは、ポート状態が UNCALIBRATED から SLAVE に変更し、PTP マスタークロックとの同期が成功したことを示しています。

ptp4l プログラムは、以下を実行してサービスとして起動することもできます。

```
~]# service ptp4l start
```

サービスとして実行する場合、オプションは /etc/sysconfig/ptp4l ファイルに指定されます。各種の異なる ptp4l オプションおよび設定ファイルの設定についての詳細は、ptp4l(8) man ページを参照してください。

デフォルトでは、メッセージは /var/log/messages に送信されます。ただし、-m オプションを指定すると標準出力へのロギングが可能になり、これはデバッグで役に立ちます。

ソフトウェアタイムスタンプを有効にするには、以下のように -S オプションを使用する必要があります。

```
~]# ptp4l -i eth3 -m -S
```

23.2.3.1. 遅延測定メカニズムの選択

遅延測定メカニズムには 2 つの異なる遅延測定メカニズムがあり、以下のように ptp4l コマンドにオプションを追加することで選択できます。

-P

-P は、 ピアツーピア (P2P)の遅延測定メカニズムを選択します。

P2P メカニズムはネットワークトポロジーの変更に高速に対応し、他のメカニズムよりも遅延の測定がより正確であるため、推奨されています。 P2P メカニズムは、各ポートが最大で他の 1 つの P2P ポートを持つ PTP メッセージを交換するトポロジーでのみ使用できます。これは、透過クロックを含む通信パス上のすべてのハードウェアでサポートされ、使用される必要があります。

-E

-E は、エンドツーエンド (E2E)の遅延測定メカニズムを選択します。これはデフォルトです。

E2E メカニズムは、遅延 「リクエスト応答」 メカニズムとも呼ばれます。

-A

-A は、遅延測定メカニズムの自動選択を有効にします。

自動オプションは、E2E モードで ptp4l を起動します。ピアの遅延リクエストを受信すると、P2P モードに変更されます。

注記

1 つの PTP 通信パス上のクロックはすべて、遅延を測定するために同じメカニズムを使用する必要があります。E2E メカニズムを使用しているポートでピアの遅延リクエストを受信すると警告が表示されます。P2P メカニズムを使用するポートで E2E の遅延リクエストを受信すると警告が表示されます。

23.3. 設定ファイルの指定

コマンドラインに設定できないコマンドラインオプションおよびその他のオプションは、オプションの設定ファイルで設定できます。

デフォルトでは、設定ファイルは読み取られないため、-f オプションを指定してランタイム時に指定する必要があります。以下に例を示します。

```
~]# ptp4l -f /etc/ptp4l.conf
```

上記の -i eth3 -m -S オプションと同等の設定ファイルは、以下のようになります。

```
~]# cat /etc/ptp4l.conf
[global]
verbose             1
time_stamping       software
[eth3]
```

23.4. PTP 管理クライアントの使用

PTP 管理クライアントである pmc を使用すると、以下のように ptp4l から追加情報を取得できま

す。

```
~]# pmc -u -b 0 'GET CURRENT_DATA_SET'
sending: GET CURRENT_DATA_SET
        90e2ba.fffe.20c7f8-0 seq 0 RESPONSE MANAGMENT CURRENT_DATA_SET
                stepsRemoved        1
                offsetFromMaster  -142.0
                meanPathDelay     9310.0
```

```
~]# pmc -u -b 0 'GET TIME_STATUS_NP'
sending: GET TIME_STATUS_NP
        90e2ba.fffe.20c7f8-0 seq 0 RESPONSE MANAGMENT TIME_STATUS_NP
                master_offset              310
                ingress_time               1361545089345029441
                cumulativeScaledRateOffset   +1.000000000
                scaledLastGmPhaseChange    0
                gmTimeBaseIndicator        0
                lastGmPhaseChange          0x0000'0000000000000000.0000
                gmPresent                  true
                gmIdentity                 00a069.fffe.0b552d
```

-b オプションを ゼロ に設定すると、ローカルで実行している **ptp4l** インスタンスへの境界を制限されます。大きな境界値が大きいと、ローカルクロックからさらに **PTP** ノードから情報を取得します。取得できる情報は次のとおりです。

- **stepsRemoved** は、グランドマスタークロックへの通信パスの数です。
- **offsetFromMaster** および **master_offset** は、マスターから最後に測定されたオフセットをナノ秒で表します。
- **meanPathDelay** は、マスターから送信された同期メッセージの遅延予測をナノ秒で表します。

- *gmPresent が true の場合、PTP クロックがマスターに同期され、ローカルクロックはグランドマスタークロックに同期されません。*

- *gmIdentity はグランドマスターのアイデンティティーです。*

pmc コマンドの完全な一覧を表示するには、root で以下を入力します。

```
~]# pmc help
```

pmc(8) man ページでは詳細情報が見られます。

23.5. クロックの同期

phc2sys プログラムは、システムクロックを NIC 上の PHC ハードウェアクロック(PHC)と同期するために使用されます。phc2sys サービスは /etc/sysconfig/phc2sys 設定ファイルで設定されます。/etc/sysconfig/phc2sys ファイルのデフォルト設定は

```
OPTIONS="-a -r"
```

です。-a オプションを使用すると、phc2sys が ptp4l アプリケーションから同期されるクロックを読み取ります。PTP ポートの状態の変更に従い、NIC ハードウェアクロック間の同期を適宜調整します。-r オプションも指定されていない限り、システムクロックは同期されません。システムクロックをタイムソースにするには、-r オプションを 2 回指定します。

/etc/sysconfig/phc2sys への変更後に、root でコマンドを実行し、コマンドラインから phc2sys サービスを再起動します。

```
~]# service phc2sys restart
```

通常の状態では、service コマンドを使用して、phc2sys サービスを起動し、停止し、再起動します。

phc2sys をサービスとして起動するには、コマンドラインからこれを起動できます。たとえば、root で以下のコマンドを入力します。

```
~]# phc2sys -a -r
```

-a オプションを使用すると、phc2sys が ptp4l アプリケーションから同期されるクロックを読み取ります。システムクロックをタイムソースにするには、-r オプションを 2 回指定します。

または、-s オプションを使用して、システムクロックを特定のインターフェースの PTP ハードウェアクロックに同期します。以下に例を示します。

```
~]# phc2sys -s eth3 -w
```

-w オプションは、実行中の ptp4l アプリケーションが PTP クロックを同期するまで待機し、ptp4l から UTC オフセットへの TAI を取得します。

通常、PTP は 国際原子時(TAI)のタイムスケールで作動し、システムクロックは 協定世界時 (UTC) で維持されます。TAI と UTC のタイムスケール間の現在のオフセットは、36 秒です。このオフセットは、うるう秒が追加もしくは取り除かれると変化します。以下のように、-w オプションを使用しない場合は、-O オプションを使用してこのオフセットを手動で設定する必要があります。

```
~]# phc2sys -s eth3 -O -36
```

phc2sys servo がロックされた状態になると、-S オプションを使用しない限り、クロックはステップされません。つまり、phc2sys プログラムは、ptp4l プログラムが PTP ハードウェアクロックを同期した後に起動すべきということになります。ただし、-w を使用すると、ptp4l の後に phc2sys を起動する必要はありません。これは、クロックの同期を待つためです。

phc2sys プログラムは、以下のコマンドを実行してサービスとしても起動できます。

```
~]# service phc2sys start
```

サービスとして実行する場合、オプションは /etc/sysconfig/phc2sys ファイルに指定されます。phc2sys の他のオプションの詳細は、 phc2sys (8) man ページを参照してください。

本セクションの例では、コマンドがスレーブシステムまたはスレーブポートで実行されている想定であることに注意してください。

23.6. 時間同期の検証

PTP 時間同期が適切に機能すると、オフセットと頻度調整を含む新しいメッセージが ptp4l および phc2sys に定期的に出力され（ハードウェアタイムスタンプが使用される場合）。これらの値は、最終

的に短期間後に収束されます。これらのメッセージは /var/log/messages ファイルで確認できます。ptp4l の出力例 :

```
ptp4l[352.359]: selected /dev/ptp0 as PTP clock
ptp4l[352.361]: port 1: INITIALIZING to LISTENING on INITIALIZE
ptp4l[352.361]: port 0: INITIALIZING to LISTENING on INITIALIZE
ptp4l[353.210]: port 1: new foreign master 00a069.fffe.0b552d-1
ptp4l[357.214]: selected best master clock 00a069.fffe.0b552d
ptp4l[357.214]: port 1: LISTENING to UNCALIBRATED on RS_SLAVE
ptp4l[359.224]: master offset       3304 s0 freq      +0 path delay      9202
ptp4l[360.224]: master offset       3708 s1 freq  -29492 path delay      9202
ptp4l[361.224]: master offset      -3145 s2 freq  -32637 path delay      9202
ptp4l[361.224]: port 1: UNCALIBRATED to SLAVE on MASTER_CLOCK_SELECTED
ptp4l[362.223]: master offset       -145 s2 freq  -30580 path delay      9202
ptp4l[363.223]: master offset       1043 s2 freq  -29436 path delay      8972
ptp4l[364.223]: master offset        266 s2 freq  -29900 path delay      9153
ptp4l[365.223]: master offset        430 s2 freq  -29656 path delay      9153
ptp4l[366.223]: master offset        615 s2 freq  -29342 path delay      9169
ptp4l[367.222]: master offset       -191 s2 freq  -29964 path delay      9169
ptp4l[368.223]: master offset        466 s2 freq  -29364 path delay      9170
ptp4l[369.235]: master offset         24 s2 freq  -29666 path delay      9196
ptp4l[370.235]: master offset       -375 s2 freq  -30058 path delay      9238
ptp4l[371.235]: master offset        285 s2 freq  -29511 path delay      9199
ptp4l[372.235]: master offset        -78 s2 freq  -29788 path delay      9204
```

phc2sys の出力例 :

```
phc2sys[526.527]: Waiting for ptp4l...
phc2sys[527.528]: Waiting for ptp4l...
phc2sys[528.528]: phc offset     55341 s0 freq      +0 delay   2729
phc2sys[529.528]: phc offset     54658 s1 freq  -37690 delay   2725
phc2sys[530.528]: phc offset       888 s2 freq  -36802 delay   2756
phc2sys[531.528]: phc offset      1156 s2 freq  -36268 delay   2766
phc2sys[532.528]: phc offset       411 s2 freq  -36666 delay   2738
phc2sys[533.528]: phc offset       -73 s2 freq  -37026 delay   2764
phc2sys[534.528]: phc offset        39 s2 freq  -36936 delay   2746
phc2sys[535.529]: phc offset        95 s2 freq  -36869 delay   2733
phc2sys[536.529]: phc offset      -359 s2 freq  -37294 delay   2738
phc2sys[537.529]: phc offset      -257 s2 freq  -37300 delay   2753
phc2sys[538.529]: phc offset       119 s2 freq  -37001 delay   2745
phc2sys[539.529]: phc offset       288 s2 freq  -36796 delay   2766
phc2sys[540.529]: phc offset      -149 s2 freq  -37147 delay   2760
phc2sys[541.529]: phc offset      -352 s2 freq  -37395 delay   2771
phc2sys[542.529]: phc offset       166 s2 freq  -36982 delay   2748
phc2sys[543.529]: phc offset        50 s2 freq  -37048 delay   2756
phc2sys[544.530]: phc offset       -31 s2 freq  -37114 delay   2748
phc2sys[545.530]: phc offset      -333 s2 freq  -37426 delay   2747
phc2sys[546.530]: phc offset       194 s2 freq  -36999 delay   2749
```

ptp4l の場合、出力を減らして統計のみを出力するディレクティブ summary_interval もあります。通常、2 秒ごとにメッセージを出力します。たとえば、出力を 1024 秒ごとに減らすには、以下の行を /etc/ptp4l.conf ファイルに追加します。

```
summary_interval 10
```

summary_interval 6 を含む ptp4l の出力例：

```
ptp4l: [615.253] selected /dev/ptp0 as PTP clock
ptp4l: [615.255] port 1: INITIALIZING to LISTENING on INITIALIZE
ptp4l: [615.255] port 0: INITIALIZING to LISTENING on INITIALIZE
ptp4l: [615.564] port 1: new foreign master 00a069.fffe.0b552d-1
ptp4l: [619.574] selected best master clock 00a069.fffe.0b552d
ptp4l: [619.574] port 1: LISTENING to UNCALIBRATED on RS_SLAVE
ptp4l: [623.573] port 1: UNCALIBRATED to SLAVE on MASTER_CLOCK_SELECTED
ptp4l: [684.649] rms  669 max 3691 freq -29383 ± 3735 delay  9232 ± 122
ptp4l: [748.724] rms  253 max  588 freq -29787 ± 221 delay  9219 ± 158
ptp4l: [812.793] rms  287 max  673 freq -29802 ± 248 delay  9211 ± 183
ptp4l: [876.853] rms  226 max  534 freq -29795 ± 197 delay  9221 ± 138
ptp4l: [940.925] rms  250 max  562 freq -29801 ± 218 delay  9199 ± 148
ptp4l: [1004.988] rms  226 max  525 freq -29802 ± 196 delay  9228 ± 143
ptp4l: [1069.065] rms  300 max  646 freq -29802 ± 259 delay  9214 ± 176
ptp4l: [1133.125] rms  226 max  505 freq -29792 ± 197 delay  9225 ± 159
ptp4l: [1197.185] rms  244 max  688 freq -29790 ± 211 delay  9201 ± 162
```

phc2sys からの出力を減らすには、以下のように -u オプションを使用してこれを呼び出します。

```
~]# phc2sys -u summary-updates
```

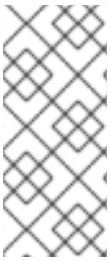
ここでの summary-updates は、サマリー統計に含めるクロック更新数です。例を示します。

```
~]# phc2sys -s eth3 -w -m -u 60
phc2sys[700.948]: rms 1837 max 10123 freq -36474 ± 4752 delay  2752 ±  16
phc2sys[760.954]: rms  194 max  457 freq -37084 ± 174 delay  2753 ±  12
phc2sys[820.963]: rms  211 max  487 freq -37085 ± 185 delay  2750 ±  19
phc2sys[880.968]: rms  183 max  440 freq -37102 ± 164 delay  2734 ±  91
phc2sys[940.973]: rms  244 max  584 freq -37095 ± 216 delay  2748 ±  16
phc2sys[1000.979]: rms  220 max  573 freq -36666 ± 182 delay  2747 ±  43
phc2sys[1060.984]: rms  266 max  675 freq -36759 ± 234 delay  2753 ±  17
```

23.7. NTP を使用した PTP 時間の実行

ntpd デーモンは、ptp4l または phc2sys で同期されたシステムクロック、または LOCAL 参照クロックドライバーを使用して同期されたシステムクロックの時間を分散するよう設定できます。ntpd がシステムクロックを調整するのを防ぐには、ntp.conf ファイルで NTP サーバーを指定しないでください。以下は、ntp.conf の最小限の例です。

```
~]# cat /etc/ntp.conf
server   127.127.1.0
fudge    127.127.1.0 stratum 0
```



注記

DHCP クライアントプログラムである dhclient が DHCP サーバーから NTP サーバーの一覧を受信したら、ntp.conf に追加され、サービスを再起動します。この機能を無効にするには、PEERNTP=no を /etc/sysconfig/network に追加します。

23.8. PTP を使った NTP 時間の実行

逆方向の NTP から PTP への同期も可能です。ntpd を使用してシステムクロックを同期する場合、ptp4l は priority1 オプション（または、最適なマスタークロックアルゴリズムに含まれるその他のクロックオプション）でグランドマスタークロックとして設定し、システムクロックから PTP を介して時間を分配できます。

```
~]# cat /etc/ptp4l.conf
[global]
priority1 127
[eth3]
# ptp4l -f /etc/ptp4l.conf
```

ハードウェアタイムスタンプでは、phc2sys を使って PTP ハードウェアクロックをシステムクロックに同期させる必要があります。

```
~]# phc2sys -c eth3 -s CLOCK_REALTIME -w
```

PTP クロックの周波数の迅速な変更を防ぐため、システムクロックへの同期は、PI servo の小規模な P（比例）および I（積分）定数を使用して緩めることができます。

```
~]# phc2sys -c eth3 -s CLOCK_REALTIME -w -P 0.01 -I 0.0001
```

```
]# phc2sys -s eth3 -c CLOCK_REALTIME -w -r -r 0.01 -r 0.0001
```

23.9. TIMEMASTER を使用した PTP または NTP 時間への同期

複数の PTP ドメインがネットワーク上で利用できるか、または NTP へのフォールバックが必要な場合、timemaster プログラムを使用して、利用可能なすべてのタイムソースにシステムクロックを同期できます。PTP 時間は、共有メモリードライバー （ chronyd または ntpd へのSHM リファレンスクロック）を介して phc2sys および ptp4l で提供されます（システム上で設定されている NTP デーモンにより異なります）。NTP デーモンは次に、すべてのタイムソース、PTP および NTP の両方を比較でき、システムクロックを同期するのに最適なソースを使用できます。

開始時に、timemaster は NTP および PTP タイムソースを指定する設定ファイルを読み取り、独自のクロックを持つか、または PTP ハードウェアクロック(PHC)を共有するネットワークインターフェースを確認し、ptp4l および chronyd または ntpd の設定ファイルを生成し、必要に応じて ptp4l、phc2sys、および chronyd または ntpd プロセスを起動します。終了時には生成された設定ファイルは削除されます。chronyd、ntpd、ptp4l の設定ファイルを /var/run/timemaster/ に書き込みます。

23.9.1. timemaster をサービスとして起動

timemaster をサービスとして起動するには、root で以下のコマンドを発行します。

```
~]# service timemaster start
```

これにより、 /etc/timemaster.conf のオプションが読み取られます。Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;LinuxRed Hat Enterprise Linux 6nbsp;6 でのシステムサービスの管理に関する詳細は、「Managing Services with systemd」を参照してください。

23.9.2. timemaster 設定ファイルの概要

Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux は、デフォルトの /etc/timemaster.conf ファイルにデフォルトオプションを含む数多くのセクションを提供します。このセクションの見出しは括弧で囲まれます。

デフォルト設定を表示するには、以下のコマンドを実行します。

```
~]$ less /etc/timemaster.conf
# Configuration file for timemaster

#[ntp_server ntp-server.local]
#minpoll 4
```

```
#maxpoll 4

#[ptp_domain 0]
#interfaces eth0

[timemaster]
ntp_program chronyd

[chrony.conf]
include /etc/chrony.conf

[ntp.conf]
includefile /etc/ntp.conf

[ptp4l.conf]

[chronyd]
path /usr/sbin/chronyd
options -u chrony

[ntpd]
path /usr/sbin/ntpd
options -u ntp:ntp -g

[phc2sys]
path /usr/sbin/phc2sys

[ptp4l]
path /usr/sbin/ptp4l
```

```
[ntp_server address]
```

という名前のセクションに注目してください。これは NTP サーバーセクションの例で、「ntp-server.local」 はローカル LAN 上の NTP サーバーのホスト名の例です。セクション名の一部としてホスト名または IP アドレスを使用して、必要に応じてセクションを追加します。セクション例の短いポーリング値はパブリックサーバーに適していないことに注意してください。適切な minpoll 値および maxpoll 値の説明は、22章ntpd を使用した NTP 設定 を参照してください。

```
[ptp_domain number]
```

A 「PTP domain」 は、互いに同期する 1 つ以上の PTP クロックのグループです。それらは別のドメインでクロックに同期される場合もありますが、同期されない場合もあります。同じドメイン番号で設定されているクロックがドメインを構成します。これには、PTP グランドマスタークロックが含まれま

す。各 「PTP ドメインセクションのドメイン」 番号は、ネットワーク上に設定される PTP ドメインのいずれかに対応している必要があります。

ptp4l のインスタンスは、独自の PTP クロックを持つすべてのインターフェースで起動し、ハードウェアのタイムスタンプは自動的に有効にされます。ハードウェアのタイムスタンプをサポートするインターフェースには PTP クロック(PHC)が割り当てられます。同じ PHC を共有するインターフェースグループごと、ソフトウェアのタイムスタンプのみをサポートする各インターフェースに対して、別の ptp4l インスタンスが起動します。すべての ptp4l インスタンスは、スレーブとして実行するように設定されます。ハードウェアのタイムスタンプが設定されたインターフェースが 1 つ以上の PTP ドメインに指定される場合、作成される最初の ptp4l インスタンスのみで、ハードウェアのタイムスタンプが有効にされます。

[timemaster]

という名前のセクションに注目してください。デフォルトの timemaster 設定には、アクセス制限および認証キーの設定を組み込むためにシステムの ntpd および chrony 設定（/etc/ntp.conf または /etc/chronyd.conf）が含まれます。つまり、そこに指定されたすべての NTP サーバーが timemaster で使用されます。

セクションの見出しは以下のようになります。

- *[ntp_SERVER ntp-server.local]: このサーバーのポーリング間隔を指定します。必要に応じて追加のセクションを作成します。セクション見出しのホスト名または IP アドレスを含めます。*
- *[ptp_domain 0]: このドメインに PTP クロックを設定するインターフェースを指定します。必要に応じて、適切なドメイン番号で追加のセクションを作成します。*
- *[timemaster]: 使用する NTP デーモンを指定します。使用できる値は chronyd および ntpd です。*
- *[chrony.conf]: chronyd に生成される設定ファイルにコピーされる追加設定を指定します。*
- *[ntp.conf]: ntpd に生成される設定ファイルにコピーされる追加設定を指定します。*

- *[ptp4l.conf]: ptp 4l に生成される設定ファイルにコピーされるオプションを指定します。*
- *[chronyd]: chronyd に対してコマンドラインで渡される追加設定を指定します。*
- *[ntpd]: ntpd に対してコマンドラインで渡される追加設定を指定します。*
- *[phc2sys]: phc2sys に対してコマンドラインで渡される追加設定を指定します。*
- *[ptp4l]: ptp4l のすべてのインスタンスに対してコマンドラインで渡される追加設定を指定します。*

セクション見出しおよびその内容の詳細は、timemaster(8) man ページで説明されています。

23.9.3. timemaster オプションの設定

手順23.1 timemaster 設定ファイルの編集

1. *デフォルト設定を変更するには、root で編集するために /etc/timemaster.conf ファイルを開きます。*

```
~]# vi /etc/timemaster.conf
```

2. *timemaster を使用して制御する必要のある各 NTP サーバーについて、[ntp_server address] セクションを作成します。この例の短いポーリング値はパブリックサーバーに適していません。適切な minpoll 値および maxpoll 値の説明は、22章ntpd を使用した NTP 設定 を参照してください。*

3. *ドメインで使用するインターフェースを追加するには、#[ptp_domain 0] セクションを編集してインターフェースを追加します。必要に応じて追加のドメインを作成します。以下に例を示します。*

```
[ptp_domain 0]
    interfaces eth0
```

```
[ptp_domain 1]
interfaces eth1
```

4. このシステムの NTP デーモンとして ntpd を使用する必要がある場合は、[timemaster] セクションのデフォルトエントリーを chronyd から ntpd に変更します。ntpd と chronyd の相違点は、「chrony スイートを使用した NTP 設定」を参照してください。

5. このシステムで chronyd を NTP サーバーとして使用する場合は、[chrony.conf] セクションのデフォルトの include /etc/chrony.conf エントリーの下に追加オプションを追加します。/etc/chrony.conf へのパスが変更された場合は、デフォルトの include エントリーを編集します。

6. このシステムで ntpd を NTP サーバーとして使用する場合は、[ntp.conf] セクションのデフォルトの include /etc/ntp.conf エントリーの下に追加オプションを追加します。/etc/ntp.conf へのパスが変更された場合は、デフォルトの include エントリーを編集します。

7. [ptp4l.conf] セクションに、ptp4l に生成される設定ファイルにコピーされるオプションを追加します。本章では、共通オプションについて説明します。詳細は、ptp4l(8) man ページを参照してください。

8. [chronyd] セクションに、timemaster で呼び出される場合に chronyd に渡されるコマンドラインオプションを追加します。chronyd の使用に関する詳細は、「chrony スイートを使用した NTP の設定」を参照してください。

9. [ntpd] セクションに、timemaster で呼び出される場合に ntpd に渡されるコマンドラインオプションを追加します。ntpd の使用方法は、22章ntpd を使用した NTP 設定 を参照してください。

10. [phc2sys] セクションに、timemaster で呼び出される場合に phc2sys に渡すコマンドラインオプションを追加します。本章では、共通オプションについて説明します。詳細は、phy2sys(8) man ページを参照してください。

11. [ptp4l] セクションに、timemaster で呼び出される場合に ptp4l に渡されるコマンドラインオプションを追加します。本章では、共通オプションについて説明します。詳細は、ptp4l(8) man ページを参照してください。

12. 設定ファイルを保存し、**timemaster** を再起動するには、**root** で以下のコマンドを実行します。

```
~]# service timemaster restart
```

23.10. 精度の向上

これまでテストの結果では、ティックレスカーネル機能を無効にするとシステムクロックの安定性が大幅に改善され、**PTP** 同期の精度が向上します（電力消費量は高まります）。カーネル起動オプションパラメーターに **nohz=off** を追加することで、カーネルティックモードを無効にすることができます。ただし、**kernel-3.10.0-197.el7** に適用される最近の改善により、システムクロックの安定性が大幅に改善され、**nohz=off** のあるクロックの安定性の相違点が、ほとんどのユーザーに対して大幅に小さくなるはずです。

ptp4l および **phc2sys** アプリケーションは、新しい適応サービスを使用するように設定できます。**PI** サーボに対するこれの利点は、適正に機能させるために **PI** 定数を設定する必要がない点にあります。これを **ptp4l** に使用するには、**/etc/ptp4l.conf** ファイルに以下の行を追加します。

```
clock_servo linreg
```

/etc/ptp4l.conf に変更を加えた後に、**root** として以下のコマンドを実行し、コマンドラインから **ptp4l** サービスを再起動します。

```
~]# service ptp4l restart
```

これを **phc2sys** に使用するには、**/etc/sysconfig/phc2sys** ファイルに以下の行を追加します。

```
-E linreg
```

/etc/sysconfig/phc2sys に変更を加えた後に、**root** として以下のコマンドを実行し、コマンドラインから **phc2sys** サービスを再起動します。

```
~]# service phc2sys restart
```

23.11. その他のリソース

以下の情報ソースで PTP および ptp4l ツールに関する追加リソースが提供されています。

23.11.1. インストールされているドキュメント

- ptp4l(8) man ページ： 設定ファイルの形式を含む ptp4l オプションを説明しています。
- pmc(8) man ページ： PTP 管理クライアントおよびそのコマンドオプションを説明しています。
- phc2sys(8) man ページ： システムクロックを PTP ハードウェアクロック(PHC)に同期させるツールを説明しています。

23.11.2. 便利な Web サイト

http://linuxptp.sourceforge.net/

Linux PTP プロジェクト。

http://www.nist.gov/el/isd/ieee/ieee1588.cfm

IEEE 1588 規格

パート VII. 監視と自動化

ここでは、システム管理者がシステムパフォーマンスのモニター、システムタスクの自動化、およびバグの報告を行うための各種ツールを説明します。

第24章 システムモニタリングツール

システムを設定するには、多くの場合システム管理者は空きメモリの容量、空きディスク領域、ハードディスクのパーティション設定状況、実行中のプロセスを決定する必要があります。

24.1. システムプロセスの表示

24.1.1. ps コマンドの使用

ps コマンドは、実行中のプロセスについての情報を表示します。静的な一覧、すなわちコマンドを実行するときに実行しているプロセスのスナップショットです。実行中のプロセスを定期的に更新した場合は、top コマンドまたは System Monitor アプリケーションを代わりに使用します。

他のユーザーが所有しているプロセスを含め、現在システム上で実行中の全プロセスを一覧表示するには、シェルプロンプトで以下を入力します。

```
ps ax
```

一覧のプロセスごとに、ps ax コマンドがプロセス ID(PID)、それに関連付けられているターミナル(TTY)、現在のステータス(STAT)、累積された CPU 時間(TIME)、および実行可能ファイルの名前(COMMAND)を表示します。以下に例を示します。

```
~]$ ps ax
  PID TTY      STAT   TIME COMMAND
    1 ?        Ss     0:01 /sbin/init
    2 ?        S      0:00 [kthreadd]
    3 ?        S      0:00 [migration/0]
    4 ?        S      0:00 [ksoftirqd/0]
    5 ?        S      0:00 [migration/0]
    6 ?        S      0:00 [watchdog/0]
[output truncated]
```

各プロセスと同時に所有者も表示するには、以下のコマンドを使用します。

```
ps aux
```

ps ax コマンドで提供される情報とは別に、ps aux はプロセス所有者の実効ユーザー名(USER)、CPU のパーセンテージ(%CPU)およびメモリー(%MEM)の使用状況、キロバイト単位で仮想メモリーサイズ（キロバイト単位）、キロバイト単位での非スワップ物理メモリー容量(VSZ)、プロセスの開始日時を表示します。RSSたとえば、以下のようになります。

```
~]$ ps aux
USER       PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.1  19404   832 ?        Ss   Mar02   0:01 /sbin/init
root         2  0.0  0.0      0     0 ?        S    Mar02   0:00 [kthreadd]
root         3  0.0  0.0      0     0 ?        S    Mar02   0:00 [migration/0]
root         4  0.0  0.0      0     0 ?        S    Mar02   0:00 [ksoftirqd/0]
root         5  0.0  0.0      0     0 ?        S    Mar02   0:00 [migration/0]
root         6  0.0  0.0      0     0 ?        R    Mar02   0:00 [watchdog/0]
[output truncated]
```

ps コマンドを grep と組み合わせて使用して、特定のプロセスが実行中かどうかを確認することもできます。たとえば、Emacs が実行中かどうかを確認するには、以下を入力します。

```
~]$ ps ax | grep emacs
12056 pts/3    S+     0:00 emacs
12060 pts/2    S+     0:00 grep --color=auto emacs
```

利用可能なコマンドラインオプションの一覧は、ps(1)の man ページを参照してください。

24.1.2. top コマンドの使用

top コマンドは、システムで実行しているプロセスのリアルタイムの一覧を表示します。また、システムのアップタイム、現在の CPU およびメモリ使用率、実行中のプロセスの合計数についての追加情報も表示します。さらには、一覧の並び替えやプロセスの kill などの操作も実行できます。

top コマンドを実行するには、シェルプロンプトで以下を入力します。

```
top
```

一覧表示された各プロセスについて top コマンドはプロセス ID(PID)、プロセス所有者の実効ユーザー名(USER)、優先度(PR)、優れた値(NI)、プロセスが使用する仮想メモリー容量(VIRT)、プロセスが使用する非スワップ物理メモリー容量(RES)、プロセスが使用する共有メモリー容量(SHR)、プロセスステータスフィールド S）を表示します。 CPU(%CPU)およびメモリー(%MEM)の使用量の割合、累積された CPU 時間(TIME+)、実行可能ファイルの名前(COMMAND)。以下に例を示します。

```
~]$ top
top - 02:19:11 up 4 days, 10:37,  5 users,  load average: 0.07, 0.13, 0.09
Tasks: 160 total,   1 running, 159 sleeping,   0 stopped,   0 zombie
Cpu(s): 10.7%us,  1.0%sy,  0.0%ni, 88.3%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:    760752k total,   644360k used,   116392k free,     3988k buffers
Swap:  1540088k total,    76648k used,  1463440k free,   196832k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
14401 jhradile  20   0  313m  10m 5732 S  5.6  1.4   6:27.29 gnome-system-mo
```

```
 1764 root      20   0  133m  23m 4756 S  5.3  3.2   6:32.66 Xorg
13865 jhradile  20   0 1625m 177m 6628 S  0.7 23.8   0:57.26 java
   20 root      20   0     0    0    0 S  0.3  0.0   4:44.39 ata/0
 2085 root      20   0 40396  348  276 S  0.3  0.0   1:57.13 udisks-daemon
    1 root      20   0 19404  832  604 S  0.0  0.1   0:01.21 init
    2 root      20   0     0    0    0 S  0.0  0.0   0:00.01 kthreadd
    3 root      RT   0     0    0    0 S  0.0  0.0   0:00.00 migration/0
    4 root      20   0     0    0    0 S  0.0  0.0   0:00.02 ksoftirqd/0
    5 root      RT   0     0    0    0 S  0.0  0.0   0:00.00 migration/0
    6 root      RT   0     0    0    0 S  0.0  0.0   0:00.00 watchdog/0
    7 root      20   0     0    0    0 S  0.0  0.0   0:01.00 events/0
    8 root      20   0     0    0    0 S  0.0  0.0   0:00.00 cpuset
    9 root      20   0     0    0    0 S  0.0  0.0   0:00.00 khelper
   10 root      20   0     0    0    0 S  0.0  0.0   0:00.00 netns
   11 root      20   0     0    0    0 S  0.0  0.0   0:00.00 async/mgr
   12 root      20   0     0    0    0 S  0.0  0.0   0:00.00 pm
[output truncated]
```

表24.1「インタラクティブな top コマンド」には、top で使用できる便利な対話型のコマンドが含まれています。詳細は、top(1)の man ページを参照してください。

表24.1 インタラクティブな top コマンド

コマンド	詳細
Enter、Space	表示を最新の情報に直ちに更新します。
h, ?	ヘルプ画面を表示します。
k	プロセスを強制終了します。プロセス ID およびプロセスに送信するシグナルがプロンプトされます。
-n	表示されるプロセス番号を変更します。番号を入力するようプロンプトされます。
u	一覧をユーザー別に並べ替えます。
M	一覧をメモリ使用率で並べ替えます。
%P	一覧を CPU 使用率で並べ替えます。
q	ユーティリティーを終了して、シェルプロンプトに戻ります。

24.1.3. システムモニターツールの使用

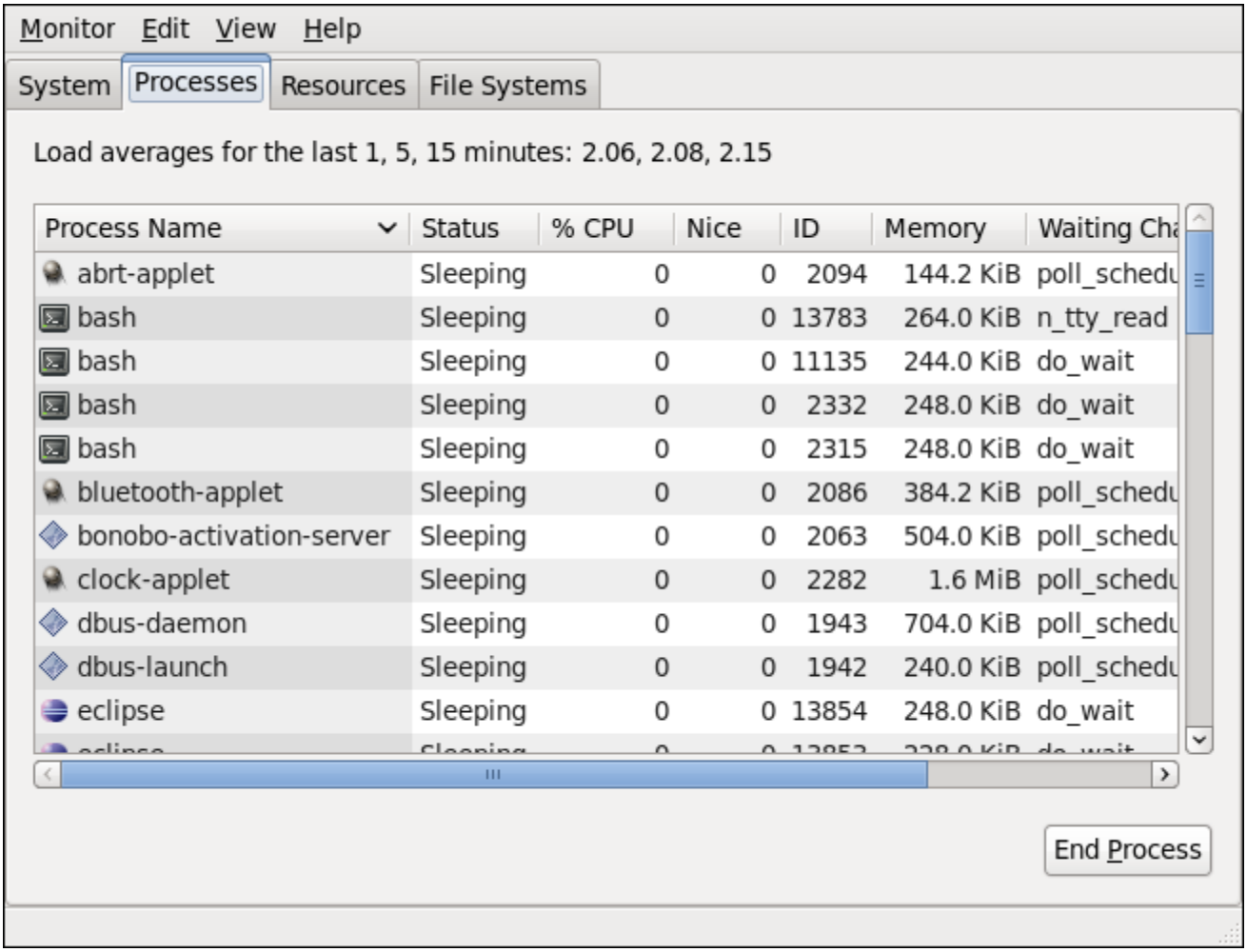
System Monitor ツールの Processes タブを使用すると、グラフィカルユーザーインターフェースからプロセスの表示、検索、優先度の変更、kill を行うことができます。このツールをインストールす

るには、root で以下のコマンドを発行します。

```
~]# yum install gnome-system-monitor
```

System Monitor ツールを起動するには、パネルから Applications → System Tools → System Monitor を選択するか、シェルプロンプトで gnome-system-monitor と入力します。次に Processes タブをクリックして実行中プロセスの一覧を表示します。

図24.1 システムモニター — プロセス



[D]

一覧のプロセスごとに、System Monitor ツールは 名前(Process Name)、現在のステータス（ステータス）、CPU 使用率のパーセンテージ(% CPU)、nice 値(Nice)、プロセス ID(ID)、メモリー使用量（メモリー）、プロセスが待機中のチャネル(Waiting Channel)、およびセッション（セッション）に関する追加情報を表示します。特定の列で昇順で情報を並べ替えるには、その列の名前をクリックします。特定のコラム別に情報を昇順で並び替えるには、コラム名をクリックします。

デフォルトでは、System Monitor ツールは現在のユーザーが所有しているプロセスの一覧を表示

します。View メニューからさまざまなオプションを選択すると、以下が可能になります。

- *実行中のプロセスのみの表示*
- *すべてのプロセスの表示*
- *ユーザーのプロセスの表示*
- *プロセスの依存関係の表示*
- *選択したプロセスのメモリーマップの表示*
- *選択したプロセスで開いているファイルの表示*
- *プロセスの一覧を更新します。*

さらに、Edit メニューのさまざまなオプションにより、以下が可能になります。

- *プロセスを停止します。*
- *停止したプロセスの実行を継続します。*
- *プロセスを終了します。*
- *プロセスを強制終了します。*
- *選択したプロセスの優先度を変更する*

- プロセスの一覧の更新間隔や表示する情報など、システムモニター の設定を編集します。

また、一覧からプロセスを選択し、End Process ボタンをクリックしてプロセスを終了することもできます。

24.2. メモリ使用量の表示

24.2.1. free コマンドの使用

free コマンドは、システムのメモリの空き容量と使用量を表示します。シェルプロンプトで以下を入力してください。

```
free
```

free コマンドは、物理メモリー(Mem)とスワップ領域(Swap)に関する情報を提供します。メモリーの合計量(total)、使用メモリー容量(used)、空きメモリー容量(free)、共有(shared)、カーネルバッファー(buffers)、およびキャッシュメモリー容量(cached)を表示します。以下に例を示します。

```
~]$ free
             total       used       free     shared    buffers     cached
Mem:        760752     661332      99420          0       6476     317200
-/+ buffers/cache:     337656     423096
Swap:      1540088     283652    1256436
```

デフォルトでは、free は値をキロバイトで表示します。メガバイトで値を表示するには、-m コマンドラインオプションを指定します。

```
free -m
```

たとえば、以下のようになります。

```
~]$ free -m
             total       used       free     shared    buffers     cached
Mem:           742        646         96          0          6        309
-/+ buffers/cache:        330        412
Swap:         1503        276       1227
```

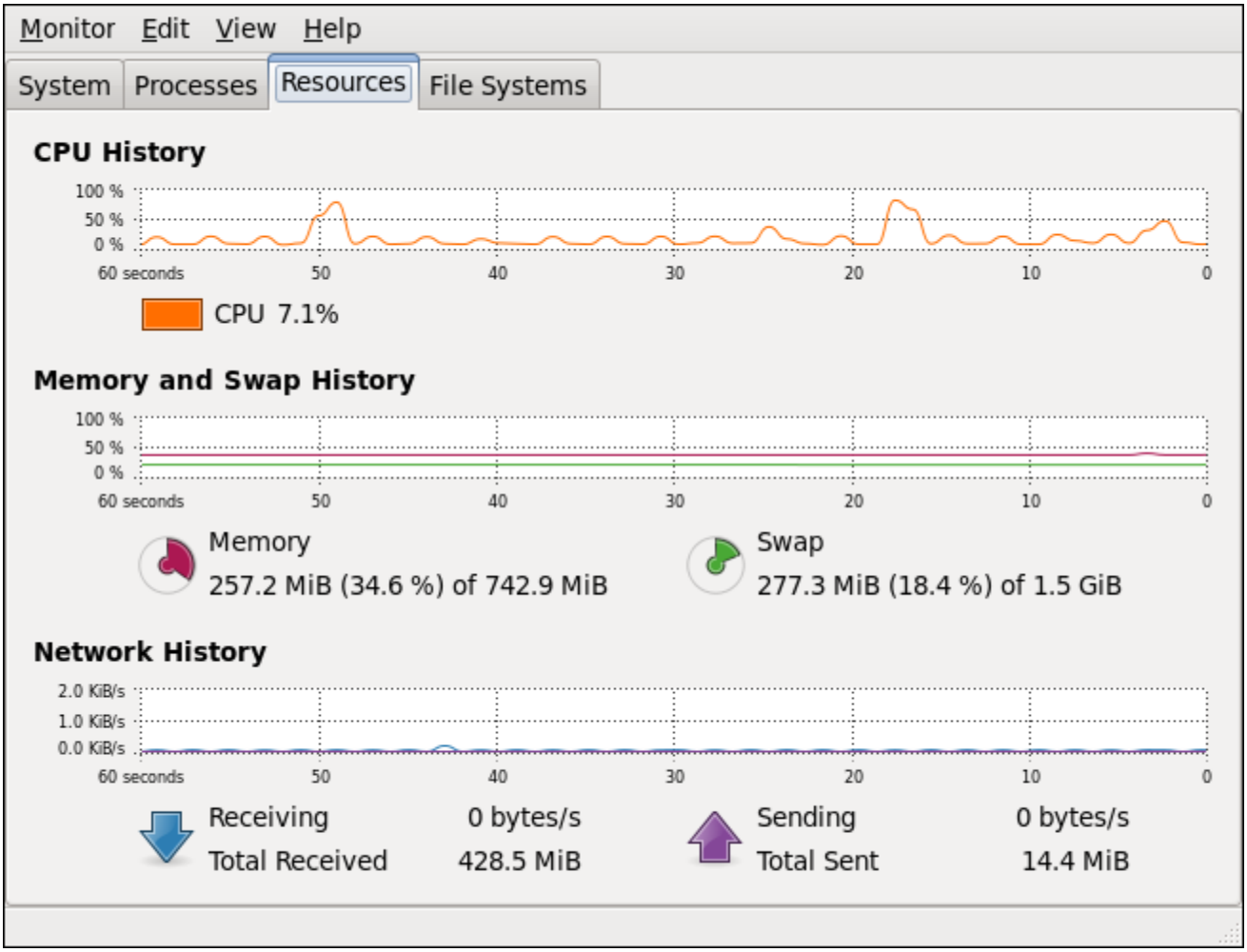
利用可能なコマンドラインオプションの一覧は、free(1)の man ページを参照してください。

24.2.2. システムモニターツールの使用

System Monitor ツールの Resources タブは、システムのメモリの空き容量と使用量を表示します。

System Monitor ツールを起動するには、パネルから Applications → System Tools → System Monitor を選択するか、シェルプロンプトで gnome-system-monitor と入力します。次に Resources タブをクリックしてシステムのメモリー使用率を表示します。

図24.2 システムモニター — リソース



[D]

Memory and India History セクションでは、System Monitor ツールがメモリーとスワップの使用量履歴をグラフィック表示します。また、物理メモリー（メモリー）とスワップ領域(memory)の合計量、および使用中の量が表示されます。

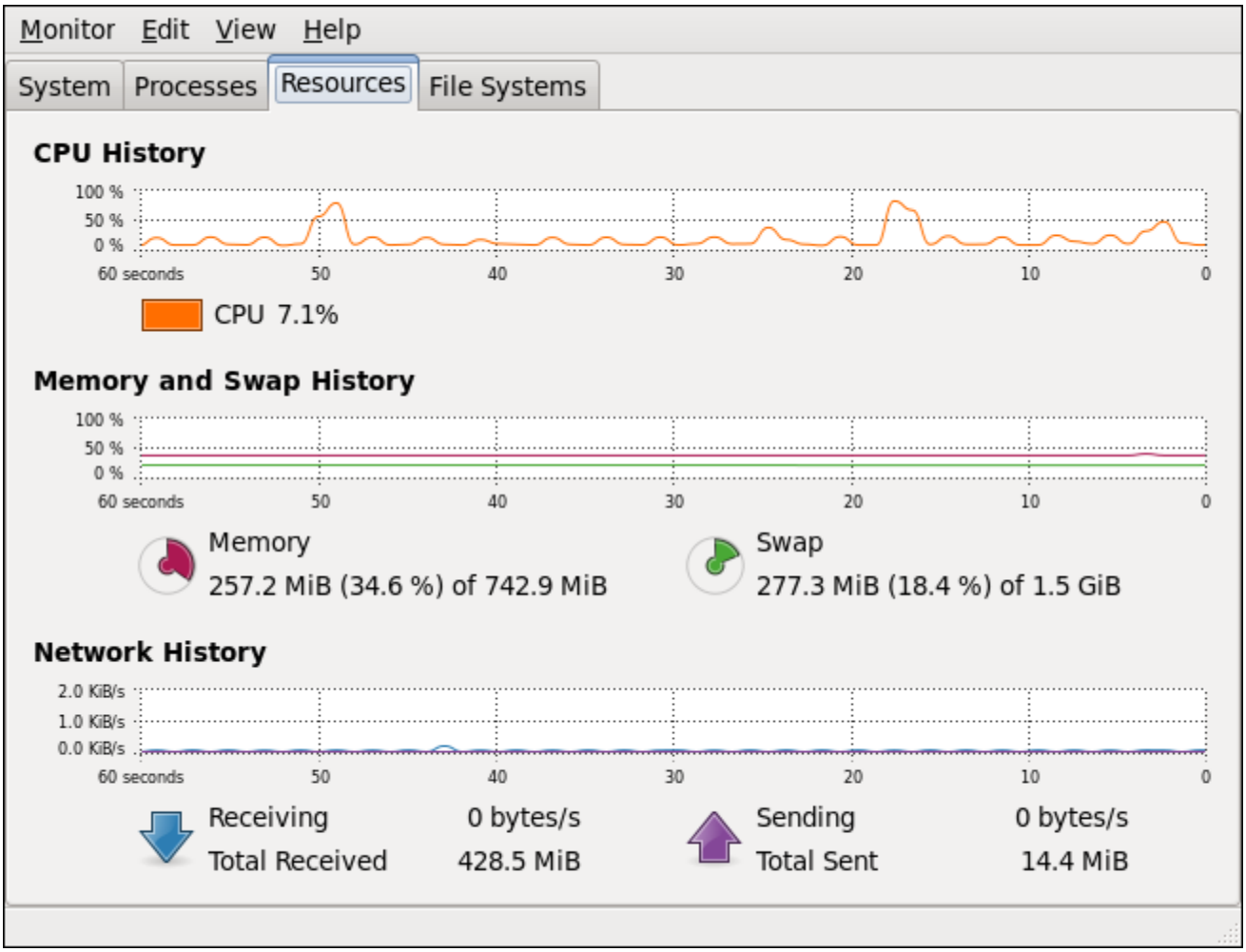
24.3. CPU 使用率の表示

24.3.1. システムモニターツールの使用

System Monitor ツールの Resources タブは、システムの現在の CPU 使用率を表示します。

System Monitor ツールを起動するには、パネルから Applications → System Tools → System Monitor を選択するか、シェルプロンプトで gnome-system-monitor と入力します。次に Resources タブをクリックしてシステムの CPU 使用率を表示します。

図24.3 システムモニター — リソース



[D]

CPU History セクションで、System Monitor ツールは CPU 使用率履歴をグラフィックで表し、現在使用中の CPU 容量のパーセンテージを表示します。

24.4. ブロックデバイスとファイルシステムの表示

24.4.1. lsblk コマンドの使用

lsblk コマンドを使用すると、利用可能なブロックデバイスの一覧を表示できます。シェルプロンプトで以下を入力してください。

```
lsblk
```

一覧表示された各ブロックデバイスについて lsblk コマンドが表示するのは次のとおりです。デバイス名(NAME)、メジャーおよびマイナーデバイス番号(MAJ:MIN)、デバイスがリムーバブル場合は(RM)、サイズが読み取り専用(SIZE)、そのタイプ(RO)、およびデバイスがマウントされている場所(TYPE)を表示します。MOUNTPOINT以下に例を示します。

```
~]$ lsblk
NAME                  MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
sr0                    11:0    1  1024M  0 rom
vda                   252:0    0    20G  0 rom
|-vda1                252:1    0   500M  0 part /boot
`-vda2                252:2    0  19.5G  0 part
  |-vg_kvm-lv_root (dm-0) 253:0    0    18G  0 lvm  /
  `-vg_kvm-lv_swap (dm-1) 253:1    0   1.5G  0 lvm  [SWAP]
```

デフォルトでは、lsblk はツリーのような形式でブロックデバイスを一覧表示します。情報を通常のリストとして表示するには、コマンドラインオプション -l を追加します。

```
lsblk -l
```

たとえば、以下のようになります。

```
~]$ lsblk -l
NAME                MAJ:MIN RM   SIZE RO TYPE MOUNTPOINT
sr0                  11:0    1  1024M  0 rom
vda                 252:0    0    20G  0 rom
vda1                252:1    0   500M  0 part /boot
vda2                252:2    0  19.5G  0 part
vg_kvm-lv_root (dm-0) 253:0    0    18G  0 lvm  /
vg_kvm-lv_swap (dm-1) 253:1    0   1.5G  0 lvm  [SWAP]
```

利用可能なコマンドラインオプションの一覧は、lsblk(8)man ページを参照してください。

24.4.2. blkid コマンドの使用

blkid コマンドを使用すると、利用可能なブロックデバイスに関する情報を表示できます。これを行うには、root で次のコマンドを実行します。

```
blkid
```

一覧表示された各ブロックデバイスについて blkid コマンドは、汎用一意識別子 (UUID)、ファイルシステムタイプ(TYPE)、ボリュームラベル(LABEL)などの属性を表示します。以下に例を示します。

```
~]# blkid
/dev/vda1: UUID="7fa9c421-0054-4555-b0ca-b470a97a3d84" TYPE="ext4"
/dev/vda2: UUID="7IvYzk-TnnK-oPjf-ipdD-cofz-DXaJ-gPdgBW" TYPE="LVM2_member"
/dev/mapper/vg_kvm-lv_root: UUID="a07b967c-71a0-4925-ab02-aebcad2ae824" TYPE="ext4"
/dev/mapper/vg_kvm-lv_swap: UUID="d7ef54ca-9c41-4de4-ac1b-4193b0c1ddb6" TYPE="swap"
```

デフォルトでは、blkid コマンドは利用可能なすべてのブロックデバイスを一覧表示します。特定のデバイスの情報のみを表示するには、コマンドラインでデバイス名を指定します。

```
blkid device_name
```

たとえば、/dev/vda1 に関する情報を表示するには、以下を入力します。

```
~]# blkid /dev/vda1
/dev/vda1: UUID="7fa9c421-0054-4555-b0ca-b470a97a3d84" TYPE="ext4"
```

また、上記のコマンドを -p および -o udev のコマンドラインオプションと共に使用して、詳細情報を取得することもできます。このコマンドを実行するには、root 権限が必要な点に注意してください。

```
blkid -po udev device_name
```

以下に例を示します。

```
~]# blkid -po udev /dev/vda1
ID_FS_UUID=7fa9c421-0054-4555-b0ca-b470a97a3d84
ID_FS_UUID_ENC=7fa9c421-0054-4555-b0ca-b470a97a3d84
ID_FS_VERSION=1.0
ID_FS_TYPE=ext4
ID_FS_USAGE=filesystem
```

利用可能なコマンドラインオプションの一覧は、blkid(8)man ページを参照してください。

24.4.3. findmnt コマンドの使用

findmnt コマンドは、現在マウントされているファイルシステムの一覧を表示します。シェルプロンプトで以下を入力してください。

```
findmnt
```

一覧表示された各ファイルシステムについて findmnt コマンドが表示するのは、ターゲットマウントポイント(TARGET)、ソースデバイス(SOURCE)、ファイルシステムタイプ(FSTYPE)、および関連するマウントオプション(OPTIONS)を表示します。以下に例を示します。

```
~]$ findmnt
TARGET                  SOURCE                  FSTYPE   OPTIONS
/                       /dev/mapper/vg_kvm-lv_root ext4     rw,relatime,sec
|-/proc                 /proc                   proc     rw,relatime
| |-/proc/bus/usb       /proc/bus/usb           usbfs    rw,relatime
| `-/proc/sys/fs/binfmt_misc                    binfmt_m rw,relatime
|-/sys                  /sys                    sysfs    rw,relatime,sec
|-/selinux                                      selinuxf rw,relatime
|-/dev                  udev                    devtmpfs rw,relatime,sec
| `-/dev                udev                    devtmpfs rw,relatime,sec
|   |-/dev/pts          devpts                  devpts   rw,relatime,sec
|   `-/dev/shm          tmpfs                   tmpfs    rw,relatime,sec
|-/boot                 /dev/vda1               ext4     rw,relatime,sec
|-/var/lib/nfs/rpc_pipefs  sunrpc               rpc_pipe rw,relatime
|-/misc                 /etc/auto.misc          autofs   rw,relatime,fd=
`-/net                  -hosts                  autofs   rw,relatime,fd=
[output truncated]
```

デフォルトでは、findmnt はツリーのような形式でファイルシステムを一覧表示します。情報を通常のリストとして表示するには、コマンドラインオプション -l を追加します。

```
findmnt -l
```

たとえば、以下のようになります。

```
~]$ findmnt -l
TARGET                  SOURCE                  FSTYPE   OPTIONS
/proc                   /proc                   proc     rw,relatime
/sys                    /sys                    sysfs    rw,relatime,seclabe
/dev                    udev                    devtmpfs rw,relatime,seclabe
/dev/pts                devpts                  devpts   rw,relatime,seclabe
/dev/shm                tmpfs                   tmpfs    rw,relatime,seclabe
/                       /dev/mapper/vg_kvm-lv_root ext4     rw,relatime,seclabe
/selinux                                        selinuxf rw,relatime
/dev                    udev                    devtmpfs rw,relatime,seclabe
/proc/bus/usb           /proc/bus/usb           usbfs    rw,relatime
/boot                   /dev/vda1               ext4     rw,relatime,seclabe
/proc/sys/fs/binfmt_misc                        binfmt_m rw,relatime
/var/lib/nfs/rpc_pipefs  sunrpc                 rpc_pipe rw,relatime
/misc                   /etc/auto.misc          autofs   rw,relatime,fd=7,pg
/net                    -hosts                  autofs   rw,relatime,fd=13,p
[output truncated]
```

特定のタイプのファイルシステムのみを一覧表示するよう選択することも可能です。これを行うに

は、-t コマンドラインオプションに続けてファイルシステムタイプを追加します。

```
findmnt -t type
```

たとえば、ext4 ファイルシステムの一覧を表示するには、次のコマンドを実行します。

```
~]$ findmnt -t ext4
TARGET SOURCE                  FSTYPE OPTIONS
/      /dev/mapper/vg_kvm-lv_root ext4   rw,relatime,seclabel,barrier=1,data=ord
/boot  /dev/vda1                  ext4   rw,relatime,seclabel,barrier=1,data=ord
```

利用可能なコマンドラインオプションの一覧は、findmnt(8)の man ページを参照してください。

24.4.4. df コマンドの使用

df コマンドは、システムのディスク領域の使用量についての詳しいレポートを表示します。シェルプロンプトで以下を入力してください。

```
df
```

一覧表示された各ファイルシステムについて df コマンドが表示するのは次のとおりです。名前(Filesystem)、サイズ（1K-blocks または Size）、使用容量(Used)、利用可能な領域のサイズ(Available)、領域の使用量の割合(Use%)、およびファイルシステムがマウントされている場所(Mounted on)。以下に例を示します。

```
~]$ df
Filesystem              1K-blocks      Used Available Use% Mounted on
/dev/mapper/vg_kvm-lv_root  18618236   4357360  13315112  25% /
tmpfs                     380376       288    380088   1% /dev/shm
/dev/vda1                   495844     77029    393215  17% /boot
```

デフォルトでは、df コマンドは 1 キロバイトブロック単位でパーティションサイズと、キロバイト単位で使用中および利用可能なディスク領域の容量を表示します。メガバイトおよびギガバイトで情報を表示するには、コマンドラインオプション -h を指定して、df が人間が判読可能な形式で値を表示します。

```
df -h
```

たとえば、以下のようになります。

```
~]$ df -h
```

```
Filesystem            Size  Used Avail Use% Mounted on
/dev/mapper/vg_kvm-lv_root   18G  4.2G   13G  25% /
tmpfs                372M  288K  372M   1% /dev/shm
/dev/vda1            485M   76M  384M  17% /boot
```

利用可能なコマンドラインオプションの一覧は、df(1)の man ページを参照してください。

24.4.5. du コマンドの使用

du コマンドはディレクトリー内のファイルが使用している領域を表示します。現在の作業ディレクトリー内のサブディレクトリーごとのディスク使用量を表示するには、オプションなしで以下のコマンドを実行します。

```
du
```

以下に例を示します。

```
~]$ du
14972   ./Downloads
4       ./.gnome2
4       ./.mozilla/extensions
4       ./.mozilla/plugins
12      ./.mozilla
15004   .
```

デフォルトでは、du コマンドはキロバイト単位でディスク使用量を表示します。メガバイトおよびギガバイトで情報を表示するには、コマンドラインオプション -h を指定して、ユーティリティーが人間が判読可能な形式で値を表示します。

```
du -h
```

たとえば、以下のようになります。

```
~]$ du -h
15M     ./Downloads
4.0K    ./.gnome2
4.0K    ./.mozilla/extensions
4.0K    ./.mozilla/plugins
12K     ./.mozilla
15M     .
```

du コマンドは、一覧の最後で常に現在のディレクトリーの合計量を表示します。この情報のみを表示するには、-s コマンドラインオプションを指定します。

```
du -sh
```

以下に例を示します。

```
~]$ du -sh
15M     .
```

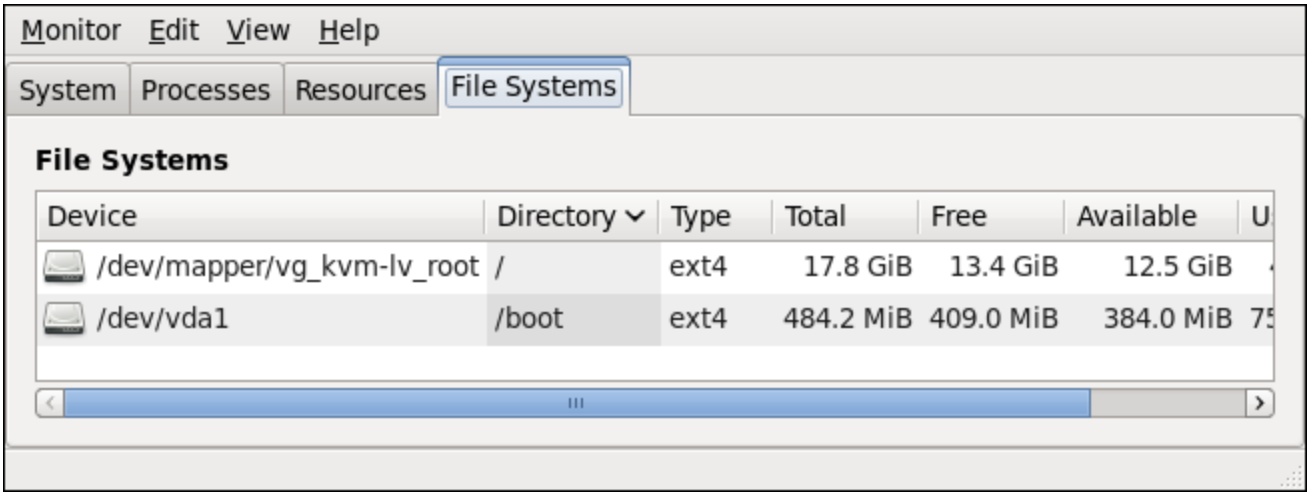
利用可能なコマンドラインオプションの一覧は、du(1)の man ページを参照してください。

24.4.6. システムモニターツールの使用

System Monitor ツールの File Systems タブは、グラフィカルユーザーインターフェースでファイルシステムおよびディスク領域の使用量を表示します。

System Monitor ツールを起動するには、パネルから Applications → System Tools → System Monitor を選択するか、シェルプロンプトで gnome-system-monitor と入力します。次に、ファイルシステム タブを クリックしてファイルシステムの一覧を表示します。

図24.4 システムモニター — ファイルシステム



[D]

一覧表示された各ファイルシステムについて、System Monitor ツールはソースデバイス（デバイス）、ターゲットマウントポイント（ディレクトリー）、およびファイルシステムタイプ（タイプ）、そのサイズ(Total)、利用可能な領域（空き）、使用済み（使用）を表示します。

24.4.7. gamin によるファイルおよびディレクトリーの監視

Red Hat Enterprise Linux 6.8 以降、GLib システムライブラリーはファイルとディレクトリーの監視、NFS ファイルシステムでの変更の検出に gamin を使用します。デフォルトでは、gamin は inotify ではなく NFS ファイルシステムのポーリングを使用します。他のファイルシステムの変更は、GLib に直接実装される inotify monitor により監視されます。

File Alteration Monitor(FAM)システムのサブセットとして、 inotify Linux カーネルサブシステムで FAM 仕様を再実装します。これは GNOME プロジェクトですが、GNOME 依存関係はありません。glib2 パッケージと gamin パッケージの両方はデフォルトでインストールされます。

デフォルトでは、gamin は設定なしで機能し、Linux では /mnt/ または / media/ * に一致するすべてのパスのポーリングの使用に戻ります。ユーザーは、以下の設定をオーバーライドするか、または拡張するには、以下のいずれかの設定ファイルの内容を変更します。*

- */etc/gamin/gaminrc*
- *$HOME/.gaminrc*
- */etc/gamin/mandatory_gaminrc*

設定ファイルは以下のコマンドのみを受け入れます。

設定ファイルで使用できるコマンド

notify

カーネルモニタリングを一致するパスに使用する必要があることを示すには、以下を行います。

poll

一致するパスにポーリングを使用することを表すには、以下を行います。

fsset

ファイルシステムタイプで使用される通知メソッドを制御する。

このような設定ファイルの例は、以下を参照してください。

```
# configuration for gamin
# Can be used to override the default behaviour.
# notify filepath(s) : indicate to use kernel notification
# poll filepath(s)   : indicate to use polling instead
# fsset fsname method poll_limit : indicate what method of notification for the file system
#                          kernel - use the kernel for notification
#                          poll - use polling for notification
#                          none - don't use any notification
#                          the poll_limit is the number of seconds
#                          that must pass before a resource is polled again.
#                          It is optional, and if it is not present the previous
#                          value will be used or the default.
notify /mnt/local* /mnt/pictures* # use kernel notification on these paths
poll /temp/*                 # use poll notification on these paths
fsset nfs poll 10             # use polling on nfs mounts and poll once every 10 seconds
```

この順序で、3 つの設定ファイルが読み込まれます。

i.
/etc/gamin/gaminrc

ii.
~/.gaminrc

iii.
/etc/gamin/mandatory_gaminrc

/etc/gamin/mandatory_gaminrc 設定ファイルを使用すると、システム管理者は、ユーザーが設定した潜在的な危険性のある設定をオーバーライドできます。ポーリングまたはカーネル通知のいずれを使用すべきかを推測すべきパスを確認すると、gamin は最初に、設定ファイル内のユーザーが指定したルールの順番をチェックしてから、事前定義済みのルールをチェックします。これにより、例にある /mnt/local* の最初の宣言は、/ mnt/* のデフォルトの宣言を上書きします。

gamin が特定のパスでポーリング通知を使用するように設定されていない場合は、パスが存在するファイルシステムに基づいて決定されます。

24.5. ハードウェア情報の表示

24.5.1. lspci コマンドの使用

lspci コマンドは、PCI バスおよびそれらに接続されているデバイスの情報を表示します。システム

内にあるすべての PCI デバイスを一覧表示するには、シェルプロンプトで以下を入力してください。

```
lspci
```

これは、以下のようにデバイスのシンプルな一覧を表示します。

```
~]$ lspci
00:00.0 Host bridge: Intel Corporation 82X38/X48 Express DRAM Controller
00:01.0 PCI bridge: Intel Corporation 82X38/X48 Express Host-Primary PCI Express Bridge
00:1a.0 USB Controller: Intel Corporation 82801I (ICH9 Family) USB UHCI Controller #4 (rev
02)
00:1a.1 USB Controller: Intel Corporation 82801I (ICH9 Family) USB UHCI Controller #5 (rev
02)
00:1a.2 USB Controller: Intel Corporation 82801I (ICH9 Family) USB UHCI Controller #6 (rev
02)
[output truncated]
```

-v コマンドラインオプションを使用して詳細出力を表示することもできます。詳細の出力には -vv を指定することもできます。

```
lspci -v|-vv
```

たとえば、システムのビデオカードの製造元、モデル、およびメモリーサイズを確認するには、以下を入力します。

```
~]$ lspci -v
[output truncated]

01:00.0 VGA compatible controller: nVidia Corporation G84 [Quadro FX 370] (rev a1) (prog-if
00 [VGA controller])
        Subsystem: nVidia Corporation Device 0491
        Physical Slot: 2
        Flags: bus master, fast devsel, latency 0, IRQ 16
        Memory at f2000000 (32-bit, non-prefetchable) [size=16M]
        Memory at e0000000 (64-bit, prefetchable) [size=256M]
        Memory at f0000000 (64-bit, non-prefetchable) [size=32M]
        I/O ports at 1100 [size=128]
        Expansion ROM at <unassigned> [disabled]
        Capabilities: <access denied>
        Kernel driver in use: nouveau
        Kernel modules: nouveau, nvidiafb

[output truncated]
```

利用可能なコマンドラインオプションの一覧は、lspci(8)man ページを参照してください。

24.5.2. lsusb コマンドの使用

lsusb コマンドは、USB バスおよびそれらに接続されているデバイスの情報を表示します。システム内にあるすべての USB デバイスを一覧表示するには、シェルプロンプトで以下を入力してください。

```
lsusb
```

これは、以下のようにデバイスのシンプルな一覧を表示します。

```
~]$ lsusb
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
[output truncated]
Bus 001 Device 002: ID 0bda:0151 Realtek Semiconductor Corp. Mass Storage Device
(Multicard Reader)
Bus 008 Device 002: ID 03f0:2c24 Hewlett-Packard Logitech M-UAL-96 Mouse
Bus 008 Device 003: ID 04b3:3025 IBM Corp.
```

-v コマンドラインオプションを使用すると、さらに詳細な出力を表示することもできます。

```
lsusb -v
```

たとえば、以下のようになります。

```
~]$ lsusb -v
[output truncated]

Bus 008 Device 002: ID 03f0:2c24 Hewlett-Packard Logitech M-UAL-96 Mouse
Device Descriptor:
  bLength                18
  bDescriptorType         1
  bcdUSB               2.00
  bDeviceClass            0 (Defined at Interface level)
  bDeviceSubClass         0
  bDeviceProtocol         0
  bMaxPacketSize0         8
  idVendor           0x03f0 Hewlett-Packard
  idProduct          0x2c24 Logitech M-UAL-96 Mouse
  bcdDevice            31.00
  iManufacturer           1
  iProduct                2
  iSerial                 0
  bNumConfigurations      1
  Configuration Descriptor:
```

```
    bLength                 9
    bDescriptorType         2
[output truncated]
```

利用可能なコマンドラインオプションの一覧は、lsusb(8)man ページを参照してください。

24.5.3. lspcmcia コマンドの使用

lspcmcia コマンドを使用すると、システムに存在するすべての PCMCIA デバイスを一覧表示できます。シェルプロンプトで以下を入力してください。

```
lspcmcia
```

以下に例を示します。

```
~]$ lspcmcia
Socket 0 Bridge:      [yenta_cardbus]      (bus ID: 0000:15:00.0)
```

-v コマンドラインオプションを使用して詳細情報を表示するか、-vv を使用して詳細レベルをさらに増やすこともできます。

```
lspcmcia -v|-vv
```

たとえば、以下のようになります。

```
~]$ lspcmcia -v
Socket 0 Bridge:      [yenta_cardbus]      (bus ID: 0000:15:00.0)
      Configuration:  state: on      ready: unknown
```

利用可能なコマンドラインオプションの一覧は、pccardctl(8)man ページを参照してください。

24.5.4. lscpu コマンドの使用

lscpu コマンドを使用すると、CPU の数、アーキテクチャー、ベンダー、ファミリー、モデル、CPU キャッシュなど、システムに存在する CPU に関する情報を一覧表示できます。シェルプロンプトで以下を入力してください。

```
lscpu
```

以下に例を示します。

```
~]$ lscpu
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:            Little Endian
CPU(s):                4
On-line CPU(s) list:   0-3
Thread(s) per core:    1
Core(s) per socket:    4
Socket(s):             1
NUMA node(s):          1
Vendor ID:             GenuineIntel
CPU family:            6
Model:                 23
Stepping:              7
CPU MHz:               1998.000
BogoMIPS:              4999.98
Virtualization:        VT-x
L1d cache:             32K
L1i cache:             32K
L2 cache:              3072K
NUMA node0 CPU(s):     0-3
```

利用可能なコマンドラインオプションの一覧は、lscpu(1)の man ページを参照してください。

24.6. NET-SNMP を使用したパフォーマンスのモニタリング

Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;LinuxRed Hat Enterprise Linux 6nbsp;6 には、柔軟で拡張可能な Simple Network Management Protocol (SNMP)エージェントを含む Net-SNMP ソフトウェアスイートが含まれます。このエージェントと関連ユーティリティーを使用すると、多くのシステムからのパフォーマンスデータを SNMP プロトコルによるポーリングに対応する各種ツールに提供することができます。

このセクションでは、ネットワーク上でパフォーマンスデータを安全に提供するための Net-SNMP エージェントの設定方法、SNMP プロトコルを使用したデータの取得方法、カスタムのパフォーマンスメトリックを提供するための SNMP エージェントの拡張方法について説明します。

24.6.1. Net-SNMP のインストール

Net-SNMP ソフトウェアスイートは、Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux ソフトウェアディストリビューションの RPM パッケージセットとして利用できます。表24.2「利用可能な Net-SNMP パッケージ」 は、各パッケージと内容を要約したものです。

表24.2 利用可能な Net-SNMP パッケージ

パッケージ	提供する項目
net-snmp	SNMP Agent Daemon とドキュメント。このパッケージは、パフォーマンスデータをエクスポートするために必要です。
net-snmp-libs	netsnmp ライブラリーと、同梱の MIB（Management Information Base: 管理情報ベース）。このパッケージは、パフォーマンスデータをエクスポートするために必要です。
net-snmp-utils	snmpget や snmpwalk などの SNMP クライアント。SNMP 経由でシステムのパフォーマンスデータをクエリーするには、このパッケージが必要です。
net-snmp-perl	mib2c ユーティリティーおよび NetSNMP Perl モジュール。
net-snmp-python	Python 向け SNMP クライアントライブラリー。

これらのパッケージをインストールするには、以下の形式で yum コマンドを使用します。

```
yum install package
```

たとえば、本セクションで使用される SNMP Agent Daemon および SNMP クライアントをインストールするには、シェルプロンプトで以下を入力します。

```
~]# yum install net-snmp net-snmp-libs net-snmp-utils
```

このコマンドを実行するには、スーパーユーザーの権限 (つまり root としてログイン) が必要であることに注意してください。Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux に新しいパッケージをインストールする方法の詳細は、「パッケージのインストール」 を参照してください。

24.6.2. Net-SNMP Daemon の実行

net-snmp パッケージには、SNMP Agent Daemon である snmpd が含まれています。本セクションでは、snmpd サービスを起動、停止、再起動する方法と、特定のランレベルで有効にする方法を説明します。ランレベルの概念と Red Hat Enterprise Linux でシステムサービスを管理する方法は、12 章サービスおよびデーモン を参照してください。

24.6.2.1. サービスの起動

現行のセッションで snmpd サービスを実行するには、シェルプロンプトで root として以下を入力します。

```
service snmpd start
```

起動時にサービスが自動的に起動するよう設定するには、以下のコマンドを使用します。

```
chkconfig snmpd on
```

これにより、ランレベル 2、3、4、および 5 でサービスが有効になります。または、「サービスの有効化および無効化」 の説明に従って Service Configuration ユーティリティーを使用できます。

24.6.2.2. サービスの停止

実行中の snmpd サービスを停止するには、シェルプロンプトで root として以下を入力します。

```
service snmpd stop
```

ブート時のサービスの起動を無効にするには、以下のコマンドを使用します。

```
chkconfig snmpd off
```

これにより、すべてのランレベルでサービスが無効になります。または、「サービスの有効化および無効化」 の説明に従って Service Configuration ユーティリティーを使用できます。

24.6.2.3. サービスの再起動

実行中の snmpd サービスを再起動するには、シェルプロンプトで以下を入力します。

```
service snmpd restart
```

これにより、サービスが停止し、再起動が連続して開始します。サービスを停止せずに設定の再読み込みだけを行いたい場合は、代わりに以下のコマンドを実行します。

```
service snmpd reload
```

これにより、実行中の snmpd サービスが設定を再読み込みします。

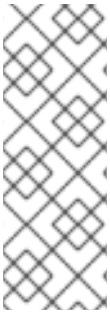
または、「サービスの起動、再起動、停止」 の説明に従って Service Configuration ユーティリティーを使用できます。

24.6.3. Net-SNMP の設定

Net-SNMP Agent Daemon 設定を変更するには、/etc/snmp/snmpd.conf 設定ファイルを編集します。Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;Linux Red Hat Enterprise Linux 6nbsp;6 に同梱されるデフォルトの snmpd.conf ファイルは、多くのコメントが含まれているため、エージェント設定の際のスタート地点となります。

本セクションでは、システム情報と認証の設定という 2 つの一般的なタスクにフォーカスしています。利用可能な設定ディレクティブの詳細については、snmpd.conf(5)man ページを参照してください。また、snmpconf という名前の net-snmp パッケージにも、有効なエージェント設定を対話形式で生成するのに使用できるユーティリティーがあります。

本セクションで説明されている snmpwalk ユーティリティーを使用するには、net-snmp-utils パッケージをインストールする必要があることに注意してください。



変更の適用

設定ファイルの変更を反映させるには、root で以下のコマンドを実行し、snmpd サービスに設定の再読み取りを強制します。

```
service snmpd reload
```

24.6.3.1. システム情報の設定

Net-SNMP は、system ツリー経由で基本的なシステム情報を提供します。たとえば、次の snmpwalk コマンドは、デフォルトのエージェント設定を持つ system ツリーを示しています。

```
~]# snmpwalk -v2c -c public localhost system
SNMPv2-MIB::sysDescr.0 = STRING: Linux localhost.localdomain 2.6.32-122.el6.x86_64 #1
SMP Wed Mar 9 23:54:34 EST 2011 x86_64
SNMPv2-MIB::sysObjectID.0 = OID: NET-SNMP-MIB::netSnmpAgentOIDs.10
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (99554) 0:16:35.54
SNMPv2-MIB::sysContact.0 = STRING: Root <root@localhost> (configure
/etc/snmp/snmp.local.conf)
SNMPv2-MIB::sysName.0 = STRING: localhost.localdomain
SNMPv2-MIB::sysLocation.0 = STRING: Unknown (edit /etc/snmp/snmpd.conf)
```

デフォルトでは、sysName オブジェクトはホスト名に設定されています。sysLocation および sysContact オブジェクトは、syslocation ディレクティブおよび syscontact ディレクティブの値を変

更することで、/etc/snmp/snmpd.conf ファイルで設定できます。以下に例を示します。

```
syslocation Datacenter, Row 3, Rack 2
syscontact UNIX Admin <admin@example.com>
```

設定ファイルに変更を加えた後は、再度 snmpwalk コマンドを実行して設定を再読み込みしてテストします。

```
~]# service snmpd reload
Reloading snmpd: [  OK  ]
~]# snmpwalk -v2c -c public localhost system
SNMPv2-MIB::sysDescr.0 = STRING: Linux localhost.localdomain 2.6.32-122.el6.x86_64 #1
SMP Wed Mar 9 23:54:34 EST 2011 x86_64
SNMPv2-MIB::sysObjectID.0 = OID: NET-SNMP-MIB::netSnmpAgentOIDs.10
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (158357) 0:26:23.57
SNMPv2-MIB::sysContact.0 = STRING: UNIX Admin <admin@example.com>
SNMPv2-MIB::sysName.0 = STRING: localhost.localdomain
SNMPv2-MIB::sysLocation.0 = STRING: Datacenter, Row 3, Rack 2
```

24.6.3.2. 認証の設定

Net-SNMP Agent Daemon は SNMP プロトコルの 3 つの全バージョンに対応します。最初の 2 つのバージョン（1 および 2c）は、コミュニティー文字列 を使用した簡易認証を提供します。この文字列は、エージェントとクライアントのユーティリティー間で共有される秘密です。この文字列は、ネットワーク上でクリアテキストで渡されますが、安全であるとはみなされません。SNMP プロトコルのバージョン 3 は、各種プロトコルを使用したユーザー認証とメッセージの暗号化に対応しています。Net-SNMP エージェントは、SSH でのトンネリング、X.509 証明書を使用した TLS 認証、および Kerberos 認証にも対応しています。

SNMP Version 2c Community の設定

SNMP version 2c community を設定するには、/etc/snmp/snmpd.conf 設定ファイルの rocommunity または rwcommunity ディレクティブを使用します。ディレクティブの形式は次のとおりです。

```
directive community [source [OID]]
```

community は使用するコミュニティー文字列で、source は IP アドレスまたはサブネットで、OID はアクセスを提供する SNMP ツリーです。たとえば、次のディレクティブは、ローカルマシンのコミュニティー文字列 「redhat」 を使用するクライアントに システム ツリーへの読み取り専用アクセスを提供します。

```
rocommunity redhat 127.0.0.1 .1.3.6.1.2.1.1
```

設定をテストするには、-v オプションおよび -c オプションを指定して snmpwalk コマンドを使用

します。

```
~]# snmpwalk -v2c -c redhat localhost system
SNMPv2-MIB::sysDescr.0 = STRING: Linux localhost.localdomain 2.6.32-122.el6.x86_64 #1
SMP Wed Mar 9 23:54:34 EST 2011 x86_64
SNMPv2-MIB::sysObjectID.0 = OID: NET-SNMP-MIB::netSnmpAgentOIDs.10
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (158357) 0:26:23.57
SNMPv2-MIB::sysContact.0 = STRING: UNIX Admin <admin@example.com>
SNMPv2-MIB::sysName.0 = STRING: localhost.localdomain
SNMPv2-MIB::sysLocation.0 = STRING: Datacenter, Row 3, Rack 2
```

SNMP Version 3 User の設定

SNMP version 3 ユーザー を設定するには、net-snmp-create-v3-user コマンドを 使用します。このコマンドにより、/var/lib/net-snmp/snmpd.conf ファイルおよび /etc/snmp/snmpd.conf ファイルにエントリーを追加します。net-snmp-create-v3-user コマンドは、エージェントが実行されていない場合にのみ実行できることに注意してください。以下の例では、パスワードが 「redhatsnmp」 の 「admin」 ユーザーを作成します。

```
~]# service snmpd stop
Stopping snmpd:                                [  OK  ]
~]# net-snmp-create-v3-user
Enter a SNMPv3 user name to create:
admin
Enter authentication pass-phrase:
redhatsnmp
Enter encryption pass-phrase:
  [press return to reuse the authentication pass-phrase]

adding the following line to /var/lib/net-snmp/snmpd.conf:
   createUser admin MD5 "redhatsnmp" DES
adding the following line to /etc/snmp/snmpd.conf:
   rwuser admin
~]# service snmpd start
Starting snmpd:                                [  OK  ]
```

net-snmp-create-v3-user が /etc/snmp/snmpd.conf に追加される rwuser ディレクティブ（ - ro コマンドラインオプションを指定する場合は rouser ）は、rwcommunity ディレクティブおよび rocommunity ディレクティブと同様の形式になります。

```
directive user [noauth|auth|priv] [OID]
```

user はユーザー名で、OID はアクセスを提供する SNMP ツリーです。デフォルトでは、Net-SNMP Agent Daemon は認証済み要求のみを許可します（auth オプション）。noauth オプションを使用すると、認証されていない要求を許可でき、priv オプションは暗号化の使用を強制します。authpriv オプションは要求の認証と応答の暗号化が必要であることを指定します。

たとえば、以下の行では、「admin」 にツリー全体への読み取り/書き込みアクセスを付与しま

す。

```
rwuser admin authpriv .1
```

設定をテストするには、ユーザーのホームディレクトリーに .snmp ディレクトリーを作成して、そのディレクトリーに次の行を含む snmp.conf という名前の設定ファイルを作成します (~/.snmp/snmp.conf)。

```
defVersion 3
defSecurityLevel authPriv
defSecurityName admin
defPassphrase redhatsnmp
```

これで、エージェントのクエリー時に、snmpwalk コマンドがこれらの認証設定を使用するようになりました。

```
~]$ snmpwalk -v3 localhost system
SNMPv2-MIB::sysDescr.0 = STRING: Linux localhost.localdomain 2.6.32-122.el6.x86_64 #1
SMP Wed Mar 9 23:54:34 EST 2011 x86_64
[output truncated]
```

24.6.4. SNMP によるパフォーマンスデータの取得

Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux の Net-SNMP Agent は、SNMP プロトコルによりさまざまなパフォーマンス情報を提供します。さらにエージェントは、システム上のインストールされた RPM パッケージの一覧、システム上で現在実行中のプロセス一覧、またはシステムのネットワーク設定をクエリーすることもできます。

本セクションでは、SNMP による利用可能なパフォーマンスチューニングに関連する OID の概要について説明します。ここでは、net-snmp-utils パッケージがインストールされ、ユーザーが「認証の設定」 で説明されているように SNMP ツリーへのアクセスが許可されていることを前提としています。

24.6.4.1. ハードウェアの設定

Net-SNMP に含まれる Host Resources MIB は、ホストの現在のハードウェアおよびソフトウェア設定に関する情報をクライアントユーティリティーに表示します。表24.3「利用可能な OID」 は、その MIB で利用可能なさまざまな OID の概要を示します。

表24.3 利用可能な OID

OID	詳細
HOST-RESOURCES-MIB::hrSystem	アップタイム、ユーザー数、実行中のプロセス数などのシステム情報全般が含まれています。
HOST-RESOURCES-MIB::hrStorage	メモリおよびファイルシステムの使用に関するデータが含まれています。
HOST-RESOURCES-MIB::hrDevices	すべてのプロセッサー、ネットワークデバイス、ファイルシステムの一覧が含まれています。
HOST-RESOURCES-MIB::hrSWRun	実行中の全プロセス一覧が含まれています。
HOST-RESOURCES-MIB::hrSWRunPerf	HOST-RESOURCES-MIB::hrSWRun からのプロセステーブル上のメモリと CPU 統計が含まれています。
HOST-RESOURCES-MIB::hrSWInstalled	RPM データベースの一覧が含まれています。

入手可能な情報の概要を取得するために使用できる Host Resources MIB には、多くの SNMP テーブルがあります。以下の例では、HOST-RESOURCES-MIB::hrFSTable を表示しています。

```
~]$ snmptable -Cb localhost HOST-RESOURCES-MIB::hrFSTable
SNMP table: HOST-RESOURCES-MIB::hrFSTable

 Index MountPoint RemoteMountPoint                 Type
   Access Bootable StorageIndex LastFullBackupDate LastPartialBackupDate
     1       "/"              "" HOST-RESOURCES-TYPES::hrFSLinuxExt2
 readWrite     true          31      0-1-1,0:0:0.0         0-1-1,0:0:0.0
     5 "/dev/shm"              ""     HOST-RESOURCES-TYPES::hrFSOther
 readWrite    false          35      0-1-1,0:0:0.0         0-1-1,0:0:0.0
     6    "/boot"             "" HOST-RESOURCES-TYPES::hrFSLinuxExt2
 readWrite    false          36      0-1-1,0:0:0.0         0-1-1,0:0:0.0
```

HOST-RESOURCES-MIB の詳細は、/usr/share/snmp/mibs/HOST-RESOURCES-MIB.txt ファイルを参照してください。

24.6.4.2. CPU およびメモリー情報

ほとんどのシステムパフォーマンスデータは、UCD SNMP MIB で利用できます。systemStats OID は、プロセッサー使用率に関する多くのカウンターを提供します。

```
~]$ snmpwalk localhost UCD-SNMP-MIB::systemStats
UCD-SNMP-MIB::ssIndex.0 = INTEGER: 1
UCD-SNMP-MIB::ssErrorName.0 = STRING: systemStats
UCD-SNMP-MIB::ssSwapIn.0 = INTEGER: 0 kB
UCD-SNMP-MIB::ssSwapOut.0 = INTEGER: 0 kB
UCD-SNMP-MIB::ssIOSent.0 = INTEGER: 0 blocks/s
UCD-SNMP-MIB::ssIOReceive.0 = INTEGER: 0 blocks/s
UCD-SNMP-MIB::ssSysInterrupts.0 = INTEGER: 29 interrupts/s
UCD-SNMP-MIB::ssSysContext.0 = INTEGER: 18 switches/s
UCD-SNMP-MIB::ssCpuUser.0 = INTEGER: 0
UCD-SNMP-MIB::ssCpuSystem.0 = INTEGER: 0
UCD-SNMP-MIB::ssCpuIdle.0 = INTEGER: 99
UCD-SNMP-MIB::ssCpuRawUser.0 = Counter32: 2278
UCD-SNMP-MIB::ssCpuRawNice.0 = Counter32: 1395
UCD-SNMP-MIB::ssCpuRawSystem.0 = Counter32: 6826
UCD-SNMP-MIB::ssCpuRawIdle.0 = Counter32: 3383736
UCD-SNMP-MIB::ssCpuRawWait.0 = Counter32: 7629
UCD-SNMP-MIB::ssCpuRawKernel.0 = Counter32: 0
UCD-SNMP-MIB::ssCpuRawInterrupt.0 = Counter32: 434
UCD-SNMP-MIB::ssIORawSent.0 = Counter32: 266770
UCD-SNMP-MIB::ssIORawReceived.0 = Counter32: 427302
UCD-SNMP-MIB::ssRawInterrupts.0 = Counter32: 743442
UCD-SNMP-MIB::ssRawContexts.0 = Counter32: 718557
UCD-SNMP-MIB::ssCpuRawSoftIRQ.0 = Counter32: 128
UCD-SNMP-MIB::ssRawSwapIn.0 = Counter32: 0
UCD-SNMP-MIB::ssRawSwapOut.0 = Counter32: 0
```

特に、ssCpuRawUser、ssCpuRawSystem、ssCpuRawWait および ssCpuRawIdle の OID は、システムがカーネルスペース、ユーザー空間、または I/O でプロセッサー時間の大半を費やしているかどうかを判断する際に役立つカウンターを提供します。Ss RawSwapIn および ssRawSwapOut は、システムがメモリーを使い切っているかどうかを判断する際に役立ちます。

より多くのメモリー情報は、free コマンドに同様のデータを提供する UCD-SNMP-MIB::memory OID で利用できます。

```
~]$ snmpwalk localhost UCD-SNMP-MIB::memory
UCD-SNMP-MIB::memIndex.0 = INTEGER: 0
UCD-SNMP-MIB::memErrorName.0 = STRING: swap
UCD-SNMP-MIB::memTotalSwap.0 = INTEGER: 1023992 kB
UCD-SNMP-MIB::memAvailSwap.0 = INTEGER: 1023992 kB
UCD-SNMP-MIB::memTotalReal.0 = INTEGER: 1021588 kB
UCD-SNMP-MIB::memAvailReal.0 = INTEGER: 634260 kB
UCD-SNMP-MIB::memTotalFree.0 = INTEGER: 1658252 kB
UCD-SNMP-MIB::memMinimumSwap.0 = INTEGER: 16000 kB
UCD-SNMP-MIB::memBuffer.0 = INTEGER: 30760 kB
UCD-SNMP-MIB::memCached.0 = INTEGER: 216200 kB
UCD-SNMP-MIB::memSwapError.0 = INTEGER: noError(0)
UCD-SNMP-MIB::memSwapErrorMsg.0 = STRING:
```

負荷平均は、UCD SNMP MIB でも利用可能です。SNMP テーブル UCD-SNMP-MIB::laTable に

は、1 分、5 分、15 分間の負荷平均の一覧があります。

```
~]$ snmptable localhost UCD-SNMP-MIB::laTable
SNMP table: UCD-SNMP-MIB::laTable

 laIndex laNames laLoad laConfig laLoadInt laLoadFloat laErrorFlag laErrMessage
       1  Load-1   0.00    12.00         0    0.000000     noError
       2  Load-5   0.00    12.00         0    0.000000     noError
       3 Load-15   0.00    12.00         0    0.000000     noError
```

24.6.4.3. ファイルシステムとディスク情報

Host Resources MIB は、ファイルシステムのサイズと使用量についての情報を提供します。HOST-RESOURCES-MIB::hrStorageTable テーブルには、各ファイルシステム（および各メモリープール）のエントリーがあります。

```
~]$ snmptable -Cb localhost HOST-RESOURCES-MIB::hrStorageTable
SNMP table: HOST-RESOURCES-MIB::hrStorageTable

 Index                           Type          Descr
AllocationUnits    Size   Used AllocationFailures
     1          HOST-RESOURCES-TYPES::hrStorageRam Physical memory
1024 Bytes 1021588 388064                ?
     3 HOST-RESOURCES-TYPES::hrStorageVirtualMemory  Virtual memory
1024 Bytes 2045580 388064                ?
     6        HOST-RESOURCES-TYPES::hrStorageOther  Memory buffers
1024 Bytes 1021588  31048                ?
     7        HOST-RESOURCES-TYPES::hrStorageOther   Cached memory
1024 Bytes  216604 216604                ?
    10 HOST-RESOURCES-TYPES::hrStorageVirtualMemory      Swap space
1024 Bytes 1023992      0                ?
    31     HOST-RESOURCES-TYPES::hrStorageFixedDisk              /
4096 Bytes 2277614 250391                ?
    35     HOST-RESOURCES-TYPES::hrStorageFixedDisk        /dev/shm
4096 Bytes  127698      0                ?
    36     HOST-RESOURCES-TYPES::hrStorageFixedDisk          /boot
1024 Bytes  198337  26694                ?
```

HOST-RESOURCES-MIB::hrStorageSize および HOST-RESOURCES-MIB::hr StorageUsed の OID を使用して、マウントされた各ファイルシステムの残りの容量を算出することができます。

I/O データは UCD-SNMP-MIB::systemStats （ssIORawSent.0 と ssIORawRecieved.0）と UCD-DISKIO-MIB::diskIOTable の両方で利用できます。後者は、前者と比べてより粒度の細かいデータを提供します。このテーブルには、diskIONReadX および diskIONWrittenX の OID があり、システムブートから問題のブロックデバイスに対し読み取りおよび書き込みを実行したバイト数のカウンターを提供します。

```
~]$ snmptable -Cb localhost UCD-DISKIO-MIB::diskIOTable
```

```
SNMP table: UCD-DISKIO-MIB::diskIOTable

 Index Device     NRead  NWritten Reads Writes LA1 LA5 LA15    NReadX NWrittenX
...
   25    sda 216886272 139109376 16409   4894   ?   ?    ? 216886272 139109376
   26   sda1   2455552      5120   613      2   ?   ?    ?   2455552      5120
   27   sda2   1486848         0   332      0   ?   ?    ?   1486848         0
   28   sda3 212321280 139104256 15312   4871   ?   ?    ? 212321280 139104256
```

24.6.4.4. ネットワーク情報

Interfaces MIB はネットワークデバイスの情報を提供します。IF-MIB::ifTable は、SNMP テーブルにシステム上の各インターフェースのエントリー、インターフェースの設定、インターフェース用の各種パケットカウンターを提供します。以下の例は、2 つの物理ネットワークインターフェースを持つシステム上の ifTable の最初の数コラムを示しています。

```
~]$ snmptable -Cb localhost IF-MIB::ifTable
SNMP table: IF-MIB::ifTable

 Index Descr          Type   Mtu    Speed      PhysAddress AdminStatus
     1    lo softwareLoopback 16436 10000000                       up
     2  eth0   ethernetCsmacd  1500        0 52:54:0:c7:69:58         up
     3  eth1   ethernetCsmacd  1500        0 52:54:0:a7:a3:24       down
```

ネットワークトラフィックは、IF-MIB::ifOutOctets および IF-MIB::ifInOctets の OID で利用できます。以下の SNMP クエリーは、このシステム上の各インターフェースに対するネットワークトラフィックを取得します。

```
~]$ snmpwalk localhost IF-MIB::ifDescr
IF-MIB::ifDescr.1 = STRING: lo
IF-MIB::ifDescr.2 = STRING: eth0
IF-MIB::ifDescr.3 = STRING: eth1
~]$ snmpwalk localhost IF-MIB::ifOutOctets
IF-MIB::ifOutOctets.1 = Counter32: 10060699
IF-MIB::ifOutOctets.2 = Counter32: 650
IF-MIB::ifOutOctets.3 = Counter32: 0
~]$ snmpwalk localhost IF-MIB::ifInOctets
IF-MIB::ifInOctets.1 = Counter32: 10060699
IF-MIB::ifInOctets.2 = Counter32: 78650
IF-MIB::ifInOctets.3 = Counter32: 0
```

24.6.5. Net-SNMP の拡張

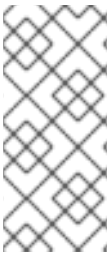
Net-SNMP Agent は、raw システムメトリックに加えてアプリケーションメトリックを提供するために拡張することができます。これにより、パフォーマンス問題のトラブルシューティングだけでなく容量計画も行うことができます。たとえば、試験中に電子メールシステムの 5 分の負荷平均が 15 であったことを把握しておくことは役に立つかもしれませんが、毎秒 80,000 メッセージの処理中に電子メールシステムの負荷平均が 15 であることを知っておく方がはるかに役立ちます。アプリケーションメトリックがシステムメトリックと同じインターフェースで使用可能な場合、システムパフォーマンス

の様々な負荷状況の影響も視覚化することができます (たとえば 10,000 メッセージが追加されると、負荷平均は 100,000 まで直線的に増加します)。

Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux に同梱されるアプリケーションの多くは、Net-SNMP Agent を拡張し、SNMP を介してアプリケーションメトリックを提供します。カスタムアプリケーション用にエージェントを拡張する方法もいくつかあります。本セクションでは、シェルスクリプトおよび Perl プラグインでエージェントを拡張する方法を説明します。これは、net-snmp-utils および net-snmp-perl パッケージがインストールされていることを前提としており、「認証の設定」 の説明に従ってユーザーに SNMP ツリーへのアクセスが許可されていることを前提としています。

24.6.5.1. シェルスクリプトによる Net-SNMP の拡張

Net-SNMP Agent は、任意のシェルスクリプトをクエリーするために使用できる拡張 MIB(NET-SNMP-EXTEND-MIB)を提供します。実行するシェルスクリプトを指定するには、/etc/snmp/snmpd.conf ファイルの extend ディレクティブを使用します。定義されると、Agent は SNMP により終了コードとコマンドの出力を提供します。以下の例は、プロセステーブルの httpd プロセスの数を決定するスクリプトを使用したこの仕組みを示しています。



PROC ディレクティブの使用

Net-SNMP Agent は、proc ディレクティブでプロセステーブルをチェックする組み込みメカニズムも提供します。詳細は、snmpd.conf(5)man ページを参照してください。

以下のシェルスクリプトの終了コードは、任意の時点におけるシステムで実行している httpd プロセスの数です。

```
#!/bin/sh

NUMPIDS=`pgrep httpd | wc -l`

exit $NUMPIDS
```

SNMP 経由でこのスクリプトを利用できるようにするには、スクリプトをシステムパス上の場所にコピーし、実行可能なビットを設定して、/etc/snmp/snmpd.conf ファイルに extend ディレクティブを追加します。extend ディレクティブの形式は次のとおりです。

```
extend name prog args
```

name は拡張の識別文字列、prog は実行するプログラム、args はプログラムを提供する引数です。たとえば、上記のシェルスクリプトが /usr/local/bin/check_apache.sh にコピーされた場合、以下

のディレクティブは SNMP ツリーにスクリプトを追加します。

```
extend httpd_pids /bin/sh /usr/local/bin/check_apache.sh
```

スクリプトは NET-SNMP-EXTEND-MIB::nsExtendObjects でクエリーできます。

```
~]$ snmpwalk localhost NET-SNMP-EXTEND-MIB::nsExtendObjects
NET-SNMP-EXTEND-MIB::nsExtendNumEntries.0 = INTEGER: 1
NET-SNMP-EXTEND-MIB::nsExtendCommand."httpd_pids" = STRING: /bin/sh
NET-SNMP-EXTEND-MIB::nsExtendArgs."httpd_pids" = STRING:
/usr/local/bin/check_apache.sh
NET-SNMP-EXTEND-MIB::nsExtendInput."httpd_pids" = STRING:
NET-SNMP-EXTEND-MIB::nsExtendCacheTime."httpd_pids" = INTEGER: 5
NET-SNMP-EXTEND-MIB::nsExtendExecType."httpd_pids" = INTEGER: exec(1)
NET-SNMP-EXTEND-MIB::nsExtendRunType."httpd_pids" = INTEGER: run-on-read(1)
NET-SNMP-EXTEND-MIB::nsExtendStorage."httpd_pids" = INTEGER: permanent(4)
NET-SNMP-EXTEND-MIB::nsExtendStatus."httpd_pids" = INTEGER: active(1)
NET-SNMP-EXTEND-MIB::nsExtendOutput1Line."httpd_pids" = STRING:
NET-SNMP-EXTEND-MIB::nsExtendOutputFull."httpd_pids" = STRING:
NET-SNMP-EXTEND-MIB::nsExtendOutNumLines."httpd_pids" = INTEGER: 1
NET-SNMP-EXTEND-MIB::nsExtendResult."httpd_pids" = INTEGER: 8
NET-SNMP-EXTEND-MIB::nsExtendOutLine."httpd_pids".1 = STRING:
```

終了コード（この例では「8」）は INTEGER タイプとして提供され、出力は STRING タイプとして提供されることに注意してください。複数のメトリクスを整数として公開するには、extend ディレクティブを使用してスクリプトに異なる引数を指定します。例えば、以下のシェルスクリプトを使用すると、任意の文字列に一致するプロセスの数を見つけ出すことができ、プロセスの数を示すテキスト文字列も出力します。

```
#!/bin/sh

PATTERN=$1
NUMPIDS=`pgrep $PATTERN | wc -l`

echo "There are $NUMPIDS $PATTERN processes."
exit $NUMPIDS
```

以下の /etc/snmp/snmpd.conf ディレクティブは、上記のスクリプトが /usr/local/bin/check_proc.sh にコピーされたときに httpd PID の数と snmpd PID の数の両方を提供します。

```
extend httpd_pids /bin/sh /usr/local/bin/check_proc.sh httpd
extend snmpd_pids /bin/sh /usr/local/bin/check_proc.sh snmpd
```

以下の例は、nsExtendObjects OID の snmpwalk の出力を示しています。

-

```
~]$ snmpwalk localhost NET-SNMP-EXTEND-MIB::nsExtendObjects
NET-SNMP-EXTEND-MIB::nsExtendNumEntries.0 = INTEGER: 2
NET-SNMP-EXTEND-MIB::nsExtendCommand."httpd_pids" = STRING: /bin/sh
NET-SNMP-EXTEND-MIB::nsExtendCommand."snmpd_pids" = STRING: /bin/sh
NET-SNMP-EXTEND-MIB::nsExtendArgs."httpd_pids" = STRING: /usr/local/bin/check_proc.sh
httpd
NET-SNMP-EXTEND-MIB::nsExtendArgs."snmpd_pids" = STRING:
/usr/local/bin/check_proc.sh snmpd
NET-SNMP-EXTEND-MIB::nsExtendInput."httpd_pids" = STRING:
NET-SNMP-EXTEND-MIB::nsExtendInput."snmpd_pids" = STRING:
...
NET-SNMP-EXTEND-MIB::nsExtendResult."httpd_pids" = INTEGER: 8
NET-SNMP-EXTEND-MIB::nsExtendResult."snmpd_pids" = INTEGER: 1
NET-SNMP-EXTEND-MIB::nsExtendOutLine."httpd_pids".1 = STRING: There are 8 httpd
processes.
NET-SNMP-EXTEND-MIB::nsExtendOutLine."snmpd_pids".1 = STRING: There are 1 snmpd
processes.
```

整数の終了コードは制限されています

整数の終了コードは 0 から 255 の範囲に制限されています。256 を超える可能性がある値については、スクリプトの標準出力 (文字列として入力されるもの) を使用するか、エージェントを拡張するという別の方法を実行してください。

この最後の例では、システムの空きメモリと httpd プロセスの数のクエリーを示しています。このクエリーは、パフォーマンステスト中にメモリ負担に与えるプロセス数の影響を知るために使用することができます。

```
~]$ snmpget localhost \
   'NET-SNMP-EXTEND-MIB::nsExtendResult."httpd_pids"' \
   UCD-SNMP-MIB::memAvailReal.0
NET-SNMP-EXTEND-MIB::nsExtendResult."httpd_pids" = INTEGER: 8
UCD-SNMP-MIB::memAvailReal.0 = INTEGER: 799664 kB
```

24.6.5.2. Perl による Net-SNMP の拡張

extend ディレクティブを使用したシェルスクリプトの実行は、SNMP によるカスタムアプリケーションメトリックを公開する非常に限定的な方法です。Net-SNMP エージェントは、カスタムオブジェクトを公開するための埋め込み Perl インターフェースも提供します。net-snmp-perl パッケージは、Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux で組み込み Perl プラグインを作成するために使用される NetSNMP::agent Perl モジュールを提供します。

NetSNMP::agent Perl モジュールは、エージェントの OID ツリーの一部に対する要求を処理する

ために使用される agent オブジェクトを提供します。agent オブジェクトのコンストラクターには、エージェントを snmpd のサブエージェントまたはスタンドアロンエージェントとして実行するためのオプションがあります。埋め込みエージェントを作成するために必要な引数はありません。

```
use NetSNMP::agent (':all');

my $agent = new NetSNMP::agent();
```

agent オブジェクトには、コールバック関数を特定の OID に登録するために使用される register メソッドがあります。register 関数は、名前、OID、コールバック関数へのポインターを取ります。以下の例では、hello_handler という名前のコールバック関数を OID .1.3.6 .1.4.1.8072.9999.9999 で要求を処理する SNMP Agent に登録しています。

```
$agent->register("hello_world", ".1.3.6.1.4.1.8072.9999.9999",
                 \&hello_handler);
```

ルート OID の取得

通常、OID .1.3.6.1.4.1.8072.9999.9999 (NET-SNMP-MIB::netSnmpPlaypen)はデモ目的でのみ使用されます。お客様の組織に root OID がない場合は、ISO Name Registration Authority (米国では ANSI) にご連絡いただくと取得できます。

ハンドラー関数は、HANDLER、REGISTRATION_INFO、REQUEST_INFO、および REQUESTS の 4 つのパラメーターで呼び出されます。REQUESTS パラメーターには、現在の呼び出しの要求一覧が含まれており、反復されデータが追加されるはずです。一覧の request オブジェクトには get メソッドおよび set メソッドがあるため、リクエストの OID および value を操作することができます。たとえば、以下の呼び出しは要求オブジェクトの値を文字列 「hello world」 に設定します。

```
$request->setValue(ASN_OCTET_STR, "hello world");
```

ハンドラー関数は、GET 要求と GETNEXT 要求という 2 種類の SNMP 要求に応答できます。要求のタイプは、ハンドラー関数に第 3 のパラメーターとして渡される getMode オブジェクトの request_info メソッドを呼び出すことで決定されます。要求が GET 要求である場合、呼び出し元は、ハンドラーに要求の OID に応じて value オブジェクトの request を設定するよう求めます。要求が GETNEXT 要求である場合、呼び出し元は、ハンドラーに要求の OID をツリー内で次に利用可能な OID に設定するよう求めます。以下のコードは、この例を示しています。

```
my $request;
my $string_value = "hello world";
my $integer_value = "8675309";

for($request = $requests; $request; $request = $request->next()) {
  my $oid = $request->getOID();
  if ($request_info->getMode() == MODE_GET) {
    if ($oid == new NetSNMP::OID(".1.3.6.1.4.1.8072.9999.9999.1.0")) {
```

```
    $request->setValue(ASN_OCTET_STR, $string_value);
   }
   elsif ($oid == new NetSNMP::OID(".1.3.6.1.4.1.8072.9999.9999.1.1")) {
    $request->setValue(ASN_INTEGER, $integer_value);
   }
  } elsif ($request_info->getMode() == MODE_GETNEXT) {
   if ($oid == new NetSNMP::OID(".1.3.6.1.4.1.8072.9999.9999.1.0")) {
    $request->setOID(".1.3.6.1.4.1.8072.9999.9999.1.1");
    $request->setValue(ASN_INTEGER, $integer_value);
   }
   elsif ($oid < new NetSNMP::OID(".1.3.6.1.4.1.8072.9999.9999.1.0")) {
    $request->setOID(".1.3.6.1.4.1.8072.9999.9999.1.0");
    $request->setValue(ASN_OCTET_STR, $string_value);
   }
  }
}
```

getMode が MODE_GET を返すと、ハンドラーは getOID オブジェクトの request 呼び出しの値を分析します。value の request は、OID が .1 「.0」 で終わる場合は string_value に、OID が .1.1 で終わる場合は integer_value に設定され 「ます。」getMode が MODE_GETNEXT を返す場合、ハンドラーは要求の OID が .1.0 かどうかを判断し、「.」 1.1 の OID と値を設定し 「ます。」ツリーで要求が .1.0 を超える場合、「.1.0」 の OID および値が 「設定」 されます。実際、これはツリーの 「次」 の値を返すため、snmpwalk のようなプログラムは構造に関する事前知識なくツリーをトラバースできます。

変数のタイプは NetSNMP::ASN からの定数を使用して設定されます。利用可能な定数の全一覧については、NetSNMP::ASN の perldoc を参照してください。

この例の Perl プラグインのコード全一覧は、以下のとおりです:

```
#!/usr/bin/perl

use NetSNMP::agent (':all');
use NetSNMP::ASN qw(ASN_OCTET_STR ASN_INTEGER);

sub hello_handler {
 my ($handler, $registration_info, $request_info, $requests) = @_;
 my $request;
 my $string_value = "hello world";
 my $integer_value = "8675309";

 for($request = $requests; $request; $request = $request->next()) {
  my $oid = $request->getOID();
  if ($request_info->getMode() == MODE_GET) {
   if ($oid == new NetSNMP::OID(".1.3.6.1.4.1.8072.9999.9999.1.0")) {
    $request->setValue(ASN_OCTET_STR, $string_value);
   }
   elsif ($oid == new NetSNMP::OID(".1.3.6.1.4.1.8072.9999.9999.1.1")) {
    $request->setValue(ASN_INTEGER, $integer_value);
   }
```

```
    } elsif ($request_info->getMode() == MODE_GETNEXT) {
      if ($oid == new NetSNMP::OID(".1.3.6.1.4.1.8072.9999.9999.1.0")) {
        $request->setOID(".1.3.6.1.4.1.8072.9999.9999.1.1");
        $request->setValue(ASN_INTEGER, $integer_value);
      }
      elsif ($oid < new NetSNMP::OID(".1.3.6.1.4.1.8072.9999.9999.1.0")) {
        $request->setOID(".1.3.6.1.4.1.8072.9999.9999.1.0");
        $request->setValue(ASN_OCTET_STR, $string_value);
      }
    }
  }
}

my $agent = new NetSNMP::agent();
$agent->register("hello_world", ".1.3.6.1.4.1.8072.9999.9999",
                 \&hello_handler);
```

プラグインをテストするには、上記のプログラムを /usr/share/snmp/hello_world.pl にコピーし、以下の行を /etc/snmp/snmpd.conf 設定ファイルに追加します。

```
perl do "/usr/share/snmp/hello_world.pl"
```

新しい Perl プラグインをロードするためには、SNMP Agent Daemon を再起動する必要があります。再起動したら、snmpwalk が新しいデータを返すはずです。

```
~]$ snmpwalk localhost NET-SNMP-MIB::netSnmpPlaypen
NET-SNMP-MIB::netSnmpPlaypen.1.0 = STRING: "hello world"
NET-SNMP-MIB::netSnmpPlaypen.1.1 = INTEGER: 8675309
```

snmpget を使用すると、ハンドラーの他のモードを使用することも可能です。

```
~]$ snmpget localhost \
   NET-SNMP-MIB::netSnmpPlaypen.1.0 \
   NET-SNMP-MIB::netSnmpPlaypen.1.1
NET-SNMP-MIB::netSnmpPlaypen.1.0 = STRING: "hello world"
NET-SNMP-MIB::netSnmpPlaypen.1.1 = INTEGER: 8675309
```

24.7. 関連資料

システム情報の収集方法についてさらに知るには、以下のリソースを参照してください。

24.7.1. インストールされているドキュメント

- ps(1): ps コマンドの man ページです。

- *top(1): top コマンドの man ページです。*

- *Free(1)- free コマンドの man ページです。*

- *df(1)- df コマンドの man ページです。*

- *du(1): du コマンドの man ページです。*

- *lspci(8): lspci コマンドの man ページです。*

- *snmpd(8): snmpd サービスの man ページです。*

- *snmpd.conf(5): 利用可能な設定ディレクティブについての全ドキュメントを含む /etc/snmp/snmpd.conf ファイルの man ページです。*

第25章 ログファイルの表示と管理

ログファイル は、システム（カーネル、サービス、および実行中のアプリケーションなど）に関するメッセージが含まれるファイルです。各情報にはそれぞれ異なるログファイルがあります。例えば、デフォルトのシステムログファイル、セキュリティメッセージ専用のログファイル、cron タスク用のログファイルなどです。

ログファイルは、カーネルドライバーのロードを試行するなどシステムの問題を解決する場合や、システムへの無許可のログイン試行を探す場合に役立ちます。本章では、ログファイルの場所、ログファイルの閲覧方法、ログファイルの注意すべき項目を説明します。

一部のログファイルは、rsyslogd と呼ばれるデーモンによって制御されます。rsyslogd デーモンは、以前の sysklogd の拡張版であり、拡張されたフィルタリング、暗号化で保護されたメッセージのリレー、様々な設定オプション、入出力モジュール、TCP プロトコルまたは UDP プロトコルを介した転送のサポートを提供します。rsyslog は sysklogd と互換性があることに注意してください。

25.1. RSYSLOG のインストール

rsyslog パッケージで提供される rsyslog のバージョン 5 は、Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;LinuxRed Hat Enterprise Linux 6nbsp;6 にデフォルトでインストールされます。必要な場合は、インストールされていることを確認するために root で以下のコマンドを発行します。

```
~]# yum install rsyslog
Loaded plugins: product-id, refresh-packagekit, subscription-manager
Package rsyslog-5.8.10-10.el6_6.i686 already installed and latest version
Nothing to do
```

25.1.1. rsyslog バージョン 7 へのアップグレード

rsyslog7 パッケージで提供される rsyslog のバージョン 7 は、Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;LinuxRed Hat Enterprise Linux 6nbsp;6 で利用できます。バージョン 5 に対する機能強化が数多く追加されました。特に、より多くのプラグインの処理とサポートを特に改善しています。必要に応じて、バージョン 7 に変更するには、以下のように yum shell ユーティリティーを使用します。

手順25.1 rsyslog 7 へのアップグレード

rsyslog バージョン 5 から rsyslog バージョン 7 にアップグレードするには、関連するパッケージを同時にインストールおよび削除する必要があります。これは、yum shell ユーティリティーを使用し

て実行できます。

1. root で以下のコマンドを入力し、yum シェルを起動します。

```
~]# yum shell
Loaded plugins: product-id, refresh-packagekit, subscription-manager
>
```

yum シェルプロンプトが表示されます。

2. 以下のコマンドを入力して rsyslog7 パッケージをインストールし、rsyslog パッケージを削除します。

```
> install rsyslog7
> remove rsyslog
```

3. run を入力してプロセスを開始します。

```
> run
--> Running transaction check
---> Package rsyslog.i686 0:5.8.10-10.el6_6 will be erased
---> Package rsyslog7.i686 0:7.4.10-3.el6_6 will be installed
--> Finished Dependency Resolution

================================================================================
=
 Package       Arch     Version          Repository                Size
================================================================================
=
Installing:
 rsyslog7      i686     7.4.10-3.el6_6   rhel-6-workstation-rpms    1.3 M
Removing:
 rsyslog       i686     5.8.10-10.el6_6  @rhel-6-workstation-rpms   2.1 M

Transaction Summary
================================================================================
=
Install  1 Package
Remove   1 Package

Total download size: 1.3 M
Is this ok [y/d/N]:y
```

4.
アップグレードを開始するプロンプトが出されたら、y を入力します。

5.
アップグレードが完了すると、yum シェル プロンプトが表示されます。exit または exit を入力してシェルを終了します。

```
Finished Transaction
> quit
Leaving Shell
~]#
```

rsyslog バージョン 7 が提供する新しい構文の使用方法は、「新規設定フォーマットの使用」 を参照してください。

25.2. ログファイルの場所の特定

rsyslogd が維持するログファイルの一覧は、/etc/rsyslog.conf 設定ファイルにあります。ほとんどのログファイルは /var/log/ ディレクトリーにあります。httpd や samba などの一部のアプリケーションでは、ログファイル用のディレクトリーが /var/log/ 内にあります。

/var/log/ ディレクトリー内には番号が付いた複数のファイル（例： cron-20100906）があることに気付くかもしれません。これらの番号はローテーションを行ったログファイルに追加されたタイムスタンプを表します。ログファイルは、ファイルサイズが大きくなり過ぎないようにローテーションが行われます。logrotate パッケージには cron タスクが含まれており、/etc/logrotate.conf 設定ファイルと /etc/logrotate.d/ ディレクトリー内の設定ファイルに従って自動的にログファイルをローテーションします。

25.3. RSYSLOG の基本設定

rsyslog の主な設定ファイルは /etc/rsyslog.conf です。ここでは、グローバルディレクティブ、モジュール、および フィルター 部分および アクション 部分で構成される ルール を指定できます。また、ハッシュ記号 (#) の後にテキスト形式でコメントを追加することもできます。

25.3.1. フィルター

ルールは、syslog メッセージのサブセットを選択する フィルター の部分と、選択したメッセージで何をするかを指定する アクション の部分で指定されます。/etc/rsyslog.conf 設定ファイルでルールを定義するには、フィルターとアクションの両方を 1 行で定義し、1 つ以上の空白またはタブで区切ります。

rsyslog は、選択したプロパティーに従って syslog メッセージをフィルターする様々な方法を提供します。利用可能なフィルタリングの方法は、Facility/Priority ベース、Property ベース、Expression ベース の 3 種類のフィルターに分けられます。

Facility (ファシリティー) /Priority (優先度) ベースのフィルター

syslog メッセージのフィルターに最もよく知られた方法は、syslog メッセージをフィルターするファシリティー/優先度ベースのフィルターを使用して、2 つの条件（ ファシリティー および 優先度 をドットで区切る）を使用することです。セレクターを作成するには以下の構文を使用します。

```
FACILITY.PRIORITY
```

詳細は以下のようになります。

- FACILITY は、特定の syslog メッセージを生成するサブシステムを指定します。たとえば、mail サブシステムはメール関連のすべての syslog メッセージを処理します。FACILITY は、以下のキーワード（または数値コード）のいずれかで表示できます。kern (0), user (1), mail (2), daemon (3), auth (4), syslog (5), lpr (6)、news (7)、uucp (8)、cron (9)、authpriv (10)、ftp (11)、および local0 through local7 (16 - 23)

- PRIORITY は、syslog メッセージの優先度を指定します。PRIORITY は以下のキーワード（または数字）のいずれかで表示できます。debug (7)、info (6)、notice (5)、warning (4)、err (3)、crit (2)、alert (1)、および emerg (0)。

上記の構文は、定義された優先度もしくは より高い 優先度で syslog メッセージを選択します。いずれかの優先度のキーワード前に等号 (=) を付けると、指定された優先度の syslog メッセージのみ選択されることが指定されます。他のすべての優先度は無視されます。反対に、感嘆符 (!) を優先度のキーワードの前に付けると、この優先度以外のすべての syslog メッセージが選択されます。

上記で指定されているキーワード以外に、アスタリスク (*) を使用してすべてのファシリティーもしくは優先度を定義することもできます (アスタリスクをコンマの前か後に配置するかによる)。優先度キーワード none を指定すると、優先度のないファシリティーを指定することになります。ファシリティーおよび優先度の条件は、どちらも大文字と小文字を区別しません。

定義するファシリティーや優先度が複数になる場合は、コンマ (,) を使用して区切ります。同じ行に複数のセレクターを定義するには、セミコロン (;) を使用して区切ります。セレクターフィールド内の各セレクターは、以前のものを上書きすることに注意してください。これにより、パターンから優先度が除外される可能性があります。

例25.1 ファシリティー/優先度ベースのフィルター

以下は、/etc/rsyslog.conf で指定できる単純な facility/priority ベースのフィルターの例です。優先度ですべてのカーネル syslog メッセージを選択するには、設定ファイルに以下のテキストを追加します。

```
kern.*
```

優先度が crit 以上になるメール syslog メッセージすべてを選択するには、以下の形式を使用します。

```
mail.crit
```

info または debug の優先度以外のすべての cron syslog メッセージを選択するには、設定ファイルで以下の形式を設定します。

```
cron.!info,!debug
```

プロパティーベースのフィルター

プロパティーベースのフィルターを使用すると、timegenerated や syslogtag などのプロパティーで syslog メッセージをフィルターできます。プロパティーに関する詳細情報は、「プロパティー」 を参照してください。指定された各プロパティーは、表25.1「プロパティベースの比較処理」 で一覧表示されている比較処理のいずれかを使用して特定の値に対して比較できます。プロパティー名と比較処理はどちらも大文字と小文字を区別します。

プロパティーベースのフィルターは、コロン (:) で開始する必要があります。フィルターの定義には、以下の構文を使用します。

```
:PROPERTY, [!]COMPARE_OPERATION, "STRING"
```

詳細は以下のようになります。

- PROPERTY 属性は希望するプロパティーを指定します。
- オプションの感嘆符 (!) は比較処理の出力を無効にします。他のブール値演算子は現在、プロパティーベースのフィルターではサポートされていません。

- COMPARE_OPERATION 属性は、表25.1「プロパティベースの比較処理」 に記載の比較処理のいずれかを指定します。

- STRING 属性は、プロパティーが提供するテキストの比較先となる値を指定します。この値は、引用符で囲む必要があります。この文字列内の特定の文字をエスケープさせるには (たとえば、引用符 (")、バックスラッシュ (\) を使用します。

表25.1 プロパティベースの比較処理

比較処理	詳細
contains	提供された文字列が、プロパティーで提供されたテキストのいずれかの部分に適合するかどうかをチェックします。大文字と小文字を区別しない比較を実行するには、contains_i を使用します。
isequal	用意された文字列をプロパティーで提供されたすべてのテキストと比較します。これら 2 つの値が適合するには、完全に等しいものである必要があります。
startswith	提供された文字列が、プロパティーで提供されたテキストのちょうど最初にあるかどうかをチェックします。大文字と小文字を区別しない比較を実行するには、startswith_i を使用します。
regex	指定された POSIX BRE (Basic Regular Expression) をプロパティーが提供したテキストと比較します。
ereregex	指定された POSIX ERE (Extended Regular Expression) 正規表現をプロパティーが提供したテキストと比較します。
isempty	プロパティーが空かどうかをチェックします。値は破棄されます。これは、いくつかのフィールドが正規化の結果に基づいて設定される正規化データでの作業時に特に有用です。

例25.2 プロパティーベースのフィルター

以下は、/etc/rsyslog.conf で指定できるプロパティーベースのフィルターの例です。syslog メッセージのテキストに文字列 error が含まれているものを選択するには、以下を使用します。

```
:msg, contains, "error"
```

以下のフィルターは、ホスト名 host1 から受信した syslog メッセージを選択します。

```
:hostname, isequal, "host1"
```

(fatal lib error など) fatal と error の間にテキストがあるかどうかに関わらず、これらを含まない syslog メッセージを選択するには、以下を入力します。

```
:msg, !regex, "fatal .* error"
```

式ベースのフィルター

式ベースのフィルターは、定義されている算術演算、ブール演算、または文字列演算に従って syslog メッセージを選択します。式ベースのフィルターは、rsyslog の独自のスクリプト言語 RainerScript を使用して複雑なフィルターを構築します。

式ベースのフィルターの基本的な構文は、以下のようになります。

```
if EXPRESSION then ACTION else ACTION
```

詳細は以下のようになります。

- EXPRESSION 属性は、$msg startswith 'DEVNAME' または $syslogfacility-text == 'local0' などの評価される式を表します。and および or 演算子を使用して、1 つのフィルターに複数の式を指定できます。
- ACTION 属性は、式が true の値を返す場合に実行されるアクションを表します。これは単一のアクションの場合と、波括弧で囲まれた任意の複雑なスクリプトになる場合があります。
- 式ベースのフィルターは、行の最初の if キーワードで示されます。then キーワードは、EXPRESSION を ACTION から分離します。オプションで、else キーワードを使用して条件が満たされない場合に実行されるアクションを指定することもできます。

式ベースのフィルターでは、例25.3「式ベースのフィルター」 にあるように、波括弧に囲まれ

た式を使うことで条件をネスト化することができます。このスクリプトでは、式内で facility/priority-based フィルターを使用することができます。ただし、ここで property-based フィルターを使用することは推奨されません。RainerScript は、特殊な関数 re_match() および re_extract() を含む正規表現をサポートします。

例25.3 式ベースのフィルター

以下の式には、ネスト化された条件が 2 つ含まれています。prog1 と呼ばれるプログラムが生成したログファイルが、メッセージ内の文字列 "test" の有無に基づいて 2 つのファイルに分割されます。

```
if $programname == 'prog1' then {
  action(type="omfile" file="/var/log/prog1.log")
  if $msg contains 'test' then
    action(type="omfile" file="/var/log/prog1test.log")
  else
    action(type="omfile" file="/var/log/prog1notest.log")
}
```

様々な式ベースのフィルターの他の例は、「オンラインドキュメント」 を参照してください。RainerScript は rsyslog の新規設定形式の基盤となります。を参照してください。 「新規設定フォーマットの使用」

25.3.2. アクション

アクションは、明確に定義されたセレクターでフィルターされたメッセージで何を実行するかを指定します。以下にルール内で定義できるアクションをいくつか示します。

ログファイルへの syslog メッセージの保存

アクションの大半は、どのログファイルに syslog メッセージを保存するかを指定します。これは定義済みセレクターの後にファイルパスを指定することで行います。

FILTER PATH

ここでの FILTER はユーザーが指定したセレクターを表し、PATH はターゲットファイルのパスを表します。

たとえば、以下のルールは、すべての cron syslog メッセージを選択するセレクターと、それらを /var/log/cron.log ログファイルに保存するアクションで構成されています。

```
cron.* /var/log/cron.log
```

デフォルトでは、syslog メッセージの生成時に毎回ログファイルは同期されます。同期を省略する場合は、ダッシュ記号 (-) を該当するファイルパスの接頭辞として使います。

```
FILTER -PATH
```

書き込みの直後にシステムが終了すると、情報が失われる場合があることに注意してください。ただし、この設定では、特に非常に詳細なログメッセージを生成するプログラムを実行する場合には、パフォーマンスも改善されます。

指定したファイルパスは、static または dynamic のいずれかになります。静的ファイルは、上記の例で示されているように固定ファイルパスで示されます。動的ファイルパスは、受け取ったメッセージによって異なります。動的ファイルパスは、テンプレートと疑問符 (?) の接頭辞で示されます。

```
FILTER ?DynamicFile
```

ここでは、DynamicFile は出力パスを変更する事前定義テンプレートの名前です。ダッシュ記号 (-) の接頭辞を使うと同期を無効にでき、またコロン (;) 区切りで複数のテンプレートを使用できます。テンプレートに関する詳細は 「動的なファイル名の生成」 を参照してください。

指定したファイルが既存の terminal または /dev/console デバイスである場合、syslog メッセージは標準出力（特別な terminal処理を使用）へ送信されるか、X Window システムの使用時には（特別な /dev/console-handling を使用）標準出力に送信されます。

ネットワークを使った syslog メッセージの送信

rsyslog を使用すると、ネットワーク経由で syslog メッセージを送受信できます。この機能により、1 台のマシンで複数ホストの syslog メッセージを管理できます。syslog メッセージをリモートマシンに転送するには、以下の構文を使用します。

```
@[(zNUMBER)]HOST:[PORT]
```

詳細は以下のようになります。

- アットマーク(@)は、syslog メッセージが UDP プロトコルを使用してホストへ転送

されることを示します。TCP プロトコルを使用するには、2 つのアットマークを空白なしで使用します(@@)。

- オプションの zNUMBER 設定を使用すると、syslog メッセージの zlib 圧縮が可能になります。NUMBER 属性は圧縮のレベルを指定します（最低 1 から 9 まで）。圧縮が得られた値は rsyslogd によって自動的にチェックされます。メッセージが圧縮されるのは圧縮され、60 バイト未満のメッセージは圧縮されません。

- HOST 属性は、選択した syslog メッセージを受信するホストを指定します。

- PORT 属性は、ホストマシンのポートを指定します。

IPv6 アドレスをホストとして指定する場合は、アドレスを角括弧([,])で囲みます。

例25.4 ネットワークを使用した syslog メッセージの送信

以下の例は、ネットワーク上で syslog メッセージを転送するアクションです (注記: すべてのアクションの前には、いずれかの優先度を持つすべてのメッセージを選択するセレクターが付いています)。UDP プロトコルを介してメッセージを 192.168.0.1 に転送するには、以下を入力します。

```
*.* @192.168.0.1
```

ポート 6514 と TCP プロトコルを使用してメッセージを "example.com" に転送するには、以下を使用します。

```
*.* @@example.com:6514
```

以下ではメッセージを zlib （レベル 9 圧縮）で圧縮し、UDP プロトコルを使用して 2001:db8::1 に転送します。

```
*.* @(z9)[2001:db8::1]
```

出力チャンネル

出力チャンネルは主にログファイルの最大サイズを指定するために使用されます。これは、ログファイルのローテーションに非常に便利です (詳細は 「ログローテーション」 を参照してくださ

い）。出力チャンネルは基本的に出力アクションに関する情報を集めたものです。出力チャンネルは、**$outchannel** ディレクティブで定義されます。**/etc/rsyslog.conf** で出力チャンネルを定義するには、以下の構文を使用します。

```
$outchannel NAME, FILE_NAME, MAX_SIZE, ACTION
```

詳細は以下のようになります。

- **NAME** 属性は、出力チャンネル名を指定します。
- **FILE_NAME** 属性は、出力ファイル名を指定します。出力チャンネルはファイルにのみ書き込み可能で、パイプやターミナル、その他の出力には書き込みできません。
- **MAX_SIZE** 属性は、（**FILE_NAME**内の）指定されたファイルの最大サイズを表します。この値は バイト 単位で指定します。
- **ACTION** 属性は、**MAX_SIZE** で定義された最大サイズに達すると実行するアクションを指定します。

定義済みの出力チャンネルをルール内のアクションとして使用するには、以下を入力します。

```
FILTER :omfile:$NAME
```

例25.5 出力チャンネルのログローテーション

以下の出力は、出力チャンネルを使用した簡単なログローテーションを示しています。まず、出力チャンネルは **$outchannel** ディレクティブで定義されます。

```
$outchannel log_rotation, /var/log/test_log.log, 104857600, /home/joe/log_rotation_script
```

その後に、優先度を持つすべての **syslog** メッセージを選択し、取得した **syslog** メッセージ上の事前定義された出力チャンネルを実行するルール内で使用されます。

```
*.* :omfile:$log_rotation
```

制限（この例では **100 MB**）に達すると、**/home/joe/log_rotation_script** が実行されます。このスクリプトには、ファイルを異なるフォルダーに移動することやその中の特別なコンテ

ンツを編集すること、単にそれを削除することなど、様々なタスクを含めることができます。

特定ユーザーへの syslog メッセージの送信

Rsyslog は、（ 例25.7「複数アクションの指定」のように）メッセージを送信するユーザーのユーザー名を指定することで、syslog メッセージを特定のユーザーに送信できます。複数のユーザーを指定するには、各ユーザー名をコンマ (,) で区切ります。現在ログオンしている全ユーザーにメッセージを送るには、アスタリスク (*) を使用します。

プログラムの実行

rsyslog を使用すると、選択した syslog メッセージに対してプログラムを実行し、system（） 呼び出しを使用してシェルでプログラムを実行できます。実行するプログラムを指定するには、そのプログラムの前に キャレット文字(^) を付けます。その後に、受信したメッセージをフォーマットしてそれを 1 行のパラメーターとして指定した実行ファイルに渡すテンプレートを指定します (テンプレートに関する詳細は 「テンプレート」 を参照)。

```
FILTER ^EXECUTABLE; TEMPLATE
```

ここでは、FILTER 条件の出力は EXECUTABLE で表されるプログラムによって処理されます。このプログラムは、有効な実行ファイルであればどれでも構いません。TEMPLATE をフォーマットするテンプレートの名前に置き換えます。

例25.6 プログラムの実行

以下の例では、すべての優先度の syslog メッセージが選択され、template テンプレートでフォーマットされ、パラメーターとして test-program プログラムに渡されます。その後、提供されているパラメーターで実行されます。

```
*.* ^test-program;template
```

シェル実行アクションを使用する場合は注意してください。

ホストからメッセージを受信して、シェル実行アクションを使用する際には、コマンドインジェクションに対する脆弱性があります。ユーザーが自身のアクションで実行されるように指定しているプログラム内で、攻撃者が別のコマンドの挿入と実行を試みる可能性があります。セキュリティー脅威の可能性を回避するには、シェル実行アクションの使用をよく検討してください。

syslog メッセージのデータベースでの保存

選択された syslog メッセージは、データベースライター の動作を使用して、直接データベーステーブルに書き込むことができます。データベースライターは、以下の構文を使用します。

```
:PLUGIN:DB_HOST,DB_NAME,DB_USER,DB_PASSWORD;[TEMPLATE]
```

詳細は以下のようになります。

- *PLUGIN はデータベースの書き込みを処理する指定プラグインを呼び出します（ ommysql プラグインなど）。*
- *DB_HOST 属性はデータベースのホスト名を指定します。*
- *DB_NAME 属性はデータベースの名前を指定します。*
- *DB_USER 属性はデータベースユーザーを指定します。*
- *DB_PASSWORD 属性は上記のデータベースユーザーで使用されるパスワードを指定します。*
- *TEMPLATE 属性は syslog メッセージを変更するテンプレートのオプション使用を指定します。テンプレートに関する詳細は 「テンプレート」 を参照してください。*

MYSQL および POSTGRESQL の使用

現在、rsyslog は MySQL および PostgreSQL データベースにのみ対応しています。MySQL および PostgreSQL のデータベースライター機能を使用するには、rsyslog-mysql および rsyslog-pgsql パッケージをそれぞれインストールします。また、/etc/rsyslog.conf 設定ファイルに適切なモジュールを読み込んでください。

```
$ModLoad ommysql    # Output module for MySQL support
$ModLoad ompgsql    # Output module for PostgreSQL support
```

rsyslog モジュールの詳細は、「Rsyslog モジュールの使用」 を参照してください。

または、omlibdb モジュールが提供する汎用のデータベースインターフェースを使用することもできます（サポート対象： Firebird/Interbase、MS SQL、Sybase、SQLLite、Ingres、Oracle、mSQL）。

syslog メッセージの破棄

選択したメッセージを破棄する場合は、チルダ文字(~)を使用します。

```
FILTER ~
```

破棄するアクションは、ほとんどの場合、さらに処理をする前にメッセージをフィルタリングするために使用されます。繰り返されるメッセージを省略しなければログファイルがいっぱいになってしまう場合に、これは効果的です。破棄アクションの結果は、指定された設定ファイル内の場所によって異なります。最善の結果を得るためにも、これらのアクションは、アクションリストの上に配置します。一旦破棄したメッセージを、設定ファイルの後の行で回復することはできないことに注意してください。

たとえば、以下のルールは cron syslog メッセージを破棄します。

```
cron.* ~
```

複数アクションの指定

各セレクターで、複数のアクションを指定できます。1 つのセレクターに複数アクションを指定するには、各アクションを別々の行に書き込んでそれらの先頭にアンパサンド文字 (&) を付けます。

```
FILTER ACTION
& ACTION
& ACTION
```

指定されたセレクターが評価されるのは 1 回のみであるため、複数の動作を指定すると、希望する結果の全体的なパフォーマンスが向上します。

例25.7 複数アクションの指定

以下の例では、重要な優先度(crit)を持つすべてのカーネル syslog メッセージはユーザー user1 に送信され、テンプレート temp によって処理され、test-program 実行可能ファイルに渡され、UDP プロトコルを介して 192.168.0. 1 に転送されます。

```
kern.=crit user1
& ^test-program;temp
& @192.168.0.1
```

すべてのアクションで、メッセージをフォーマットするテンプレートが後に続きます。テンプレートを指定するには、アクションの末尾にセミコロン (;) を付け、続けてテンプレートの名前を指定します。テンプレートに関する詳細は 「テンプレート」 を参照してください。

テンプレートの使用

テンプレートはアクションで使用される前に定義する必要があり、アクションの後に定義しても無視されます。つまり、/etc/rsyslog.conf では、テンプレート定義が常にルール定義の前にくるようにする必要があります。

25.3.3. テンプレート

rsyslog で生成される出力はすべて、テンプレート を使用して、ニーズに合わせて変更およびフォーマットできます。テンプレートを作成するには、/etc/rsyslog.conf で以下の構文を使用します。

```
$template TEMPLATE_NAME,"text %PROPERTY% more text", [OPTION]
```

ここでは、以下のようになります。

- *$template は、その後のテキストを示すテンプレートディレクティブで、テンプレートを定義します。*

- *TEMPLATE_NAME テンプレートの名前です。この名前は、テンプレートを参照します。*

- *2 つの引用符（「...」）の間はすべて実際のテンプレートテキストです。このテキスト内では、改行文字の \n、キャリッジリターンの \r などの特殊文字を使用できます。% または " などのその他の文字を使用する場合は、それらの文字を文字どおりエスケープする必要があります。*

- *2 つのパーセントマーク(%)の間にあるテキストは、syslog メッセージの特定のコンテンツにアクセスできるようにする プロパティー を指定します。プロパティーに関する詳細情報は、「プロパティー」 を参照してください。*

- *OPTION 属性は、テンプレート機能を修正するオプションを指定します。現在サポートされているテンプレートオプションは、テキストを SQL クエリーとしてフォーマットするのに使用される sql および stdsql です。*

SQL オプションおよび STDSQL オプション

データベースライターは、sql オプションまたは stdsql オプションがテンプレートで指定されているかどうかをチェックします。指定されていないと、データベースライターはアクションを実行しません。これは SQL インジェクションなどのセキュリティーの脅威を回避するためです。

詳細は、『「アクション」 の「syslog メッセージのデータベースで』 の保存」を参照してください。

動的なファイル名の生成

テンプレートを使用して、動的なファイル名を生成できます。プロパティーをファイルパスの一部として指定すると、それぞれ一意のプロパティーに対して新しいファイルが作成されます。これは、syslog メッセージを分類する便利な方法です。

たとえば、メッセージからタイムスタンプを抽出する timegenerated プロパティーを使用して、各 syslog メッセージに一意のファイル名を生成します。

```
$template DynamicFile,"/var/log/test_logs/%timegenerated%-test.log"
```

$template ディレクティブは単にテンプレートを指定するだけです。効果を反映するためには、それをルール内で使用しなければなりません。/etc/rsyslog.conf で、アクション定義でクエスチョンマーク(?)を使用して、動的ファイル名テンプレートをマークします。

```
*.* ?DynamicFile
```

プロパティー

テンプレート内（2 つのパーセントマーク(%)の内側）で定義されたプロパティーにより、プロパティー置換関数 を使用して syslog メッセージの各種コンテンツにアクセスできるようになります。テンプレート内（2 つの引用符（「...」）でプロパティーを定義するには、以下の構文を使用します。

```
%PROPERTY_NAME[:FROM_CHAR:TO_CHAR:OPTION]%
```

ここでは、以下のようになります。

- *PROPERTY_NAME 属性は、プロパティー名を指定します。利用可能なすべてのプロパティーとその詳細な説明は、man ページの rsyslog.conf(5) の「Available Properties 」セクションにあります。*
- *FROM_CHAR 属性および TO_CHAR 属性は、指定したプロパティーが動作する文字の範囲を表します。他の方法では、正規表現を使用して文字の範囲を指定することもできます。これを行うには、文字 R を FROM_CHAR 属性として設定し、希望する正規表現を TO_CHAR 属性として指定します。*
- *OPTION 属性は、入力を 小 文字に変換する lowercase オプションなどのプロパティーオプションを指定します。利用可能なすべてのプロパティーオプションとその詳細な説明は、man ページの rsyslog.conf(5) の Property Options セクションでご覧になれます。*

簡単なプロパティーの例を以下に示します。

- *以下のプロパティーは、syslog メッセージのメッセージテキスト全体を取得します。*

  ```
  %msg%
  ```

- *以下のプロパティは、syslog メッセージにあるメッセージテキストの最初の 2 文字を取得します。*

```
%msg:1:2%
```

- 以下のプロパティは、syslog メッセージの全メッセージテキストを取得して、最後のラインフィード文字を省きます。

```
%msg:::drop-last-lf%
```

- 以下のプロパティーは、syslog メッセージの受信時に生成されるタイムスタンプの最初の 10 文字を取得し、『RFC 3999』 日付標準に従ってフォーマットします。

```
%timegenerated:1:10:date-rfc3339%
```

テンプレートの例

このセクションでは、rsyslog テンプレートの例をいくつか紹介します。

例25.8「詳細な syslog メッセージのテンプレート」 は、syslog メッセージをフォーマットし、メッセージの重大度、ファシリティー、メッセージの受信時のタイムスタンプ、ホスト名、メッセージタグ、メッセージテキストを出力し、改行で終了するようにテンプレートを示しています。

例25.8 詳細な syslog メッセージのテンプレート

```
$template verbose, "%syslogseverity%, %syslogfacility%, %timegenerated%,
%HOSTNAME%, %syslogtag%, %msg%\n"
```

例25.9「ウォールメッセージのテンプレート」 は、従来のウォールメッセージ（ログインしていてその mesg(1) パーミッションが yesに設定されている全ユーザーに送信されるメッセージ）に似ているテンプレートを示しています。このテンプレートは改行後 (\r と \nを使用) にメッセージテキストと共にホスト名、メッセージタグ、およびタイムスタンプを出力してベル (\7 を使用) を鳴らします。

例25.9 ウォールメッセージのテンプレート

```
$template wallmsg,"\r\n\7Message from syslogd@%HOSTNAME% at %timegenerated%
...\r\n %syslogtag% %msg%\n\r"
```

例25.10「データベースフォーマットが設定されたメッセージのテンプレート」 は、syslog メッセージのフォーマットを作成してデータベースクエリーとして使用できるようにするテンプレートを示しています。テンプレートの末尾でテンプレートオプションとして指定されている sql オプションに注

目してください。これは、メッセージを MySQL SQL クエリーとしてフォーマットするようにデータベースライターに指示します。

例25.10 データベースフォーマットが設定されたメッセージのテンプレート

```
$template dbFormat,"insert into SystemEvents (Message, Facility, FromHost, Priority,
DeviceReportedTime, ReceivedAt, InfoUnitID, SysLogTag) values ('%msg%',
%syslogfacility%, '%HOSTNAME%', %syslogpriority%, '%timereported:::date-mysql%',
'%timegenerated:::date-mysql%', %iut%, '%syslogtag%')", sql
```

rsyslog には、RSYSLOG_ 接頭辞で識別される事前定義のテンプレートのセットも含まれています。これらは syslog の使用用に予約されており、競合を回避するためにこの接頭辞を使用してテンプレートを作成しないことが推奨されます。以下の一覧では、これらの事前定義のテンプレートとその定義を示しています。

RSYSLOG_DebugFormat

プロパティー問題のトラブルシューティングに使われる特別なフォーマット。

```
"Debug line with all properties:\nFROMHOST: '%FROMHOST%', fromhost-ip: '%fromhost-
ip%', HOSTNAME: '%HOSTNAME%', PRI: %PRI%,\nsyslogtag '%syslogtag%',
programname: '%programname%', APP-NAME: '%APP-NAME%', PROCID: '%PROCID%',
MSGID: '%MSGID%',\nTIMESTAMP: '%TIMESTAMP%', STRUCTURED-DATA:
'%STRUCTURED-DATA%',\nmsg: '%msg%'\nescaped msg: '%msg:::drop-cc%'\nrawmsg:
'%rawmsg%'\n\n\"
```

RSYSLOG_SyslogProtocol23Format

IETF のインターネットドラフト ietf-syslog-protocol-23 で指定されるフォーマット。これは、新しい syslog 標準 RFC になると想定されています。

```
"%PRI%1 %TIMESTAMP:::date-rfc3339% %HOSTNAME% %APP-NAME% %PROCID%
%MSGID% %STRUCTURED-DATA% %msg%\n\"
```

RSYSLOG_FileFormat

TraditionalFileFormat に類似した新しいスタイルのログファイルフォーマットですが、タイムスタンプとタイムゾーン情報の精度がより高くなります。

```
"%TIMESTAMP:::date-rfc3339% %HOSTNAME% %syslogtag%%msg:::sp-if-no-1st-
sp%%msg:::drop-last-lf%\n\"
```

RSYSLOG_TraditionalFileFormat

タイムスタンプの精度の低い古いスタイルのデフォルトのログファイルフォーマット。

```
"%TIMESTAMP% %HOSTNAME% %syslogtag%%msg:::sp-if-no-1st-sp%%msg:::drop-last-lf%\n\"
```

RSYSLOG_ForwardFormat

高精度のタイムスタンプとタイムゾーン情報が含まれる転送フォーマット。

```
"%PRI%%TIMESTAMP:::date-rfc3339% %HOSTNAME% %syslogtag:1:32%%msg:::sp-if-no-1st-sp%%msg%\"
```

RSYSLOG_TraditionalForwardFormat

精度の低いタイムスタンプを使用する従来の転送フォーマット。

```
"%PRI%%TIMESTAMP% %HOSTNAME% %syslogtag:1:32%%msg:::sp-if-no-1st-sp%%msg%\"
```

25.3.4. グローバルディレクティブ

グローバルディレクティブは、rsyslogd デーモンに適用される設定オプションです。通常、これらは rsyslogd デーモンの動作に影響を与える事前定義された特定の変数の値、またはそれに続くルールを指定します。グローバルディレクティブはすべて、ドル記号($)で開始する必要があります。1 行ごとに指定できるディレクティブは 1 つのみです。以下は、syslog メッセージキューの最大サイズを指定するグローバルディレクティブの例です。

```
$MainMsgQueueSize 50000
```

このディレクティブに定義したデフォルトのサイズ（10,000 メッセージ）は、別の値を指定することで上書きできます（上記の例を参照）。

/etc/rsyslog.conf 設定ファイルで複数のディレクティブを定義できます。1 つのディレクティブは、同じディレクティブの発生が再度検出されるまですべての設定オプションの動作に影響します。グローバルディレクティブは、アクション、キュー、デバッグの設定に使用できます。利用可能なすべての設定ディレクティブの一覧は、「オンラインドキュメント」 でご覧になれます。現在、$ ベースの構文 (「新規設定フォーマットの使用」 を参照) に代わる新たな設定フォーマットが開発されています。ただし、従来のグローバルディレクティブはレガシーフォーマットとして引き続きサポートされます。

25.3.5. ログローテーション

以下は、/etc/logrotate.conf 設定ファイルのサンプルです。

```
# rotate log files weekly
weekly
# keep 4 weeks worth of backlogs
rotate 4
# uncomment this if you want your log files compressed
compress
```

上記の設定ファイル内のすべての行は、すべてのログファイルに適用されるグローバルオプションを定義しています。この例では、ログファイルは週ごとにローテーションされ、ローテーションされたログファイルは 4 週間保持され、ローテーションされるログファイルはすべて gzip により .gz 形式に圧縮されます。ハッシュマーク (#) で始まる行はすべてコメントで、これは処理されません。

特定のログファイル用に設定オプションを定義して、それをグローバルオプションの下に置くこともできます。ただし、/etc/logrotate.d/ ディレクトリーに、特定ログファイル用の個別の設定ファイルを作成し、そこに設定オプションを定義することが推奨されます。

/etc/logrotate.d/ ディレクトリーに設定ファイルが保存されている例を以下に示します。

```
/var/log/messages {
    rotate 5
    weekly
    postrotate
    /usr/bin/killall -HUP syslogd
    endscript
}
```

このファイル内の設定オプションは、/var/log/messages ログファイル専用の特有なものです。ここで指定された設定は、可能な場合はグローバルオプションを上書きします。そのため、ローテートされた /var/log/messages ログファイルは、グローバルオプションで定義された 4 週間ではなく、5 週間保管されます。

以下は、logrotate 設定ファイル内で指定できるディレクティブの一覧です。

- weekly - ログファイルの週ごとのローテーションを指定します。同様なディレクティブには以下のものがあります。
  - daily

- *monthly*
- *yearly*

- *compress - ローテートしたログファイルの圧縮を有効にします。同様なディレクティブには以下のものがあります。*
  - *nocompress*
  - *compresscmd - 圧縮に使用するコマンドを指定します。*
  - *uncompresscmd*
  - *compressext - 圧縮に使用する拡張子を指定します。*
  - *compressoptions - 使用する圧縮プログラムに渡すオプションを指定します。*
  - *delaycompress - ログファイルの圧縮を次回のログファイルのローテーションまで延期します。*

- *rotate INTEGER - ログファイルが削除される、または特定のアドレスに送信されるまでにログファイルがローテーションされる回数を指定します。0 の値を指定すると、ローテーションではなく古いログファイルが削除されます。*

- *mail ADDRESS - このオプションは、rotate ディレクティブで定義された回数だけローテーションされたログファイルを特定のアドレスへメール送信できるようにします。同様な*

ディレクティブには以下のものがあります。

- nomail

- mailfirst - 間もなく期限切れになるログファイルではなく、ローテートされたログファイルがメール送信されるよう指定します。

- maillast - 交代されたばかりのログファイルではなく、間もなく期限切れになるログファイルがメール送信されるよう指定します。mail が有効な場合は、これがデフォルトのオプションです。

ディレクティブおよび設定オプションの一覧は、man ページの logrotate(5) を参照してください。

25.4. 新規設定フォーマットの使用

rsyslog バージョン 7 では、Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;LinuxRed Hat Enterprise Linux 6nbsp;6 の rsyslog7 パッケージで利用可能な新しい設定構文が導入されています。この新しい設定形式の目的は、より強力かつ直感的なものにすることと、特定の無効なコンストラクトを許可しないことによりよくあるミスを防ぐことです。この構文の拡張は、RainerScript に依存する新たな設定プロセッサーによって可能になります。以前の形式は引き続き完全にサポートされ、/etc/rsyslog.conf 設定ファイルでデフォルトで使用されます。rsyslog 7 をインストールするには、「rsyslog バージョン 7 へのアップグレード」 を参照してください。

RainerScript は、ネットワークイベントの処理と rsyslog などのイベントプロセッサーの設定用に設計されたスクリプト言語です。rsyslog バージョン 5 の RainerScript のバージョンは、式ベースのフィルターを定義するために使用されます。例25.3「式ベースのフィルター」 を参照してください。rsyslog バージョン 7 の RainerScript のバージョンは、input() ステートメントおよび ruleset() ステートメントを実装し、/etc/rsyslog.conf 設定ファイルを新しい構文で記述できます。新しい構文は主に構成される点で異なります。パラメーターは input、action、template、module load などのステートメントへの引数として渡されます。オプションの範囲はブロックによって制限されます。オプションのスコープはブロックにより制限されます。これにより、可読性が増し、設定の間違えによって発生するバグの数が減ります。一部の機能は両方の構文で公開され、他の機能は新しい構文でのみ公開されます。

以前のスタイルのパラメーターで記述された設定を比較します。

```
$InputFileName /tmp/inputfile
$InputFileTag tag1:
$InputFileStateFile inputfile-state
$InputRunFileMonitor
```

同じ設定を新たなフォーマットステートメントを使用すると、以下のようになります。

```
input(type="imfile" file="/tmp/inputfile" tag="tag1:" statefile="inputfile-state")
```

これにより、設定で使用されるパラメーター数が大幅に削減され、読みやすさが向上するとともに、実行速度も速まります。RainerScript ステートメントおよびパラメーターに関する詳細な情報は、「オンラインドキュメント」 を参照してください。

25.4.1. ルールセット

特別なディレクティブを残すと、rsyslog は、フィルター条件と、条件が true の場合に実行されるアクションで構成される ルール で定義されているようにメッセージを処理します。従来の /etc/rsyslog.conf ファイルでは、すべてのルールは、すべての入力メッセージの外観順に評価されます。このプロセスは最初のルールで開始し、すべてのルールが処理されるか、ルールのいずれかがメッセージを破棄するまで続きます。

ただし、ルールは rulesets と呼ばれるシーケンスにグループ化できます。ルールセットでは、特定のルールの効果を選択した入力のみに制限したり、特定の入力にバインドされる個別のアクションセットを定義して rsyslog のパフォーマンスを強化したりできます。つまり、特定の種類のメッセージでは必然的に false と評価されるフィルター条件をスキップできます。/etc/rsyslog.conf のレガシールールセット定義は以下のようになります。

```
$RuleSet rulesetname
rule
rule2
```

ルールは、別のルールが定義されたとき、または以下のようにデフォルトのルールセットが呼び出されたときに終了します。

```
$RuleSet RSYSLOG_DefaultRuleset
```

rsyslog 7 の新しい設定形式では、この操作のために input() ステートメントおよび ruleset() ステートメントが予約されます。/etc/rsyslog.conf の新しい形式のルールセット定義は以下のようになります。

```
ruleset(name="rulesetname") {
    rule
    rule2
    call rulesetname2
    ...
}
```

rulesetname を、ルールセットの識別子に置き換えます。この名前空間は rsyslog で使用するために予約されているため、ルールセット名は RSYSLOG_ で始めることはできません。RSzySLOG_DefaultRuleset は、メッセージが他のルールセットを割り当てていない場合に実行するデフォルトのルールセットを定義します。rule と rule 2 では、上記の filter-action 形式でルールを定義できます。call パラメーターでは、他の ruleset ブロック内から ruleset を呼び出すことでこれらをネスト化できます。

ruleset の作成後は、これが適用される入力を指定する必要があります。

```
input(type="input_type" port="port_num" ruleset="rulesetname");
```

ここでは、メッセージを収集する入力モジュールである input_type か、ポート番号である port_num で入力メッセージを特定できます。file や tag などの他のパラメーターは、input() 用に指定できます。rulesetname を、メッセージに対して評価されるルールセットの名前に置き換えます。入力メッセージが明示的に ruleset にバインドされていない場合は、デフォルトの ruleset が適用されます。

レガシーフォーマットを使用して ruleset を定義することもできます。詳細は 「オンラインドキュメント」 を参照してください。

例25.11 ルールセットの使用

以下の ruleset は、異なるポートからのリモートメッセージが確実に異なる方法で処理されるようにします。以下を /etc/rsyslog.conf に追加します。

```
ruleset(name="remote-6514") {
   action(type="omfile" file="/var/log/remote-6514")
}

ruleset(name="remote-601") {
   cron.* action(type="omfile" file="/var/log/remote-601-cron")
   mail.* action(type="omfile" file="/var/log/remote-601-mail")
}

input(type="imtcp" port="6514" ruleset="remote-6514");
input(type="imtcp" port="601" ruleset="remote-601");
```

上記の例に示されるルールセットは、2 つのポートからのログ宛先を定義します。ポート 601 の場合、メッセージはファシリティーに従ってソートされます。次に、TCP 入力が有効になり、ルールセットにバインドされます。この設定が機能するには、必須モジュール (imtcp) の読み込みが必要なことに注意してください。

25.4.2. sysklogd との互換性

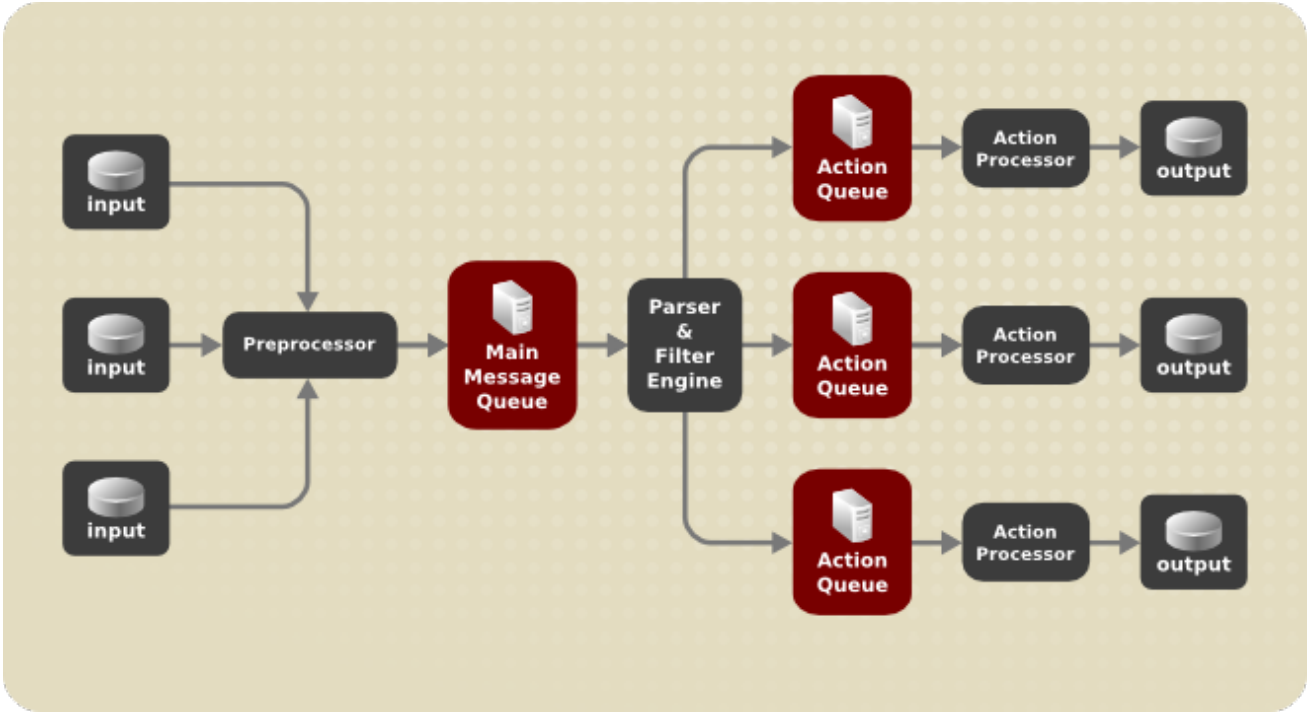
-c オプションで指定される互換性モードは rsyslog バージョン 5 にありますが、バージョン 7 には含まれません。また、sysklogd-style コマンドラインオプションは非推奨となり、これらのコマンドラインオプションを使用した rsyslog の設定を回避する必要があります。ただし、複数のテンプレートおよびディレクティブを使用して、rsyslogd が sysklogd のような動作をエミュレートするように設定できます。

rsyslogd オプションの詳細は、man ページの rsyslogd(8)を参照してください。

25.5. RSYSLOG でのキュー (QUEUE) を使った操作

キューは、rsyslog のコンポーネント間でコンテンツ（主に syslog メッセージ）を渡すために使用されます。キューを使うと、rsyslog は複数のメッセージを同時に処理することができ、複数のアクションを単一メッセージに一度に適用できます。rsyslog 内のデータフローは、以下のようになります。

図25.1 Rsyslog 内のメッセージフロー



[D]

rsyslog がメッセージを受信するたびに、このメッセージをプリプロセッサーに渡して、メインのメッセージキュー に配置します。メッセージはそこでキューから取り出され、rule プロセッサー に渡されるのを待ちます。

rule processor は、分析およびフィルタリングのエンジンです。ここでは、/etc/rsyslog.conf で定義されたルールが適用されます。これらのルールに基づいて、rule processor はどのアクションが実行

されるかを評価します。アクションにはそれぞれ、独自のアクションキューがあります。メッセージはこのキューにより各アクションプロセッサーに渡され、これが最終的な出力を作成します。この時点では、1 つのメッセージに関して複数のアクションが同時に実行可能であることに注意してください。このためにメッセージは複製され、複数のアクションプロセッサーに渡されます。

1 アクションにつき、1 つのキューのみが可能です。設定により、メッセージはアクションキューなしですぐにアクションプロセッサーに送信されることもあります。これはダイレクトキュー (direct queues) (下記参照) と呼ばれる動作です。出力アクションが失敗すると、アクションプロセッサーがアクションキューに通知を行い、それにより未処理要素が戻され、しばらくのインターバルの後、アクションが再度試されます。

まとめると、rsyslog では 2 つの位置があります。ルールプロセッサーの前にある単一の メインメッセージキュー の前か、さまざまなタイプの出力アクションの前に アクションキュー の前かのいずれかです。キューは、メッセージ処理のパフォーマンス向上につながる以下の 2 つの利点を提供します。

- rsyslog構造で切り離されたプロデューサーとコンシューマーのバッファーとして機能します。
- メッセージで実行されるアクションの 並列化 を可能にします。

それ以外では、キューを複数のディレクティブで設定して、システムに最適なパフォーマンスを提供することができます。これらの設定オプションは、以下の項で説明されています。

警告

出力プラグインがメッセージを提供できない場合、メッセージは、先行のメッセージキューに保存されます。キューがいっぱいになると、空きができるまで入力がブロックされます。これにより、ブロックされたキューを使用して新しいメッセーがログに記録されることが回避されます。個別のアクションキューが存在しないため、SSH ロギングが阻止され、SSH アクセスが阻止されるなどの重大な問題が発生することがあります。したがって、ネットワークを介して転送される、またはデータベースに転送される出力専用アクションキューを使用することが推奨されます。

25.5.1. キューの定義

メッセージが格納されている場所に基づいて、最も一般的に使用される ダイレクト、インメモ

リー、ディスク、ディスクアシストメモリー キューの複数のタイプのキューがあります。最もよく使用されるのは、direct、in-memory、disk、および disk-assisted in-memory のキューです。以下を /etc/rsyslog.conf に追加します。

```
$objectQueueType queue_type
```

ここでは、メインメッセージキューの設定（ object を MainMsgに置き換え）またはアクションキュー（ オブジェクト を Actionに置き換える）を適用することができます。queue_type を direct、linkedlist または fixedarray （インメモリーキュー）または ディスク のいずれかに置き換えます。

メインメッセージキューのデフォルト設定は、FixedArray queue で 10,000 メッセージの制限があります。アクションキューはデフォルトで、Direct queue に設定されます。

ダイレクトキュー (Direct Queue)

出力をローカルファイルに書き出すなど、多くの単純な操作では、アクションの前にキューを構築することは不要です。キューの使用を回避するには、以下を使用します。

```
$objectQueueType Direct
```

object を MainMsg または Action に置き換えて、このオプションをメインメッセージキューまたはアクションキューにそれぞれ使用します。ダイレクトキューを使用すると、メッセージはプロデューサーからコンシューマーに直ちに直接渡されます。

ディスクキュー

ディスクキューはメッセージを厳密にハードドライブに保存します。これにより、信頼性が高くなりますが、すべてのキューモードが最も遅くなります。こうすることで信頼性は非常に高くなりますが、queuing モードのなかでは一番遅くなります。ただし、ほとんどのユースケースでは、ディスクキューは推奨されません。ディスクキューを設定するには、/etc/rsyslog.conf に以下を入力します。

```
$objectQueueType Disk
```

object を MainMsg または Action に置き換えて、このオプションをメインメッセージキューまたはアクションキューにそれぞれ使用します。ディスクキューは、デフォルトサイズの 10 Mb で部分で記述されます。このデフォルトサイズは、以下の設定ディレクティブで変更できます。

```
$objectQueueMaxFileSize size
```

ここでの size は、ディスクキューの一部の指定サイズを表します。定義されたサイズ制限は制限されず、rsyslog は、サイズ制限に違反する場合でも常に 1 つの完全なキューエントリーを書き込みま

す。ディスクキューの各部分は、個別ファイルに適合します。これらファイルの命名ディレクティブは、以下のとおりです。

```
$objectQueueFilename name
```

これでファイルに name プレフィックスが設定され、1 から始まる 7 桁の数字が設定され、ファイルごとに増えます。

メモリー内キュー

インメモリーキューでは、キューに格納されたメッセージをメモリーに保持し、プロセスを高速化します。コンピュータの電源を切ってすぐに入れ直すか、シャットダウンが行われると、キューのデータは失われます。ただし、$ActionQueueSaveOnShutdown 設定を使用して、シャットダウン前にデータを保存することができます。インメモリーキューには 2 種類あります。

- **FixedArray キュー:** *メインメッセージキューのデフォルトモードで、10,000 要素の制限があります。このタイプのキューは、キュー要素へのポインターを保有する固定かつ事前割り当てのアレイを使用します。これらのポインターのために、キューが空であってもある程度のメモリーが消費されます。しかし、FixedArray は、最善のランタイムパフォーマンスを提供し、比較的少ないキューに登録済みのメッセージと高パフォーマンスを期待する場合に最適なものです。*

- **LinkedList キュー:** *ここではすべての構造物はリンクされたリストに動的に割り当てられるので、メモリーは必要な場合にのみ割り当てられます。LinkedList キューは時折発生するメッセージバーストにもうまく対応します。*

一般的に、疑いが残る場合は LinkedList キューを使用してみてください。FixedArray と比べてメモリー消費量が少なく、プロセスオーバヘッドが低くて済みます。

以下の構文を使用して、インメモリーキューを設定します。

```
$objectQueueType LinkedList
```

```
$objectQueueType FixedArray
```

object を MainMsg または Action に置き換えて、このオプションをメインメッセージキューまたはアクションキューにそれぞれ使用します。

ディスク補助のインメモリーキュー (Disk-Assisted In-memory Queues)

ディスクとインメモリーキューの両方に利点があり、rsyslog を使用すると、ディスク補助のイン

メモリーキュー でそれらを組み合わせることができます。これを行うには、通常のインメモリーキューを設定し、$objectQueueFileName ディレクティブを追加してディスクのファイル名を定義します。このキューは disk-assisted となり、インメモリーキューとディスクキューが連携します。

インメモリーキューがいっぱい、もしくはシャットダウン後にも継続する必要がある場合に、ディスクキューがアクティブ化されます。ディスク補助のキューでは、ディスク固有およびインメモリー固有の設定パラメーターの両方を設定できます。このタイプのキューはおそらく最も広く使われているもので、潜在的に長期間実行され、信頼性が低いアクションで特に便利です。

ディスク補助インメモリーキューの機能を指定するには、いわゆる ウォーターマーク を使用します。

> $objectQueueHighWatermark number
>
> $objectQueueLowWatermark number

object を MainMsg または Action に置き換えて、このオプションをメインメッセージキューまたはアクションキューにそれぞれ使用します。number をキューに格納されたメッセージ数に置き換えます。インメモリーキューがハイウォーターマークで定義された数字に到達すると、ディスクへのメッセージの書き込みが開始し、インメモリーキューのサイズがローウォーターマークで定義された数字になるまで続きます。watermarks を正しく設定すると、ディスクファイルへの書き込みが長くなるため、不要なディスク書き込みが最小限に抑えられます。したがって、高い基準は $objectQueueSize で設定されたキュー容量全体よりも低くなければなりません。そのために、ハイウォーターマークは queue.size で設定されているキューのキャパシティ全体より低い必要があります。一方で、ハイウォーターマークを低く設定しすぎると、不必要なディスク補助が頻繁に発生してしまいます。

> 例25.12 サーバーへのログメッセージの確実な転送
>
> 多くの場合、Rsyslog は、ログメッセージがネットワークを介してサーバーに転送される中央ロギングシステムを保持するために使用されます。サーバーが利用できない場合にメッセージが失われないように、転送アクションに対するアクションキューを設定することが推奨されます。これにより、送信に失敗したメッセージは、サーバーが再度到達可能になるまでローカルに保存されます。このようなキューは UDP プロトコルを使用している接続には設定できないことに注意してください。ロギングサーバーがプライベートネットワーク外にある場合などに、完全に信頼できる接続を確立するには、「RELP の使用」 で説明されている RELP プロトコルの使用を検討してください。
>
> 手順25.2 単一サーバーへの転送
>
> ログメッセージをシステムからホスト名が example.com のサーバーに転送し、サーバー停止時にメッセージをバッファーするようアクションキューを設定するタスクを考えてみます。このタスクを実行するには、以下の手順を実行します。
>
> - /etc/rsyslog.conf の以下の設定を使用するか、/etc/rsyslog.d/ ディレクトリーに以下

の内容のファイルを作成します。

```
$ActionQueueType LinkedList
$ActionQueueFileName example_fwd
$ActionResumeRetryCount -1
$ActionQueueSaveOnShutdown on
*.*       @@example.com:6514
```

詳細は以下のようになります。

- **$ActionQueueType** は **LinkedList** インメモリーキューを有効にします。
- **$ActionFileName** はディスクストレージを定義します。この場合、バックアップファイルが **example_fwd** 接頭辞を持つ **/var/lib/rsyslog/** ディレクトリーに作成されます。
- **$ActionResumeRetryCount -1** 設定は、サーバーが応答しない場合に接続を再試行するときに **rsyslog** がメッセージを破棄しないようにします。
- **rsyslog** がシャットダウンすると、有効になっている **$ActionQueueSaveOnShutdown** がインメモリーデータを保存します。
- 最後の行は受信メッセージをすべてロギングサーバーに転送します。ポートの指定は任意です。

上記の設定では、**rsyslog** は、リモートサーバーに到達できない場合にメッセージをメモリーに保持します。ディスク上にあるファイルは、設定されたメモリーキュー領域が **rsyslog** で不足するか、シャットダウンする必要がある場合にのみ作成されます。これにより、システムパフォーマンスが向上します。

手順25.3 複数のサーバーへの転送

複数のサーバーにログメッセージを転送するプロセスは前の手順に似ています。

- 各送信先サーバーでは、個別の転送ルール、アクションキュー指定、ディスク上のバックアップファイルが必要です。たとえば、**/etc/rsyslog.conf** で以下の設定を使用するか、**/etc/rsyslog.d/** ディレクトリーに以下の内容のファイルを作成します。

```
$ActionQueueType LinkedList
$ActionQueueFileName example_fwd1
$ActionResumeRetryCount -1
$ActionQueueSaveOnShutdown on
*.*       @@example1.com

$ActionQueueType LinkedList
$ActionQueueFileName example_fwd2
$ActionResumeRetryCount -1
$ActionQueueSaveOnShutdown on
*.*       @@example2.com
```

25.5.2. rsyslog ログファイルの新しいディレクトリーの作成

rsyslog は syslogd デーモンとして実行され、SELinux により管理されます。したがって、rsyslog が書き込む必要があるすべてのファイルでは適切な SELinux ファイルコンテキストが設定されている必要があります。

手順25.4 新規作業用ディレクトリーの作成

1. 作業用ファイルを格納する別のディレクトリーを使用する必要がある場合は、以下のようにディレクトリーを作成します。

```
~]# mkdir /rsyslog
```

2. SELinux ポリシーを管理するためにユーティリティーをインストールします。

```
~]# yum install policycoreutils-python
```

3. SELinux ディレクトリーコンテキストタイプを /var/lib/rsyslog/ ディレクトリーと同じものに設定します。

```
~]# semanage fcontext -a -t syslogd_var_lib_t /rsyslog
```

4. SELinux コンテキストを適用します。

```
~]# restorecon -R -v /rsyslog
restorecon reset /rsyslog context unconfined_u:object_r:default_t:s0-
>unconfined_u:object_r:syslogd_var_lib_t:s0
```

5. 必要な場合は、以下のように SELinux コンテキストを確認します。

```
~]# ls -Zd /rsyslog
drwxr-xr-x. root root system_u:object_r:syslogd_var_lib_t:s0   /rsyslog
```

6. 必要に応じてサブディレクトリーを作成します。以下に例を示します。

```
~]# mkdir /rsyslog/work
```

サブディレクトリーが親ディレクトリーと同じ SELinux コンテキストで作成されます。

7. /etc/rsyslog.conf を有効にする直前にそのファイルに次の行を追加します。

```
$WorkDirectory /rsyslog/work
```

この設定は、設定ファイルを解析するときに次の WorkDirectory ディレクティブが検出されるまで有効になります。

25.5.3. キューの管理

キューはすべてのタイプで、使用中の要件に合致するようにさらなる設定が可能です。複数のディレクティブを使用してアクションキューとメインメッセージキューの両方を修正できます。現時点では、20 以上のキューパラメーターが利用可能です。「オンラインドキュメント」 を参照してください。これらの設定の一部は一般的に使用されます。ワーカースレッド管理などでは、キューの動作をより詳細に制御でき、上級ユーザー用に予約されています。高度な設定では、rsyslog の「パフォーマンス、スケジュールキュー」を最適化したり、システムシャットダウン時のキューの動作を変更したりすることができます。

キューのサイズ制限

キューが保有できるメッセージ数は、以下の設定で制限できます。

```
$objectQueueHighWatermark number
```

- *object を MainMsg または Action に置き換えて、このオプションをメインメッセージキューまたはアクションキューにそれぞれ使用します。number をキューに格納されたメッセージ数に置き換えます。キューサイズを実際のメモリーサイズではなくメッセージの数として設定します。デフォルトのキューサイズは、メインメッセージキューとルールセットキューの場合は 10,000 メッセージ、アクションキューの場合は 1,000 となります。*

  ディスク補助キューはデフォルトで無制限で、このディレクティブでは制限できませんが、以下の設定で物理ディスク領域をバイト単位で確保することができます。

  ```
  $objectQueueMaxDiscSpace number
  ```

  object を MainMsg または Action に置き換えます。数 で指定したサイズ制限がヒットすると、キューデキューのメッセージで十分な領域が解放されるまでメッセージは破棄されます。

  メッセージの破棄

  *キューのメッセージがある数に達すると、重要でないメッセージを破棄して、より優先度が高いエントリーのためにキューのスペースを節約できます。破棄プロセスを開始するしきい値は、*discard mark *と呼ばれるもので設定できます。*

  ```
  $objectQueueDiscardMark number
  ```

  object を MainMsg または Action に置き換えて、このオプションをメインメッセージキューまたはアクションキューにそれぞれ使用します。ここでは、number は、破棄プロセスを開始するためにキューにある必要がある多数のメッセージを表します。破棄するメッセージを定義するには、以下を使用します。

  ```
  $objectQueueDiscardSeverity priority
  ```

  priority を、以下のキーワード（または数字）のいずれかに置き換えます。debug (7)、info (6)、notice (5)、warning (4)、err (3)、crit (2)、alert (1)、および emerg (0)。この設定により、定義されたプライオリティーを下回る、新しく受信したメッセージおよびすでにキューに格納されたメッセージは、破棄マークに到達すると直ちにキューから消去されます。

  タイムフレームの使用

  特定の期間中にキューを処理するように rsyslog を設定できます。このオプションを使用すると、たとえば処理をオフピーク時に移すことができます。時間帯を定義するには、以下の構文を使用します。

  ```
  $objectQueueDequeueTimeBegin hour
  ```

-

$objectQueueDequeueTimeEnd hour

hour では、時間枠を区切る時間を指定できます。分は指定せず、24 時間形式を用います。

ワーカースレッドの設定

ワーカースレッド はキューに格納されたメッセージに対して指定されたアクションを実行します。たとえば、メインメッセージキューでは、ワーカーのタスクは、入ってくるメッセージにフィルター論理を適用し、関連のアクションキューに入れることです。メッセージが届くと、ワーカースレッドは自動的に開始します。メッセージ数がある数に達すると、別のワーカースレッドがオンになります。この数字を指定するには、以下を使用します。

$objectQueueWorkerThreadMinimumMessages number

number を、補助ワーカースレッドをトリガーするメッセージ数に置き換えます。たとえば、number を 100 に設定すると、100 を超えるメッセージが到達すると、新しいワーカースレッドが起動します。200を超えるメッセージが到着すると、3番目のワーカースレッドが開始されます。ただし、並行して実行している作業スレッド数が多すぎるため、以下を使用してそれらの最大スレッド数を制限できます。

$objectQueueWorkerThreads number

ここでの number は、並行して実行できる作業スレッドの最大数を表します。メインメッセージキューのデフォルトは、1 スレッドです。ワーカースレッドが一旦起動すると、非アクティブタイムアウトが現れるまで、実行し続けます。タイムアウトを設定するには、以下を入力します。

$objectQueueWorkerTimeoutThreadShutdown time

time をミリ秒単位の期間設定に置き換えます。この設定がないと、ゼロタイムアウトが適用され、ワーカースレッドはメッセージが不足するとすぐに終了します。time を -1 として指定すると、スレッドは閉じられません。

バッチのデキュー

パフォーマンスを向上させるために、rsyslog を設定して、複数のメッセージを一度にデキューします。このデキューの最大値を設定するには、以下を使用します。

$objectQueueDequeueBatchSize number

number を、同時にデキューできるメッセージの最大数に置き換えます。この数字を高く設定して、許可されるワーカースレッドの結果を大きくすると、メモリー消費量が大きくなることに注意してください。

キューの終了

メッセージを含んでいるキューを終了する際には、ワーカースレッドがキューの処理を完了する間隔を指定することで、データ損失を最小限に抑えることができます。

```
$objectQueueTimeoutShutdown time
```

time をミリ秒単位で指定します。この期間の後にまだキューに入っているメッセージがある場合、ワーカーは現在のデータ要素を完了してから終了します。このため、未処理のメッセージは失われます。ワーカーが最終要素を完了する間隔も設定できます。

```
$objectQueueTimeoutActionCompletion time
```

このタイムアウトが切れると、残りのワーカーはシャットダウンします。シャットダウン時にデータを保存するには、以下を使用します。

```
$objectQueueTimeoutSaveOnShutdown time
```

これが設定されている場合、rsyslog の終了前にすべてのキュー要素がディスクに保存されます。

25.5.4. rsyslog キューの新規構文の使用

rsyslog 7 で利用可能な新しい構文では、キューは /etc/rsyslog.conf で個別に使用するか、またはルールセット内部で使用できる action() オブジェクト内に定義されます。アクションキューの形式は以下のようになります。

```
action(type="action_type" queue.size="queue_size" queue.type="queue_type"
queue.filename="file_name")
```

action_type を、アクションを実行するモジュールの名前に置き換え、queue_size をキューに含めることができるメッセージの最大数に置き換えます。queue_type には、disk を選択するか、インメモリーキュー（direct、Linkedlist、または fixedarray ）のいずれかを選択します。file_name には、パスではなくファイル名のみを指定します。ログファイルを保持する新規ディレクトリーを作成する場合は、SELinux コンテキストを設定する必要があることに注意してください。例は、「rsyslog ログファイルの新しいディレクトリーの作成」 を参照してください。

例25.13 アクションキューの定義

出力アクションを、最大 10,000 個のメッセージを保持できる非同期リンクリストベースのアクションキューで設定するには、以下のようにコマンドを入力します。

```
action(type="omfile" queue.size="10000" queue.type="linkedlist" queue.filename="logfile")
```

直接アクションキューの rsyslog 7 構文は以下のとおりです。

```
*.* action(type="omfile" file="/var/lib/rsyslog/log_file
    )
```

複数のパラメーターがアクションキュー用 rsyslog 7 構文は以下のように記述できます。

```
*.* action(type="omfile"
           queue.filename="log_file"
           queue.type="linkedlist"
           queue.size="10000"
    )
```

デフォルトの作業ディレクトリー、または最後に設定した作業ディレクトリーが使用されます。別の作業ディレクトリーを使用する必要がある場合は、アクションキューの前に以下の行を追加します。

```
global(workDirectory="/directory")
```

例25.14 新規構文を使用した単一サーバーへの転送

以下の例は、手順25.2「単一サーバーへの転送」 の手順に基づき、従来の構文と rsyslog 7 の構文の違いを示しています。omfwd プラグインは、UDP または TCP を介した転送を提供するために使用されます。デフォルトは UDP です。プラグインは組み込まれているため、ロードする必要がありません。

/etc/rsyslog.conf の以下の設定を使用するか、/etc/rsyslog.d/ ディレクトリーに以下の内容のファイルを作成します。

```
*.* action(type="omfwd"
     queue.type="linkedlist"
     queue.filename="example_fwd"
     action.resumeRetryCount="-1"
     queue.saveOnShutdown="on"
     target="example.com" port="6514" protocol="tcp"
    )
```

詳細は以下のようになります。

- queue.type="linkedlist" は、LinkedList インメモリーキューを有効にします。
- queue.filename はディスクストレージを定義します。バックアップファイルは、前のグローバルの workDirectory ディレクティブで指定された作業ディレクトリーに example_fwd 接頭辞を付けて作成されます。
- action.resumeRetryCount -1 設定は、サーバーが応答しない場合に接続を再試行するときに rsyslog がメッセージを破棄しないようにします。
- rsyslog がシャットダウンすると、有効になっている queue.saveOnShutdown="on" はインメモリーデータを保存します。
- 最後の行は受信メッセージをすべてロギングサーバーに転送します。ポートの指定は任意です。

25.6. ロギングサーバーでの RSYSLOG の設定

rsyslog サービスは、ロギングサーバーを実行する機能と、個別のシステムがログファイルをロギングサーバーに送信するように設定する機能の両方を提供します。クライアントの rsyslog 設定の詳細は、例25.12「サーバーへのログメッセージの確実な転送」 を参照してください。

rsyslog サービスは、ロギングサーバーとして使用するシステムと、そのシステムにログを送信するように設定する全システムにインストールする必要があります。rsyslog は、Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;LinuxRed Hat Enterprise Linux 6nbsp;6 にデフォルトでインストールされます。必要な場合は、確実にインストールするために root で以下のコマンドを入力します。

```
~]# yum install rsyslog
```

syslog トラフィックのデフォルトのプロトコルおよびポートは、/etc/services ファイルに記載されている UDP および 514 です。ただし、rsyslog はデフォルトで、ポート 514 で TCP を使用するよう

に設定されています。設定ファイル /etc/rsyslog.conf では、TCP は @@ で示され ます。

例では他のポートが使用されることがありますが、SELinux には、デフォルトで以下のポートでの送受信のみを許可するように設定されています。

```
~]# semanage port -l | grep syslog
syslogd_port_t          tcp     6514, 601
syslogd_port_t          udp     514, 6514, 601
```

semanage ユーティリティーは、policycoreutils-python パッケージの一部として提供されます。必要な場合は、以下のようにパッケージをインストールします。

```
~]# yum install policycoreutils-python
```

さらに、デフォルトでは rsyslog の SELinux タイプであるrsyslogd_t は、SELinux タイプが rsh_port_t のリモートシェル(rsh)ポートでの送受信を許可するよう設定されます（デフォルトではポート 514 の TCP に設定されます）。したがって、semanage を使用してポート 514 で TCP を明示的に許可する必要はありません。たとえば、SELinux がポート 514 でそのTCPを許可するよう設定されているかを確認するには、以下のコマンドを入力します。

```
~]# semanage port -l | grep 514
output omitted
rsh_port_t              tcp     514
syslogd_port_t          tcp     6514, 601
syslogd_port_t          udp     514, 6514, 601
```

SELinux の詳細は、『『Red Hat Enterprise Linux 6 SELinux ユーザーガイド』』 を参照してください。

ロギングサーバーとして使用するシステムで以下の手順を実行します。これらの手順はすべて root ユーザーで実行する必要があります。

手順25.5 ポートで rsyslog トラフィックを許可する SELinux の設定

rsyslog トラフィックに新しいポートを使用する必要がある場合は、ロギングサーバーとクライアントでこの手順を実行します。たとえば、ポート 10514 で TCP トラフィックを送受信するには、以下の手順を実行します。

1.

```
~]# semanage port -a -t syslogd_port_t -p tcp 10514
```

2.
以下のコマンドを入力して SELinux ポートを確認します。

```
~]# semanage port -l | grep syslog
```

3.
新しいポートがすでに /etc/rsyslog.conf に設定されている場合は、rsyslog を再起動して変更を反映します。

```
~]# service rsyslog restart
```

4.
rsyslog が現在リッスンしているポートを確認します。

```
~]# netstat -tnlp | grep rsyslog
tcp        0      0 0.0.0.0:10514           0.0.0.0:*   LISTEN      2528/rsyslogd
tcp        0      0 :::10514                :::*        LISTEN      2528/rsyslogd
```

semanage port コマンドの詳細は、man ページの semanage-port(8) を参照してください。

手順25.6 iptables ファイアウォールの設定

iptables ファイアウォールが、受信 rsyslog トラフィックを許可するように設定します。たとえば、ポート 10514 で TCP トラフィックを許可するには、以下の手順を実行します。

1.
テキストエディターで /etc/sysconfig/iptables ファイルを開きます。

2.

ポート 10514 の TCP トラフィックを許可する INPUT ルールを追加します。新しいルールは、REJECT トラフィックに REJECT ルールの前 に 表示される必要があります。

```
-A INPUT -m state --state NEW -m tcp -p tcp --dport 10514 -j ACCEPT
```

3. /etc/sysconfig/iptables ファイルへの変更を保存します。

4. ファイアウォールの変更を有効にするために iptables サービスを再起動します。

```
~]# service iptables restart
```

手順25.7 rsyslog で、リモートログメッセージを受信してソートするように設定

1. テキストエディターで /etc/rsyslog.conf ファイルを開き、以下の手順を実行します。

   a. モジュールセクションと Provides UDP syslog reception セクションの間に以下の行を追加します。

```
# Define templates before the rules that use them

### Per-Host Templates for Remote Systems ###
$template TmplAuthpriv,
"/var/log/remote/auth/%HOSTNAME%/%PROGRAMNAME:::secpath-replace%.log"
$template TmplMsg,
"/var/log/remote/msg/%HOSTNAME%/%PROGRAMNAME:::secpath-replace%.log"
```

   b. デフォルトの Provides TCP syslog reception セクションを、以下の内容に置き換えます。

```
# Provides TCP syslog reception
$ModLoad imtcp
# Adding this ruleset to process remote messages
$RuleSet remote1
authpriv.*   ?TmplAuthpriv
*.info;mail.none;authpriv.none;cron.none   ?TmplMsg
$RuleSet RSYSLOG_DefaultRuleset   #End the rule set by switching back to the default
rule set
$InputTCPServerBindRuleset remote1  #Define a new input and bind it to the "remote1"
rule set
$InputTCPServerRun 10514
```

/etc/rsyslog.conf ファイルへの変更を保存します。

2.
rsyslog サービスは、ロギングサーバーと、そのサーバーにログ記録を試みるシステムの両方で実行する必要があります。

a.
service コマンドを使用して rsyslog サービスを起動します。

```
~]# service rsyslog start
```

b.
rsyslog サービスが今後自動的に起動されるようにするには、root で以下のコマンドを入力します。

```
~]# chkconfig rsyslog on
```

環境内の他のシステムからログファイルを受け取り、保存するように、ログサーバーが設定されています。

25.6.1. ロギングサーバーでの新規テンプレート構文の使用

rsyslog 7 にはテンプレートスタイルが多数あります。文字列テンプレートは従来の形式に最もよく似ています。文字列形式を使用して上記の例からテンプレートを生成する場合は、以下のようになります。

```
template(name="TmplAuthpriv" type="string"
      string="/var/log/remote/auth/%HOSTNAME%/%PROGRAMNAME:::secpath-replace%.log"
      )

template(name="TmplMsg" type="string"
      string="/var/log/remote/msg/%HOSTNAME%/%PROGRAMNAME:::secpath-replace%.log"
      )
```

また、これらのテンプレートは、以下のようにリスト形式で記述することもできます。

```
template(name="TmplAuthpriv" type="list") {
   constant(value="/var/log/remote/auth/")
   property(name="hostname")
   constant(value="/")
   property(name="programname" SecurePath="replace")
   constant(value=".log")
```

```
    }

template(name="TmplMsg" type="list") {
    constant(value="/var/log/remote/msg/")
    property(name="hostname")
    constant(value="/")
    property(name="programname" SecurePath="replace")
    constant(value=".log")
    }
```

このテンプレートテキスト形式は、rsyslog の初心者にとって理解しやすいかもしれません。したがって、要件が変更したら簡単に変更できます。

新規構文への変更を完了するには、モジュールロードコマンドを再作成し、ルールセットを追加して、プロトコル、ポート、およびルールセットにルールセットをバインドする必要があります。

```
module(load="imtcp")

ruleset(name="remote1"){
    authpriv.*   action(type="omfile" DynaFile="TmplAuthpriv")
     *.info;mail.none;authpriv.none;cron.none action(type="omfile" DynaFile="TmplMsg")
}

input(type="imtcp" port="10514" ruleset="remote1")
```

25.7. RSYSLOG モジュールの使用

モジュラー設計により、rsyslog は、追加機能を提供するさまざまな モジュール を提供します。モジュールはサードパーティーによる書き込みが可能であることに注意してください。ほとんどのモジュールは、追加入力 (以下の **Input Modules** 参照) または出力 (以下の **Output Modules** 参照) を提供します。その他のモジュールは、各モジュールに固有の機能を提供します。モジュールは、モジュールの読み込み後に利用可能になる追加の設定ディレクティブを提供する場合があります。モジュールを読み込むには、以下の構文を使用します。

```
$ModLoad MODULE
```

$ModLoad は指定されたモジュールをロードするグローバルディレクティブで、MODULE は希望のモジュールを表します。たとえば、rsyslog を有効にして標準テキストファイルを syslog メッセージに変換する Text File Input Module(imfile)を読み込む場合は、/etc/rsyslog.conf 設定ファイルで以下の行を指定します。

```
$ModLoad imfile
```

rsyslog は、以下の主なカテゴリーに分割されるモジュールを多数提供します。

- 入力モジュール: 入力モジュールは、さまざまなソースからメッセージを収集します。入力モジュールの名前は、常に imfile のように接頭辞 im で始まります。

- 出力モジュール: 出力モジュールはメッセージをネットワーク上に送信したり、データベース内に保存したり、暗号化するなど、様々なターゲットにメッセージを発行するための機能を提供します。出力モジュールの名前は常に omsnmp や om relp などのように接頭辞 om で始まります。

- パーサーモジュール: これらのモジュールは、カスタムの解析ルールの作成や不正な形式のメッセージ解析に使用されます。C プログラミング言語についてある程度の知識があれば、独自のメッセージパーサーが作成できます。パーサーモジュールの名前は、常に pm rfc5424 やpmrfc3164 などの接頭辞 pm で始まります。

- メッセージ修正モジュール: メッセージ修正モジュールは、syslog メッセージの内容を変更します。このモジュールの名前は、mm 接頭辞で始まります。mmanon、mmnormalize、mmjsonparse などのメッセージ修正モジュールは、メッセージの匿名化や正規化に使用されます。

- 文字列生成モジュール - 文字列生成モジュールは、メッセージの内容に基づいて文字列を生成し、rsyslog が提供するテンプレート機能と密接に連携します。テンプレートに関する詳細は 「テンプレート」 を参照してください。文字列生成モジュールの名前は、 sm file や smtradfile のように常に接頭辞 sm で始まります。

- ライブラリーモジュール - ライブラリーモジュールは、他の読み込み可能なモジュール用の機能を提供します。これらのモジュールは、必要でユーザーが設定できない場合に rsyslog によって自動的にロードされます。

利用可能なモジュールの一覧とその詳細な説明は、http://www.rsyslog.com/doc/rsyslog_conf_modules.html を参照してください。

警告

rsyslog はモジュールを読み込む際に、モジュールに対して一部の機能とデータへのアクセスを提供します。これはセキュリティー上の脅威となる可能性があります。セキュリティーリスクを最小限にするために、信頼できるモジュールのみを使用するようにしてください。

25.7.1. テキストファイルのインポート

テキストファイル入力モジュールは imfile と省略され、rsyslog がテキストファイルを syslog メッセージのストリームに変換できるようにします。imfile を使用して、独自のテキストファイルログを作成するアプリケーションからログメッセージをインポートできます。imfile を読み込むには、以下を /etc/rsyslog.conf に追加します。

```
$ModLoad imfile
$InputFilePollInterval int
```

複数のファイルをインポートする場合でも、imfile を一度読み込むだけで十分です。$InputFilePollInterval global ディレクティブは、接続済みテキストファイルの変更に対する rsyslog チェックの頻度を指定します。デフォルトの間隔は 10 秒で、変更するには int を秒単位で指定した時間間隔に置き換えます。

インポートするテキストファイルを特定するには、/etc/rsyslog.conf で以下の構文を使用します。

```
# File 1
$InputFileName path_to_file
$InputFileTag tag:
$InputFileStateFile state_file_name
$InputFileSeverity severity
$InputFileFacility facility
$InputRunFileMonitor

# File 2
$InputFileName path_to_file2
...
```

入力テキストファイルを指定するには、以下の 4 つの設定が必要です。

- *path_to_file を、テキストファイルへのパスに置き換えます。*

- *tag: を、このメッセージのタグ名に置き換えます。*

- *state_file_name は、状態ファイル の一意の名前に置き換えます。rsyslog の作業ディレクトリーに保管されている 状態ファイル は、監視対象ファイルのカーソルを保持し、すでに処理されたパーティションを示します。これらを削除すると、ファイル全体が再度読み込まれます。存在していない名前を指定してください。*

- *ファイルモニタリングを有効にする $InputRunFileMonitor ディレクティブを追加します。この設定がないと、テキストファイルは無視されます。*

必要なディレクティブ以外に、テキスト入力に適用可能な設定がいくつかあります。severity を適切なキーワードに置き換えて、インポートされたメッセージの重大度を設定します。facility を、メッセージを生成したサブシステムを定義するキーワードに置き換えます。重要度と機能のキーワードは、機能または優先度ベースのフィルターで使用されたものと同じものです (「フィルター」 を参照)。

> *例25.15 テキストファイルのインポート*
>
> *Apache HTTP サーバーはログファイルをテキスト形式で作成します。rsyslog の処理機能を apache エラーメッセージに適用するには、まず imfile モジュールを使用してメッセージをインポートします。以下を /etc/rsyslog.conf に追加します。*
>
> ```
> $ModLoad imfile
>
> $InputFileName /var/log/httpd/error_log
> $InputFileTag apache-error:
> $InputFileStateFile state-apache-error
> $InputRunFileMonitor
> ```

25.7.2. データベースへのメッセージのエクスポート

ログデータの処理は、テキストファイルではなくデータベース内で実行するとより速く、より便利なものになります。使用される DBMS のタイプに基づいて、ommysql、ompgsql、omoracle、ommongodb などのさまざまな出力モジュールを選択します。代わりに、libdbi ライブラリーに依存する一般的な omlibdbi 出力モジュールを使用します。omlibdbi モジュールは、Firebird/Interbase、MS SQL、Sybase、SQLite、Ingres、Oracle、mSQL、MySQL、および PostgreSQL のデータベースシステムをサポートします。

> *例25.16 データベースへの rsyslog メッセージのエクスポート*
>
> *rsyslog メッセージを MySQL データベースに保存するには、以下を /etc/rsyslog.conf に追加します。*

```
$ModLoad ommysql

$ActionOmmysqlServerPort 1234
*.* :ommysql:database-server,database-name,database-userid,database-password
```

最初に出力モジュールが読み込まれ、その後に通信ポートが指定されます。上記の例では、サーバー名、データベース名、認証データなどの追加情報は、最後の行で指定されています。

25.7.3. 暗号化トランスポートの有効化

ネットワーク転送の機密性と統合性は、**TLS** または **GSSAPI** 暗号化プロトコルのいずれかによって提供されます。

Transport Layer Security (*TLS*) は、ネットワーク上の通信セキュリティーを提供するために設計された暗号プロトコルです。**TLS** を使用する場合、**rsyslog** メッセージは送信前に暗号化され、送信者と受信者間で相互認証が行われます。

Generic Security Service API (*GSSAPI*) は、プログラムがセキュリティサービスにアクセスするためのアプリケーションプログラミングインターフェースです。**rsyslog** との接続で使用するには、機能している **Kerberos** 環境が必要です。

25.7.4. RELP の使用

Reliable Event Logging Protocol (*RELP*) は、コンピューターネットワークにおけるデータロギング用のネットワーキングプロトコルです。信頼性のあるイベントメッセージの配信を提供するように設計されています。これは、メッセージ損失が許されない環境で便利なものです。

RELP を設定するには、まずサーバーとクライアントの両方で **rsyslog-relp** パッケージをインストールします。

```
~]# yum install rsyslog-relp
```

次に、サーバーとクライアントの両方を設定します。

1. クライアントを設定するには、以下を設定します。

- 必要なモジュールの読み込み
- TCP 入力ポート
- トランスポート設定

以下の設定を /etc/rsyslog.conf ファイルに追加します。

```
$ModLoad omrelp
$ModLoad imuxsock
$ModLoad imtcp
$InputTCPServerRun "port"
*.*  :omrelp:"target_IP":"target_port"
```

port を、必要なポートでリスナーを開始します。

target_IP および target_port をターゲットサーバーを識別する IP アドレスとポートに置き換えます。

2. サーバーを設定するには、以下を実行します。

- モジュールの読み込みの設定
- クライアント設定と同様の TCP 入力を設定します。
- ルールを設定し、実行するアクションを選択します。

以下の設定を /etc/rsyslog.conf ファイルに追加します。

```
$ModLoad imuxsock
$ModLoad imrelp
$RuleSet relp
```

```
*.*   "log_path"
$InputRELPServerBindRuleset relp
$InputRELPServerRun "target_port"
```

target_port をクライアントと同じ値に置き換えます。

上記の例では、log_path はメッセージを保存するためのパスを指定します。

25.8. RSYSLOG のデバッグ

rsyslogd をデバッグモードで実行するには、以下のコマンドを使用します。

```
rsyslogd -dn
```

このコマンドでは、rsyslogd がデバッグ情報を作成し、標準出力に印刷します。-n は「no fork」を意味します。たとえば、デバッグ出力をログファイルに保存して、環境変数を使用してデバッグを変更できます。rsyslogd を起動する前に、コマンドラインで次のコマンドを実行します。

```
export RSYSLOG_DEBUGLOG="path"
export RSYSLOG_DEBUG="Debug"
```

path を、デバッグ情報をログ記録するファイルの場所に置き換えます。RSYSLOG_DEBUG 変数に利用可能なオプションの一覧は、man ページの rsyslogd(8) の関連セクションを参照してください。

/etc/rsyslog.conf ファイルで使用される構文が有効かどうかを確認するには、以下を使用します。

```
rsyslogd -N 1
```

1 は、出力メッセージの長さのレベルを表します。現在提供されているのは 1 つのレベルのみなので、これは前方互換性オプションになります。ただし、検証を実行するには、この引数を追加する必要があります。

25.9. グラフィカル環境でのログファイルの管理

上記のコマンドラインユーティリティー以外に、Red Hat Enterprise Linux 6 はログメッセージの管理にアクセス可能な GUI を提供します。

25.9.1. ログファイルの表示

ほとんどのログファイルはプレーンなテキスト形式で保存されるため、Vi や Emacs などのテキストエディターで表示できます。Vi や Emacs などのテキストエディターで表示できます。一部のログファイルは、システム上のすべてのユーザーが読み取り可能ですが、ほとんどのログファイルを読み取るには root 権限が必要になります。

インタラクティブなリアルタイムアプリケーションでシステムのログファイルを表示するには、Log File Viewer を使用します。

GNOME-SYSTEM-LOG パッケージのインストール

Log File Viewer を使用するには、最初に root で以下を実行して gnome-system-log パッケージがインストールされていることを確認します。

```
~]# yum install gnome-system-log
```

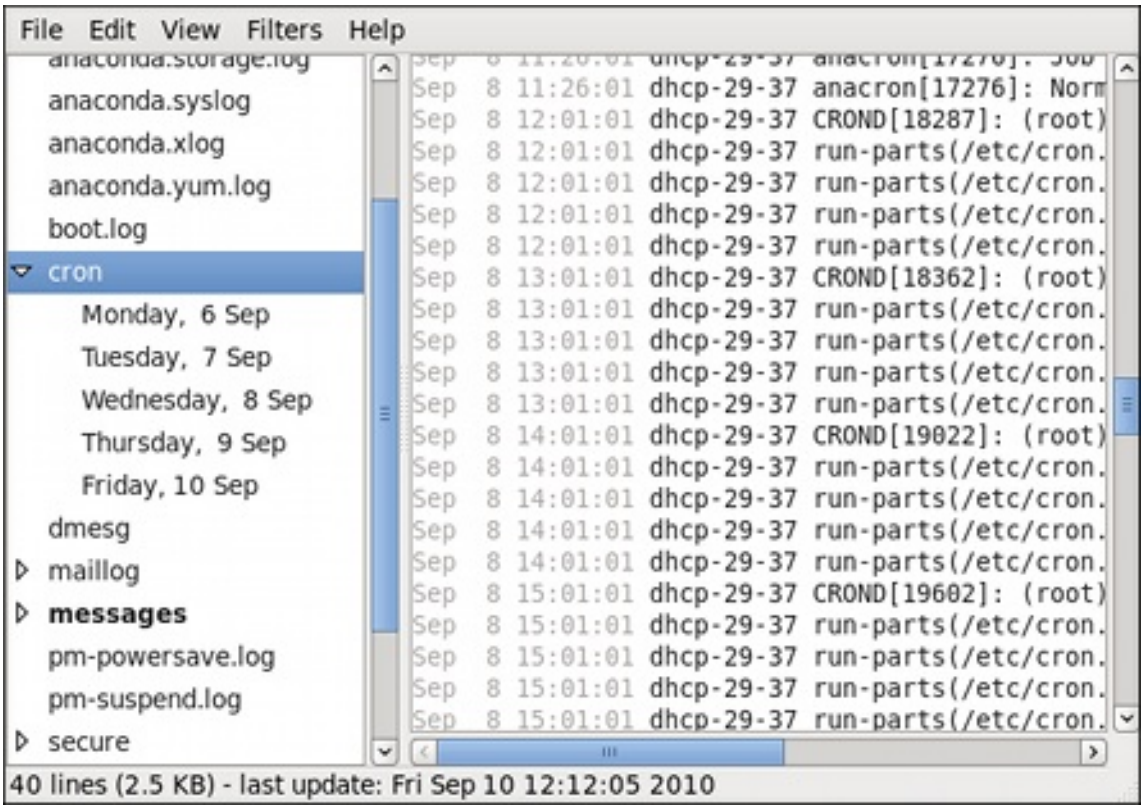
gnome-system-log パッケージは、インストール前に有効にする必要がある Optional サブスクリプションチャンネルで提供されます。Red Hat 追加チャンネルの詳細は、「Optional および Supplementary リポジトリーの追加」 を参照してください。yum を使用したパッケージのインストールは 「パッケージのインストール」 を参照してください。

gnome-system-log パッケージをインストールしたら、Applications → System Tools → Log File Viewer をクリックしてログファイルビューアーを開くか、シェルプロンプトで以下のコマンドを入力します。

```
~]$ gnome-system-log
```

このアプリケーションは、存在するログファイルのみを表示します。そのため、図25.2「ログファイルビューアー」 で表示されている一覧とは異なる場合があります。

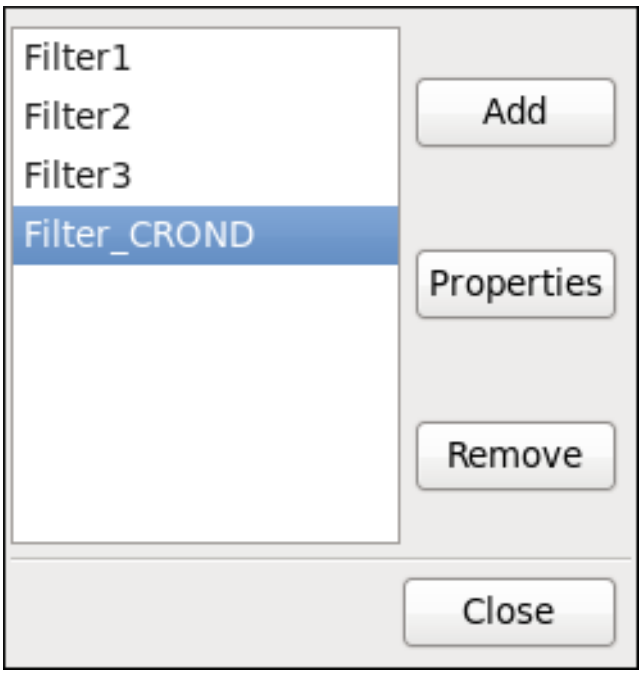
図25.2 ログファイルビューアー



[D]

*ログファイルビューアーアプリケーションを 使用すると、既存のログファイルをフィルタリングできます。メニューから **Filters** をクリックし、**Manage Filters** を選択して必要なフィルターを定義または編集します。*

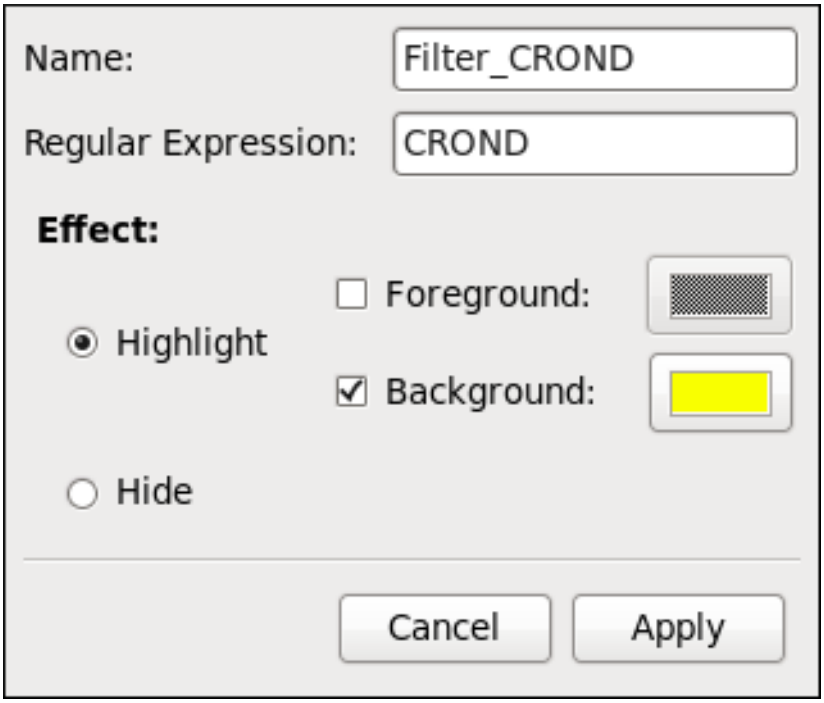
図25.3 ログファイルビューアー： フィルター



[D]

フィルターを追加または編集することで、図25.4「ログファイルビューアー： フィルターの定義」のようにパラメーターを定義できます。

図25.4 ログファイルビューアー： フィルターの定義



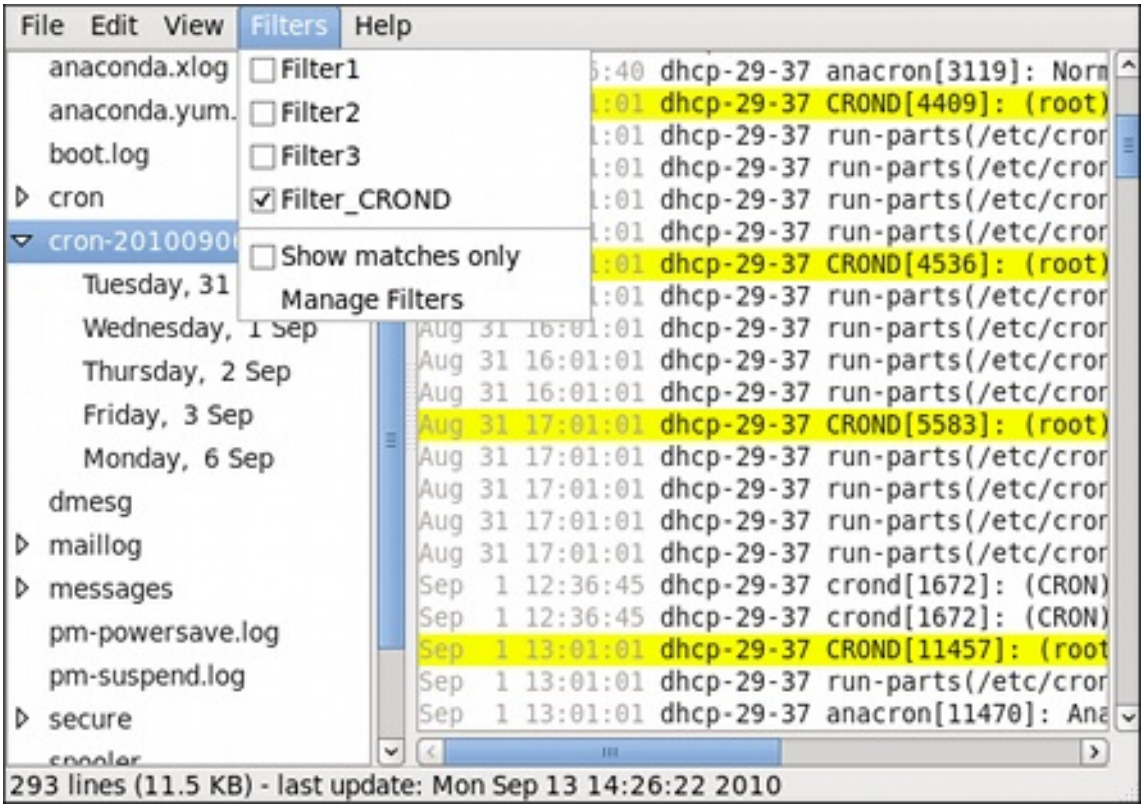
[D]

フィルターを定義する際には、以下のパラメーターを編集できます。

- name: フィルターの名前を指定します。
- 正規表現 - ログファイルに適用され、その中の実行可能なテキストの文字列に一致するよう試行する正規表現を指定します。
- Effect
  - highlight: これが有効な場合、検索結果は選択した色で 強調表示 されます。テキストのバックグラウンドまたはフォアグラウンドを強調するかどうかを選択できます。
  - Hide: これが有効な場合は、検索結果は閲覧中のログファイルから非表示になります。

少なくとも 1 つのフィルターが定義されていれば、フィルター メニューからそれを選択して自動的にフィルターで定義した文字列を検索し、現在表示しているログファイル内でのマッチを強調表示または非表示にします。

図25.5 ログファイルビューアー： フィルターの有効化



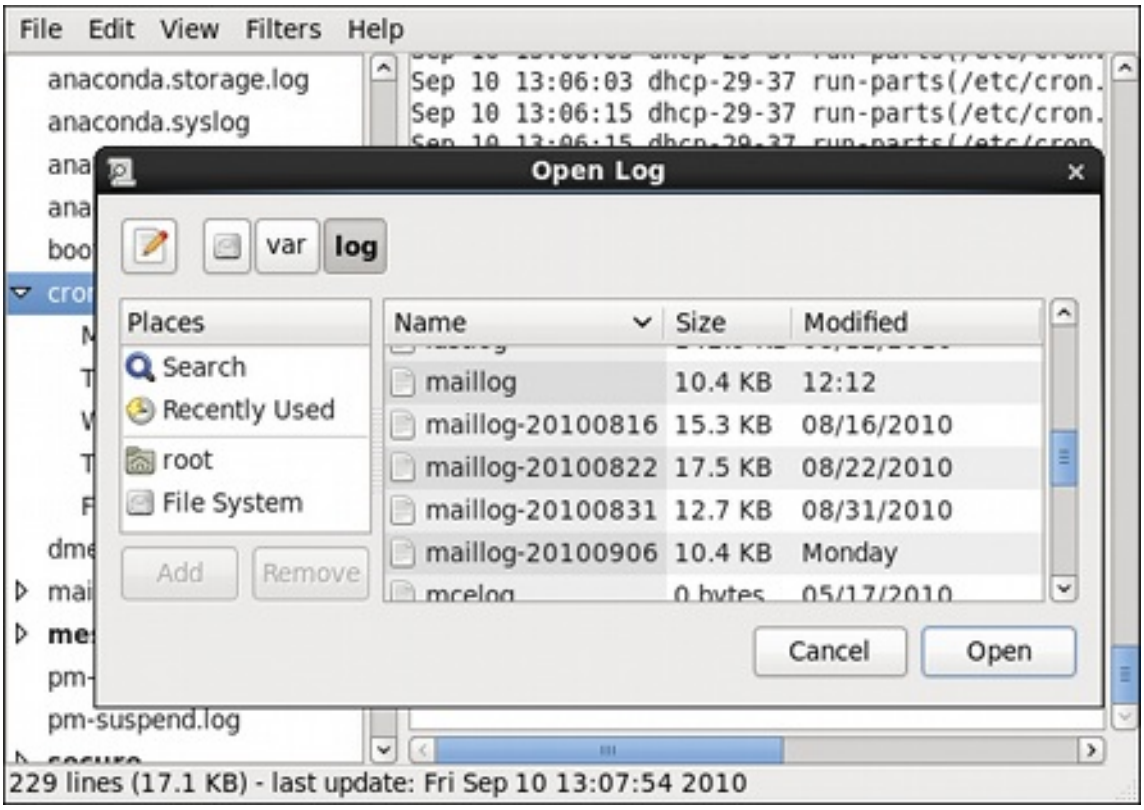
[D]

***Show matches only** オプションを選択すると、現在表示されているログファイルに一致する文字列のみが表示されます。*

25.9.2. ログファイルの追加

*一覧に表示するログファイルを追加するには、**File → Open** を選択します。これにより、表示するログファイルのディレクトリーおよびファイル名を選択できる **Open Log** ウィンドウが表示されます。図25.6「ログファイルビューアー： ログファイルの追加」 では、**Open Log** ウィンドウを示しています。*

図25.6 ログファイルビューアー： ログファイルの追加



[D]

ファイルを開くには、開く ボタンをクリックします。ファイルは表示中の一覧に直ちに追加されるため、そのファイルを選択してコンテンツを表示できます。

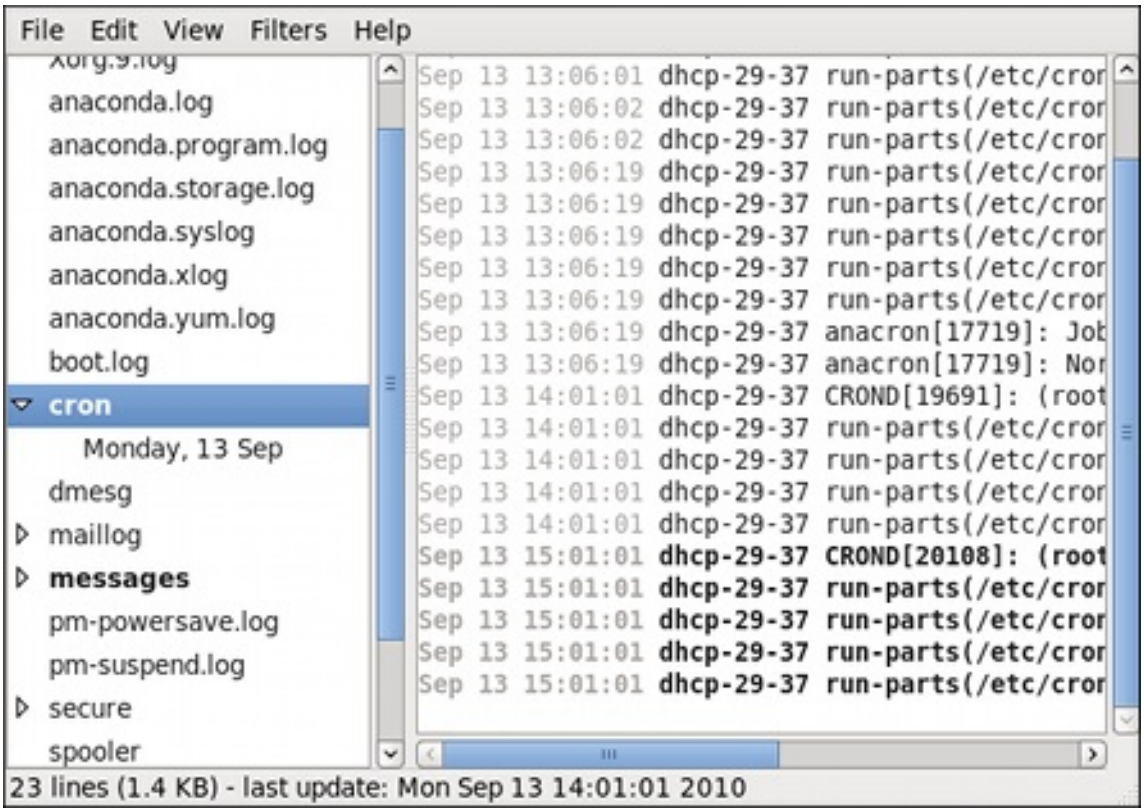
ZIP 形式のログファイルの読み取り

ログファイルビューアーを使用 すると、.gz 形式のログファイルを開くことができます。

25.9.3. ログファイルのモニタリング

ログファイルビューアーは、 デフォルトで開いているすべてのログを監視します。監視されているログファイルに新しい行が追加されると、そのログ名はログの一覧に太字で表示されます。ログファイルが選択または表示されると、ログファイルの下部に新しい行が太字で表示されます。図25.7「ログファイルビューアー： 新しいログアラート」 は、cron ログファイルと messages ログファイル内の新しいアラートを示しています。cron ログファイルをクリックすると、ファイル内のログが表示されます（新しい行は太字で表示されます）。

図25.7 ログファイルビューアー： 新しいログアラート



[D]

25.10. その他のリソース

rsyslog デーモンの設定方法、およびログファイルの場所の特定、表示、監視方法に関する詳細情報は、以下に記載のリソースを参照してください。

インストールされているドキュメント

- *rsyslogd(8): rsyslogd デーモンの man ページは、その使用方法を説明しています。*
- *rsyslog.conf(5): rsyslog.conf の man ページでは、利用可能な設定オプションが説明されています。*
- *logrotate(8): logrotate ユーティリティーの man ページは、その設定方法と使用方法を詳細に説明しています。*

オンラインドキュメント

rsyslog ホームページには、追加のドキュメント、設定例、および動画チュートリアルが含まれます。使用しているバージョンに関連するドキュメントを参照してください。

- *rsyslog ホームページの rsyslog バージョン 5 ドキュメント - Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;LinuxRed Hat Enterprise Linux 6nbsp;6 のデフォルトの rsyslog はバージョン 5 です。*

- *rsyslog ホームページの rsyslog バージョン 7 ドキュメント - Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;LinuxRed Hat Enterprise Linux 6nbsp;6 では、rsyslog のバージョン 7 が利用できます。rsyslog7*

- *Description of queues on the rsyslog Home Page: さまざまなタイプのメッセージキューおよびその使用方法に関する全般情報。*

その他の参考資料

- *4章権限の取得 では、su および sudo コマンドを使用して管理者権限を取得する方法を説明しています。*

第26章 MYSQL のアップグレード

Red Hat は、アップストリームのセキュリティーおよびバグ修正が利用可能であれば、メンテナンスサポート 2 フェーズが終了するまで、現在 Red Hat Enterprise Linux に同梱されていたアップストリームバージョンの MySQL を完全にサポートしています。Red Hat Enterprise Linux のライフサイクルの概要は、を参照してください https://access.redhat.com/support/policy/updates/errata#Maintenance_Support_2_Phase。

最新バージョンの MySQL、MySQL 5.6、および MySQL 5.7 は、rh-mysql56 および rh-mysql57 Software Collections として提供されます。これらのコンポーネントは、AMD64 および Intel 64 のアーキテクチャー上の Red Hat Enterprise Linux 6 のサポートされるすべてのリリースで利用可能な Red Hat Software Collections の一部です。

Red Hat Software Collections へのアクセス方法は、『Red 『Hat Software Collections Release Notes』』 を参照してください。

個々のコンポーネントのサポート期間の詳細は、「『Red Hat Software Collections の製品ライフサイクル』」を参照してください。

MySQL 5.1 から現在サポートされているバージョンに直接移行できないことに注意してください。MySQL 5 .『1 から MySQL 5.5、MySQL 5.5、MySQL 5.5、MySQL 5』.『6、および MySQL 5.6 から MySQL』 5.7 に移行する 詳細な手順を参照してください。

第27章 システムタスクの自動化

タスクは、ジョブ としても知られており、指定期間、またはシステムの負荷平均が 0.8 を下回る場合に自動的に実行するように設定できます。

Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux は、システムの更新を維持するために重要なシステムタスクを実行するように事前設定されています。たとえば、locate コマンドで使用される s locate データベースが毎日更新されます。システム管理者は、自動化されたタスクを使用して定期的なバックアップの実行、システムの監視、カスタムスクリプトの実行などを実行できます。

Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux には、cron、anacron、および batch などの自動化されたタスクユーティリティーが含まれてい ます。

すべてのユーティリティーは、異なるジョブタイプをスケジュールすることを目的としています。Cron および Anacron の繰り返しジョブのスケジュールは、At と Batch が 1 回限りのジョブをスケジュールするためのものです（それぞれ 「cron および Anacron」 と 「at および Batch」 を参照してください）。

27.1. CRON および ANACRON

Cron と Anacron はどちらもデーモンであり、正確な時間、日、曜日、および週で定義される特定の時点に対して繰り返し実行するタスクをスケジュールできるデーモンです。

Cron ジョブは毎分のみ実行できます。ただし、このユーティリティーは、システムが継続的に実行されていることを前提とし、ジョブがスケジュールされている時にシステムがオンになっていない場合は、ジョブは実行されません。

一方、anacron は、ジョブのスケジュール時にシステムが実行していない場合にスケジュールされたジョブを記憶します。その後、システムは起動したらすぐにジョブが実行されます。ただし、anacron は 1 日 1 回のみジョブを実行できます。

27.1.1. cron および Anacron のインストール

Cron および Anacron をインストールするには、Cron と Anacron を含む cronie パッケージ（cronie-anacron はcronie-anacron のサブパッケージ）をインストールする必要があります。cronie

パッケージがすでにシステムにインストールされているかどうかを確認するには、rpm -q cronie cronie-anacron コマンドを実行します。パッケージがすでにインストールされている場合は、cronie

および cronie-anacron パッケージのフルネームを返します。もしくはパッケージが利用できないことを通知します。

パッケージをインストールするには、以下の形式で yum コマンドを使用します。

```
yum install package
```

たとえば、Cron と Anacron の両方をインストールするには、シェルプロンプトで以下を入力します。

```
~]# yum install cronie cronie-anacron
```

このコマンドを実行するには、スーパーユーザーの権限 (つまり root としてログイン) が必要であることに注意してください。Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux に新しいパッケージをインストールする方法の詳細は、「パッケージのインストール」 を参照してください。

27.1.2. crond サービスの実行

cron ジョブと anacron ジョブの両方が crond サービスによって選択されます。本セクションでは、crond サービスを起動、停止、および再起動する方法を説明し、特定のランレベルでそのサービスを有効にする方法を説明します。ランレベルの概念と Red Hat Enterprise Linux でシステムサービスを管理する方法は、12章サービスおよびデーモン を参照してください。

27.1.2.1. cron サービスの起動と停止

サービスが実行中かどうかを確認するには、コマンド service crond status を使用します。

現行セッションで crond サービスを実行するには、root で次のコマンドを実行します。

```
service crond start
```

起動時にサービスが自動的に起動するよう設定するには、以下のコマンドを使用します。

```
chkconfig crond on
```

このコマンドは、ランレベル 2、3、4、および 5 でサービスを有効にします。または、「サービスの有効化および無効化」 の説明に従って Service Configuration ユーティリティーを使用できます。

27.1.2.2. cron サービスの停止

crond サービスを停止するには、rootで次のコマンドを実行します。

```
service crond stop
```

ブート時のサービスの起動を無効にするには、以下のコマンドを使用します。

```
chkconfig crond off
```

このコマンドは、すべてのランレベルでサービスを無効にします。または、「サービスの有効化および無効化」 の説明に従って Service Configuration ユーティリティーを使用できます。

27.1.2.3. cron サービスの再起動

crond サービスを再起動するには、シェルプロンプトで以下を入力します。

```
service crond restart
```

このコマンドで、サービスの停止と再起動が連続して行われます。

27.1.3. Anacron ジョブの設定

ジョブをスケジュールするための主な設定ファイルは /etc/anacrontab ファイルです。このファイルは root ユーザーのみがアクセスできます。このファイルには、以下が含まれます。

```
SHELL=/bin/sh
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
# the maximal random delay added to the base delay of the jobs
RANDOM_DELAY=45
# the jobs will be started during the following hours only
START_HOURS_RANGE=3-22

#period in days   delay in minutes   job-identifier   command
1        5     cron.daily    nice run-parts /etc/cron.daily
7        25    cron.weekly   nice run-parts /etc/cron.weekly
@monthly  45    cron.monthly  nice run-parts /etc/cron.monthly
```

最初の 3 行は、anacron タスクを実行する環境を設定する変数を定義します。

- *SHELL - ジョブの実行に使用するシェル環境（この例では Bash シェル）*

- *PATH - 実行可能なプログラムへのパス*

- *MAILTO - メールで anacron ジョブの出力を受信するユーザーのユーザー名*

  MAILTO 変数が定義されていない場合(MAILTO=)、メールは送信されません。

次の 2 つの変数は、定義されたジョブのスケジュール時間を変更します。

- *RANDOM_DELAY - 各ジョブに指定される delay in minutes 変数に追加される最大分単位数*

  遅延の最小値は、デフォルトでは 6 分に設定されます。

  たとえば、RANDOM_DELAY を 12 に設定すると、特定の分析内のジョブごとに、6 分から 12 分の間に delay in minutes が追加されます。RANDOM_DELAY 0 など、6 以下の値に設定することもできます。0 に設定すると、ランダムな遅延は追加されません。これは、1 つのネットワーク接続を共有する複数のコンピューターが毎日同じデータをダウンロードする必要がある場合などに便利です。

- *START_HOURS_RANGE - スケジュールされたジョブを実行できる間隔（時間単位）*

  電源障害などの理由で時間間隔がないと、スケジュールされたジョブはその日に実行されません。

/etc/anacrontab ファイルの残りの行はスケジュールされたジョブを表し、以下の形式に従います。

```
period in days   delay in minutes   job-identifier   command
```

- *period in days - ジョブ実行の頻度（日数単位）*

プロパティーの値は整数またはマクロ（@daily、@weekly、@monthly）として定義できます。ここで、@daily は整数 1 として、@weekly は 7 と同じ値、@monthly は月の長さが 1 度実行されることを指定します。

- delay in minutes - ジョブを実行する前に、anacron が待機する時間

  プロパティーの値は整数として定義されます。値が 0 に設定されている場合、遅延は適用されません。

- job-identifier - ログファイルで使用される特定のジョブを参照する一意の名前

- command - 実行するコマンド

  コマンドは、ls /proc >> /tmp/proc などの コマンド、またはカスタムスクリプトを実行するコマンドのいずれかです。

ハッシュマーク (#) で始まる行はすべてコメントで、これは処理されません。

27.1.3.1. Anacron ジョブの例

以下の例は、簡単な /etc/anacrontab ファイルを示しています。

```
SHELL=/bin/sh
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root

# the maximal random delay added to the base delay of the jobs
RANDOM_DELAY=30
# the jobs will be started during the following hours only
START_HOURS_RANGE=16-20

#period in days   delay in minutes   job-identifier   command
1        20    dailyjob      nice run-parts /etc/cron.daily
7        25    weeklyjob     /etc/weeklyjob.bash
@monthly  45    monthlyjob    ls /proc >> /tmp/proc
```

この anacrontab ファイルで定義されているすべてのジョブは 6-30 分によって無作為に遅延し、16:00 から 20:00 の間で実行できます。

最初に定義されたジョブは 16:26 から 16:50（RANDOM_DELAY は 6 分から 30 分の間であり、delay in minutes プロパティーが 20 分の間です）。このジョブに指定されたコマンドは、run-parts スクリプトを使用して、/etc/cron.daily ディレクトリーに存在するすべてのプログラムを実行します（run-parts スクリプトはディレクトリーをコマンドライン引数として受け入れ、ディレクトリー内のすべてのプログラムを順次実行します）。

2 番目のジョブは、1 週間 1 回 /etc ディレクトリーで weeklyjob.bash スクリプトを実行します。

3 番目のジョブはコマンドを実行し、/proc の内容を 1 カ月 /tmp/proc ファイル(ls /proc >> /tmp/proc)に書き込みます。

27.1.4. cron ジョブの設定

cron ジョブの設定ファイルは /etc/crontab で、root ユーザーのみが変更できます。このファイルには、以下が含まれます。

```
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
HOME=/
# For details see man 4 crontabs
# Example of job definition:
# .---------------- minute (0 - 59)
# | .------------- hour (0 - 23)
# | | .---------- day of month (1 - 31)
# | | | .------- month (1 - 12) OR jan,feb,mar,apr ...
# | | | | .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# | | | | |
# * * * * * username  command to be executed
```

最初の 3 行には、SHELL、PATH、および MAILTO の anacrontab ファイルと同じ変数定義が含まれます。これらの変数に関する詳細は、「Anacron ジョブの設定」 を参照してください。

また、このファイルは HOME 変数を定義できます。HOME 変数はディレクトリーを定義します。このディレクトリーは、コマンドまたはジョブで実行されるスクリプトの実行時にホームディレクトリーとして使用されます。

/etc/crontab ファイルの残りの行はスケジュールされたジョブを表し、以下の形式となっています。

```
minute   hour   day   month   day of week   username   command
```

以下は、ジョブの実行時間を定義します。

- minute - 0 から 59 までの任意の整数
- hour - 0 から 23 までの整数
- day - 1 から 31 までの整数（月を指定する場合は有効な日付である必要があります）
- month - 1 から 12 までの整数（または jan または feb などの月の省略名）
- day of week - 0 から 7 までの任意の整数。0 または 7 は日曜日を表します（または sun または mon などの曜日の短縮名）。

以下は、他のジョブプロパティーを定義します。

- username - ジョブが実行されるユーザーを指定します。
- command - 実行するコマンド

コマンドは、ls /proc /tmp/proc などのコマンド、またはカスタムスクリプトを実行するコマンドのいずれかです。

上記のいずれの値でも、アスタリスク(*)を使用して有効な値をすべて指定できます。たとえば、月の値をアスタリスクとして定義する場合、ジョブは、他の値の制約内で毎月実行されます。

整数間のハイフン(-)は整数の範囲を指定します。たとえば、1-4 は整数の 1、2、3、および 4 を意味します。

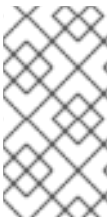
コンマで区切られた値の一覧(,)はリストを指定します。たとえば、3, 4, 6, 8 はこれら 4 つの整数を正確に示します。

スラッシュ(/)を使用して step の値を指定できます。整数の値は、/integer の範囲の後に続く範囲内で省略されます。たとえば、0-59/2 で定義される 1 分間の値は、分ごとの分を表します。ステップ値はアスタリスクでも使用することができます。たとえば、月の値が */3 として定義されている場合、タスクは 3 月ごとに実行されます。

ハッシュマーク (#) で始まる行はすべてコメントで、これは処理されません。

root 以外のユーザーは、crontab ユーティリティーを使用して cron タスクを設定できます。ユーザー定義の crontab は /var/spool/cron/ ディレクトリーに保存され、作成したユーザーが実行したように実行されます。

ユーザーとして crontab を作成するには、そのユーザーでログインし、VISUAL または EDITOR 環境変数で指定されたエディターでユーザーの crontab を編集します。このファイルは、/etc/crontab と同じ形式を使用します。crontab への変更が保存されると、crontab はユーザー名として保存され、/var/spool/cron/ ユーザー名のファイルに書き込まれます。crontab ファイルの内容を一覧表示するには、crontab -l コマンドを使用します。



ユーザーを指定しないでください。

crontab ユーティリティーを使用してジョブを定義するときにユーザーを指定しないでください。

/etc/cron.d/ ディレクトリーには、/etc/crontab ファイルと同じ構文を持つファイルが含まれます。このディレクトリーでファイルを作成および変更できるのは、root のみです。

変更を適用するデーモンを再起動しません。

cron デーモンは /etc/anacrontab ファイル、/etc/crontab ファイル、/etc/cron.d/ ディレクトリー、および /var/spool/cron/ ディレクトリーが変更されるたびにチェックし、検出された変更はメモリーに読み込まれます。そのため、anacrontab ファイルまたは crontab ファイルに変更を加えた後にデーモンを再起動する必要はありません。

27.1.5. cron へのアクセスの制御

Cron へのアクセスを制限するには、/etc/cron.allow ファイルおよび /etc/cron.deny ファイルを使用できます。これらのアクセス制御ファイルは、各行にユーザー名が同じ形式を使用します。いずれのファイルでも、空白文字は許可されません。

cron.allow ファイルが存在する場合は、ファイルに一覧表示されているユーザーのみが cron を使用でき、cron.deny ファイルは無視されます。

cron.allow ファイルが存在しない場合は、cron.deny ファイルにリストされているユーザーは Cron を使用することができません。

アクセス制御ファイルを変更した場合でも、Cron デーモン(crond)を再起動する必要はありません。アクセス制御ファイルは、ユーザーが cron ジョブの追加または削除を試みるたびにチェックされます。

root ユーザーは、アクセス制御ファイルに記載されているユーザー名に関係なく、常に cron を使用できます。

PAM（プラグ可能な認証モジュール）を使用してアクセスを制御することもできます。この設定は /etc/security/access.conf ファイルに保存されます。たとえば、以下の行をファイルに追加しますが、他のユーザーは作成できませんが、root ユーザーは crontabs を作成することができます。

```
-:ALL EXCEPT root :cron
```

禁止されているジョブは、適切なログファイルにログインするか、標準出力に返される「crontab -e」を使用する場合に記録されます。詳細は、access.conf.5 (man 5 access.conf)を参照してください。

27.1.6. cron ジョブのブラックリストおよびホワイトリスト

ジョブのブラックリストとホワイトリストは、実行する必要がないジョブの一部を定義するために使用されます。これは、/etc/cron.daily などの Cron ディレクトリーで run-parts スクリプトを呼び出す場合に便利です。ユーザーがディレクトリーにあるプログラムをジョブのブラックリストに追加すると、run-parts スクリプトはこれらのプログラムを実行しません。

ブラックリストを定義するには、run-parts スクリプトが実行されるディレクトリーに jobs.deny ファイルを作成します。たとえば、/etc/cron.daily から特定のプログラムを省略する必要がある場合は、/etc/cron.daily/jobs.deny ファイルを作成します。このファイルで、実行から省略するプログラムの名前を指定します（同じディレクトリーにあるプログラムのみを登録できます）。run-parts /etc/cron.daily などの cron.daily ディレクトリーからプログラムを実行するコマンドを実行すると、jobs.deny ファイルに定義したプログラムは実行されません。

ホワイトリストを定義するには、job .allow ファイル を作成します。

jobs.deny および jobs.allow の原則は、「cron へのアクセスの制御」 セクションで説明されている cron.deny および cron.allow の原則と同じです。

27.2. AT および BATCH

Cron は繰り返し実行されるタスクのスケジュールに使用されますが、At ユーティリティーは 1 回限りのタスクを特定の時間にスケジュールするために使用されます。batch ユーティリティーは、システムの負荷平均が 0.8 未満の場合に 1 回限りのタスクが実行されるようスケジュールするために使用されます。

27.2.1. at および Batch のインストール

at パッケージがシステムにインストールされているかどうかを確認するには、rpm -q コマンドを実行します。パッケージがすでにインストールされている場合は、at パッケージのフルネームを返します。もしくはパッケージが利用できないことを通知します。

パッケージをインストールするには、以下の形式で yum コマンドを使用します。

```
yum  install  package
```

At と Batch をインストールするには、シェルプロンプトで以下を入力します。

```
~]# yum install at
```

このコマンドを実行するには、スーパーユーザーの権限 (つまり root としてログイン) が必要であることに注意してください。Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux に新しいパッケージをインストールする方法の詳細は、「パッケージのインストール」 を参照してください。

27.2.2. at サービスの実行

At および Batch ジョブはいずれも atd サービスにより選択されます。本セクションでは、atd サービスを起動、停止、再起動する方法と、特定のランレベルでそのサービスを有効にする方法を説明します。ランレベルの概念と Red Hat Enterprise Linux でシステムサービスを管理する方法は、12章サービスおよびデーモン を参照してください。

27.2.2.1. at サービスの起動および停止

サービスが実行中かどうかを確認するには、コマンド service atd status を使用します。

現行セッションで atd サービスを実行するには、root で次のコマンドを実行します。

```
service atd start
```

システムの起動時にサービスが自動的に起動するようにするには、以下のコマンドを使用します。

```
chkconfig atd on
```

注記

システムの起動時にサービスを自動的に開始することが推奨されます。

このコマンドは、ランレベル 2、3、4、および 5 でサービスを有効にします。または、「サービスの有効化および無効化」 の説明に従って Service Configuration ユーティリティーを使用できます。

27.2.2.2. at サービスの停止

atd サービスを停止するには、rootで次のコマンドを実行します。

```
service atd stop
```

ブート時のサービスの起動を無効にするには、以下のコマンドを使用します。

```
chkconfig atd off
```

このコマンドは、すべてのランレベルでサービスを無効にします。または、「サービスの有効化および無効化」 の説明に従って Service Configuration ユーティリティーを使用できます。

27.2.2.3. at サービスの再起動

atd サービスを再起動するには、シェルプロンプトで以下を入力します。

```
service atd restart
```

このコマンドで、サービスの停止と再起動が連続して行われます。

27.2.3. at ジョブの設定

At ユーティリティーを使用して特定の時間に 1 回限りのジョブをスケジュールするには、以下を実行します。

1. コマンドラインで、 TIMEで コマンドを入力します。TIME は、コマンドを実行するタイミングになります。

   TIME 引数は、以下のいずれかの形式で定義できます。

   - HH:MM は正確な時間と分を指定します。たとえば、04 : 00 は 4:00 を指定します。
   - 午前 0 時 は午前 12:00 を指定します。
   - noon は 12:00 p.m を指定します。
   - teatime は 4:00 p.m を指定します。
   - MONTHDAYYEAR 形式。たとえば、1 月 15 日は 2012 年 1 月 15 日と指定しています。year の値は任意です。
   - MMDDYY、MM/DD/YY、または MM.DD の形式。たとえば、16 年 1 月 15 日の場合は 011512 です。
   - TIME は 整数および値タイプ（minute、hour、days、または weeks）として定義されるようになりました。たとえば、now + 5 days は、今後 5 日に同時にコマンドが実行されるように指定します。

     最初に時間を指定し、その後にオプションの日付を指定する必要があります。時間形式の詳細は、/usr/share/doc/at- <version> /timespec テキストファイルを参照してください。

   指定した時間を過去にすると、ジョブは次の日に実行されます。

2. 表示される at> プロンプトで、ジョブコマンドを定義します。

   - ジョブが実行すべきコマンドを入力して、Enter を押します。必要に応じて、手順を繰り返して複数のコマンドを提供します。

   - プロンプトでシェルスクリプトを 入力 し、スクリプトの各行で Enter を押します。

     ジョブは、ユーザーの SHELL 環境、ユーザーのログインシェル、または /bin/sh（最初に見つかったもの）に設定したシェルを使用します。

3. 完了したら、空の行で Ctrl+D を押してプロンプトを終了します。

コマンドセットやスクリプトが標準出力に情報を表示しようとする場合、その出力はユーザーにメールで送信されます。

保留中のジョブ一覧を表示するには、atq コマンドを使用します。詳細は、「保留中のジョブの表示」 を参照してください。

at コマンドの使用を制限することもできます。詳細は、「at と batch へのアクセスの制御」 を参照してください。

27.2.4. バッチジョブの設定

Batch アプリケーションは、システム負荷平均が 0.8 未満の場合に、定義した 1 回限りのタスクを実行します。

バッチジョブを定義するには、以下を行います。

1. コマンドラインで、バッチ を入力します。

2. 表示される at> プロンプトで、ジョブコマンドを定義します。

   - ジョブが実行すべきコマンドを入力して、Enter を押します。必要に応じて、手順を繰り返して複数のコマンドを提供します。

   - プロンプトでシェルスクリプトを 入力 し、スクリプトの各行で Enter を押します。

     スクリプトを入力すると、ジョブはユーザーの SHELL 環境、ユーザーのログインシェル、または /bin/sh （最初に見つかったもの）に設定したシェルを使用します。

3. 完了したら、空の行で Ctrl+D を押してプロンプトを終了します。

コマンドセットやスクリプトが標準出力に情報を表示しようとする場合、その出力はユーザーにメールで送信されます。

保留中のジョブ一覧を表示するには、atq コマンドを使用します。詳細は、「保留中のジョブの表示」 を参照してください。

batch コマンドの使用を制限することもできます。詳細は、「at と batch へのアクセスの制御」 を参照してください。

27.2.5. 保留中のジョブの表示

保留中の At ジョブおよび batch ジョブを表示するには、atq コマンドを実行します。atq コマンドは、保留中のジョブ一覧と、各ジョブが別々の行に表示されます。各行は、ジョブ番号、日付、時間、

ジョブクラス、およびユーザー名の形式に従います。ユーザーは独自のジョブのみを表示できます。root ユーザーが atq コマンドを実行すると、全ユーザーのジョブがすべて表示されます。

27.2.6. 追加のコマンドラインオプション

at および batch の他のコマンドラインオプションには、以下が含まれます。

表27.1 at および batch のコマンドラインオプション

オプション	説明
-f	プロンプトで指定する代わりに、ファイルからコマンドまたはシェルスクリプトを読み取ります。
-m	ジョブが完了したらユーザーにメールを送信します。
-v	ジョブが実行される時間を表示します。

27.2.7. at と batch へのアクセスの制御

at および batch コマンドへのアクセスを制限するには、/etc/at.allow ファイルおよび /etc/at.deny ファイルを使用します。これらのアクセス制御ファイルは、各行にユーザー名を定義するのと同じ形式を使用します。いずれのファイルでも、空白は許可されません。

at.allow ファイルが存在する場合は、ファイルに一覧表示されているユーザーのみが at または batch を使用でき、at.deny ファイルは無視されます。

at.allow が存在しない場合は、at.deny に一覧表示されているユーザーは at または batch を使用できません。

アクセス制御ファイルを変更した場合でも、at デーモン(atd)を再起動する必要はありません。アクセス制御ファイルは、ユーザーが at または batch のコマンドの実行を試みるたびに読み込まれます。

root ユーザーは、アクセス制御ファイルの内容に関係なく、常に at コマンドおよび batch コマンドを実行できます。

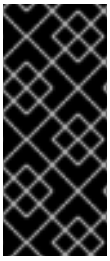
27.3. その他のリソース

自動タスクの設定に関する詳細は、以下のインストール済みドキュメントを参照してください。

- *cron の man ページには cron の概要が記載されています。*
- *セクション 1 および 5 の crontab man ページ：*
  - *セクション 1 の man ページには、crontab ファイルの概要が記載されています。*
  - *セクション 5 の man ページには、ファイルの形式とエントリーの例が記載されています。*
- *anacron の man ページには、anacron の概要が記載されています。*
- *anacrontab man ページには、anacrontab ファイルの概要が記載されています。*
- */usr/share/doc/at- <version> /timespec には、cron ジョブ定義で使用可能な時間値に関する詳細情報が含まれます。*
- *man ページ で は、at および batch の説明とそのコマンドラインオプションが記載されています。*

第28章 自動バグレポーティングツール(ABRT)

Automatic Bug Reporting Tool は通常 ABRT として省略され、abrtd デーモンと、検出された問題を処理、分析、および報告する多数のシステムサービスおよびユーティリティーで構成されます。デーモンは、ほとんどの期間のバックグラウンドでサイレントに実行され、アプリケーションがクラッシュしたり、カーネルの oops が検出されると、springs がアクションになります。次に、デーモンは、コアファイルがある場合、クラッシュアプリケーションのコマンドラインパラメーター、フォレンジックユーティリティーのその他のデータがある場合は、コアファイルなどの関連する問題データを収集します。最も重要な ABRT コンポーネントの概要は、表28.1「ABRT の基本コンポーネント」 を参照してください。



ABRT バージョン 2.0 への移行

Red Hat Enterprise Linux 6.2 では、Automatic Bug Reporting Tool がバージョン 2.0 にアップグレードされました。ABRT 2 シリーズでは、自動バグ検出およびレポートに大きな改善が加えられています。

表28.1 ABRT の基本コンポーネント

コンポーネント	パッケージ	説明
abrtd	abrt	root ユーザー下でバックグラウンドサービスとして実行する ABRT デーモン。
abrt-applet	abrt-gui	**abrtd** からメッセージを受信し、新しい問題が発生するたびに通知するプログラム。
abrt-gui	abrt-gui	収集した問題データを表示し、さらに処理できる GUI アプリケーション。
abrt-cli	abrt-cli	GUI と同様の機能を提供するコマンドラインインターフェースです。
abrt-ccpp	abrt-addon-ccpp	C/C++ の問題をアナライザーを提供する ABRT サービス。
abrt-oops	abrt-addon-kerneloops	カーネル oopses アナライザーを提供する ABRT サービス。
abrt-vmcore	abrt-addon-vmcore[a]	カーネルパニックアナライザーおよびレポーターを提供する ABRT サービス。

[a] abrt-addon-vmcore パッケージは、Optional サブスクリプションチャンネルで提供されます。Red Hat 追加チャンネルの詳細は、「Optional および Supplementary リポジトリーの追加」 を参照してください。

ABRT は現在、C/C++ 言語および Python 言語で記述されたアプリケーションにおけるクラッシュとカーネルの oops の検出をサポートしています。Red Hat Enterprise Linux 6.3 では、追加の abrt-addon-vmcore パッケージがインストールされ、kdump クラッシュダンプメカニズムが有効で、それに応じてシステムに設定した場合に、ABRT はカーネルパニックを検出することもできます。

ABRT は問題をリモートの問題トラッカーに報告できます。レポートは、問題が検出されるたびに自動的に行われるように設定したり、問題データをユーザーが手動で確認、レビュー、報告、および削除できるように設定できます。レポートツールは、Bugzilla データベースである Red Hat テクニカルサポート(RHTSupport)サイトに問題データを送信したり、FTP/SCP を使用してアップロードしたり、電子メールを行ったり、ファイルに書き込んだりすることができます。

（新規の問題データの作成などとは対照的に）既存の問題データを処理する ABRT の一部は、別のプロジェクト libreport に基づいました。libreport ライブラリーは、問題を分析および報告するための一般的なメカニズムを提供し、ABRT 以外のアプリケーションによって使用されます。ただし、ABRT および libreport 操作と設定は密接に統合されています。したがって、これらについては、本書で説明しています。

問題が検出されると、ABRT はその問題を既存の問題データと比較し、同じ問題が記録されているかどうかを判断します。存在する場合は、既存の問題データが更新され、最新の（重複している）問題は再度記録されません。この問題が ABRT で認識されない場合は、問題データディレクトリー が作成されます。問題データディレクトリーは、通常、analyzer、アーキテクチャー、コアダンプ、cmdline、実行可能な 、 kernel、os_release、理由、time、uid などのファイルで構成されます。

backtrace などの他のファイルは、分析中に使用する分析方法とその構成設定に応じて作成できます。これは、使用するアナライザーメソッドやその設定によって異なります。たとえば、kernel ファイルには、クラッシュしたカーネルのバージョンが記録されます。

問題ディレクトリーが作成され、問題データが収集されると、ABRT GUI またはコマンドラインの abrt-cli ユーティリティーを使用して問題をさらに処理し、分析し、報告することができます。これらのツールの詳細は、それぞれ「グラフィカルユーザーインターフェースの使用」 と 「コマンドラインインターフェースの使用」 を参照してください。

REPORT ユーティリティーがサポート対象外になりました。

ABRT を使用して検出された問題をさらに分析して報告せず、代わりにレガシー問題レポートツールを使用して問題を報告する場合は、新しいバグを提出できないことに注意してください。report ユーティリティーを使用すると、RHTSupport または Bugzilla データベースの既存のバグに新しいコンテンツを割り当てることができます。これを行うには、次のコマンドを使用します。

```
report [-v] --target target --ticket ID file
```

ここでの target は、Bugzilla に報告できるように RHTSupport または bugzilla に報告するための strata です。ID は、各データベースで既存の問題のケースを特定する数字を表し、ファイル は問題のケースに追加する情報が含まれるファイルです。

新しい問題を報告し、abrt-cli を使用しない場合は、report の代わりに report-cli ユーティリティーを使用できるようになりました。以下のコマンドを実行し、report-cli で問題の報告プロセスをガイドさせます。

```
report-cli -r dump_directory
```

ここでの dump_directory は、ABRT が作成した問題データディレクトリー、または libreport を使用する他のアプリケーションです。report-cli の詳細は、man report-cli を参照してください。

28.1. ABRT のインストールとサービスの起動

この使用の前提条件として、abrtd デーモンでは、abrt ユーザーが /var/spool/abrt ディレクトリーにファイルシステム操作に存在する必要があります。abrt パッケージをインストールすると、UID と GID が 173 の abrt ユーザーが存在していない場合は自動的に作成されます。それ以外の場合は、abrt ユーザーを手動で作成できます。この場合、abrtd は特定の UID および GID を要求しないため、任意の UID および GID を選択できます。

ABRT を使用する最初の手順として、以下のコマンドを root ユーザーとして実行して、abrt-desktop パッケージがシステムにインストールされていることを確認する必要があります。

```
~]# yum install abrt-desktop
```

abrt-desktop をインストールすると、グラフィカルインターフェースでのみ ABRT を使用できます。コマンドラインで ABRT を使用する場合は、abrt-cli パッケージをインストールします。

```
~]# yum install abrt-cli
```

Yum パッケージマネージャーでパッケージをインストールする方法の詳細は、「パッケージのインストール」 を参照してください。

次の手順では、abrtd が実行していることを確認する必要があります。デーモンは通常、ブート時に起動するように設定されています。以下のコマンドを root として使用し、現在のステータスを確認できます。

```
~]# service abrtd status
abrtd (pid 1535) is running...
```

service コマンドが abrt is stopped メッセージを返すと、デーモンは実行していません。以下のコマンドを入力して、現行セッションで開始できます。

```
~]# service abrtd start
Starting abrt daemon:                                  [  OK  ]
```

同様に、ABRT で C/C++ のクラッシュをキャッチしたい場合は、同じ手順に従って abrt-ccpp サービスを確認し、起動することができます。ABRT がカーネル oops を検出するように設定するには、abrt-oops サービスに同じ手順を使用します。このサービスは、システムが失敗しなくなったり、すぐに再起動したりするカーネル oops を検出できないことに注意してください。ABRT でこのようなカーネル oops を検出できるようにするには、abrt-vmcore サービスをインストールする必要があります。この機能が必要な場合は、詳細は 「カーネルパニックを検出するための ABRT の設定」 を参照してください。

ABRT パッケージをインストールすると、各 ABRT サービスが ランレベル 3 と 5 に対して自動的に有効になります。chkconfig ユーティリティーを使用して必要なランレベルの ABRT サービスを無効または有効にできます。詳細は、「chkconfig ユーティリティーの使用」 を参照してください。

ABRT のインストールが CORE_PATTERNを上書きする

ABRT パッケージをインストールすると、コアダンプファイルの命名に使用するテンプレートを含む /proc/sys/kernel/core_pattern ファイルが上書きされることに注意してください。このファイルのコンテンツは、以下のように上書きされます。

```
|/usr/libexec/abrt-hook-ccpp %s %c %p %u %g %t e
```

最後に、グラフィカルデスクトップ環境で **ABRT** を実行すると、**ABRT** 通知アプレット が実行中であることを確認できます。

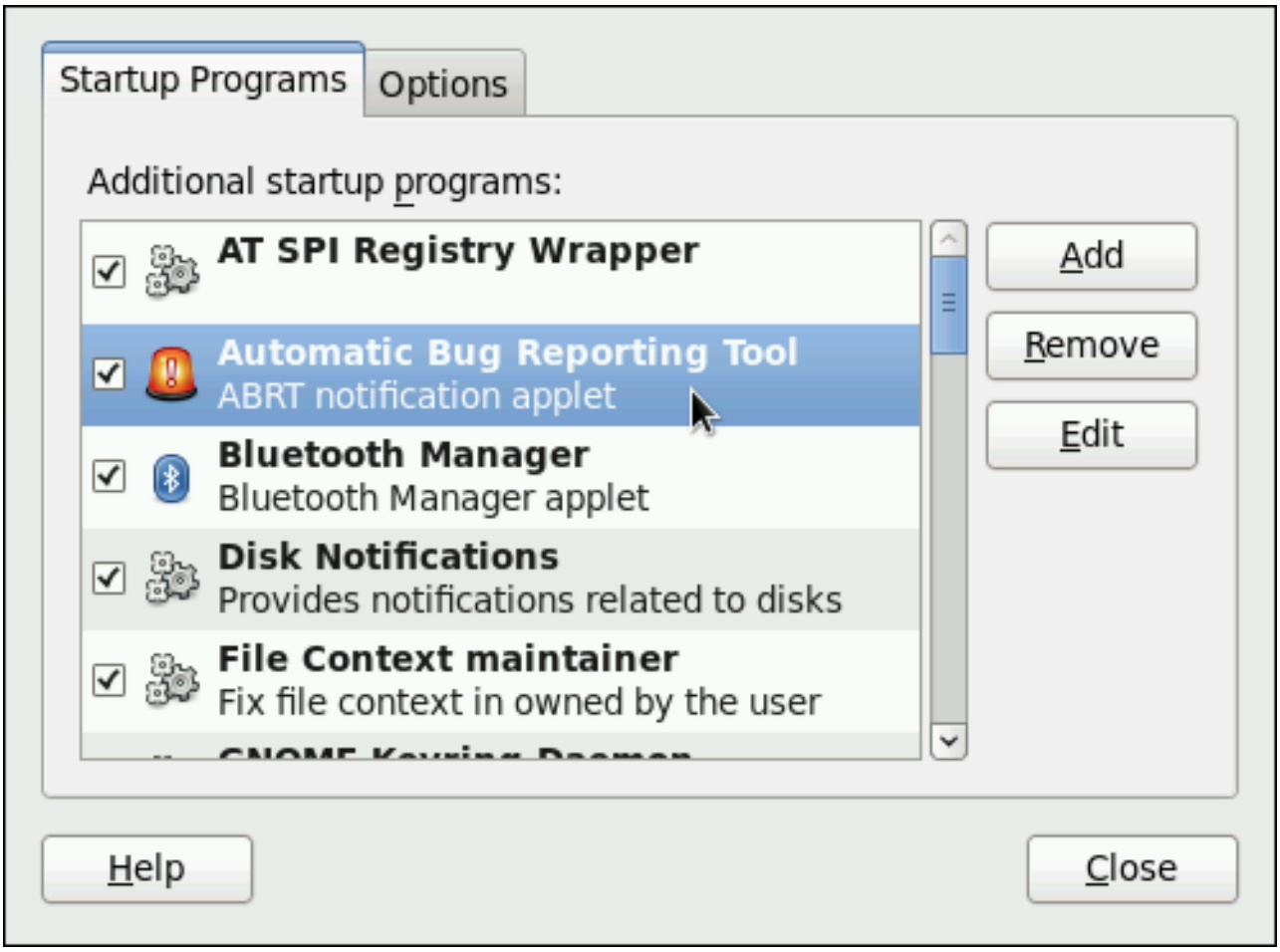
```
~]$ ps -el | grep abrt-applet
0 S   500  2036  1824  0  80   0 - 61604 poll_s ?       00:00:00 abrt-applet
```

ABRT 通知アプレットが実行されていない場合は、**abrt-applet** プログラムを実行すると、現在のデスクトップセッションで手動で起動できます。

```
~]$ abrt-applet &
[1] 2261
```

アプレットは、グラフィカルデスクトップセッションの開始時に自動的に開始するように設定できます。上部のパネルの **System** → **Preferences** → **Startup Applications** メニューを選択すると、**ABRT** 通知アプレットがプログラムの一覧に追加され、システム起動時に実行するように選択することができます。

図28.1 ABRT 通知アプレットが自動的に実行されるように設定。



[D]

28.2. グラフィカルユーザーインターフェースの使用

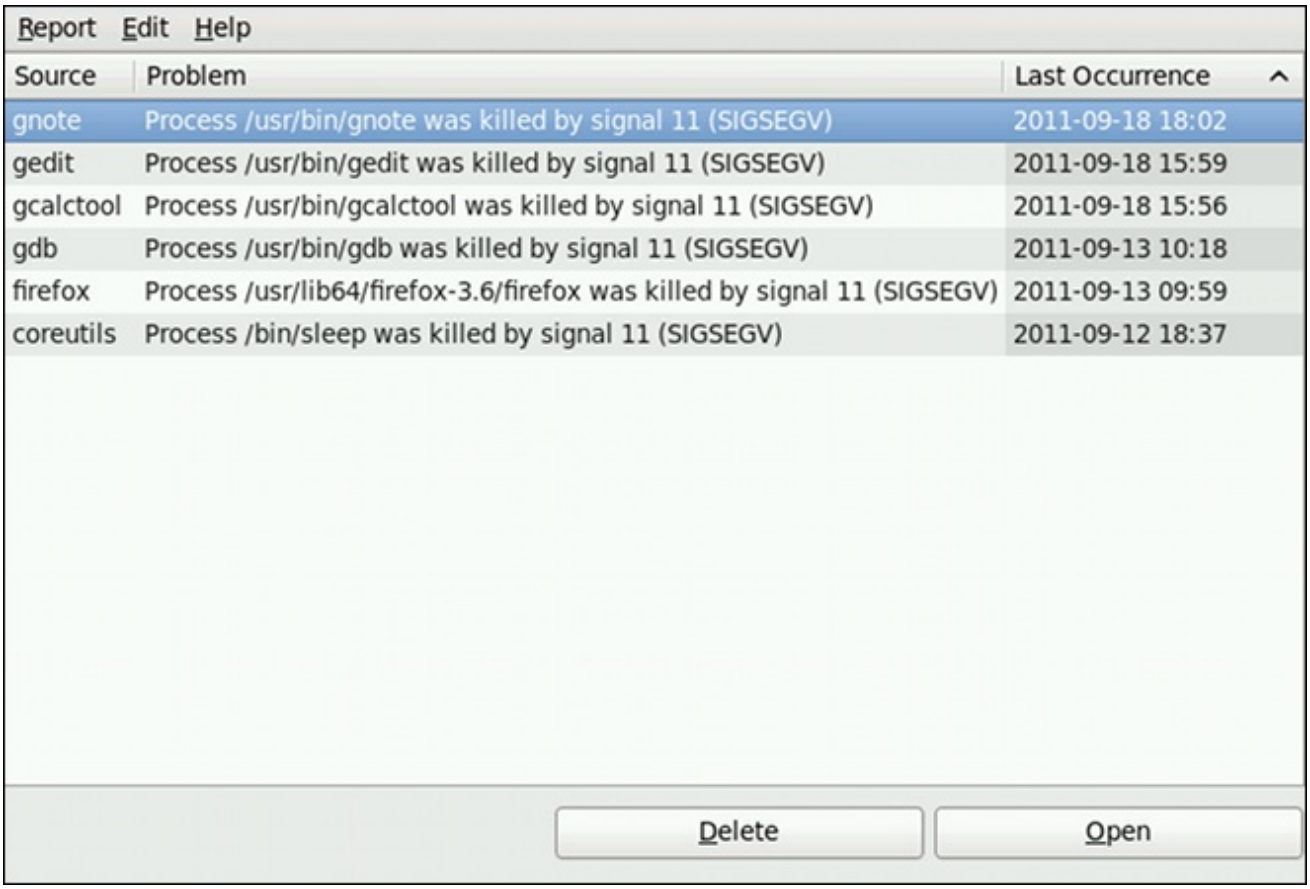
ABRT デーモンは、問題レポートの作成時に常にブロードキャスト D-Bus メッセージを送信します。ABRT 通知アプレットが実行されている場合、このメッセージを取得し、Notification Area にオレンジアラームアイコンを表示します。このアイコンを使用すると ABRT GUI アプリケーションを開くことができます。別の方法として、Application → System Tools → Automatic Bug Reporting Tool メニュー項目を選択して ABRT GUI を表示できます。

別の方法として、以下のようにコマンドラインから ABRT GUI を実行できます。

```
~]$ abrt-gui &
```

ABRT GUI は、報告された問題の表示、報告、および削除を簡単かつ直感的に行う方法を提供します。ABRT ウィンドウは、検出された問題の一覧を表示します。各問題エントリーは、障害が発生したアプリケーション名、アプリケーションがクラッシュした理由、その問題が最後に発生した日付で構成されます。

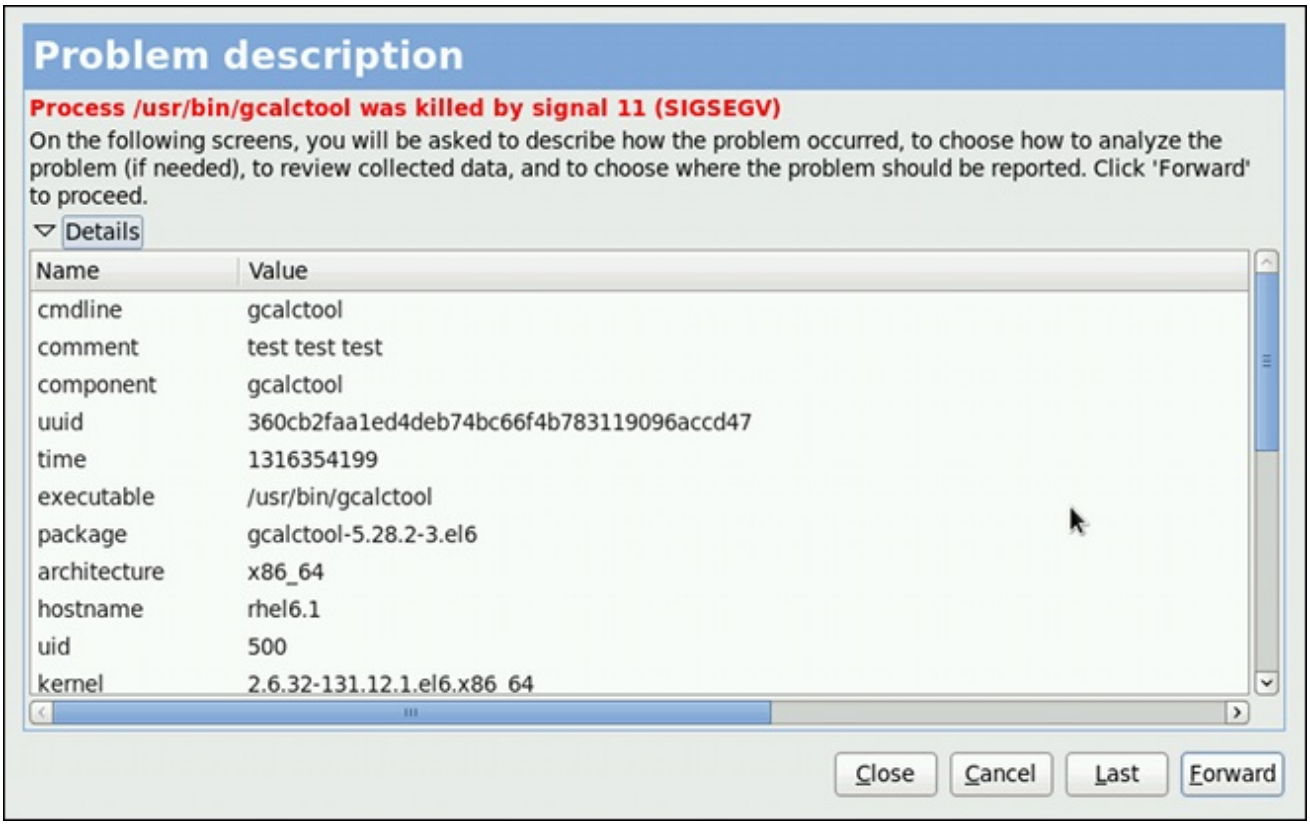
図28.2 ABRT GUI の実行例。



[D]

問題レポートの行をダブルクリックして、詳細な問題の説明にアクセスし、問題の分析方法、およびその報告先を決定するプロセスに進むことができます。

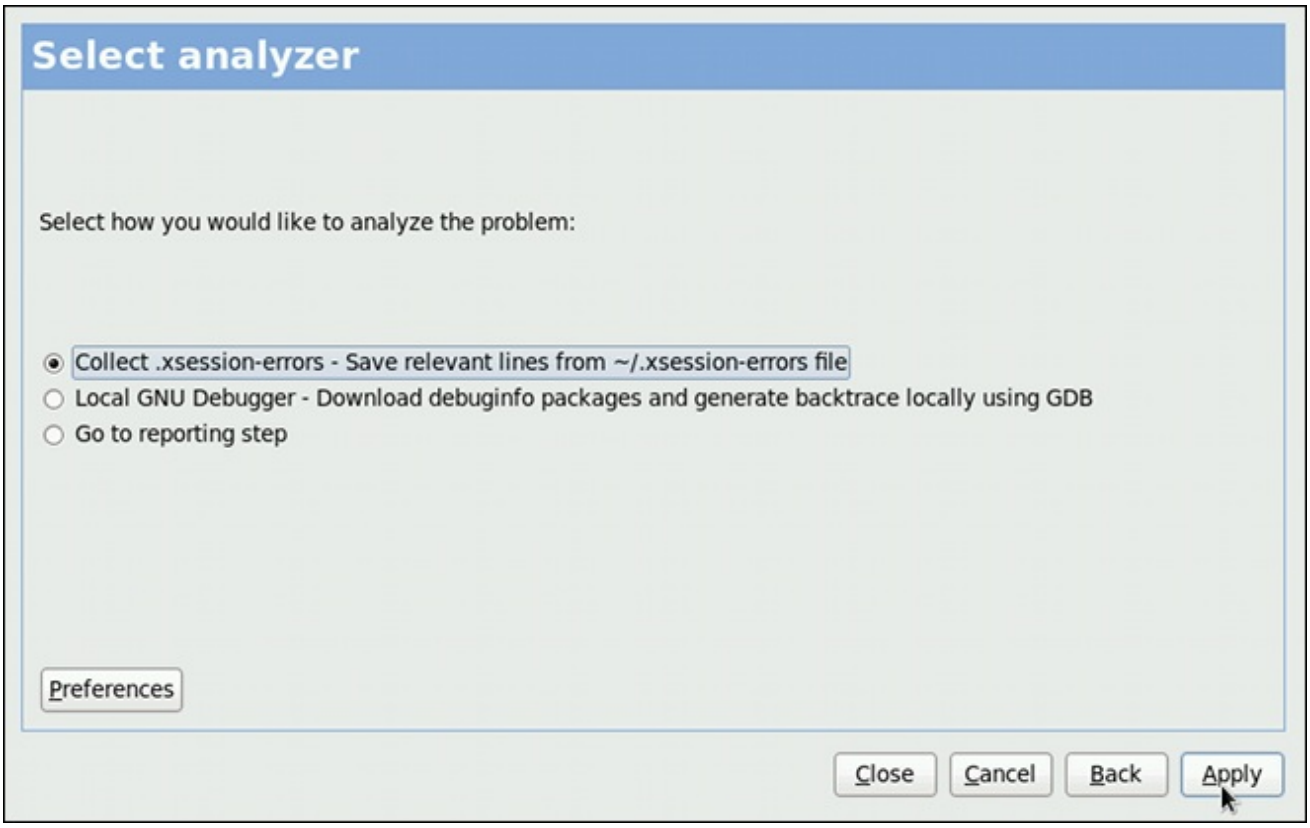
図28.3 問題データの詳細な例



[D]

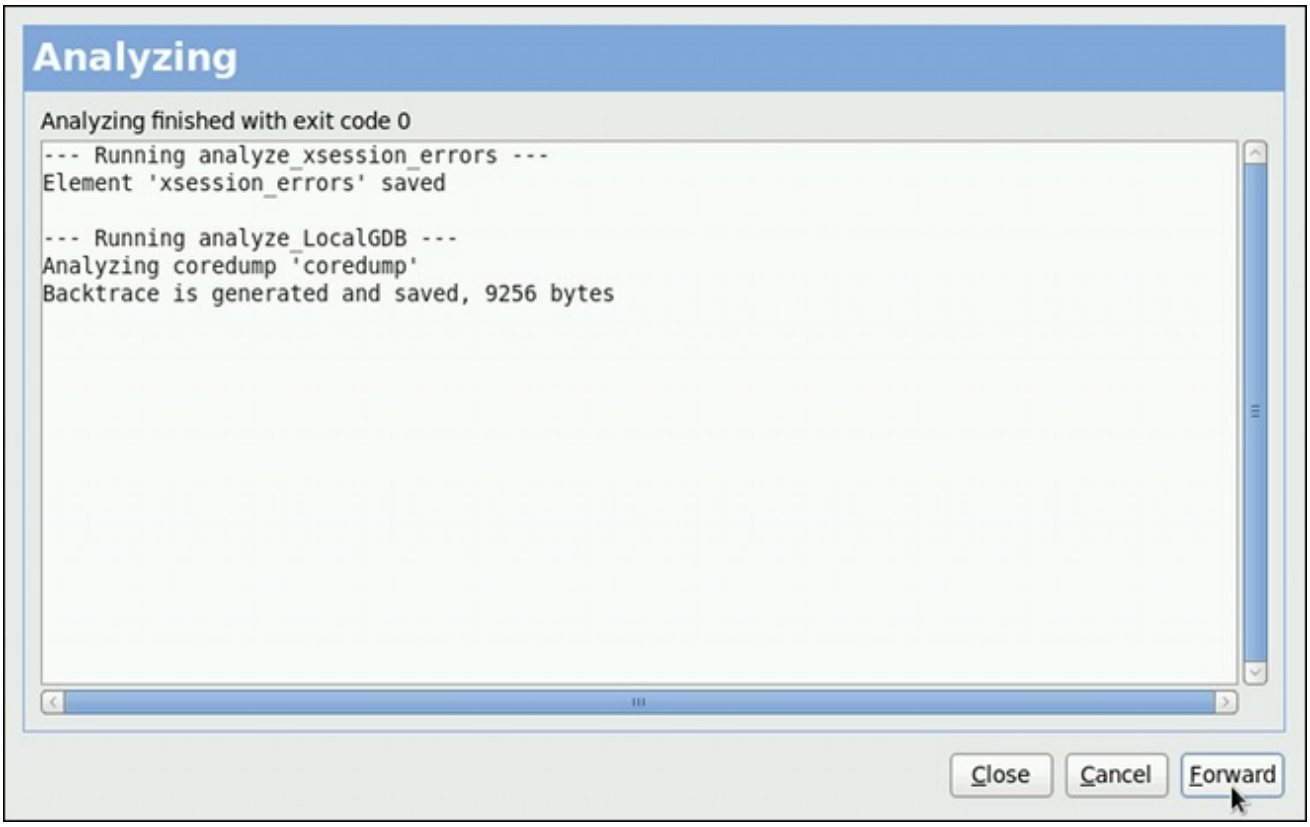
最初に発生した問題に関する追加情報を提供するよう求められます。問題の再現方法に関する詳細情報と、問題を再現するためにどの手順を実施すべきかを説明します。次の手順では、問題の分析方法を選択し、設定に応じてバックトレースを生成します。解析とバックトレース生成の手順は省略できますが、開発者はできるだけ多くの問題に関する情報が必要になることに注意してください。問題データを送信する前に、バックトレースを常に変更し、提供しない機密情報を削除できます。

図28.4 問題の分析方法の選択



[D]

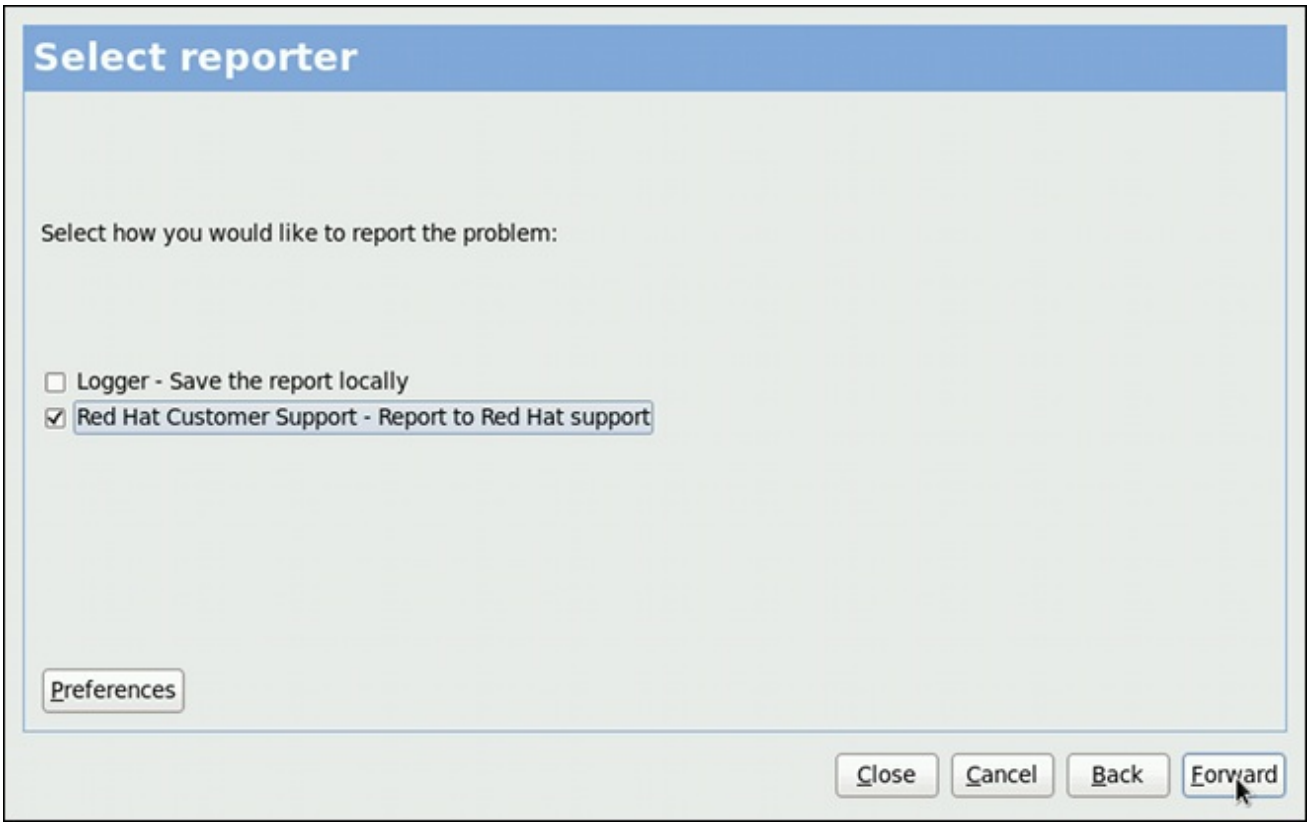
図28.5 ABRT が問題を分析



[D]

次に、問題を報告する方法を選択します。Red Hat Enterprise Linux を使用している場合は、Red Hat カスタマーサポートが推奨されます。

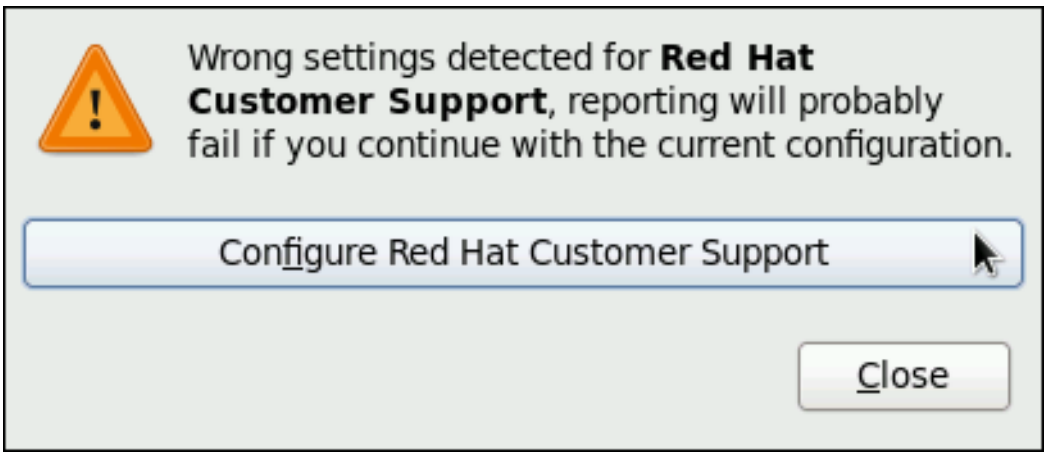
図28.6 問題レポーターの選択



[D]

Red Hat カスタマーサポートへの報告を選択し、このイベントを設定していないと、このイベントが適切に設定されていないこと、またこれを行うオプションが提供されることが警告されます。

図28.7 警告： Red Hat カスタマーサポートの設定がない

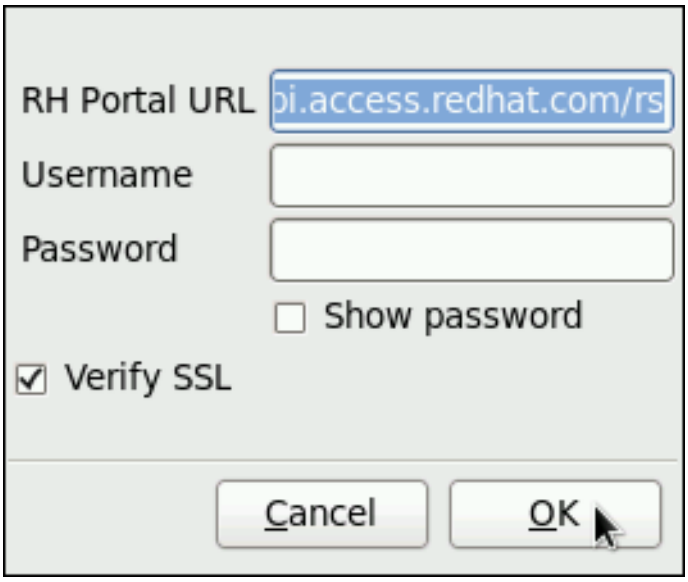


[D]

ここでは、Red Hat のログイン情報を提供する必要があります（取得方法とこのイベントの設定方法は 「ABRT GUI でのイベント設定」 を参照してください）。そうしないと、問題を報告できませ

ん。

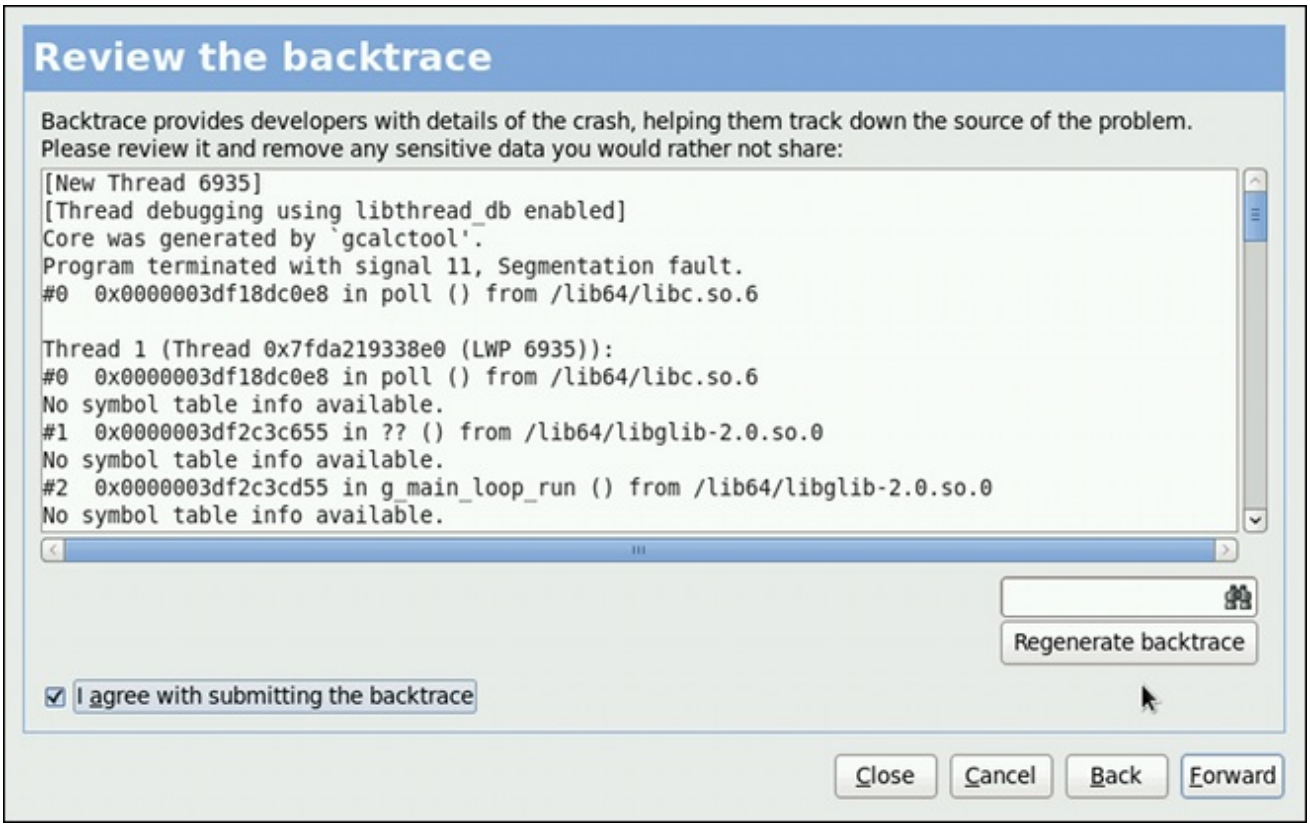
図28.8 Red Hat カスタマーサポート設定ウィンドウ



[D]

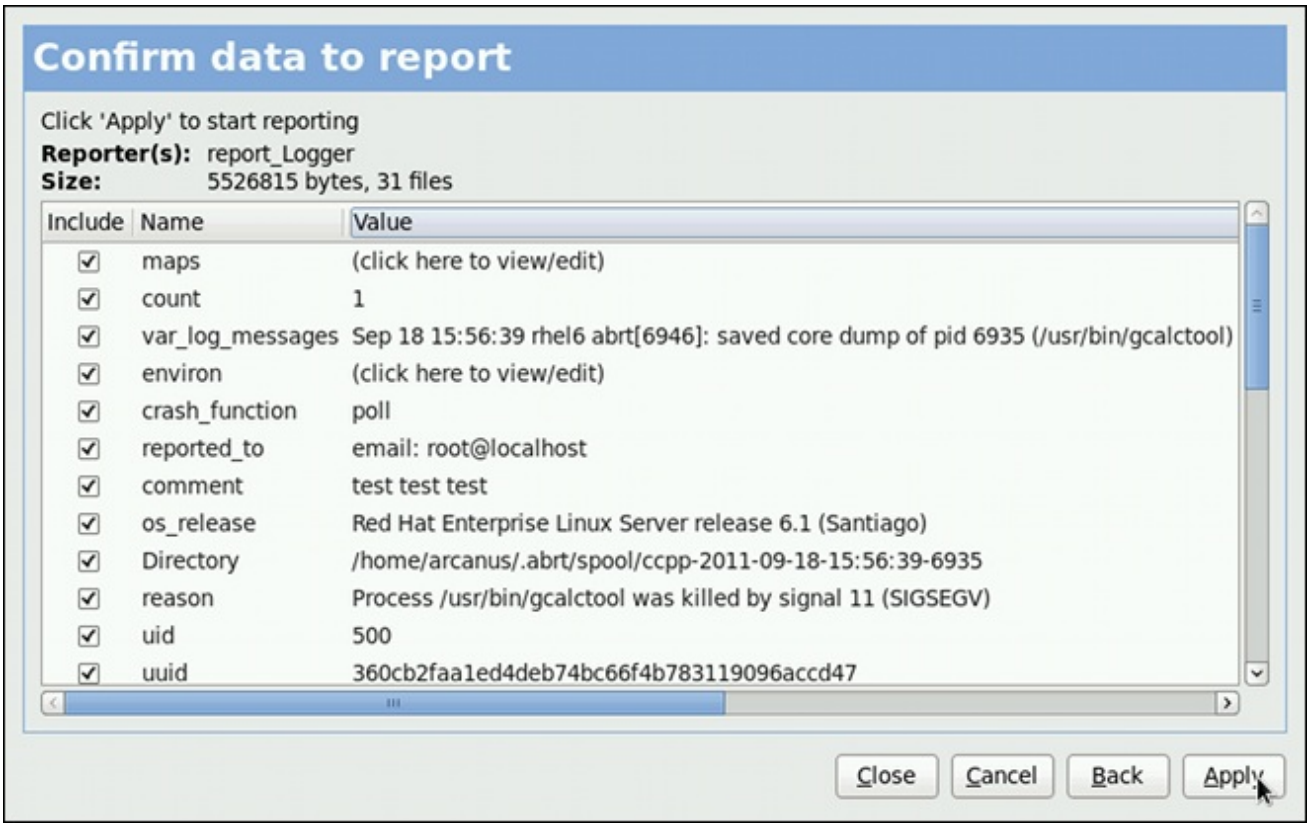
レポート方法を選択し、これが正しく設定されていることを確認したら、バックトレースを確認し、報告されたデータを確認します。

図28.9 問題バックトレースの確認



[D]

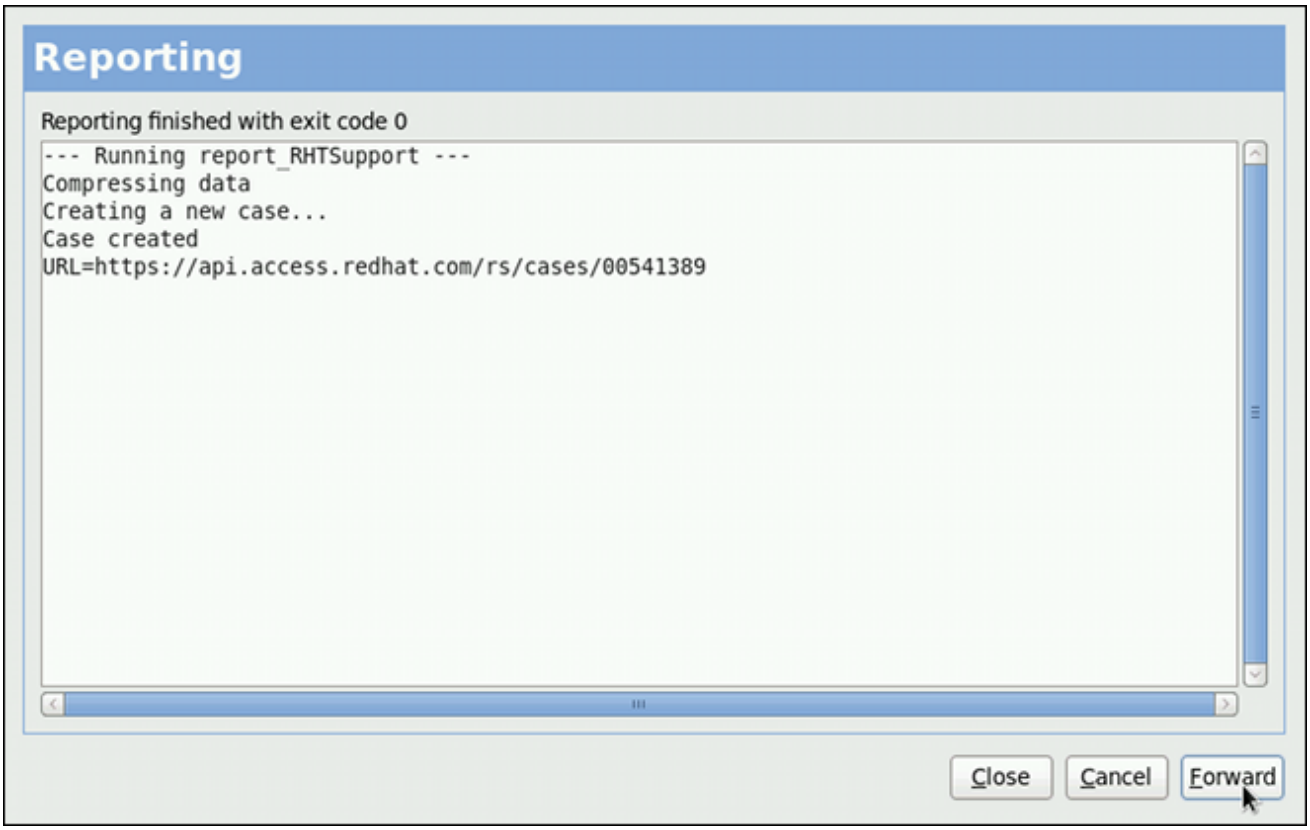
図28.10 レポートするデータの確認



[D]

最後に、問題データが選択された宛先に送信され、この問題の別の利用可能な方法を使用して問題を報告を継続するか、またはこの問題の作業を終了するかどうかを決定できるようになりました。問題を Red Hat カスタマーサポートデータベースに報告すると、問題のケースがデータベースに作成されます。これ以降は、報告プロセス時に指定したメールで問題の解決の進捗状況が通知されます。問題ケースは、問題のケース作成時に ABRT GUI によって提供される URL または Red Hat サポートから受信したメールを使用して確認することもできます。

図28.11 問題が Red Hat カスタマーサポートデータベースに報告されています。



[D]

28.3. コマンドラインインターフェースの使用

abrtd が保存した問題データは、コマンドラインインターフェースを使用して表示し、報告し、削除することができます。

abrt-cli ツールの一般的な使用方法は、以下の構文を使用して記述できます。

```
abrt-cli [--version] command [args]
```

ここでの args は、問題データディレクトリーや、コマンドの変更オプション/またはオプションを表し、コマンド は以下のサブコマンドのいずれかになります。

- *list: 問題を一覧表示し、問題データを表示します。*
- *report: 問題を分析して報告します。*

- ***rm -** 不要な問題を削除します。*

- ***info:** 特定の問題に関する情報を提供します。*

*特定の **abrt-cli** コマンドのヘルプを表示するには、以下を使用します。*

```
abrt-cli command --help
```

***abrt-cli** で使用される残りのコマンドは、以下のセクションで説明します。*

28.3.1. 問題の表示

*検出された問題を表示するには、**abrt-cli list** コマンドを入力します。*

```
~]# abrt-cli list
Directory:      /var/spool/abrt/ccpp-2011-09-13-10:18:14-2895
count:          2
executable:     /usr/bin/gdb
package:        gdb-7.2-48.el6
time:           Tue 13 Sep 2011 10:18:14 AM CEST
uid:            500

Directory:      /var/spool/abrt/ccpp-2011-09-21-18:18:07-2841
count:          1
executable:     /bin/bash
package:        bash-4.1.2-8.el6
time:           Wed 21 Sep 2011 06:18:07 PM CEST
uid:            500
```

- ***Directory -** 問題に関する情報をすべて含む問題データディレクトリーを表示します。*

- ***count -** この特定の問題が発生した回数を表示します。*

- ***executable -** クラッシュしたバイナリーまたは実行可能なスクリプトを示します。*

- ***package -** 問題の原因となったプログラムを含むパッケージ名を表示します。*

- *time - 問題の最後の日時を表示します。*

- *uid - クラッシュしたプログラムを実行したユーザーの ID を表示します。*

以下の表は、abrt-cli list コマンドで利用可能なオプションを示しています。すべてのオプションは相互に包含されるため、必要に応じて組み合わせることができます。すべてのオプションを組み合わせると、コマンド出力が最も包括的なものとなり、追加オプションを使用しない場合には、最も詳細が表示されます。

表28.2 abrt-cli list コマンドのオプション

オプション	説明
	追加のオプションがないと、**abrt-cli list** コマンドは、まだ報告されていない問題の基本情報のみを表示します。
-d、**--detailed**	既に生成されていれば**バックトレース**など、一覧表示された問題に関するすべての保存された情報を表示します。
-f,**--full**	すでに報告されたすべての問題の基本情報を表示します。
-v、**--verbose**	アクションに関する追加情報を提供します。

特定の問題についての情報のみを表示するには、以下のコマンドを使用できます。

abrt-cli info directory

ここでの ディレクトリー は、確認中の 問題の問題データディレクトリー を表します。以下の表は、abrt-cli info コマンドで利用可能なオプションを示しています。すべてのオプションは相互に包含されるため、必要に応じて組み合わせることができます。すべてのオプションを組み合わせると、コマンド出力が最も包括的なものとなり、追加オプションを使用しない場合には、最も詳細が表示されます。

表28.3 abrt-cli info コマンドのオプション

オプション	説明
	追加のオプションがないと、**abrt-cli info** コマンドは、問題**データディレクトリー**の引数で指定された問題の基本情報のみを表示します。

オプション	説明
-d、--detailed	問題 **データディレクトリー** の引数で指定された問題の保存された情報をすべて表示します。これには、すでに生成された場合は **backtrace** が含まれます。
-v、--verbose	**abrt-cli info** は、そのアクションに関する追加情報を提供します。

28.3.2. 問題の報告

特定の問題を報告するには、以下のコマンドを使用します。

```
abrt-cli report directory
```

... ここでの directory は、報告されている問題の問題 データディレクトリー を表します。以下に例を示します。

```
~]$ abrt-cli report /var/spool/abrt/ccpp-2011-09-13-10:18:14-2895
How you would like to analyze the problem?
1) Collect .xsession-errors
2) Local GNU Debugger
Select analyzer: _
```

ABRT は、報告されている問題のアナライザーイベントを選択するように求められます。イベントを選択すると、問題は分析されます。これにはかなり時間がかかる可能性があります。問題レポートの準備ができたら、abrt-cli がレポートの内容を含むテキストエディターを開きます。これでレポート内容を確認でき、クラッシュを再現させるための指示やコメントを記入できます。また、バックトレースも確認するようにしてください。バックトレースは公開サーバーに送信され、問題レポーターイベントの設定によっては誰でも確認できるためです。

選択するテキストエディターの選択

レポートの確認に使用するテキストエディターを選択できます。abrt-cli は、ABRT_EDITOR 環境変数で定義されたエディターを使用します。変数が定義されていない場合は、VISUAL および EDITOR 変数を確認します。いずれの変数も設定されていない場合は、vi が使用されます。.bashrc 設定ファイルで優先エディターを設定することもできます。たとえば、GNU Emacs を使用する場合は、そのファイルに以下の行を追加します。

```
export VISUAL=emacs
```

レポートでの作業が終了したら、変更を保存してエディターを終了します。レポートの送信に使用

する設定済みの ABRT レポーターイベントが尋ねられます。

```
How would you like to report the problem?
1) Logger
2) Red Hat Customer Support
Select reporter(s): _
```

レポート方法の選択後に、レポートに送信されるデータの確認に進むことができます。以下の表は、abrt-cli report コマンドで利用可能なオプションを示しています。

表28.4 abrt-cli report コマンドのオプション

オプション	説明
	追加オプションがないと、**abrt-cli report** コマンドは通常の出力を提供します。
-v、**--verbose**	**abrt-cli レポート** は、そのアクションに関する追加情報を提供します。

28.3.3. 問題の削除

特定の問題を報告したくないことが分かっている場合は、その問題を削除できます。ABRT に関する情報を保持しないように問題を削除するには、以下のコマンドを使用します。

```
abrt-cli rm directory
```

... ここでの directory は、削除される問題の問題データディレクトリーを表します。以下に例を示します。

```
~]$ abrt-cli rm /var/spool/abrt/ccpp-2011-09-12-18:37:24-4413
rm '/var/spool/abrt/ccpp-2011-09-12-18:37:24-4413'
```

問題を削除すると、ABRT 通知が頻繁に発生することがあります。

ABRT は、ローカルに保存されたすべての問題と新しい問題を比較することで、重複の問題を検出することに注意してください。繰り返し発生しているクラッシュの場合、ABRT は一度だけ動作する必要があります。ただし、この問題のクラッシュダンプを削除すると、次にこの問題が発生すると、ABRT はこれを新しいクラッシュとして扱います。ABRT はユーザーに警告を出し、説明を入力し、報告するように促します。ABRT が繰り返し発生する問題について通知しないようにするには、その問題データを削除しないでください。

以下の表は、abrt-cli rm コマンドで利用可能なオプションを示しています。

表28.5 abrt-cli rm コマンドのオプション

オプション	説明
	オプションを指定しないと、**abrt-cli rm** コマンドは、指定した**問題データディレクトリー** を、そのすべてのコンテンツとともに削除します。
-v、--verbose	**abrt-cli rm** は、そのアクションに関する追加情報を提供します。

28.4. ABRT の設定

ABRT では、問題 のライフサイクルは イベント によって実行されます。以下に例を示します。

- イベント 1 - 問題データのディレクトリーが作成されます。
- イベント 2: 問題データの分析
- イベント 3 - 問題の Bugzilla 報告

問題が検出され、その定義データが保存されると、問題は問題のデータディレクトリーでイベントを実行して処理されます。イベントの詳細と、そのイベントの定義方法は、「ABRT イベント」 を参照してください。標準 ABRT インストールは、現在、問題報告プロセス時に選択して使用できるデフォルトのイベントを複数サポートします。これらのイベントの一覧を表示するには、「標準 ABRT インストールでサポートされているイベント」 を参照してください。

インストール時に、ABRT および libreport がそれぞれの設定ファイルをシステム上の複数のディレクトリーに配置します。

- /etc/libreport/: report_event.conf メイン設定ファイルが含まれます。この設定ファイルの詳細は、「ABRT イベント」 を参照してください。
- /etc/libreport/events/: 事前定義イベントのデフォルト設定を指定するファイルを保持します。

- /etc/libreport/events.d/: イベントを定義する設定ファイルを保持します。
- /etc/libreport/plugins/: イベントの一部を取るプログラムの設定ファイルが含まれます。
- /etc/abrt/ - ABRT のサービスおよびプログラムの挙動を変更するために使用される ABRT 固有の設定ファイルを保持します。特定の設定ファイルの詳細は、「ABRT 固有の設定」 を参照してください。
- /etc/abrt/plugins/: ABRT のサービスおよびプログラムのデフォルト設定を上書きするために使用される設定ファイルを保持します。一部の設定ファイルの詳細は、「ABRT 固有の設定」 を参照してください。

28.4.1. ABRT イベント

各イベントは、それぞれの設定ファイル内の単一のルール構造により定義されます。設定ファイルは通常 /etc/libreport/events.d/ ディレクトリーに保存されます。これらの設定ファイルは、メインの設定ファイル /etc/libreport/report_event.conf によって使用されます。

/etc/libreport/report_event.conf ファイルは、include ディレクティブ および ルール で構成されます。ルールは通常、/etc/libreport/events.d/ ディレクトリーの他の設定ファイルに保存されます。標準インストールでは、/etc/libreport/report_event.conf ファイルには 1 つの include ディレクティブのみが含まれます。

```
include events.d/*.conf
```

このファイルを変更する場合は、シェルのメタ文字（*、$、？など）を考慮し、その場所に対して相対パスを解釈することに注意してください。

各 ルール は空白でない先頭文字の行で始まり、その後のすべての行は スペース 文字または タブ 文字で始まるすべての後続の行は、このルールの一部と見なされます。各ルールは、条件部分とプログラム部分の2つの部分で構成されています。条件パートには、以下のいずれかの形式で条件が記載されます。

- VAR=VAL,
- VAR!=VAL または

- *VAL~= REGEX*

どこかに以下が行われます。

- *VAR は、EVENT のキーワードまたは問題データディレクトリー要素の名前です（実行可能な、package、hostname など）。*
- *VAL は、イベントまたは問題データ要素の名前です。*
- *REGEX は正規表現です。*

プログラムパートは、プログラム名とシェルが解釈可能なコードで構成されます。条件パートにある全条件が有効な場合、プログラムパートがシェルで実行されます。以下は イベント の一例です。

```
EVENT=post-create date > /tmp/dt
    echo $HOSTNAME `uname -r`
```

このイベントは、現在の日付と時刻で /tmp/dt ファイルの内容を上書きし、マシンのホスト名とカーネルバージョンを標準出力に表示します。

以下は、実際に事前定義されたイベントの 1 つであるより複雑なイベントの例です。abrt-ccpp サービスを使用してその問題を処理し、クラッシュ発生時に X11 ライブラリーが読み込まれている問題について ~/.xsession-errors ファイルから関連する行を問題レポートに保存します。

```
EVENT=analyze_xsession_errors analyzer=CCpp dso_list~=.*/libX11.*
    test -f ~/.xsession-errors || { echo "No ~/.xsession-errors"; exit 1; }
    test -r ~/.xsession-errors || { echo "Can't read ~/.xsession-errors"; exit 1; }
    executable=`cat executable` &&
    base_executable=${executable##*/} &&
    grep -F -e "$base_executable" ~/.xsession-errors | tail -999 >xsession_errors &&
    echo "Element 'xsession_errors' saved"
```

可能なイベントのセットは、ハードセットではありません。システム管理者は、必要に応じてイベントを追加できます。現在、ABRT および libreport の標準インストールでは、以下のイベント名が提供されます。

post-create

このイベントは、新たに作成される問題データディレクトリーで abrtd により実行されます。post-create イベントが実行すると、abrtd は新規の問題データの UUID 識別子が既存の問題ディレクトリーの UUID と一致するかどうかを確認します。このような問題ディレクトリーが存在する場合は、新しい問題データが削除されます。

analyze_name_suffix

ここでの name_suffix は、イベント名の調整可能な部分です。このイベントは、収集データの処理に使用されます。たとえば、analyze_LocalGDB は、アプリケーションのコアダンプで GNU Debugger(GDB)ユーティリティーを実行し、プログラムのバックトレースを生成します。分析イベントの一覧を表示し、abrt-gui を使用してこれを選択できます。

collect_name_suffix

ここでの name_suffix は、イベント名の調整可能な部分です。このイベントは、問題に関する追加情報を収集するために使用されます。収集イベントの一覧を表示し、abrt-gui を使用してこれを選択できます。

report_name_suffix

ここでの name_suffix は、イベント名の調整可能な部分です。このイベントは、問題を報告するために使用されます。レポートイベントの一覧を表示し、abrt-gui を使用してこれを選択できます。

イベントに関する追加情報（環境変数としてイベントに渡すことができるパラメーターの説明、名前、タイプなど）は、/etc/libreport/events/event_name.xml ファイルに保存されます。このファイルは、ユーザーインターフェースをより容易にするために abrt-gui および abrt-cli で使用されます。標準インストールを変更する場合以外に、このファイルは編集しないでください。

28.4.2. 標準 ABRT インストールでサポートされているイベント

現在、標準 ABRT インストールには、多くのデフォルトの分析、収集、およびレポートイベントがあります。これらのイベントの一部は、ABRT GUI アプリケーションを使用して設定可能です（ ABRT GUI を使用したイベント設定の詳細は、「ABRT GUI でのイベント設定」を参照してください）。ABRT GUI は、完全なイベント名ではなく、ユーザーが読みやすい名前の一意部分のみを表示します。たとえば、analyze_xsession_errors イベントは、ABRT GUI で Collect .xsession-errors として表示されます。以下は、ABRT の標準インストールで提供される、デフォルトの分析、収集、レポートイベントの一覧です。

analyze_VMcore - 仮想マシンコアの分析

アプリケーションの問題データに対して GDB (GNU Debugger)を実行し、カーネルの バックトレース を生成します。/etc/libreport/events.d/vmcore_event.conf 設定ファイルで定義されま

す。

analyze_LocalGDB: Local GNU Debugger

アプリケーションの問題データに対して GDB (GNU Debugger)を実行し、プログラムの バックトレース を生成します。これは /etc/libreport/events.d/ccpp_event.conf 設定ファイルで定義されます。

analyze_xsession_errors - Collect .xsession-errors

~/.xsession-errors ファイルから関連する行を問題レポートに保存します。これは /etc/libreport/events.d/ccpp_event.conf 設定ファイルで定義されます。

report_Logger - ロガー

問題レポートを作成して、指定のローカルファイルに保存します。これは /etc/libreport/events.d/print_event.conf 設定ファイルで定義されます。

report_RHTSupport - Red Hat カスタマーサポート

Red Hat テクニカルサポートシステムに問題を報告します。これは、Red Hat Enterprise Linux のユーザーを対象としています。これは、/etc/libreport/events.d/gitopssupport_event.conf 設定ファイルで定義されます。

report_Mailx — Mailx

問題レポートを Mailx ユーティリティーを介して指定のメールアドレスに送信します。これは /etc/libreport/events.d/mailx_event.conf 設定ファイルで定義されます。

report_Kerneloops — Kerneloops.org

カーネル問題を oops トラッカーに送信します。これは /etc/libreport/events.d/koops_event.conf 設定ファイルで定義されます。

report_Uploader - Report uploader

FTP または SCP プロトコルを使用して、問題データの tarball(.tar.gz)アーカイブを、選択した宛先にアップロードします。これは /etc/libreport/events.d/uploader_event.conf 設定ファイルで定義されます。

28.4.3. ABRT GUI でのイベント設定

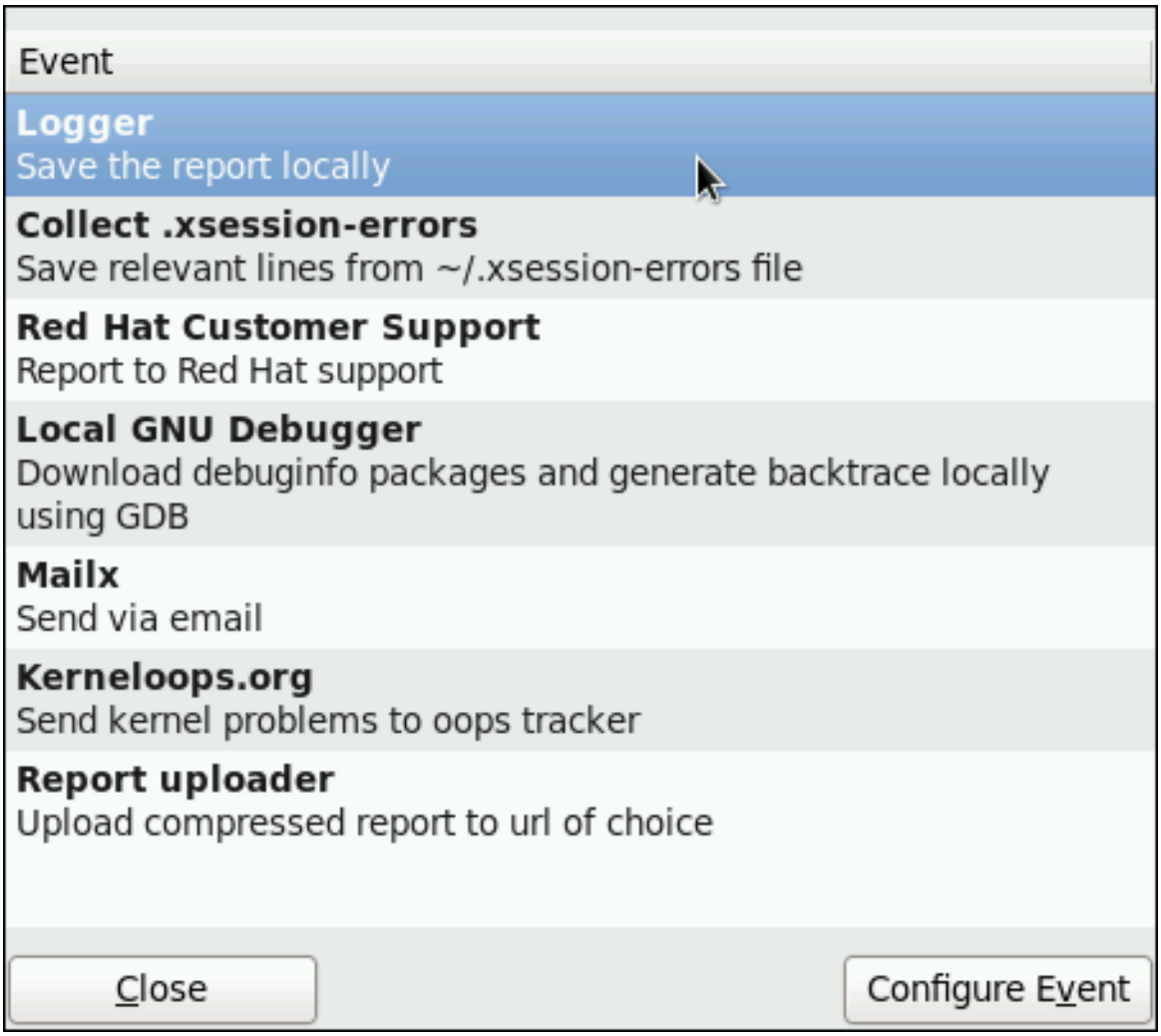
イベントは、環境変数として渡されたパラメーターを使用できます（例： report_Logger イベントは、出力ファイル名をパラメーターとして受け入れます）。それぞれの /etc/libreport/events/event_name.xml ファイルを使用すると、ABRT GUI は、選択されたイベントに対してどのパラメーターを指定できるかを判断し、ユーザーはこれらのパラメーターの値を設定できるようになります。これらの値は ABRT GUI によって保存され、後続のイベントの呼び出しで再利用されます。

Edit → Preferences をクリックして Event Configuration ウィンドウを開きます。このウィンドウでは、レポートプロセス中に選択できる利用可能なイベントの一覧が表示されます。設定可能なイベントの 1 つを選択すると、Configure Event ボタンをクリックすると、そのイベントの設定が可能になります。イベントのパラメーターを変更すると、それらが Gnome キーリングに保存され、今後の GUI セッションで使用されます。

機密データをグローバル設定ファイルに保存しないでください。

/etc/libreport/ ディレクトリー階層内のすべてのファイルは誰でも読み取り可能で、グローバル設定として使用することが意図されています。そのため、ユーザー名、パスワード、その他の機密データをそれらのデータに保存することは推奨されません。ユーザー別の設定（GUI アプリケーションで設定され、$HOME の所有者のみが読み取り可能）は Gnome キーリングに保存されるか、abrt-cli で使用するために $HOME/.abrt/*.conf のテキストファイルに保存できます。

図28.12 イベント設定ウィンドウ



[D]

以下は、ABRT GUI アプリケーションで設定可能な各イベントで利用可能な設定オプションの一覧です。

ロガー

Logger イベント設定ウィンドウで、以下のパラメーターを設定できます。

- *ログファイル: クラッシュレポートを保存するファイルを指定します（デフォルトでは /var/log/abrt.logに設定されます）。*

Append オプションにチェックが付けられると、Logger イベントは Logger ファイルオプションで指定されたログファイルに新しいクラッシュレポートを追加します。これの選択を解除すると、新しいクラッシュレポートは常に以前のクラッシュレポートに置き換わります。

Red Hat カスタマーサポート

Red Hat カスタマーサポート イベント設定ウィンドウで、以下のパラメーターを設定できます。

- RH Portal URL: クラッシュダンプの送信先となる Red Hat カスタマーサポート URL を指定します（デフォルトでは に https://api.access.redhat.com/rs設定されます）。
- Username: Red Hat カスタマーサポートにログインし、報告されたクラッシュについての Red Hat カスタマーサポートデータベースエントリーを作成するために使用されるユーザーログイン。https://www.redhat.com/en、Red Hat カスタマーポータル (https://access.redhat.com/home)、または Red Hat Network(https://rhn.redhat.com/)でアカウントを作成して、Red Hat ログイン 先を使用します。
- Password - Red Hat カスタマーサポートへのログインに使用されるパスワード（Red Hat ログイン に関連するパスワード）

SSL 検証 オプションがチェックされたら、ネットワーク経由でデータを送信する際に SSL プロトコルが使用されます。

MailX

MailX イベント設定ウィンドウで、以下のパラメーターを設定できます。

- subject - Mailx によって送信されるメールの Subject フィールドに表示される文字列（デフォルトでは「[abrt] detected a crash」に設定）。
- sender: 問題レポートメールの From フィールドに表示される文字列。
- Recipient: 問題レポートメールの受信者の電子メールアドレス。

Send Binary Data オプションを確認すると、問題レポートのメールに、その問題に関連するバイナリーファイルがすべて添付されます。コアダンプファイルは、添付ファイルとしても送信されます。

Kerneloops.org

Kerneloops.org イベント設定ウィンドウで、以下のパラメーターを設定できます。

- *KernelOops URL: カーネル の問題が報告される URL を指定します（デフォルトでは に http://submit.kerneloops.org/submitoops.php設定されます）。*

Report Uploader

Report Uploader イベント設定で、以下のパラメーターを設定できます。

- *url: 圧縮された問題データを含む tarball が FTP または SCP プロトコルを使用してアップロードされる URL を指定します（デフォルトでは ftp://localhost:/tmp/uploadに設定されます）。*

28.4.4. ABRT 固有の設定

現在、標準 ABRT インストールは以下の ABRT 固有の設定ファイルを提供します。

- */etc/abrt/abrt.conf: abrtd サービスの動作を変更できます。*
- */etc/abrt/abrt-action-save-package-data.conf: abrt-action-save-package-data プログラムの動作を変更できます。*
- */etc/abrt/plugins/CCpp.conf - ABRT の core catching フックの動作を変更できます。*

以下の設定ディレクティブは、/etc/abrt/abrt.conf ファイルでサポートされています。

WatchCrashdumpArchiveDir = /var/spool/abrt-upload

このディレクティブは、デフォルトでコメントアウトされています。abrtd で、指定したディレクトリーにある tarball アーカイブ(.tar.gz)を自動アンパッククラッシュダンプする場合は、これを有効にします。上記の例では、これは /var/spool/abrt-upload/ ディレクトリーです。このディレクティブに指定するすべてのディレクトリーは、それが存在し、abrtd に対して書き込み可能であることを確認する必要があります。ABRT デーモンはこれを自動的に作成しません。このオプションのデフォルト値を変更する場合は、ABRT の適切な機能を確保するために、このディレクトリーを DumpLocation オプションに指定されたディレクトリーと同じにし ない でください。

このオプションは SELINUX で変更しないでください。

クラッシュダンプアーカイブの場所を変更すると、SELinux ルールの変更を最初に反映しない限り、SELinux 拒否が生じます。SELinux で ABRT の実行に関する詳細は、abrt_selinux(8) man ページを参照してください。

SELinux を使用する際にこのオプションを有効にする場合は、ABRT が public_content_rw_t ドメインに書き込むように適切なブール値を設定するには、以下のコマンドを実行する必要があります。

```
setsebool -P abrt_anon_write 1
```

MaxCrashReportsSize = size_in_megabytes

このオプションは、ABRT が全ユーザーからの問題情報を保存するために使用するストレージ容量をメガバイトで設定します。デフォルト設定は 1000 MB です。ここで指定したクォータが一致したら、ABRT は継続して問題を取得し、新しいクラッシュダンプの容量を確保するために、一番古いものと最大のものを削除します。

DumpLocation = /var/spool/abrt

このディレクティブは、デフォルトでコメントアウトされています。問題データディレクトリーが作成される場所と、問題のコアダンプと、その他の問題データが格納される場所を指定します。デフォルトの場所は /var/spool/abrt ディレクトリーに設定されます。このディレクティブに指定するすべてのディレクトリーは、それが存在し、abrtd に対して書き込み可能であることを確認する必要があります。このオプションのデフォルト値を変更する場合は、ABRT の適切な機能を確保するために、このディレクトリーを WatchCrashdumpArchiveDir オプションに指定したディレクトリーと同じにし ない でください。

このオプションは SELINUX で変更しないでください。

各 SELinux ルールに変更を反映しない限り、ダンプの場所を変更すると SELinux 拒否が生じます。SELinux で ABRT の実行に関する詳細は、abrt_selinux(8) man ページを参照してください。

SELinux を使用する際にこのオプションを有効にする場合は、ABRT が public_content_rw_t ドメインに書き込むように適切なブール値を設定するには、以下のコマンドを実行する必要があります。

```
setsebool -P abrt_anon_write 1
```

以下の設定ディレクティブは、/etc/abrt/abrt-action-save-package-data.conf ファイルでサポートされています。

OpenGPGCheck = yes/no

OpenGPGCheck ディレクティブを yes（デフォルト設定）に設定すると、ABRT に対して、場所が /etc/abrt/gpg_keys ファイルにリストされている GPG キーにより署名されたパッケージで提供されるアプリケーションでのみ クラッシュを分析および処理するように指示します。OpenGPGCheck を no に設定すると、ABRT がすべてのプログラムでクラッシュを検出するように指示します。

blacklist = nspluginwrapper, valgrind, strace, [more_packages]

BlackList ディレクティブの後に一覧表示されるパッケージおよびバイナリーのクラッシュは ABRT によって処理されません。ABRT が他のパッケージやバイナリーを無視する場合は、ここにコンマで区切られたリストを指定します。

ProcessUnpackaged = yes/no

このディレクティブは、ABRT が、パッケージに属していない実行可能ファイルでクラッシュを処理するかどうかを示します。デフォルト設定は no *です。*

BlackListedPaths = /usr/share/doc/, */example**

ABRT では、これらのパスの実行可能ファイル内のクラッシュは無視されます。

以下の設定ディレクティブは、/etc/abrt/plugins/CCpp.conf ファイルでサポートされています。

MakeCompatCore = yes/no

このディレクティブは、ABRT のコアキャッチフックがインストールされていない場合に実行する可能性があるので、ABRT のコアキャッチフックがコアファイルを作成するかどうかを指定します。コアファイルは、通常、クラッシュしたプログラムの現行ディレクトリーに作成されますが、ulimit -c 設定がそれを許可する場合のみです。ディレクティブはデフォルトで yes に設定されます。

SaveBinaryImage = yes/no

このディレクティブは、ABRT のコアキャッチフックがバイナリーイメージをコアダンプに保存するかどうかを指定します。これは、削除されたバイナリーで発生したクラッシュのデバッグに役立ちます。デフォルト設定は no です。

28.4.5. カーネルパニックを検出するための ABRT の設定

Red Hat Enterprise Linux 6.3 では、ABRT は abrt-addon-vmcore パッケージが提供する abrt-vmcore サービスを使用してカーネルパニックを検出できます。このサービスはシステムの起動時に自動的に起動し、/var/crash/ ディレクトリーでコアダンプファイルを検索します。コアダンプファイルが見つかると、abrt-vmcore が /var/spool/abrt/ ディレクトリーに問題データディレクトリー を作成し、コアダンプファイルを新たに作成された問題データディレクトリーに移動します。/var/crash/ ディレクトリーを検索すると、このサービスは、次回のシステム起動まで停止します。

ABRT がカーネルパニックを検出するように設定するには、以下の手順を実行します。

1. システムで kdump サービスが有効になっている。特に、kdump カーネル用に予約されるメモリー容量が正しく設定されている必要があります。これは、system-config-kdump グラフィカルツールを使用するか、/etc/grub.conf 設定ファイルのカーネルオプションの一覧に crashkernel パラメーターを指定して設定できます。kdump の有効化と設定の詳細は、32章kdump クラッシュリカバリーサービス を参照してください。

2. yum パッケージインストーラーを使用して abrt-addon-vmcore パッケージ を インストールします。

```
~]# yum install abrt-addon-vmcore
```

これにより、各サポートファイルと設定ファイルと共に abrt-vmcore サービスがインストールされます。abrt-addon-vmcore パッケージは Optional サブスクリプションチャンネル

で提供されることに注意してください。Red Hat 追加チャンネルの詳細は、「Optional および Supplementary リポジトリーの追加」 を参照してください。

3. システムを再起動して、変更を有効にします。

ABRT が設定されていない限り、検出されたカーネルパニックの問題データは /var/spool/abrt/ ディレクトリーに保存され、他の検出されたカーネル oops として ABRT でさらに処理できるようになりました。

28.4.6. Debuginfo パッケージの自動ダウンロードとインストール

ABRT は、特定の問題のデバッグに必要なパッケージを自動的にダウンロードしてインストールするように設定できます。この機能は、会社の環境でローカルで問題をデバッグする場合に役立ちます。debuginfo の自動ダウンロードとインストールを有効にするには、システムが以下の条件を満たすようにします。

- /etc/libreport/events.d/ccpp_event.conf ファイルには、デフォルト設定でコメント解除されている以下のアナライザーイベントが含まれています。

```
EVENT=analyze_LocalGDB analyzer=CCpp
        abrt-action-analyze-core --core=coredump -o build_ids &&
        # In RHEL we don't want to install anything by default
        # and also this would fail, as the debuginfo repositories.
        # are not available without root password rhbz#759443
        # /usr/libexec/abrt-action-install-debuginfo-to-abrt-cache --size_mb=4096 &&
        abrt-action-generate-backtrace &&
        abrt-action-analyze-backtrace
```

- /etc/libreport/events.d/ccpp_event.conf ファイルには以下の行が含まれます。この行は、不要なコンテンツのインストールを回避するために、デフォルトでコメントアウトされるため、先頭の # 文字を削除して有効にする必要があります。

```
/usr/libexec/abrt-action-install-debuginfo-to-abrt-cache --size_mb=4096 &&
```

- gdb パッケージ（問題分析中にバックトレースを生成できる）がシステムにインストールされている。必要な場合は、Yum パッケージマネージャーでパッケージをインストールする方法の詳細は、「パッケージのインストール」 を参照してください。

ROOT 権限が必要

debuginfo パッケージは、root 権限が必要な rhnplugin を使用してインストールされていることに注意してください。そのため、debuginfo パッケージをインストールするには、ABRT を root で実行する必要があります。

28.4.7. 特定のタイプのクラッシュについての自動レポートの設定

ABRT は、ユーザーの対話なしに、検出された問題やクラッシュを自動的に報告するように設定できます。これを実現するには、analyst-and-report ルールを post-create ルールとして指定します。たとえば、ABRT に、ルールを有効にし、/etc/libreport/events.d/python _event.conf ファイルの EVENT=post-create 条件に置き換えて、ユーザーによる対話なしに Python のクラッシュを Bugzilla に報告 するように指示できます。新しいルールは以下のようになります。

```
EVENT=post-create analyzer=Python
        test -f component || abrt-action-save-package-data
        reporter-bugzilla -c /etc/abrt/plugins/bugzilla.conf
```

作成後の 実行が ROOT 権限で実行されます。

作成後 イベントは、通常 root 権限で実行される abrtd により実行されることに注意してください。

28.4.8. プロキシーサーバーを使用したアップロードとレポート

reporter-bugzilla ツールおよび reporter-upload ツールは、http_proxy および ftp_proxy 環境変数に準拠します。レポートイベントの一部として環境変数を使用すると、レポートを実行するプロセス（通常は abrt-gui または abrt-cli ）からその値を継承します。そのため、作業環境でこれらの変数を使用して、HTTP または FTP プロキシーサーバーを指定できます。

これらのツールを 作成後 イベントの一部として配置すると、abrtd プロセスの子として実行されます。abrtd の環境を調整するか、ルールを変更してこれらの変数を設定する必要があります。以下に例を示します。

```
EVENT=post-create analyzer=Python
        test -f component || abrt-action-save-package-data
        export http_proxy=http://proxy.server:8888/
        reporter-bugzilla -c /etc/abrt/plugins/bugzilla.conf
```

28.4.9. 自動レポーティングの設定

ABRT は faillock Reports を使用するように設定できます。このバグレポートには、以下のような利点があります。

- 有効化されると、ユーザーの対話なしに、brokersReports が自動的 に送信されます。一方、通常のレポートは、ユーザーが手動でトリガーするまで送信されません。
- faillockReports は 匿名で、機密情報は含まれません。これにより、不要なデータが自動的に送信されるリスクがなくなります。
- gitopsReport は検出された問題を JSON オブジェクトとして表します。そのため、これはマシンが読み取り可能であり、自動的に作成され、処理できます。
- gitopsReports は完全なバグレポートよりも小さくなります。
- faillockReports では、大量のデバッグ情報をダウンロードする必要はありません。

faillockReports は複数のゴールを提供します。これは、同一バグの複数の発生により作成される可能性のあるカスタマーケースの重複を防ぐのに役立ちます。さらに、faillockReports は、バグの発生の統計を収集し、異なるシステム全体で既知のバグの検索を有効にします。最後に、このセクションの最後で説明されているように 認証済み の faillockReports が有効になっている場合、ABRT はクライアントにインスタントソリューションを自動的に提示できます。ただし、pidReports は必ずしもバグを修正するための十分な情報を持つエンジニアを提供するものではなく、完全なバグレポートが必要になる場合があります。

gitopsReport は通常、以下の情報が含まれます。

- 変数のないプログラムの呼び出しスタックトレース、またはマルチスレッドの C プログラム、C++ プログラム、および Java プログラムの場合は複数のスタックトレース。
- 使用されているオペレーティングシステム
- crash に関連する RPM パッケージのバージョン

- プログラムが root ユーザーで実行するかどうか。
- カーネル oops の場合は、ホストハードウェアに関する情報がある

警告

お使いのハードウェアに関する情報を Red Hat と共有しない場合は、このレポートを有効にしないでください。

pidReport サンプルについては、Examples of faillockReports 記事を参照してください。

gitopsReports を有効にすると、クラッシュが検出されると、以下がデフォルトで実行されます。

1. ABRT は、問題に関する基本情報を Red Hat の ABRT サーバーに提出します。
2. サーバーは、問題がすでにバグデータベースにあるかどうかを判断します。
3. 存在する場合、サーバーは問題の簡単な説明と、報告されたケースの URL を返します。

   そうでない場合には、サーバーはユーザーを招待して完全な問題レポートを送信します。

すべてのユーザーについて faillockReports を有効にするには、root で以下のコマンドを実行します。

```
~]# abrt-auto-reporting enabled
```

または、以下の行を /etc/abrt/abrt.conf ファイルに追加します。

```
AutoreportingEnabled = yes
```

- *ユーザー固有の設定は $USER/.config/abrt/ ディレクトリーにあります。システム全体の設定を上書きします。*

新しい設定を適用するには、以下のコマンドを実行して ABRT サービスを再起動します。

```
~]# service abrtd restart
```

デフォルトの自動レポーティング動作 - sendReports - 変更できます。これには、/etc/abrt/abrt.conf 設定ファイルの AutoreportingEvent ディレクティブに別の ABRT イベントを割り当てます。標準イベントの概要は 「標準 ABRT インストールでサポートされているイベント」 を参照してください。

また、Red Hat Enterprise Linux 7.1 以降では、ホスト名、machine-id（ /etc/machine-id ファイルから取得）、RHN アカウント番号など、認証済み の faillockReports を送信することもできます。認証された faillockReports の利点は、通常の pidReports があるため、Red Hat のプライベートクラッシュレポートサーバーだけでなく、Red Hat カスタマーポータルに直接移動する点にあります。これにより、Red Hat は、即座にクラッシュできるようにソリューションを提供できます。

認証された 自動レポートを有効にするには、root で以下のコマンドを実行します。

```
~]# abrt-auto-reporting enabled -u RHN_username
```

RHN_username は、Red Hat Network のユーザー名に置き換えてください。このコマンドによりパスワードを要求し、プレーンテキストで /etc/libreport/plugins/gitopssupport.conf ファイルに保存します。

28.5. 集中クラッシュコレクションの設定

ABRT を設定して、クラッシュレポートを複数のシステムから収集し、さらなる処理のために専用システムに送信できます。これは、管理者が数百のシステムにログインしておらず、ABRT で見つかったクラッシュを手動でチェックしたい場合に便利です。この方法を使用するには、libreport-plugin-reportuploader プラグインをインストールする必要があります(yum install libreport-plugin-reportuploader)。ABRT の集中クラッシュコレクションを使用するようにシステムを設定する方法については、以下のセクションを参照してください。

28.5.1. 専用システムで必要な設定手順

専用（サーバー）システムで以下の手順を実行します。

1. クラッシュレポートのアップロード先のディレクトリーを作成します。通常、/var/spool/abrt-upload/ が使用されます（残りのドキュメントでは、このディレクトリーを使用していることを前提としています）。このディレクトリーが abrt ユーザーが書き込み可能であることを確認してください。

ABRT ユーザーおよびグループ

abrt-desktop パッケージをインストールすると、abrt という名前の新しいシステムユーザーとグループが作成されます。このユーザーは abrtd デーモンによって使用されます。たとえば、/var/spool/abrt/* ディレクトリーの owner:group として使用されます。

2. /etc/abrt/abrt.conf 設定ファイルで、WatchCrashdumpArchiveDir ディレクティブを以下に設定します。

```
WatchCrashdumpArchiveDir = /var/spool/abrt-upload/
```

3. FTP、SCP など、希望するアップロードメカニズムを選択します。FTP の設定方法に関する詳細は、「FTP」 を参照してください。SCP の設定方法に関する詳細は、「scp ユーティリティーの使用」 を参照してください。

アップロード方法が機能するかどうかを確認することが推奨されます。たとえば、FTP を使用する場合は、インタラクティブな FTP クライアントを使用してファイルをアップロードします。

```
~]$ ftp
ftp> open servername
Name: username
Password: password
ftp> cd /var/spool/abrt-upload
250 Operation successful
ftp> put testfile
ftp> quit
```

testfile がサーバーシステムの適切なディレクトリーに表示されているかどうかを確認します。

4. 予想されるクラッシュデータのボリュームがデフォルトの MaxCrashReportsSize MB よ

りも大きい場合は、（/etc/abrt/abrt.conf 設定ファイル内）1000 ディレクティブには大きな値を設定する必要があります。

5. C/C++ のクラッシュのバックトレースを生成するかどうかを検討します。

バックトレースを全く生成しない場合や、問題が発生した場合は、サーバーでバックトレース生成を無効にできます。標準の ABRT インストールでは、/etc/libreport/events.d/ccpp_events.conf 設定ファイルの以下のルールを使用して、C/C++ クラッシュのバックトレースが生成されます。

```
EVENT=analyze_LocalGDB analyzer=CCpp
        abrt-action-analyze-core.py --core=coredump -o build_ids &&
        abrt-action-install-debuginfo-to-abrt-cache --size_mb=4096 &&
        abrt-action-generate-backtrace &&
        abrt-action-analyze-backtrace
```

ルールに remote!=1 条件を追加して、このルールがアップロードされた問題データに適用されていないことを確認します。

6. 問題データのパッケージ情報（パッケージ および コンポーネント 要素）を収集するかどうかを決定します。集中クラッシュコレクション設定でパッケージ情報を収集する必要があるかどうか、またこれを適切に設定する必要があるかどうかを確認するには、「パッケージ情報の保存」 を参照してください。

28.5.2. クライアントシステムで必要な設定手順

中央の管理方法を使用するすべてのクライアントシステムで以下の手順を実行します。

1. バックトレースを生成したくない場合や、サーバーシステムでそれを生成する場合は、/etc/libreport/events.d/ccpp_events.conf ファイルの対応するルールを削除またはコメントアウトする必要があります。このような例は、「専用システムで必要な設定手順」 を参照してください。

2. クライアントマシンでパッケージ情報を収集しない場合は、/etc/libreport/events.d/abrt_event.conf ファイルで abrt-action-save-package-data を実行するルールをコメントアウトまたは変更します。集中クラッシュコレクション設定でパッケージ情報を収集する必要があるかどうか、またこれを適切に設定する必要があるかどうかを確認するには、「パッケージ情報の保存」 を参照してください。

3. 対応する設定ファイルのサーバーシステムに、問題レポートをアップロードするルールを

追加します。たとえば、検出される直後にすべての問題を自動的にアップロードする場合は、/etc/libreport/events.d/abrt_event.conf 設定ファイルで以下のルールを使用できます。

```
EVENT=post-create
        reporter-upload -u scp://user:password@server.name/directory
```

または、クライアントに問題データをローカルに保存し、ABRT GUI/CLI を使用して後でアップロードする場合は、reporter-upload プログラムを report_SFX イベントとして実行する同様のルールを使用することもできます。以下は、このようなイベントの例になります。

```
EVENT=report_UploadToMyServer
        reporter-upload -u scp://user:password@server.name/directory
```

28.5.3. パッケージ情報の保存

シングルマシンの ABRT インストールでは、問題は通常 RHTSupport や Bugzilla などの外部バグデータベースにレポートされます。通常、これらのバグデータベースへの報告には、問題が発生したコンポーネントおよびパッケージに関する知識が必要です。post-create イベントは、abrt-action-save-package-data ツール（他のステップ以外）を実行して、標準の ABRT インストールにこの情報を提供します。

集中クラッシュ収集システムを設定する場合、要件が大幅に異なる場合があります。必要性に応じて、2 つのオプションがあります。

問題の内部分析

問題データの収集後、パッケージ情報を外部のバグデータベースに報告せずに、インハウス内の問題を診断する予定がある場合に、収集する必要はありません。また、組織またはシステムにインストールされているサードパーティーアプリケーションが作成したプログラムで発生するクラッシュを収集することもできます。このようなプログラムが RPM パッケージの一部である場合は、クライアントシステム と 専用のクラッシュの収集システム では、それぞれの GPG キーを /etc/abrt/gpg_keys ファイルに追加するか、/etc/abrt/abrt-action-save-package-data.conf ファイルに以下の行を追加します。

```
OpenGPGCheck = no
```

プログラムが RPM パッケージに属していない場合は、クライアントシステムと 専用のクラッシュ収集システム の両方で以下の手順を実行します。

- /etc/libreport/events.d/abrt_event.conf ファイルから以下のルールを削除します。

```
EVENT=post-create component=
      abrt-action-save-package-data
```

- /etc/abrt/abrt-action-save-package-data.conf ファイルに以下のディレクティブを設定して、インストール済みのパッケージに対応しない問題データディレクトリーを削除しないでください。

```
ProcessUnpackaged = yes
```

外部バグデータベースへのレポート

または、クラッシュを RHTSupport または Bugzilla に報告することもできます。この場合は、パッケージ情報を収集する必要があります。通常、クライアントマシンと専用のクラッシュ収集システムでは、特定でないインストール済みパッケージセットがあります。そのため、クライアントからアップロードされる問題データが、専用のクラッシュ収集システムにインストールされているパッケージに対応していない可能性があります。標準の ABRT 設定では、問題データが削除されます（ABRT は、パッケージされていない実行可能ファイルでクラッシュとみなします）。これが発生しないようにするには、以下のように 専用のシステム で ABRT の設定を変更する必要があります。

- /etc/libreport/events.d/abrt_event.conf ファイルに remote!=1 条件を追加して、クライアントマシンからアップロードされる問題データのパッケージ情報の誤った収集がされないようにします。

```
EVENT=post-create remote!=1 component=
      abrt-action-save-package-data
```

- /etc/abrt/abrt-action-save-package-data.conf に以下のディレクティブを設定して、インストール済みのパッケージに対応しない問題データディレクトリーを削除しないでください。

```
ProcessUnpackaged = yes
```

専用システムにのみ必要な設定

この場合、クライアントシステムではそのような変更は不要であることに注意してください。パッケージ情報を収集し、パッケージされていない実行可能ファイルでクラッシュを無視します。

28.5.4. ABRT のクラッシュ検出のテスト

設定プロセスのすべての手順を完了すると、基本セットアップが完了します。この設定が適切に機能することをテストするには、**kill -s SEGV PID** コマンドを使用してクライアントシステムでプロセスを終了します。たとえば、以下の方法で **sleep** プロセスを開始して、**kill** コマンドでそれを終了します。

```
~]$ sleep 100 &
[1] 2823
~]$ kill -s SIGSEGV 2823
```

ABRT は、**kill** コマンドの実行直後にクラッシュを検出する必要があります。クライアントシステムの **ABRT** がクラッシュを検出していることを確認します（これは、**abrt-cli list --full** コマンドを実行するか、または **/var/spool/abrt** ディレクトリーに作成されたクラッシュダンプを調べることによって、適切な **syslog** ファイルを調べたりして確認可能）。また、サーバーシステムにコピーし、サーバーシステムに展開され、サーバーシステムで **abrt-cli** または **abrt-gui** を使用すると確認できます。

第29章 OPROFILE

OProfile は、オーバーヘッドが低い、システム全体のパフォーマンス監視ツールです。これはプロセッサー上のパフォーマンス監視ハードウェアを使用して、メモリーの参照時、L2 キャッシュ要求の数、受信したハードウェア割り込みの数など、システムのカーネルと実行可能ファイルに関する情報を取得します。Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux システムで、このツールを使用するには oprofile パッケージがインストールされている必要があります。

多くのプロセッサーには、専用のパフォーマンス監視ハードウェアが含まれます。このハードウェアを使用すると、特定のイベントの発生時（要求されたデータがキャッシュにないなど）を検出できます。ハードウェアは通常、イベントが実行されるたびにインクリメントされる 1 つ以上の カウンター の形式を取ります。カウンターの値が基本的にロールオーバーされると割り込みが生成され、パフォーマンス監視で生成される詳細（つまりオーバーヘッド）の量を制御できます。

OProfile はこのハードウェア（またはパフォーマンス監視ハードウェアが存在しない場合）を使用して、カウンターが割り込みを生成するたびにパフォーマンス関連のデータの サンプル を収集します。これらのサンプルは定期的にディスクに書き込まれます。その後、これらのサンプルに含まれるデータを使用して、システムレベルのパフォーマンスおよびアプリケーションレベルのパフォーマンスに関するレポートを生成できます。

OProfile は便利なツールですが、これを使用する場合にはいくつかの制限に注意してください。

- 共有ライブラリーの使用： 共有ライブラリー のコードの Samples は、--separate=library オプションが使用されていない限り、特定のアプリケーションに属性されません。
- パフォーマンス監視サンプルは正確に行われません。パフォーマンス監視レジスターがサンプルをトリガーすると、割り込み処理はゼロ例外による分割のように正確で はありません。プロセッサーによる命令の順不同な実行により、サンプルはほぼ近い命令に記録される場合があります。
- opreport は、インライン関数のサンプルを適切に関連付けません。opreport は単純なアドレス範囲メカニズムを使用して、アドレスがどの関数にあるかを判断します。インライン関数のサンプルは、インライン関数には属性ではなく、インライン関数が挿入された関数に対して属性されません。
- OProfile は複数の実行からのデータを累積 します。OProfile はシステム全体のプロファイラーであり、プロセスが複数回起動し、シャットダウンすることを想定します。そのため、複数の実行のサンプルが累積されました。opcontrol --reset コマンドを使用して、直前の実行からサンプルを消去します。

- ハードウェアパフォーマンスカウンターはゲスト仮想マシンでは機能しません。ハードウェアパフォーマンスカウンターは仮想システムでは使用できません。そのため、timer モードを使用する必要があります。opcontrol --deinit コマンドを実行してから、modprobe oprofile timer=1 コマンドを実行して timer モードを有効にします。

- CPU の制限のないパフォーマンスの問題: OProfile は、CPU の制限のあるプロセスの問題を見つけるために使用されます。OProfile は、ロックを待機するか、または他のイベントが発生するため、sleep であるプロセスを特定しません（例： I/O デバイスで操作を終了するなど）。

29.1. ツールの概要

表29.1「OProfile コマンド」 では、oprofile パッケージで提供されるツールの概要を説明します。

表29.1 OProfile コマンド

コマンド	説明
ophelp	システムプロセッサーで使用可能なイベントとその簡単な説明を表示します。
opimport	サンプルデータベースファイルをシステム用に外部のバイナリー形式からネイティブの形式に変換します。異なるアーキテクチャーからのサンプルデータベースを解析する場合にのみこのオプションを使用してください。
opannotate	アプリケーションがデバッグシンボルでコンパイルされている場合は、実行可能ファイルのアノテーション付きソースを作成します。詳しくは、「opannotateの使用」 を参照してください。

コマンド	説明
opcontrol	収集されるデータを設定します。詳しくは、「OProfile の設定」 を参照してください。
opreport	プロファイリングデータを取得します。詳しくは、「opreportの使用」 を参照してください。
oprofiled	デーモンとして実行して定期的にサンプルデータをディスクに書き込みます。

29.2. OPROFILE の設定

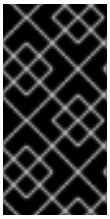
OProfile を実行する前に、設定する必要があります。少なくとも、カーネルの監視（またはカーネルを監視しない選択）を選択することが必要です。次のセクションでは、opcontrol ユーティリティーを使用して OProfile を設定する方法を説明します。opcontrol コマンドを実行すると、設定オプションが /root/.oprofile/daemonrc ファイルに保存されます。

29.2.1. カーネルの指定

まず、OProfile がカーネルを監視するかどうかを設定します。これは、OProfile を起動する前に必要となる唯一の設定オプションです。その他はすべてオプションです。

カーネルを監視するには、root で以下のコマンドを実行します。

```
~]# opcontrol --setup --vmlinux=/usr/lib/debug/lib/modules/`uname -r`/vmlinux
```



DEBUGINFO パッケージのインストール

カーネルを監視するには、カーネルの debuginfo パッケージをインストールする必要があります（非圧縮カーネルが含まれます）。

カーネルを監視しないように OProfile を設定するには、root で以下のコマンドを実行します。

```
~]# opcontrol --setup --no-vmlinux
```

このコマンドは、oprofile カーネルモジュールも読み込みます（まだ読み込まれていない場合は）、/dev/oprofile/ ディレクトリーを作成します（存在しない場合）。このディレクトリーの詳細は、「/dev/oprofile/について」 を参照してください。

カーネル内でサンプルを収集するべきかどうかを設定することにより、収集されるデータのみが変更され、収集したデータが保存される方法や場所は変更されません。カーネルライブラリーおよびアプリケーションライブラリー用にさまざまなサンプルファイルを生成するには、「カーネルとユーザー空間プロファイルの分離」 を参照してください。

29.2.2. 監視するイベントの設定

ほとんどのプロセッサーには カウンター が含まれており、これは特定のイベントを監視するために OProfile によって使用されます。表29.2「OProfile Processors and Counters」 で示されているように、利用可能なカウンターの数はプロセッサーによって異なります。

表29.2 OProfile Processors and Counters

プロセッサー	cpu_type	カウンターの数
AMD64	x86-64/hammer	4
AMD Athlon	i386/athlon	4
AMD Family 10h	x86-64/family10	4
AMD Family 11h	x86-64/family11	4
AMD Family 12h	x86-64/family12	4
AMD Family 14h	x86-64/family14	4

プロセッサー	cpu_type	カウンターの数
AMD Family 15h	x86-64/family15	6
IBM eServer System i および IBM eServer System p	timer	1
IBM POWER4	ppc64/power4	8
IBM POWER5	ppc64/power5	6
IBM PowerPC 970	ppc64/970	8
IBM S/390 および IBM System z	timer	1
Intel Core i7	i386/core_i7	4
Intel Nehalem microarchitecture	i386/nehalem	4
Intel Pentium 4(hy-hyper- thread)	i386/p4	8
Intel Pentium 4（ハイパースレッド）	i386/p4-ht	4
Intel Westmere microarchitecture	i386/westmere	4
TIMER_INT	timer	1

表29.2「OProfile Processors and Counters」 を使用して、正しいプロセッサータイプが検出され、同時に監視できるイベントの数を判断します。timer プロセッサーがパフォーマンス監視ハードウェアに対応していない場合は、プロセッサータイプとして使用されます。

timer を使用する場合は、ハードウェアのパフォーマンスカウンターをサポートしないため、どのプロセッサーにもイベントも設定できません。代わりに、タイマー割り込みがプロファイリングに使用されます。

timer がプロセッサータイプとして使用されていない場合は、監視されるイベントを変更し、プロセッサーのカウンター 0 はデフォルトで時間ベースのイベントに設定されます。プロセッサーに複数のカウンターが存在する場合は、カウンター 0 以外のカウンターはデフォルトでイベントには設定されません。監視されるデフォルトイベントは、表29.3「デフォルトイベント」 に表示されます。

表29.3 デフォルトイベント

プロセッサー	カウンターのデフォルトイベント	説明
AMD Athlon および AMD64	CPU_CLK_UNHALTED	プロセッサーのクロックが停止されていない
AMD Family 10h、AMD Family 11h、AMD Family 12h	CPU_CLK_UNHALTED	プロセッサーのクロックが停止されていない
AMD Family 14h、AMD Family 15h	CPU_CLK_UNHALTED	プロセッサーのクロックが停止されていない
IBM POWER4	サイクル	プロセッサーサイクル
IBM POWER5	サイクル	プロセッサーサイクル
IBM PowerPC 970	サイクル	プロセッサーサイクル
Intel Core i7	CPU_CLK_UNHALTED	プロセッサーのクロックが停止されていない
Intel Nehalem microarchitecture	CPU_CLK_UNHALTED	プロセッサーのクロックが停止されていない
Intel Pentium 4（ハイパースレッドおよび非スレッド）	GLOBAL_POWER_EVENTS	プロセッサーが停止していない時間
Intel Westmere microarchitecture	CPU_CLK_UNHALTED	プロセッサーのクロックが停止されていない
TIMER_INT	（なし）	各タイマー割り込みの例

一度に監視できるイベントの数は、プロセッサーのカウンター数によって決定されます。ただし、1 対 1 の相関ではありません。一部のプロセッサーでは、特定のイベントを特定のカウンターにマップする必要があります。利用可能なカウンターの数を確認するには、以下のコマンドを実行します。

```
~]# ls -d /dev/oprofile/[0-9]*
```

利用可能なイベントはプロセッサーのタイプによって異なります。プロファイリングに使用できるイベントを確認するには、root で以下のコマンドを実行します（一覧はシステムのプロセッサータイプに固有のものです）。

```
~]# ophelp
```

OPROFILE が設定されていることを確認してください。

OProfile が適切に設定されていない限り、ophelp は以下のエラーメッセージを出して失敗します。

```
Unable to open cpu_type file for reading
Make sure you have done opcontrol --init
cpu_type 'unset' is not valid
you should upgrade oprofile or force the use of timer mode
```

OProfile を設定するには、「OProfile の設定」 の手順に従います。

各カウンターのイベントは、コマンドラインまたはグラフィカルインターフェースで設定できます。グラフィカルインターフェースの詳細は、「グラフィカルインターフェース」 を参照してください。カウンターを特定のイベントに設定しない場合は、エラーメッセージが表示されます。

コマンドラインで設定可能な各カウンターのイベントを設定するには、opcontrol を使用します。

```
~]# opcontrol --event=event-name:sample-rate
```

event-name を ophelp のイベントの正確な名前に置き換え、sample-rate をサンプル間のイベント数に置き換えます。

29.2.2.1. サンプリングレート

デフォルトでは、時間ベースのイベントセットが選択されます。プロセッサーごとに 100,000 クロックサイクルごとにサンプルを作成します。タイマー割り込みを使用する場合は、タイマーは jiffy レートがどれでも設定され、ユーザー設定のテーブルではありません。cpu_type が timer でない場合、各イベントには サンプリングレート を設定できます。サンプリングレートは、各サンプルスナップショット間のイベント数です。

カウンターのイベントを設定する場合、サンプルレートを指定することもできます。

```
~]# opcontrol --event=event-name:sample-rate
```

sample-rate を、再度サンプリングするまで待機するイベントの数に置き換えます。数値が小さい

ほど、サンプルが頻繁に使用されます。頻繁に発生しないイベントの場合、イベントインスタンスをキャプチャーするためにカウントが低くなることがあります。

サンプリングが頻繁になりすぎると、システムにオーバーロードする可能性がある

サンプリングレートを設定する場合は十分に注意してください。サンプリングが頻繁にシステムをオーバーロードする可能性があります。これにより、システムがフリーズするか、システムが実際にフリーズするかのようにシステムが表示されます。

29.2.2.2. ユニットマスク

一部のユーザーパフォーマンス監視イベントでも、イベントをさらに定義するのにユニットマスクが必要になる場合があります。

各イベントのユニットマスクは、**ophelp** コマンドで一覧表示されます。各ユニットマスクの値は 16 進数形式で一覧表示されます。複数のユニットマスクを指定するには、ビット単位の操作 または 操作を使用して 16 進数の値を組み合わせる必要があります。

```
~]# opcontrol --event=event-name:sample-rate:unit-mask
```

29.2.3. カーネルとユーザー空間プロファイルの分離

デフォルトでは、カーネルモードおよびユーザーモード情報は、各イベントについて収集されます。特定のカウンターのカーネルモードのイベントを無視するように **OProfile** を設定するには、以下のコマンドを実行します。

```
~]# opcontrol --event=event-name:sample-rate:unit-mask:0
```

次のコマンドを実行して、カウンターのカーネルモードのプロファイリングを開始します。

```
~]# opcontrol --event=event-name:sample-rate:unit-mask:1
```

特定のカウンターのユーザーモードのイベントを無視するように **OProfile** を設定するには、以下のコマンドを実行します。

```
~]# opcontrol --event=event-name:sample-rate:unit-mask:kernel:0
```

次のコマンドを実行して、カウンターのユーザーモードを再度プロファイリングします。

```
~]# opcontrol --event=event-name:sample-rate:unit-mask:kernel:1
```

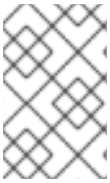
OProfile デーモンがプロファイルデータをサンプルファイルに書き込むと、カーネルとライブラリープロファイルデータをサンプルファイルに分割できます。デーモンがサンプルファイルに書き込む方法を設定するには、root で以下のコマンドを実行します。

```
~]# opcontrol --separate=choice
```

以下のいずれかを 選択 できます。

- none - プロファイルを分離することはできません（デフォルト）。
- ライブラリー - ライブラリー のアプリケーションごとのプロファイルを生成します。
- kernel - カーネルおよびカーネルモジュールのアプリケーションごとのプロファイルを生成します。
- all: ライブラリーおよびカーネルモジュールのアプリケーションごとのプロファイルを生成します。

--separate=library が使用される場合、サンプルファイル名には実行可能ファイルの名前およびライブラリーの名前が含まれます。



OPROFILE プロファイルを再起動します。

これらの設定変更は、OProfile プロファイルが再起動すると有効になります。

29.3. OPROFILE の起動および停止

OProfile を使用してシステムのモニタリングを開始するには、root で以下のコマンドを実行します。

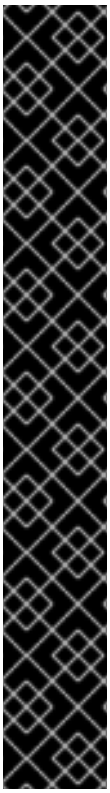
```
~]# opcontrol --start
```

以下のような出力が表示されます。

```
Using log file /var/lib/oprofile/oprofiled.log Daemon started. Profiler running.
```

/root/.oprofile/daemonrc の設定が使用されます。

OProfile デーモン oprofiled が起動し、定期的にサンプルデータを /var/lib/oprofile/samples/ ディレクトリーに書き込みます。デーモンのログファイルは /var/lib/oprofile/oprofiled.log にあります。



NMI_WATCHDOG レジスターを無効にします。

Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;LinuxRed Hat Enterprise Linux 6nbsp;6 システムでは、nmi_watchdog は perf サブシステムに登録します。このため、perf サブシステムは起動時にパフォーマンスカウンターレジスターを制御し、OProfile が機能しなくなります。

これを解決するには、nmi_watchdog=0 カーネルパラメーターセットで起動するか、以下のコマンドを実行してランタイム時に nmi_watchdog を無効にします。

```
~]# echo 0 > /proc/sys/kernel/nmi_watchdog
```

nmi_watchdog を再度有効にするには、以下のコマンドを使用します。

```
~]# echo 1 > /proc/sys/kernel/nmi_watchdog
```

プロファイラーを停止するには、root で以下のコマンドを実行します。

```
~]# opcontrol --shutdown
```

29.4. データの保存

特定の時間にサンプルを保存すると便利な場合もあります。たとえば、実行可能ファイルをプロファイリングする場合、異なる入力データセットに基づいて異なるサンプルを収集すると便利です。監視するイベントの数がプロセッサーで利用可能なカウンター数を超える場合は、OProfile の複数の実行を使用してデータを収集し、常にサンプルデータを異なるファイルに保存します。

サンプルファイルの現在のセットを保存するには、以下のコマンドを実行します。name は、現在のセッションの一意記述名に置き換えます。

```
~]# opcontrol --save=name
```

/var/lib/oprofile/samples/ ディレクトリーが作成され、現在のサンプルファイルがコピーされます。

29.5. データの分析

OProfile デーモン oprofiled は定期的にサンプルを収集し、それらを /var/lib/oprofile/samples/ ディレクトリーに書き込みます。データを読み取る前に、root で以下のコマンドを実行して、すべてのデータがこのディレクトリーに書き込まれていることを確認します。

```
~]# opcontrol --dump
```

各サンプルファイル名は、実行可能ファイルの名前に基づいています。たとえば、/bin/bash の Pentiumでのデフォルトイベントのサンプルは以下のようになります。

```
\{root\}/bin/bash/\{dep\}/\{root\}/bin/bash/CPU_CLK_UNHALTED.100000
```

以下のツールは、サンプルデータの収集後にサンプルデータのプロファイルに使用できます。

- opreport
- opannotate

これらのツールとバイナリープロファイルを使用して、さらに分析できるレポートを生成します。

実行ファイルとサンプルファイルをバックアップします。

プロファイルされる実行可能ファイルは、データを分析するためにこのツールと共に使用する必要があります。データの収集後に変更する必要がある場合は、サンプルファイルおよびサンプルファイルの作成に使用する実行ファイルをバックアップします。サンプルファイルとバイナリーが合意する必要がある点に注意してください。バックアップが一致しない場合には、バックアップは動作しません。この問題に対処するために oparchive を使用できます。

各実行可能ファイルのサンプルは、1 つのサンプルファイルに書き込まれます。動的にリンクされた各ライブラリーからのサンプルも、1 つのサンプルファイルに書き込まれます。OProfile が実行中に、変更がモニタリングされ、実行可能ファイルのサンプルファイルが存在する場合、既存のサンプルファイルが自動的に削除されます。したがって、既存のサンプルファイルが必要な場合は、実行ファイルを新バージョンに置き換える前に、作成に使用する実行ファイルと共にバックアップする必要があります。OProfile 分析ツールは、解析中にサンプルを作成した実行可能ファイルを使用します。実行ファイルが変更すると、分析ツールは関連するサンプルを分析しません。サンプルファイルをバックアップする方法は、「データの保存」 を参照してください。

29.5.1. opreportの使用

opreport ツールは、プロファイルされるすべての実行可能ファイルの概要を提供します。

以下はサンプル出力の一部です。

```
Profiling through timer interrupt
TIMER:0|
samples|      %|
------------------
25926 97.5212 no-vmlinux
359  1.3504 pi
65  0.2445 Xorg
62  0.2332 libvte.so.4.4.0
56  0.2106 libc-2.3.4.so
34  0.1279 libglib-2.0.so.0.400.7
19  0.0715 libXft.so.2.1.2
17  0.0639 bash
8  0.0301 ld-2.3.4.so
8  0.0301 libgdk-x11-2.0.so.0.400.13
6  0.0226 libgobject-2.0.so.0.400.7
5  0.0188 oprofiled
4  0.0150 libpthread-2.3.4.so
4  0.0150 libgtk-x11-2.0.so.0.400.13
3  0.0113 libXrender.so.1.2.2
```

```
3  0.0113 du
1  0.0038 libcrypto.so.0.9.7a
1  0.0038 libpam.so.0.77
1  0.0038 libtermcap.so.2.0.8
1  0.0038 libX11.so.6.2
1  0.0038 libgthread-2.0.so.0.400.7
1  0.0038 libwnck-1.so.4.9.0
```

各実行可能ファイルは各行に一覧表示されます。最初の列は、実行ファイル用に記録されたサンプル数です。2 列目は、サンプルの合計数に対するサンプルの割合です。3 列目は実行可能ファイルの名前です。

利用可能なコマンドラインオプションの一覧は、man ページの opreport を参照してください。たとえば、-r オプションでは、サンプルの最大数が最も少ない実行可能ファイルから、最大数のサンプルを持つ実行ファイルから出力をソートします。

29.5.2. 単一の実行可能ファイルでの opreport の使用

特定の実行ファイルに関するより詳細なプロファイル情報を取得するには、opreport を使用します。

```
~]# opreport mode executable
```

実行可能ファイル は、分析する実行可能ファイルへの完全パスである必要があります。モード は以下のいずれかになります。

-l

シンボルによるサンプルデータを一覧表示します。たとえば、以下は、opreport -l /lib/tls/libc-version.so コマンドを実行する際の出力の一部です。

```
samples % symbol name
12 21.4286 __gconv_transform_utf8_internal
5 8.9286 _int_malloc 4 7.1429 malloc
3 5.3571 __i686.get_pc_thunk.bx
3 5.3571 _dl_mcount_wrapper_check
3 5.3571 mbrtowc
3 5.3571 memcpy
2 3.5714 _int_realloc
2 3.5714 _nl_intern_locale_data
2 3.5714 free
2 3.5714 strcmp
1 1.7857 __ctype_get_mb_cur_max
1 1.7857 __unregister_atfork
1 1.7857 __write_nocancel
1 1.7857 _dl_addr
```

```
1 1.7857 _int_free
1 1.7857 _itoa_word
1 1.7857 calc_eclosure_iter
1 1.7857 fopen@@GLIBC_2.1
1 1.7857 getpid
1 1.7857 memmove
1 1.7857 msort_with_tmp
1 1.7857 strcpy
1 1.7857 strlen
1 1.7857 vfprintf
1 1.7857 write
```

最初の列は、シンボルのサンプル数で、2 番目のコラムは、実行可能なサンプルに対するこのシンボルのサンプルの割合で、3 番目のコラムは記号名になります。

最大数のサンプルから最小（リバース順序）に出力を並べ替えるには、-l オプションとともに -r を使用します。

-i symbol-name

シンボル名に固有のサンプルデータを一覧表示します。たとえば、以下の出力は、opreport -l -i __gconv_transform_utf8_internal /lib/tls/libc-version.so コマンドにあります。

```
samples % symbol name
12 100.000 __gconv_transform_utf8_internal
```

最初の行は、シンボル/実行可能ファイルの組み合わせの概要です。

最初の列は、メモリーシンボルのサンプル数です。2 列目は、シンボルのサンプルの合計数と相対的に、メモリーアドレスのサンプルの割合（パーセント）です。3 列はシンボル名です。

-d

-l よりも多くの詳細を含むシンボルによるサンプルデータを一覧表示します。たとえば、以下の出力は、opreport -l -d __gconv_transform_utf8_internal /lib/tls/libc-version.so コマンドにあります。

```
vma samples % symbol name
00a98640 12 100.000 __gconv_transform_utf8_internal
00a98640 1 8.3333
00a9868c 2 16.6667
00a9869a 1 8.3333
```

```
00a986c1 1 8.3333
00a98720 1 8.3333
00a98749 1 8.3333
00a98753 1 8.3333
00a98789 1 8.3333
00a98864 1 8.3333
00a98869 1 8.3333
00a98b08 1 8.3333
```

データは -l オプションと同じですが、各シンボルで、使用される各仮想メモリーアドレスが表示されます。各仮想メモリーアドレスについて、シンボルのサンプル数と基準に対するサンプル数とパーセンテージが表示されます。

-x symbol-name

出力からシンボルのコンマ区切りリストを除外します。

Session:name

/var/lib/oprofile/samples/ ディレクトリーに対するセッションまたはディレクトリーへの完全パスを指定します。

29.5.3. モジュールでの詳細な出力の取得

OProfile は、マシンで実行しているカーネルおよびユーザー空間コードについてシステム全体のデータを収集します。ただし、モジュールがカーネルに読み込まれると、カーネルモジュールの元の情報が失われます。モジュールは、起動時に initrd ファイルから、さまざまなカーネルモジュールを含むディレクトリー、またはローカルに作成されたカーネルモジュールから送られる場合があります。その結果、OProfile がモジュールのサンプルを記録すると、root ディレクトリーで実行可能なモジュールのサンプルのみが一覧表示されますが、これはモジュールの実際のコードで配置される訳ではありません。分析ツールが実行ファイルを取得できるようにするには、いくつかの手順を実行する必要があります。

モジュールのアクションをより詳細に表示するには、モジュール「トリップ解除」（カスタムビルドからインストールされる）か、カーネル用に debuginfo パッケージをインストールする必要があります。

uname -a コマンドで実行しているカーネルを確認し、適切な debuginfo パッケージを取得し、これをマシンにインストールします。

次に、以下のコマンドで、直前の実行の例をクリアします。

```
~]# opcontrol --reset
```

Westmere プロセッサーを持つマシン上で監視プロセスを開始するには、以下のコマンドを実行します。

```
~]# opcontrol --setup --vmlinux=/usr/lib/debug/lib/modules/`uname -r`/vmlinux --
event=CPU_CLK_UNHALTED:500000
```

次に、ext4 モジュールなどの詳細情報を以下のコマンドで取得できます。

```
~]# opreport /ext4 -l --image-path /lib/modules/`uname -r`/kernel
CPU: Intel Westmere microarchitecture, speed 2.667e+06 MHz (estimated)
Counted CPU_CLK_UNHALTED events (Clock cycles when not halted) with a unit mask of
0x00 (No unit mask) count 500000
warning: could not check that the binary file /lib/modules/2.6.32-
191.el6.x86_64/kernel/fs/ext4/ext4.ko has not been modified since the profile was taken.
Results may be inaccurate.
samples  %        symbol name
1622     9.8381  ext4_iget
1591     9.6500  ext4_find_entry
1231     7.4665  __ext4_get_inode_loc
783      4.7492  ext4_ext_get_blocks
752      4.5612  ext4_check_dir_entry
644      3.9061  ext4_mark_iloc_dirty
583      3.5361  ext4_get_blocks
583      3.5361  ext4_xattr_get
479      2.9053  ext4_htree_store_dirent
469      2.8447  ext4_get_group_desc
414      2.5111  ext4_dx_find_entry
```

29.5.4. opannotateの使用

opannotate ツールは、特定の手順のサンプルをソースコードの対応する行と照合しようとします。生成されたファイルには、左側の行のサンプルが必要です。また、関数の合計サンプルを一覧表示する各関数の最初のコメントに配置します。

このユーティリティーを機能させるには、実行可能ファイルに適した debuginfo パッケージがシステムにインストールされている必要があります。Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux では、debuginfo パッケージは、実行ファイルを含む対応するパッケージで自動的にインストールされません。個別に取得してインストールする必要があります。

opannotate の一般的な構文は以下のとおりです。

```
~]# opannotate --search-dirs src-dir --source executable
```

ソースコードと分析対象の実行可能ファイルを含むディレクトリーを指定する必要があります。追加のコマンドラインオプションの一覧は、man ページの opannotate を参照してください。

29.6. /DEV/OPROFILE/について

/dev/oprofile/ ディレクトリーには、OProfile のファイルシステムが含まれます。cat コマンドを使用して、このファイルシステムの仮想ファイルの値を表示します。たとえば、以下のコマンドは検出されたプロセッサー OProfile のタイプを表示します。

```
~]# cat /dev/oprofile/cpu_type
```

カウンターごとにディレクトリーが /dev/oprofile/ に存在する。たとえば、カウンターが 2 つある場合は、/dev/oprofile/0 / ディレクトリーと dev/oprofile /1/ ディレクトリーが存在 します。

カウンターの各ディレクトリーには、以下のファイルが含まれます。

- count: サンプルの間隔
- enabled: 0 の場合、カウンターはオフになり、そのカウンターは収集されません。1 の場合は、カウンターがオンになり、サンプルが収集されます。
- event: 監視するイベント。
- extra: Nehalem プロセッサーを持つマシンで使用し、さらに監視するイベントを指定します。
- kernel: 0 の場合は、プロセッサーが カーネル スペースにあるときにこのカウンターイベントのサンプルが収集されません。1 の場合は、プロセッサーがカーネルスペースにある場合でもサンプルが収集されます。

- *unit_mask: カウンターに対して有効なユニットマスクを定義します。*

- *User: 0 の場合は、プロセッサーが ユーザー 空間にあるときにカウンターイベントのサンプルが収集されません。1 の場合は、プロセッサーがユーザー空間にある場合でもサンプルが収集されます。*

このファイルの値は、cat コマンドで取得できます。以下に例を示します。

```
~]# cat /dev/oprofile/0/count
```

29.7. 使用例

OProfile は開発者がアプリケーションのパフォーマンスを分析するのに使用できますが、システム管理者がシステム分析を実行するのに使用することもできます。以下に例を示します。

- システムで使用されるアプリケーションとサービスを判別 *します。opreport を使用して、アプリケーションやサービスが使用するプロセッサー時間を判別できます。システムが複数のサービスに使用されていても、実行中である場合は、最もプロセッサー時間を消費するサービスを専用のシステムに移動できます。*

- プロセッサーの使用量の決定*: CPU_CLK_UNHALTED イベントを監視し、一定期間におけるプロセッサーの負荷を判断できます。このデータを使用して、追加のプロセッサーが追加または高速のプロセッサーがシステムパフォーマンスを向上させるかどうかを判断できます。*

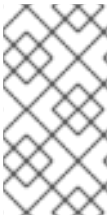
29.8. JAVA の OPROFILE サポート

OProfile を使用すると、Java 仮想マシン(JVM)の動的にコンパイルされたコード（Just-in-time または JIT コード）をプロファイルできます。Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;Linux Red Hat Enterprise Linux 6nbsp;6 の OProfile には、Java 1.5 以降をサポートする JVM Tools Interface(JVMTI)エージェントライブラリーに対する組み込みサポートが含まれています。

29.8.1. Java コードのプロファイリング

JVMTI エージェントを使用して Java 仮想マシンからの JIT コードをプロファイルするには、以下を JVM 起動パラメーターに追加します。

```
-agentlib:jvmti_oprofile
```



OPROFILE-JIT パッケージのインストール

OProfile で JIT コードをプロファイルするには、oprofile-jit パッケージがシステムにインストールされている必要があります。

OProfile での Java サポートの詳細は、「その他のリソース」 にリンクされている OProfile Manual を参照してください。

29.9. グラフィカルインターフェース

一部の OProfile 設定は、グラフィカルインターフェースで設定できます。これを起動するには、シェルプロンプトで root で oprof_start コマンドを実行します。グラフィカルインターフェースを使用するには、oprofile-gui パッケージがインストールされている必要があります。

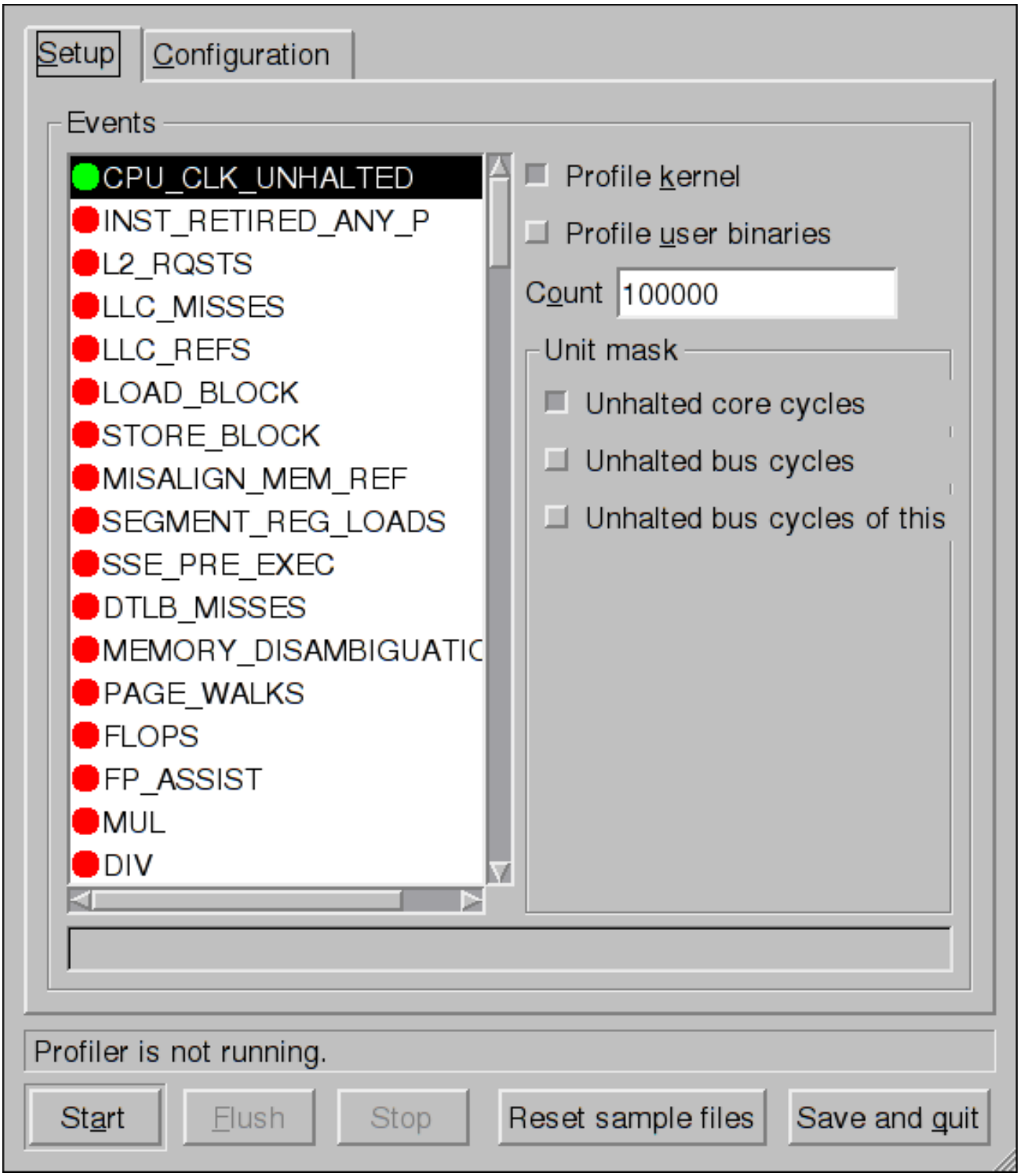
オプションのいずれかを変更したら、Save and quit ボタンをクリックして保存します。設定は /root/.oprofile/daemonrc に書き込まれ、アプリケーションが終了します。

SAVE AND QUIT ボタンをクリックします

アプリケーションを終了しても OProfile がサンプリングから削除されません。

Setup タブでプロセッサーカウンターのイベントを設定するには、「監視するイベントの設定」 プルダウンメニューからカウンターを選択し、一覧からイベントを選択します。イベントの簡単な説明が、リストの下にあるテキストボックスに表示されます。特定のカウンターと特定のアーキテクチャーで利用可能なイベントのみが表示されます。また、このインターフェースは、プロファイラーが実行されているかどうかと、それに関する簡単な統計も表示します。

図29.1 OProfile のセットアップ



[D]

*タブの右側にある **Profile kernel** オプションを選択して、現在選択されているイベントのカーネルモードでイベントをカウントします。「カーネルとユーザー空間プロファイルの分離」で説明されているように、カーネルモードでイベントをカウントします。このオプションが選択されていない場合、カーネルのサンプルは収集されません。*

*「カーネルとユーザー空間プロファイルの分離」で説明されているように、**Profile user binaries** オプションを選択して、現在選択されているイベントのイベントをユーザーモードでカウントします。*

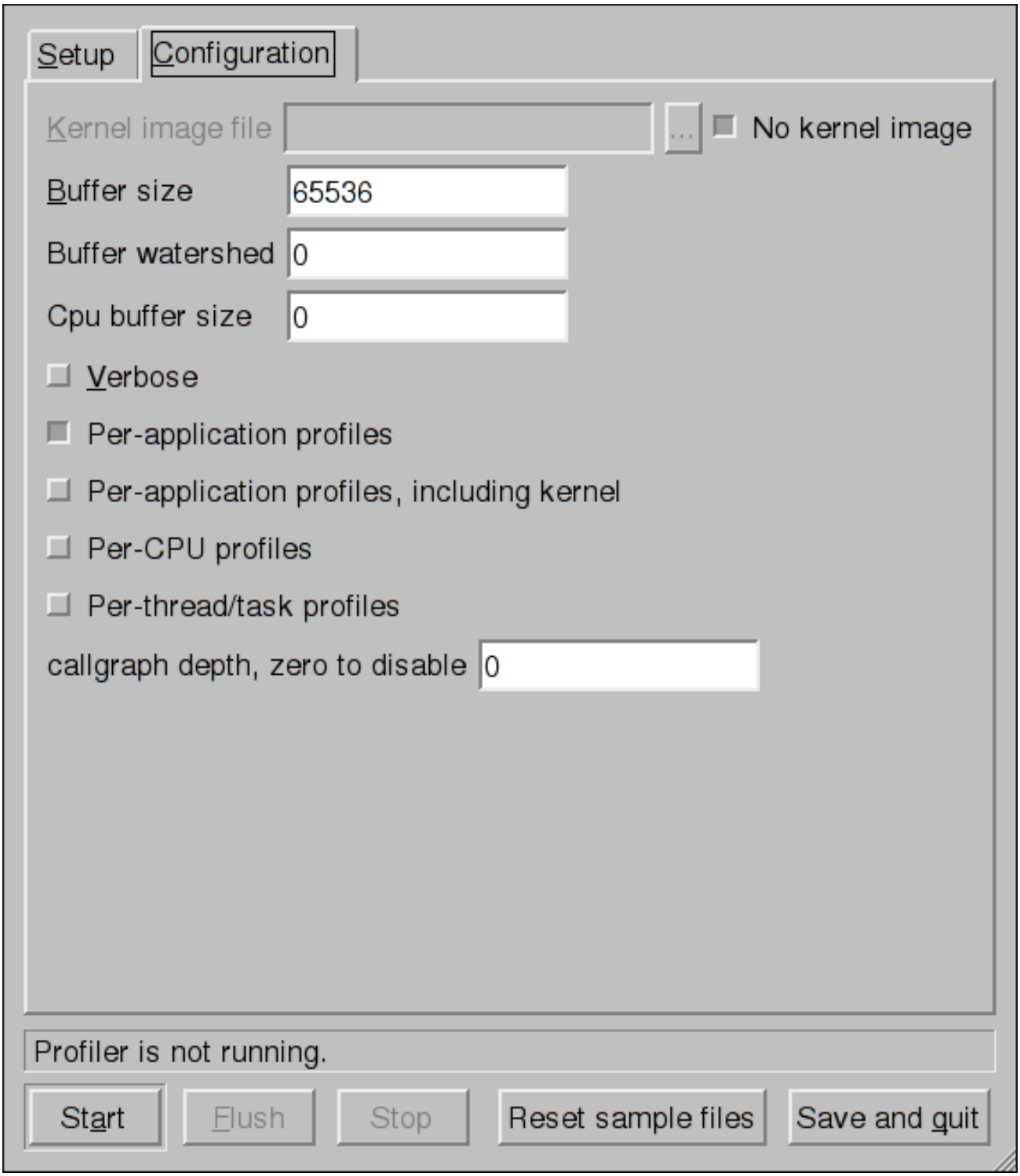
このオプションが選択されていない場合、ユーザーアプリケーションのサンプルは収集されません。

Count テキストフィールドを使用して、「サンプリングレート」 で説明されているように、現在選択されているイベントのサンプリングレートを設定します。

「ユニットマスク」 で説明されているように、現在選択されているイベントでユニットマスクが利用可能な場合は、Setup タブの右側の Unit Masks エリアに表示されます。ユニットマスクの横にあるチェックボックスを選択してイベントで有効にします。

設定 タブでカーネルのプロファイルを行うには、カーネル イメージ ファイルのテキストフィールドで監視するカーネルの vmlinux ファイルの名前と場所を入力します。OProfile がカーネルを監視しないように設定するには、No kernel image を選択します。

図29.2 OProfile の設定



[D]

Verbose オプションを選択すると、oprofiled デーモンログにより多くの情報が含まれます。

アプリケーションプロファイル per-application が選択されている場合、OProfile はライブラリーのアプリケーションごとのプロファイルを生成します。これは opcontrol --separate=library コマンドと同じです。カーネルを含むアプリケーションごとのプロファイル が選択されている場合、OProfile は「カーネルとユーザー空間プロファイルの分離」 で説明されているように、カーネルおよびカーネルモ

ジュールのアプリケーションごとのプロファイルを生成します。これは opcontrol --separate=kernel コマンドと同じです。

「データの分析」 で説明されているように、データがサンプルファイルに書き込まれるように強制するには、Flush ボタンをクリックします。これは opcontrol --dump コマンドと同じです。

グラフィカルインターフェースから OProfile を起動するには、Start をクリックします。プロファイラーを停止するには、Stop をクリックします。アプリケーションを終了しても OProfile がサンプリングから削除されません。

29.10. OPROFILE および SYSTEMTAP

SystemTap は、オペレーティングシステムのアクティビティーを詳細に調査および監視できる追跡およびプロービングツールです。netstat、ps、top、iostat などのツールの出力に似た情報を提供しますが、SystemTap は収集した情報に対してより多くのフィルタリングと分析オプションを提供するように設計されています。

OProfile の使用が推奨されますが、プロセッサーがコードの特定領域で時間を費やした場所や理由で、プロセッサーがアイドル状態のままになる理由を見つける際は、OProfile の使用が推奨されます。

コードの特定の場所をインストルメント化するときに SystemTap の使用を希望する場合があります。SystemTap ではインストルメンテーションを停止および再起動しなくてもコードインストルメンテーションを実行することができるため、カーネルとデーモンのインストルメント化には特に便利です。

SystemTap の詳細は、関連する SystemTap ドキュメントの 「便利な Web サイト」 を参照してください。

29.11. その他のリソース

本章では、OProfile と設定、使用方法のみを取り上げています。詳細は、以下のリソースを参照してください。

29.11.1. Installed Docs

- /usr/share/doc/oprofile-version/oprofile.html - 『OProfile Manual』

- *OProfile man ページ： opcontrol、opreport、opannotate、および ophelp*

29.11.2. 便利な Web サイト

- *http://oprofile.sourceforge.net/ - 最新のドキュメント、メーリングリスト、IRC チャネルなどが含まれます。*

- *SystemTap ビギナーズガイド: SystemTap を使用して Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux の異なるサブシステムを監視する基本的な手順を提供します。*

パート VIII. カーネル、モジュール、およびドライバーの設定

ここでは、管理者がカーネルのカスタマイズに役立つさまざまなツールについて説明します。

第30章 カーネルの手動によるアップグレード

Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux カーネルは、サポートしているハードウェアとの整合性と互換性を確保するために、Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux カーネルチームがカスタムを構築します。Red Hat がカーネルをリリースする前に、まず厳格な品質保証テストセットを渡す必要があります。

Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux カーネルは RPM 形式でパッケージ化されるため、Yum または PackageKit パッケージマネージャーを使用したアップグレードと検証が容易になります。PackageKit は自動的に Red Hat Network サーバーをクエリーし、カーネルパッケージなど、利用可能な更新を含むパッケージを通知します。

本章は、yum の代わりに rpm コマンドを使用して手動でカーネルパッケージを更新する必要のあるユーザーに のみ 有用です。

YUM を使用して、可能な限りカーネルをインストールします。

可能な場合は、Yum または PackageKit パッケージマネージャーを使用して新しいカーネルをインストールしてください。これは、現在のカーネルの代わりに常に新しいカーネルを インストール し、システムが起動できなくなる可能性があるためです。

カスタムカーネルのビルドはサポートされていません。

カスタムカーネルの構築は、Red Hat グローバルサービスサポートチームではサポートされないため、このマニュアルでは検討されていません。

Yum を使用したカーネルパッケージのインストールの詳細は、「パッケージの更新」 を参照してください。Red Hat Network の詳細は、6章システム登録およびサブスクリプション管理 を参照してください。

30.1. カーネルパッケージの概要

Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux には、以下のカーネルパッケージが含まれています。

- *kernel - 単一、マルチコア、マルチプロセッサーシステム用のカーネルが含まれます。*
- *kernel-debug - カーネル診断用に有効なデバッグオプションが多数含まれるカーネルが含まれますが、パフォーマンスが低下します。*
- *kernel-devel - kernel パッケージに対してモジュールを構築するのに十分なカーネルヘッダーと makefiles が含まれます。*
- *kernel-debug-devel - カーネル診断用に多くのデバッグオプションが有効になっている開発バージョンのカーネルが含まれます。ただし、パフォーマンスが低下します。*
- *kernel-doc - カーネルソースのドキュメントファイル。これらのファイルには、同梱で配布される Linux カーネルとデバイスドライバーのさまざまな部分が文書化されています。このパッケージをインストールすると、オプションへの参照が提供され、読み込み時に Linux カーネルモジュールに渡すことができます。*

  デフォルトでは、これらのファイルは /usr/share/doc/kernel-doc- <kernel_version> / ディレクトリーに配置されます。
- *kernel-headers - Linux カーネルとユーザー空間ライブラリーとプログラム間のインターフェースを指定する C ヘッダーファイルが含まれます。ヘッダーファイルは、ほとんどの標準プログラムを構築するのに必要な構造と定数を定義します。*
- *kernel-firmware - さまざまなデバイスで動作するために必要なすべてのファームウェアファイルが含まれます。*
- *perf - このパッケージには、各カーネルイメージのサブパッケージに同梱される perf ツールのサポートスクリプトとドキュメントが含まれています。*

30.2. アップグレードの準備

カーネルをアップグレードする前に、予防的な前準備手順の実行をお薦めします。

まず、システム用に作業中ブートメディアが存在することを確認します。ブートローダーが新しいカーネルを起動するように適切に設定されていない場合は、このメディアを使用して Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux で起動できます。

USB メディアは、多くの場合、ペンドライブ、サムディスク、または 鍵 と呼ばれるフラッシュデバイスの形式、または外部接続のハードディスクデバイスとして提供されます。このタイプのほとんどすべてのメディアは VFAT ファイルシステムとしてフォーマットされています。ext2、ext3、または VFAT としてフォーマットされているメディア上で起動可能な USB メディアを作成できます。

ディストリビューションのイメージファイル、または最低限ブートメディア (minimal boot media) イメージを USB メディアに転送することができます。デバイスには十分な空き領域があることを確認してください。約 4 GB はディストリビューション DVD イメージ、ディストリビューション CD イメージの場合は 700 MB、または最小ブートメディアイメージの場合は約 10 MB に必要です。

Red Hat Enterprise Linuxnbsp;Hat Enterprise Red Hat Enterprise Linuxnbsp;Linux インストール DVD、またはインストール CD-ROM #1 からの boot.iso ファイルのコピーと、空き領域の約 16 MB でフォーマットされた USB ストレージデバイスが必要です。以下の手順は、コピー先のファイルと同じパス名を持っている場合を除き、USB ストレージデバイスにある既存のファイルには影響しません。USB ブートメディアを作成するには、root で以下のコマンドを実行します。

1. USB ストレージデバイスに SYSLINUX ブートローダーをインストールします。

```
~]# syslinux /dev/sdX1
```

... sdX はデバイス名です。

2. boot.iso と USB ストレージデバイス用にマウントポイントを作成します。

```
~]# mkdir /mnt/isoboot /mnt/diskboot
```

3. boot.iso をマウントします。

```
~]# mount -o loop boot.iso /mnt/isoboot
```

4. USB ストレージデバイスをマウントします。

```
~]# mount /dev/<sdX1> /mnt/diskboot
```

5. ISOLINUX ファイルを boot.iso から USB ストレージデバイスにコピーします。

```
~]# cp /mnt/isoboot/isolinux/* /mnt/diskboot
```

6. boot.iso からの isolinux.cfg ファイルを USB デバイスの syslinux.cfg ファイルとして使用します。

```
~]# grep -v local /mnt/isoboot/isolinux/isolinux.cfg > /mnt/diskboot/syslinux.cfg
```

7. boot.iso と USB ストレージデバイスをアンマウントします。

```
~]# umount /mnt/isoboot /mnt/diskboot
```

8. ブートメディアでマシンを再起動し、起動できることを確認してから、続行してください。

別の方法として、フロッピードライブがあるシステムで mkbootdisk パッケージをインストールし、root で mkbootdisk コマンドを実行してブートディスクを作成できます。このパッケージをインストールした後に使用法について確認するには、man mkbootdisk man ページを参照してください。

どのカーネルパッケージがインストールされているかを確認するには、シェルプロンプトで yum list installed "kernel-*" コマンドを実行します。出力は、システムのアーキテクチャーに応じて以下のパッケージの一部またはすべてで構成されており、バージョン番号は異なる場合があります。

```
~]# yum list installed "kernel-*"
kernel.x86_64                2.6.32-17.el6          @rhel-x86_64-server-6
kernel-doc.noarch            2.6.32-17.el6          @rhel-x86_64-server-6
kernel-firmware.noarch       2.6.32-17.el6          @rhel-x86_64-server-6
kernel-headers.x86_64        2.6.32-17.el6          @rhel-x86_64-server-6
```

この出力から、カーネルのアップグレード用にダウンロードすべきパッケージを判断します。1 つのプロセッサーシステムの場合、必要なパッケージは kernel パッケージのみです。異なるパッケージの説明は、「カーネルパッケージの概要」 を参照してください。

30.3. アップグレードされたカーネルのダウンロード

システム用に更新されたカーネルが利用可能かを判定する手段は数種類あります。

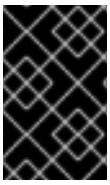
- セキュリティーエラータ - セキュリティー問題を修正するカーネルのアップグレードなど、セキュリティーエラータの詳細 http://www.redhat.com/security/updates/ は を参照してください。

- Red Hat Network: Red Hat Network にサブスクライブしているシステムの場合は、yum パッケージマネージャーが最新のカーネルをダウンロードして、システム上のカーネルをアップグレードできます。Dracut ユーティリティーは、必要に応じて初期 RAM ディスクイメージを作成し、新しいカーネルを起動するようにブートローダーを設定します。Red Hat Network からパッケージをインストールする方法は、8章Yum を参照してください。Red Hat Network にシステムを登録する方法は、6章システム登録およびサブスクリプション管理 を参照してください。

更新されたカーネルを Red Hat Network からダウンロードおよびインストールするために yum を使用している場合は、「初期 RAM ディスクイメージの確認」 および 「ブートローダーの確認」 の指示に従うようにしてください。これは、デフォルトでカーネルをブートし ない ように変更しません。カーネルはデフォルトで起動するように変更しないでください。カーネルを手動でインストールするには、「アップグレードの実行」 に進みます。

30.4. アップグレードの実行

必要なパッケージをすべて取り込んだ後は、既存カーネルをアップグレードします。



アップグレードの実行時に古いカーネルを保持する

新しいカーネルに問題がある場合を考え、古いカーネルの維持を強く推奨します。

シェルプロンプトで、カーネル RPM パッケージを格納しているディレクトリーに移動します。rpm コマンドに -i 引数を使用して古いカーネルを残します。-U オプションは、現在インストールされているカーネルを上書きして、ブートローダーの問題を作成するため、使用し ない でください。以下に例を示します。

```
~]# rpm -ivh kernel-<kernel_version>.<arch>.rpm
```

次の手順では、初期 RAM ディスクイメージが作成されていることを確認します。詳しくは、「初期 RAM ディスクイメージの確認」 を参照してください。

30.5. 初期 RAM ディスクイメージの確認

初期 RAM ディスクイメージのジョブは、IDE、SCSI、RAID などのブロックデバイスモジュールをプレロードすることです。Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;LinuxRed

Hat Enterprise Linux 6nbsp;6 システムでは、Yum、PackageKit、または RPM パッケージマネージャーのいずれかを使用して新しいカーネルがインストールされるたびに、Dracut ユーティリティーは常に、initramfs（初期 RAM ディスクイメージ）を作成するためにインストールスクリプトで呼び出されます。

IBM eServer System i（「IBM eServer System i 上の初期 RAM ディスクイメージとカーネルの検証」を参照）以外のすべてのアーキテクチャー上では、dracut コマンドを実行すると initramfs を作成できます。ただし、initramfs を手動で作成する必要はありません。このステップは、カーネルとその関連パッケージが Red Hatnbsp;Hat によって配布される RPM パッケージからインストールされているか、またはアップグレードされている場合には自動的に実行されます。

以下の手順に従って、現在のカーネルバージョンに対応する initramfs が存在し、grub.conf 設定ファイルで正しく指定されていることを確認できます。

手順30.1 初期 RAM ディスクイメージの確認

1. *root として、/boot/ ディレクトリーのコンテンツを一覧表示し、最新のバージョン番号でカーネル(vmlinuz- <kernel_version>)と initramfs- <kernel_version > を検索します。*

例30.1 カーネルと initramfs バージョンの一致を確認

```
~]# ls /boot/
config-2.6.32-17.el6.x86_64          lost+found
config-2.6.32-19.el6.x86_64          symvers-2.6.32-17.el6.x86_64.gz
config-2.6.32-22.el6.x86_64          symvers-2.6.32-19.el6.x86_64.gz
efi                                  symvers-2.6.32-22.el6.x86_64.gz
grub                                 System.map-2.6.32-17.el6.x86_64
initramfs-2.6.32-17.el6.x86_64.img   System.map-2.6.32-19.el6.x86_64
initramfs-2.6.32-19.el6.x86_64.img   System.map-2.6.32-22.el6.x86_64
initramfs-2.6.32-22.el6.x86_64.img   vmlinuz-2.6.32-17.el6.x86_64
initrd-2.6.32-17.el6.x86_64kdump.img vmlinuz-2.6.32-19.el6.x86_64
initrd-2.6.32-19.el6.x86_64kdump.img vmlinuz-2.6.32-22.el6.x86_64
initrd-2.6.32-22.el6.x86_64kdump.img
```

例30.1「カーネルと initramfs バージョンの一致を確認」は以下の点を示しています。

- *3 つのカーネルがインストールされています（より正確には、3 つのカーネルファイルが /boot/にあります）。*
- *最新のカーネルは vmlinuz-2.6.32-22.el6.x86_64 です。*

- カーネルバージョン initramfs - 2.6.32-22.el6.x86_64.img に一致する initramfs ファイルもあります。

/BOOT ディレクトリーの INITRD ファイルが INITRAMFS ファイルと同じではない

/boot/ ディレクトリーには、initrd- <version> kdump.img ファイルが複数見つかる場合があります。これは、カーネルのデバッグ目的で Kdump メカニズムで作成される特別なファイルであり、システムの起動には使用されません。また、無視しても問題はありません。

2. （オプション） initramfs- <kernel_version > ファイルが /boot/ 内の最新カーネルのバージョンと一致しない場合、または他の特定の状況では、Dracut ユーティリティーを使用して initramfs ファイルを生成する必要がある場合があります。root としてオプションを指定せずに dracut を呼び出すと、そのディレクトリーにある最新のカーネルの initramfs ファイルが /boot/ ディレクトリーに生成されます。

```
~]# dracut
```

dracut で既存の initramfs を上書きするには、--force オプションを使用する必要があります（たとえば、initramfs が破損している場合）。それ以外の場合は、dracut は既存の initramfs ファイルの上書きを拒否します。

```
~]# dracut
Will not override existing initramfs (/boot/initramfs-2.6.32-22.el6.x86_64.img) without --
force
```

現在のディレクトリーに initramfs を作成するには、 dracut < initramfs_name& gt; <kernel_version > を呼び出します。

```
~]# dracut "initramfs-$(uname -r).img" $(uname -r)
```

事前に読み込む特定のカーネルモジュールを指定する必要がある場合には、/etc/dracut.conf 設定ファイルの add_dracutmodules="<module> [<more_modules>]" ディレクティブの括弧内にこれらのモジュールの名前（.koなどのファイル名のサフィックスを除く）を追加します。dracut で作成した initramfs イメージファイルの内容を一覧表示するには、lsinitrd < initramfs_file& gt; コマンドを使用します。

```
~]# lsinitrd initramfs-2.6.32-22.el6.x86_64.img
initramfs-2.6.32-22.el6.x86_64.img:
========================================================================
```

```
dracut-004-17.el6
==========================================================================

drwxr-xr-x  23 root     root          0 May  3 22:34 .
drwxr-xr-x   2 root     root          0 May  3 22:33 proc
-rwxr-xr-x   1 root     root       7575 Mar 25 19:53 init
drwxr-xr-x   7 root     root          0 May  3 22:34 etc
drwxr-xr-x   2 root     root          0 May  3 22:34 etc/modprobe.d
[output truncated]
```

オプションと使用方法についての詳しい情報は、man dracut および man dracut.conf を参照してください。

3. /boot/grub/ ディレクトリーの grub.conf 設定ファイルを検査して、起動しているカーネルバージョンの initrd initramfs-<kernel_version>.img が存在することを確認します。詳細は、「ブートローダーの確認」 を参照してください。

IBM eServer System i 上の初期 RAM ディスクイメージとカーネルの検証

IBM eServer System i マシンでは、初期 RAM ディスクと kernel ファイルが 1 つのファイルに統合され、これは addRamDisk コマンドで作成されます。この手順は、カーネルとその関連パッケージが Red Hatnbsp;Hat によって配布される RPM パッケージからインストールまたはアップグレードされると自動的に実行されます。そのため、手動で実行する必要はありません。作成したことを確認するには、ls -l /boot/ コマンドを使用して、/ boot/vmlinitrd- <kernel_version > ファイルがすでに存在していることを確認します（< kernel_version > はインストールしたカーネルのバージョンと一致する必要があります）。

30.6. ブートローダーの確認

rpm を使用してカーネルをインストールすると、カーネルパッケージはブートローダー設定ファイル内にその新しいカーネル用のエントリーを作成します。ただし、rpm は、新しいカーネルをデフォルトのカーネルとして起動するよう設定し ません。rpm で新しいカーネルをインストールする場合は、手動で実行する必要があります。

rpm で新しいカーネルをインストールした後に、設定が正しいことを確認することが推奨されます。そうでない場合は、システムは Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux を適切に起動できない場合があります。この場合は、先に作成したブートメディアでシステムを起動し、ブートローダーを再設定します。

システムのアーキテクチャーを探して、ブートローダーを判断し、「See」リンクをクリックしてシステムの正しい手順に移動します。

表30.1 アーキテクチャー別のブートローダー

アーキテクチャー	ブートローダー	参照
x86	GRUB	「GRUB ブートローダーの設定」
AMD AMD64 または Intel 64	GRUB	「GRUB ブートローダーの設定」
IBM eServer System i	OS/400	「OS/400 ブートローダーの設定」
IBM eServer System p	YABOOT	「YABOOT ブートローダーの設定」
IBM System z	z/IPL	

30.6.1. GRUB ブートローダーの設定

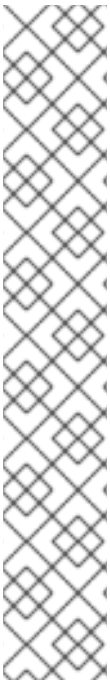
GRUB の設定ファイル /boot/grub/grub.conf には、default、timeout、splashimage、および hiddenmenu （最後のディレクティブには引数がありません）などのディレクティブを含む行がいくつか含まれています。残りのファイルには、4 行の スタンザ が含まれており、それぞれがインストール済みカーネルを参照します。これらのスタンザは、常に title エントリーで始まります。その後、関連付けられた root、kernel ディレクティブ、および initrd ディレクティブを常にインデントする必要があります。各スタンザが、同じスタンザの title 行のバージョン番号（括弧内）が含まれる kernel /vmlinuz-<version_number> で始まることを確認します。

例30.2 /boot/grub/grub.conf

```
# grub.conf generated by anaconda
[comments omitted]
default=1
timeout=0
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu

title Red Hat Enterprise Linux (2.6.32-22.el6.x86_64)
        root (hd0,0)
        kernel /vmlinuz-2.6.32-22.el6.x86_64 ro root=/dev/mapper/vg_vm6b-lv_root
rd_LVM_LV=vg_vm6b/lv_root rd_NO_LUKS rd_NO_MD rd_NO_DM LANG=en_US.UTF-8
SYSFONT=latarcyrheb-sun16 KEYBOARDTYPE=pc KEYTABLE=us rhgb quiet
crashkernel=auto
        initrd /initramfs-2.6.32-22.el6.x86_64.img

title Red Hat Enterprise Linux (2.6.32-19.el6.x86_64)
        root (hd0,0)
        kernel /vmlinuz-2.6.32-19.el6.x86_64 ro root=/dev/mapper/vg_vm6b-lv_root
rd_LVM_LV=vg_vm6b/lv_root rd_NO_LUKS rd_NO_MD rd_NO_DM LANG=en_US.UTF-8
SYSFONT=latarcyrheb-sun16 KEYBOARDTYPE=pc KEYTABLE=us rhgb quiet
crashkernel=auto
```

```
        initrd /initramfs-2.6.32-19.el6.x86_64.img

title Red Hat Enterprise Linux 6 (2.6.32-17.el6.x86_64)
        root (hd0,0)
        kernel /vmlinuz-2.6.32-17.el6.x86_64 ro root=/dev/mapper/vg_vm6b-lv_root
rd_LVM_LV=vg_vm6b/lv_root rd_NO_LUKS rd_NO_MD rd_NO_DM LANG=en_US.UTF-8
SYSFONT=latarcyrheb-sun16 KEYBOARDTYPE=pc KEYTABLE=us rhgb quiet
        initrd /initramfs-2.6.32-17.el6.x86_64.img
```

別の /boot/ パーティションを作成した場合、カーネルと initramfs イメージへのパスは /boot/ との相対パスになります。上記は 例30.2「/boot/grub/grub.conf」 で行います。したがって、最初のカーネルスタンザの initrd /initramfs-2.6.32-22.el6.x86_64.img 行は、root ファイルシステムがマウントされる際に initramfs イメージが実際に /boot/initramfs-2.6.32-22.el6.x86_64.img に置かれていることを意味します。また、grub.conf の各スタンザでカーネルパス（例： kernel /vmlinuz-2.6.32-22.el6.x86_64）に使用されます。



GRUB.CONF の INITRD ディレクティブは INITRAMFS イメージを参照します。

grub.conf のカーネルブートスタンザでは、initrd ディレクティブは、同じカーネルバージョンに対応する initramfs *ファイルの場所（別のパーティションにある場合は /boot/ ディレクトリーに対して相対的）を参照する必要があります。初期 RAM ディスクイメージを作成した以前のツール、mkinitrd が initrd と呼ばれるファイルを作成したため、この指示文は initrd と呼ばれます。したがって、grub.conf ディレクティブは、他のツールとの互換性を維持するために initrd のままです。初期 RAM ディスクイメージを作成するための dracut ユーティリティーを使用したシステムのファイル命名規則は、initramfs- <kernel_version > .img です。*

Dracut は、Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;LinuxRed Hat Enterprise Linux 6nbsp;6 で利用可能で、mkinitrd を通じて非常に多くの改良された新しいユーティリティーです。Dracut の使用方法は、「初期 RAM ディスクイメージの確認」 を参照してください。

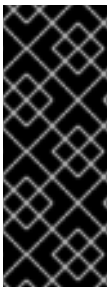
kernel /vmlinuz-<kernel_version> 行で指定されるカーネルバージョン番号が、各スタンザの initrd /initramfs-<kernel_version>.img 行に指定されている initramfs イメージのバージョン番号と一致している必要があります。詳細は、手順30.1「初期 RAM ディスクイメージの確認」 を参照してください。

default= ディレクティブは、デフォルトで 起動するカーネルが GRUB に指示します。grub.conf の各 title は起動可能なカーネルを表します。GRUB は、titleで始まる起動可能なカーネルを表す 0 d スタンザをカウントします。例30.2「/boot/grub/grub.conf」 では、default=1 の行は、GRUB がデフォルトで 2 番目 のカーネルエントリー（つまり title Red Hat Enterprise Linux (2.6.32-19.el6.x86_64) ）を起動することを示しています。

このため、例30.2「/boot/grub/grub.conf」 GRUB は、バージョン番号と比較すると、古いカーネルを起動するように設定されます。grub.conf の最初 の title エントリーである新しいカーネルを起動するには、default の値を 0 に変更する必要があります。

rpm で新しいカーネルをインストールしたら、/boot/grub/grub.conf が正しいことを確認し、default= の値を新しいカーネルに変更します（ 0からカウントするのを忘れている間は、コンピューターを新しいカーネルに再起動します）。ブートプロセスの出力を監視して、ハードウェアが検出されていることを確認します。

GRUB がエラーを表示し、デフォルトのカーネルで起動できない場合は、問題を修正できるように代替または古いカーネルでの起動を試みる方が簡単です。



GRUB 起動メニューを表示させる

grub.conf の timeout ディレクティブを 0 に設定すると、システムの起動時に GRUB が起動可能なカーネルの一覧を表示しません。システムの起動時にこの一覧を表示するには、BIOS 情報が表示された後に英数字キーを押して保持します。GRUB メニューが表示されます。

または、先に作成したブートメディアを使用してシステムを起動します。

30.6.2. ループデバイスの制限の設定

Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;LinuxRed Hat Enterprise Linux 6nbsp;6 のループバックデバイスの最大数は、max_loop カーネルオプションで設定されます。たとえば、ループバックデバイスの最大数を 64 に設定し、/etc/grub.conf ファイルを編集し、カーネル行の最後に max_loop=64 を追加します。/etc/grub.conf の行は以下のようになります。

```
kernel /vmlinuz-2.6.32-131.0.15.el6.x86_64 ro root=/dev/mapper/root rhgb quiet max_loop=64
initrd /initramfs-2.6.32-131.0.15.el6.x86_64.img
```

システムを再起動して、変更を有効にします。

デフォルトでは、8 つの /dev/loop* デバイス（/dev/loop0 から /dev/loop7への）が自動的に生成されますが、必要に応じて作成することもできます。たとえば、/dev/loop8 という名前の ninth ループデバイスを設定するには、root で以下のコマンドを実行します。

```
~]# mknod /dev/loop8 b 7 8
```

したがって、Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;LinuxRed Hat Enterprise Linux 6nbsp;6 カーネルを搭載したシステムの管理者は、必要な数のループバックデバイス

と、init スクリプト、または udev ルールを使って手動で作成できます。

ただし、システムの起動前に max_loop が設定されている場合、max_loop はループバックデバイスの数がハード制限になり、ループバックデバイスの数は制限を超えて動的に拡張できません。

30.6.3. OS/400 ブートローダーの設定

/boot/vmlinitrd- <kernel-version> ファイル は、カーネルのアップグレード時にインストールされます。ただし、dd コマンドを使用して、新しいカーネルを起動するように設定する必要があります。

1. root として、cat /proc/iSeries/mf/side コマンドを実行してデフォルトのサイドを確認します（A、B、または C のいずれか）。

2. root で、以下のコマンドを実行します。ここで、< kernel-version > は新しいカーネルのバージョンで、< side > は直前のコマンド側になります。

```
dd if=/boot/vmlinitrd-<kernel-version> of=/proc/iSeries/mf/<side>/vmlinux bs=8k
```

コンピューターを再起動して新しいカーネルのテストを開始し、メッセージを監視してハードウェアが適切に検出されるようにします。

30.6.4. YABOOT ブートローダーの設定

IBM eServer System p は、YABOOT をブートローダーとして使用します。YABOOT は、/etc/yaboot.conf を設定ファイルとして使用します。ファイルに先ほどインストールした image パッケージと同じバージョンの kernel セクションが含まれている（ initramfs イメージの場合と同様）ことを確認します。

```
boot=/dev/sda1 init-message=Welcome to RedRed Hat Enterprise Linuxnbsp;Hat
EnterpriseRed Hat Enterprise Linuxnbsp;Linux! Hit <TAB> for boot options
partition=2 timeout=30 install=/usr/lib/yaboot/yaboot delay=10 nonvram
image=/vmlinuz-2.6.32-17.EL
  label=old
  read-only
  initrd=/initramfs-2.6.32-17.EL.img
  append="root=LABEL=/"
image=/vmlinuz-2.6.32-19.EL
  label=linux
  read-only
  initrd=/initramfs-2.6.32-19.EL.img
  append="root=LABEL=/"
```

*デフォルトでは新しいカーネルが設定されていないことに注意してください。最初のイメージ内のカーネルは、デフォルトで起動します。デフォルトのカーネルを変更して、そのイメージスタンザが一覧表示された最初のものになるように変更するか、ディレクティブ **default** を追加して、新しいカーネルを含むイメージスタンザの **label** に設定します。*

コンピューターを再起動して新しいカーネルのテストを開始し、メッセージを監視してハードウェアが適切に検出されるようにします。

第31章 カーネルモジュールの使用

Linux カーネルはモジュールであり、動的に読み込まれる カーネルモジュール を使用してその機能を拡張できます。カーネルモジュールは、以下のものを提供できます。

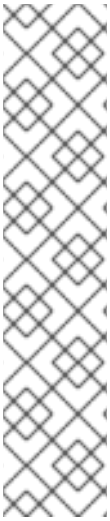
- 新しいハードウェアへのサポートを強化するデバイスドライバー
- btrfs や NFS などのファイルシステムのサポート

カーネル自体と同様に、モジュールは動作をカスタマイズするパラメーターを取ることができます。ただし、デフォルトのパラメーターでほとんどのケースで十分に機能します。また、利用可能なパラメーターに使用できる全モジュールやモジュール固有の情報をクエリーできます。さらには、実行中のカーネルに対してモジュールを動的にロード/アンロード (削除) することも可能です。module-init-tools パッケージが提供するユーティリティーの多くは、操作の実行時にモジュールの依存関係を考慮するため、手動による依存関係の追跡が必要になることはほとんどありません。

最新のシステムでは、条件を呼び出す際に、さまざまなメカニズムによりカーネルモジュールが自動的に読み込まれます。ただし、モジュールがオプションの機能を提供する場合など、モジュールを手動でロードまたはアンロードする必要がある場合、特定のモジュールは基本的な機能を提供する場合もありますが、モジュールがその他の状況で動作しない場合もあります。

本章では、以下を行う方法を説明します。

- ユーザー空間の module-init-tools パッケージを使用して、カーネルモジュールとその依存関係を表示、クエリー、読み込み、アンロードします。
- モジュールパラメーターをコマンドラインで動的に設定し、カーネルモジュールの動作をカスタマイズできるようにします。
- 起動時にモジュールをロードします。



MODULE-INIT-TOOLS パッケージのインストール

本章で説明するカーネルモジュールユーティリティーを使用するには、最初に root で以下を実行して module-init-tools パッケージがインストールされていることを確認します。

```
~]# yum install module-init-tools
```

yum を使用したパッケージのインストールは 「パッケージのインストール」 を参照してください。

31.1. 現在ロードされているモジュールの一覧表示

lsmod コマンドを実行して、現在カーネルに読み込み済みの全カーネルモジュールを一覧表示できます。

```
~]$ lsmod
Module                  Size  Used by
xfs                   803635  1
exportfs                3424  1 xfs
vfat                    8216  1
fat                    43410  1 vfat
tun                    13014  2
fuse                   54749  2
ip6table_filter         2743  0
ip6_tables             16558  1 ip6table_filter
ebtable_nat             1895  0
ebtables               15186  1 ebtable_nat
ipt_MASQUERADE          2208  6
iptable_nat             5420  1
nf_nat                 19059  2 ipt_MASQUERADE,iptable_nat
rfcomm                 65122  4
ipv6                  267017  33
sco                    16204  2
bridge                 45753  0
stp                     1887  1 bridge
llc                     4557  2 bridge,stp
bnep                   15121  2
l2cap                  45185  16 rfcomm,bnep
cpufreq_ondemand        8420  2
acpi_cpufreq            7493  1
freq_table              3851  2 cpufreq_ondemand,acpi_cpufreq
usb_storage            44536  1
sha256_generic         10023  2
aes_x86_64              7654  5
aes_generic            27012  1 aes_x86_64
cbc                     2793  1
dm_crypt               10930  1
```

```
kvm_intel              40311  0
kvm                   253162  1 kvm_intel
[output truncated]
```

lsmod 出力の各行は、以下を指定します。

- メモリーに現在読み込まれているカーネルモジュールの名前
- 使用するメモリー量（および）
- モジュールを使用しているプロセスの合計と、それに依存する他のモジュールがある場合は、そのモジュールの名前の一覧が続きます。この一覧を使用して、アンロードしたいモジュールに依存しているモジュールすべてを最初にアンロードできます。詳細は、「モジュールのアンロード」 を参照してください。

最後に、lsmod 出力は /proc/modules 擬似ファイルの内容ほど詳細ではないので、はるかに読み取りやすくなっていることに注意してください。

31.2. モジュール情報の表示

カーネルモジュールの詳細情報を表示するには、modinfo < module_name& gt; コマンドを実行します。

モジュール名が .KO で終了しない

カーネルモジュール名を module-init-tools ユーティリティーのいずれかの引数として入力する場合は、名前の末尾に拡張子 .ko を付けないでください。カーネルモジュール名には拡張子はありません。対応するファイルには拡張子があります。

たとえば、Intel PRO/1000 ネットワークドライバーである e1000e モジュールに関する情報を表示するには、以下のコマンドを実行します。

例31.1 lsmod を使用したカーネルモジュール情報の一覧表示

```
~]# modinfo e1000e
filename:       /lib/modules/2.6.32-71.el6.x86_64/kernel/drivers/net/e1000e/e1000e.ko
version:        1.2.7-k2
license:        GPL
description:    Intel(R) PRO/1000 Network Driver
```

```
author:         Intel Corporation, <linux.nics@intel.com>
srcversion:     93CB73D3995B501872B2982
alias:          pci:v00008086d00001503sv*sd*bc*sc*i*
alias:          pci:v00008086d00001502sv*sd*bc*sc*i*
[some alias lines omitted]
alias:          pci:v00008086d0000105Esv*sd*bc*sc*i*
depends:
vermagic:       2.6.32-71.el6.x86_64 SMP mod_unload modversions
parm:           copybreak:Maximum size of packet that is copied to a new buffer on receive
(uint)
parm:           TxIntDelay:Transmit Interrupt Delay (array of int)
parm:           TxAbsIntDelay:Transmit Absolute Interrupt Delay (array of int)
parm:           RxIntDelay:Receive Interrupt Delay (array of int)
parm:           RxAbsIntDelay:Receive Absolute Interrupt Delay (array of int)
parm:           InterruptThrottleRate:Interrupt Throttling Rate (array of int)
parm:           IntMode:Interrupt Mode (array of int)
parm:           SmartPowerDownEnable:Enable PHY smart power down (array of int)
parm:           KumeranLockLoss:Enable Kumeran lock loss workaround (array of int)
parm:           WriteProtectNVM:Write-protect NVM [WARNING: disabling this can lead to
corrupted NVM] (array of int)
parm:           CrcStripping:Enable CRC Stripping, disable if your BMC needs the CRC (array
of int)
parm:           EEE:Enable/disable on parts that support the feature (array of int)
```

以下は、modinfo 出力のフィールドの一部を表しています。

filename

.ko カーネルオブジェクトファイルへの絶対パス。modinfo -n を、filename フィールドのみを出力するショートカットコマンドとして使用できます。

description

モジュールの簡単な説明。modinfo -d を、説明フィールドのみを出力するショートカットコマンドとして使用できます。

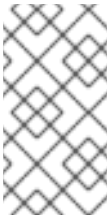
alias

alias フィールドは、モジュールのエイリアスが存在する回数だけ表示され、存在しない場合は完全に省略されます。

依存

このフィールドには、このモジュールが依存するすべてのモジュールのコンマ区切りリストが

含まれます。



DEPENDS フィールドの削除

モジュールに依存関係がない場合、depends フィールドは出力から省略される可能性があります。

parm

各 parm フィールドは、parameter_name:description の形式でモジュールパラメーターを 1 つ表示します。ここでは以下のようになります。

- PARAMETER _NAME は、コマンドラインでモジュールパラメーターとして使用する場合や、/etc/modprobe.d/ ディレクトリーの .conf ファイルのオプション行に使用すべき構文です。

- description は、パラメーターの動作を簡単に説明し、パラメーターが受け入れる値のタイプ（ int、unit、または array of intなど）を括弧で示すことが想定されています。

-p オプションを使用すると、モジュールがサポートするパラメーターの一覧を表示できます。ただし、便利な値タイプの情報は modinfo -p の出力から省略されるため、以下を実行すると便利です。

例31.2 モジュールパラメーターの一覧表示

```
~]# modinfo e1000e | grep "^parm" | sort
parm:           copybreak:Maximum size of packet that is copied to a new buffer on receive (uint)
parm:           CrcStripping:Enable CRC Stripping, disable if your BMC needs the CRC (array of int)
parm:           EEE:Enable/disable on parts that support the feature (array of int)
parm:           InterruptThrottleRate:Interrupt Throttling Rate (array of int)
parm:           IntMode:Interrupt Mode (array of int)
parm:           KumeranLockLoss:Enable Kumeran lock loss workaround (array of int)
parm:           RxAbsIntDelay:Receive Absolute Interrupt Delay (array of int)
parm:           RxIntDelay:Receive Interrupt Delay (array of int)
parm:           SmartPowerDownEnable:Enable PHY smart power down (array of int)
parm:           TxAbsIntDelay:Transmit Absolute Interrupt Delay (array of int)
parm:           TxIntDelay:Transmit Interrupt Delay (array of int)
parm:           WriteProtectNVM:Write-protect NVM [WARNING: disabling this can lead to corrupted NVM] (array of int)
```

31.3. モジュールの読み込み

カーネルモジュールを読み込むには、root で modprobe < module_name> コマンドを実行します。たとえば、wacom モジュールを読み込むには、以下のコマンドを実行します。

```
~]# modprobe wacom
```

デフォルトでは、modprobe は /lib/modules/ <kernel_version> /kernel/drivers/ ディレクトリーからモジュールを読み込もうとします。このディレクトリーでは、各タイプのモジュールには、ネットワークおよび SCSI インターフェースドライバー用の net/ や scsi/ などの独自のサブディレクトリーがあります。

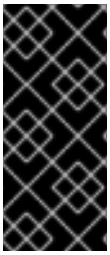
一部のモジュールには依存関係があります。これは、問題のモジュールを読み込む前に読み込む必要がある他のカーネルモジュールです。モジュール依存関係の一覧は、カーネルまたはドライバーパッケージがインストールされたときに自動的に実行される depmod プログラムにより生成および維持されます。depmod プログラムは、依存関係のリストを /lib/modules/<kernel_version>/modules.dep ファイルに保持します。modprobe コマンドは、操作の実行時に常に modules.dep ファイルを読み取ります。modprobe に特定のカーネルモジュールを読み込むよう指示すると、まずそのモジュールの依存関係（ある場合）を調べ、カーネルに読み込まれていない場合は読み込みます。modprobe は依存関係を再帰的に解決します。必要に応じて、依存関係のすべての依存関係を読み込むため、すべての依存関係が常に満たされるようにします。

-v （または --verbose）オプションを使用すると、modprobe が、モジュール依存関係のロードなど、実行内容に関する詳細情報を表示できます。以下は、イーサネットモジュールでファイバーチャネル の詳細を読み込む例です。

例31.3 modprobe -v は、読み込む際にモジュール依存関係を表示します。

```
~]# modprobe -v fcoe
insmod /lib/modules/2.6.32-71.el6.x86_64/kernel/drivers/scsi/scsi_tgt.ko
insmod /lib/modules/2.6.32-71.el6.x86_64/kernel/drivers/scsi/scsi_transport_fc.ko
insmod /lib/modules/2.6.32-71.el6.x86_64/kernel/drivers/scsi/libfc/libfc.ko
insmod /lib/modules/2.6.32-71.el6.x86_64/kernel/drivers/scsi/fcoe/libfcoe.ko
insmod /lib/modules/2.6.32-71.el6.x86_64/kernel/drivers/scsi/fcoe/fcoe.ko
```

この例では、modprobe が、fcoe を読み込む前に、scsi_tgt モジュール、scsi_transport_fc、libfc モジュール、libfcoe モジュールを依存関係としてロードしていることを示しています。また、modprobe は、モジュールを実行中のカーネルに挿入するために、より 「プリミティブ」 insmod コマンドを使用していることに注意してください。



INSMOD の代わりに MODPROBE を常に使用します。

insmod コマンドを使用してカーネルモジュールを読み込むこともできますが、依存関係は解決されません。このため、代わりに modprobe を使用してモジュールを常に読み込む必要があります。

31.4. モジュールのアンロード

root で modprobe -r <module_name> を実行すると、カーネルモジュール をアンロードできます。たとえば、wacom モジュールがすでにカーネルに読み込まれていると仮定して、以下のコマンドを実行してアンロードできます。

```
~]# modprobe -r wacom
```

ただし、プロセスが次のように使用されていると、このコマンドは失敗します。

- **wacom モジュール**
- **wacom が直接依存するモジュール、または**
- **wacom- 依存関係ツリー経由のモジュールは、間接的に依存します。**

lsmod を使用して特定のモジュールをアンロードしないようにモジュールの名前を取得する方法は、「現在ロードされているモジュールの一覧表示」 を参照してください。

たとえば、firewire_ohci モジュールをアンロードしたい場合（たとえば、システムの安定性に影響を与えるバグがあると思われるため）、ターミナルセッションは以下のようになります。

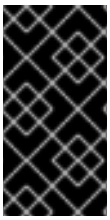
```
~]# modinfo -F depends firewire_ohci
depends:        firewire-core
~]# modinfo -F depends firewire_core
depends:        crc-itu-t
~]# modinfo -F depends crc-itu-t
depends:
```

読み込んだ Firewire モジュールの依存関係ツリー（この例ではブランチではない）を調べている。Firewire _ohci は firewire_ core に依存します。これは、それが crc-itu-t に依存します。

modprobe -v -r < module_name> コマンドを使用して firewire_ ohci をアンロードできます。-r は --remove では短縮され、--verbose の場合は -v になります。

```
~]# modprobe -r -v firewire_ohci
rmmod /lib/modules/2.6.32-71.el6.x86_64/kernel/drivers/firewire/firewire-ohci.ko
rmmod /lib/modules/2.6.32-71.el6.x86_64/kernel/drivers/firewire/firewire-core.ko
rmmod /lib/modules/2.6.32-71.el6.x86_64/kernel/lib/crc-itu-t.ko
```

この出力は、モジュールが読み込まれる逆順でアンロードされることを示しています。ただし、アンロードされるモジュールはどれもプロセスに依存しないためです。



RMMOD を直接使用しないでください！

rmmod コマンドを使用してカーネルモジュールをアンロードできますが、代わりに modprobe -r を使用することが推奨されます。

31.5. モジュールのブラックリスト登録

さまざまなパフォーマンスやセキュリティー上の理由から、システムが特定のカーネルモジュールを使用しないようにする必要がある場合があります。これは、モジュールブラックリスト （ modprobe ユーティリティーが使用するメカニズムを使用して、カーネルが特定のモジュールを自動的に読み込みできないか、モジュールがまったくロードできない）で実現できます。これは、特定のモジュールを使用している場合など、システムに対してセキュリティーリスクが発生したり、モジュールが別のモジュールと同じハードウェアまたはサービスを制御し、両方のモジュールを読み込むと、システムやそのコンポーネントの読み込みによってシステムが不安定になったり、動作しなくなったりするなどの特定の状況で役に立ちます。

モジュールをブラックリストに指定するには、root で /etc/modprobe.d/ ディレクトリーの指定の設定ファイルに以下の行を追加します。

```
blacklist <module_name>
```

&lt ;module_name> は、ブラックリストに指定されたモジュールの名前です。

デフォルトでシステムにすでに存在している /etc/modprobe.d/blacklist.conf ファイルを変更できます。ただし、特定のカーネルモジュールにのみ特定の設定が含まれる別の設定ファイル /etc/modprobe.d/ <module_name > .conf を作成することが推奨されます。

例31.4 /etc/modprobe.d/blacklist.conf の例

```
#
```

```
# Listing a module here prevents the hotplug scripts from loading it.
# Usually that'd be so that some other driver will bind it instead,
# no matter which driver happens to get probed first.  Sometimes user
# mode tools can also control driver binding.
#
# Syntax: see modprobe.conf(5).
#

# watchdog drivers
blacklist i8xx_tco

# framebuffer drivers
blacklist aty128fb
blacklist atyfb
blacklist radeonfb
blacklist i810fb
blacklist cirrusfb
blacklist intelfb
blacklist kyrofb
blacklist i2c-matroxfb
blacklist hgafb
blacklist nvidiafb
blacklist rivafb
blacklist savagefb
blacklist sstfb
blacklist neofb
blacklist tridentfb
blacklist tdfxfb
blacklist virgefb
blacklist vga16fb
blacklist viafb

# ISDN - see bugs 154799, 159068
blacklist hisax
blacklist hisax_fcpcipnp

# sound drivers
blacklist snd-pcsp

# I/O dynamic configuration support for s390x (bz #563228)
blacklist chsc_sch
```

ただし、blacklist <module_name > コマンドは、モジュールが手動でロードされないようにしたり、ブラックリスト化されていない別のカーネルモジュールの依存関係としてロードされたりしません。システムでモジュールを読み込めないようにするには、/etc/modprobe.d/ ディレクトリーの指定の設定ファイルを root として修正します。

```
install <module_name> /bin/true
```

ここで、<module_name& gt; はブラックリストに指定されたモジュールの名前です。

例31.5 モジュールブラックリストを一時的な問題のソリューションとして使用する方法

Linux カーネルの PPP over L2TP モジュール(pppol2pt)の不具合が確認され、この不具合がシステム侵害に誤用される可能性があります。システムで pppol2pt モジュールを機能させる必要がある場合は、この問題が修正されるまで pppol2pt を完全にブラックリストに指定します。

1. 以下のコマンドを実行して、カーネルに pppol2pt が現在読み込まれているかどうかを確認します。

```
~]# lsmod | grep ^pppol2tp && echo "The module is loaded" || echo "The module is not loaded"
```

2. モジュールがロードされた場合は、誤った使用を防ぐために、モジュールとそのすべての依存関係をアンロードする必要があります。安全にアンロードする方法は、「モジュールのアンロード」 を参照してください。

3. 以下のコマンドを実行して、pppol2pt をカーネルに読み込みできないようにします。

```
~]# echo "install pppol2tp /bin/true" > /etc/modprobe.d/pppol2tp.conf
```

このコマンドは、システムに /etc/modprobe.d/pppol2tp.conf ファイルの内容を上書きすることに注意してください。このコマンドを実行する前に、既存の pppol2tp.conf を確認してバックアップします。また、モジュールをアンロードできない場合は、このコマンドを有効にするためにシステムを再起動する必要があります。

pppol2pt モジュールの問題が正しく修正されたら、/etc/modprobe.d/pppol2tp.conf ファイルを削除するか、以前のコンテンツを復元できます。これにより、システムが元の設定で pppol2pt モジュールをロードできるようになります。

カーネルモジュールをブラックリストに登録する場合には注意してください。

カーネルモジュールをブラックリストに登録する前に、現在のシステム設定が正しく機能するようにモジュールが必ず重要ではないことを確認してください。主要なカーネルモジュールを正しくブラックリストに指定しないと、システムが不安定になったり、システムが動作しなくなったりすることがあります。

31.6. モジュールパラメーターの設定

また、カーネル自体と同様に、モジュールは動作を変更するパラメーターを取ることもできます。多くの場合、デフォルトのパラメーターは適切に機能しますが、モジュール用のカスタムパラメーターを設定する必要がある場合があります。実行中のカーネルにすでにロードされているモジュールにはパラメーターを動的に設定できないため、設定方法は 2 つあります。

1. コマンドラインで modprobe コマンドを実行し、カスタマイズされたパラメーターの一覧を実行して、カーネルモジュールを読み込みます。モジュールがすでにロードされている場合は、modprobe -r コマンドを使用して、最初にすべての依存関係とモジュール自体をアンロードする必要があります。この方法では、変更を永続化せずに、特定の設定でカーネルモジュールを実行できます。詳細は、「カスタマイズされたモジュールの読み込み： 一時的な変更」 を参照してください。

2. または、/etc/modprobe.d/ ディレクトリーにある、既存のファイルまたは新たに作成されたファイルで、カスタマイズしたパラメーターの一覧を指定します。この方法では、再起動または modprobe コマンドごとに、モジュールが読み込まれるたびに指定のパラメーターを設定すると、モジュールのカスタマイズが永続化されます。詳細は、「カスタマイズされたモジュールの読み込み - 永続的な変更」 を参照してください。

31.6.1. カスタマイズされたモジュールの読み込み： 一時的な変更

特定の設定でカーネルモジュールを一時的に実行するのに役に立つ場合もあります。現在のシステムセッション用にカスタマイズしたパラメーターでカーネルモジュールを読み込むか、モジュールが別のパラメーターでリロードされるまでは、root で以下の形式の modprobe を実行します。

```
~]# modprobe <module_name> [parameter=value]
```

[parameter=value] は、そのモジュールで利用可能なカスタマイズされたパラメーターのリストを表します。コマンドラインでカスタムパラメーターを使用してモジュールを読み込む場合は、以下の点に注意してください。

- 複数のパラメーターおよび値を入力するには、それらをスペースで区切ります。

- モジュールパラメーターによっては、コンマ区切りの値の一覧を引数として要求するものもあります。値の一覧を入力する際には、コンマごとにスペースを挿入し ない でください。または、modprobe は、以下のスペースの値を追加のパラメーターとして誤って解釈します。

- モジュールが正常に読み込まれた場合、また はモジュールがすでに カーネルに読み込まれている場合は、modprobe コマンドは、終了ステータス 0 で警告なしで成功します。したがって、カスタムパラメーターで読み込む前に、モジュールがまだロードされていないことを確認する必要があります。modprobe コマンドは、モジュールを自動的に再読み込みしません。もしくは、すでに読み込まれていることを警告します。

以下の手順は、たとえば、Intel PRO/1000 ネットワークアダプターのネットワークドライバーである e1000e モジュールのカスタムパラメーターを使用してカーネルモジュールを読み込むのに推奨される手順を説明します。

手順31.1 カスタムパラメーターを使用したカーネルモジュールの読み込み

1. 以下のコマンドを実行して、モジュールがまだカーネルに読み込まれていないことを確認します。

```
~]# lsmod|grep e1000e
e1000e              236338  0
ptp                 9614  1 e1000e
```

この例のコマンドの出力は、e1000e モジュールがすでにカーネルに読み込まれていることを示しています。また、このモジュールには ptp モジュールの依存関係が 1 つあることも示されています。

2. モジュールがすでにカーネルに読み込まれている場合は、次のステップに進む前にモジュールとそのすべての依存関係をアンロードする必要があります。安全にアンロードする方法は、「モジュールのアンロード」 を参照してください。

3. モジュールを読み込み、モジュール名の後にすべてのカスタムパラメーターを一覧表示します。たとえば、割り込みスロットルレートで Intel PRO/1000 ネットワークドライバーを読み込む場合は、ドライバーの 1 番目、秒、および 3 番目のインスタンスに対して、割り込みスロットルを 3000 割り込みに設定し、EEE(EEE)がオンになります。[5]root で以下のコマンドを実行します。

```
~]# modprobe e1000e InterruptThrottleRate=3000,3000,3000 EEE=1
```

この例では、コンマで複数の値を区切り、それらの間のスペースを省略して、複数の値を 1 つのパラメーターに渡す方法を示しています。

31.6.2. カスタマイズされたモジュールの読み込み - 永続的な変更

カーネルモジュールが常に特定の設定で読み込まれるようにするには、/etc/modprobe.d/ ディレクトリーの既存ファイルまたは新たに作成されたファイルを、以下の形式の行で修正します。

```
~]# options <module_name> [parameter=value]
```

[parameter=value] は、そのモジュールで利用可能なカスタマイズされたパラメーターのリストを表します。

以下の手順では、ワイヤレスネットワーク用に Open Firmware 用の b43 モジュールにカスタムパラメーターを使用してカーネルモジュールを読み込む手順を説明します。これにより、モジュールの再読み込み間で変更が永続化されるようになります。

手順31.2 カスタムパラメーターを使用したカーネルモジュールの読み込み - 永続的な変更

1. *以下の行を /etc/modprobe.d/openfwwf.conf ファイルに追加します。これにより、b43 モジュールは常に QoS で読み込まれ、ハードウェアアクセラレーション暗号が無効になります。*

```
options b43 nohwcrypt=1 qos=0
```

2. *以下のコマンドを実行して、モジュールがまだカーネルに読み込まれていないことを確認します。*

```
~]# lsmod|grep ^b43
~]#
```

この例のコマンドの出力は、モジュールが現在カーネルに読み込まれていないことを示しています。

3. *モジュールがすでにカーネルに読み込まれている場合は、次のステップに進む前にモジュールとそのすべての依存関係をアンロードする必要があります。安全にアンロードする方法は、「モジュールのアンロード」 を参照してください。*

4. *以下のコマンドを実行して b43 モジュールを読み込みます。*

```
~]# modprobe b43
```

31.7. 永続的なモジュールの読み込み

例31.1「lsmod を使用したカーネルモジュール情報の一覧表示」 で示されているように、多くのカーネルモジュールは、システムの起動時に自動的に読み込まれます。読み込む追加のモジュールを指定するには、/etc/sysconfig/modules/ ディレクトリーに新しい < file_name > .modules ファイルを作成します。ここで、< file_name > は任意の説明的な名前です。&lt ;file_name > .modules ファイルは

システム起動スクリプトがシェルスクリプトとして扱われ、そのためには最初の行として インタープリターディレクティブ（「bang 行」とも呼ばれます）で開始する必要があります。

例31.6 file_name.modules ファイルの最初の行

```
#!/bin/sh
```

さらに、< file_name> .modules ファイルは実行可能である必要があります。以下を実行して実行可能にします。

```
modules]# chmod +x <file_name>.modules
```

たとえば、以下の bluez-uinput.modules スクリプトは uinput モジュールを読み込みます。

例31.7 /etc/sysconfig/modules/bluez-uinput.modules

```
#!/bin/sh

if [ ! -c /dev/input/uinput ] ; then
        exec /sbin/modprobe uinput >/dev/null 2>&1
fi
```

3 行目の if-conditional ステートメントは、/dev/input/uinput ファイルが存在し ない ことを確認します（この条件は ! 記号を否定）し、その場合は exec /sbin/modprobe uinput を呼び出すことで uinput モジュールをロードします。uinput モジュールは /dev/input/uinput ファイルを作成するため、そのファイルが存在するかどうかをテストし、uinput モジュールがカーネルに読み込まれているかどうかを検証します。

その行の最後 にある以下の >/dev/null 2>&1 句は、その行の最後で出力を /dev/null にリダイレクトすることで、modprobe コマンドが quiet のままになるようにします。

31.8. 特定のカーネルモジュール機能

本セクションでは、さまざまなカーネルモジュールを使用して特定のカーネル機能を有効にする方法を説明します。

31.8.1. チャンネルボンディングの使用

Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux を使用すると、管理者は ボンディングカーネルモジュールと、チャンネルボンディング インターフェースと呼ばれる特別なネット

ワークインターフェースを使用して、NIC を 1 つの チャンネル にバインドできます。このチャンネルボンディングにより、複数のネットワークインターフェースが 1 つとして機能できるようになり、また同時に帯域幅が増加し、冗長性を提供します。

複数のネットワークインターフェースをボンディングにチャネルするには、管理者は以下の手順を実行する必要があります。

1. 「チャンネルボンディングインターフェース」 の説明に従って、チャネルボンディングインターフェースを設定します。

2. パフォーマンスを強化するには、利用可能なモジュールオプションを調節して、最適な組み合わせを確認します。特に miimon、arp_interval、arp_ip_target パラメーターに注意してください。利用可能なオプション一覧と使用しているボンディングされたインターフェースに最適なオプションを迅速に決定する方法については、「ボンディングモジュールのディレクティブ」を参照してください。

31.8.1.1. ボンディングモジュールのディレクティブ

ボンディングインターフェース設定ファイル（例 : ifcfg-bond0 ）で BONDING_OPTS="<bonding parameters>" ディレクティブに追加する前に、ボンディングインターフェースにどのチャンネルボンディングモジュールパラメーターが最適であるかをテストすることが推奨されます。ボンディングされたインターフェースのパラメーターは、sysfs ファイルシステム内のファイルを操作することで、ボンディングモジュールをアンロード (およびリロード) することなく設定できます。

sysfs は、カーネルオブジェクトをディレクトリー、ファイル、シンボリックリンクとして表す仮想ファイルシステムです。sysfs を使用すると、カーネルオブジェクトに関する情報のクエリーや、通常のファイルシステムコマンドを使用することでこれらのオブジェクトを操作することもできます。sysfs 仮想ファイルシステムには /etc/fstab に行があり、/sys/ ディレクトリー下にマウントされます。ボンディングインターフェースはすべて、/sys/class/net/ ディレクトリー下にあるファイルと対話したり、これらを操作することで動的に設定できます。

ボンディングインターフェースに最適なパラメーターを決定するには、「チャンネルボンディングインターフェース」 の説明に従って ifcfg-bond0 などのチャンネルボンディングインターフェースファイルを作成します。ボンディングされている各インターフェースの設定ファイルに SLAVE=yes ディレクティブおよび MASTER=bond0 ディレクティブを bond0 に挿入します。これが完了すると、パラメーターの確認に進むことができます。

まず、ifconfig bond < N>upas root を実行して、作成したボンディング を起動します。

```
~]# ifconfig bond0 up
```

ifcfg-bond0 ボンディングインターフェースファイルを正しく作成している場合は、ifconfig の出力に bond0 がリストされているはずです（オプションはありません）。

```
~]# ifconfig
bond0     Link encap:Ethernet HWaddr 00:00:00:00:00:00
          UP BROADCAST RUNNING MASTER MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
eth0      Link encap:Ethernet  HWaddr 52:54:00:26:9E:F1
          inet addr:192.168.122.251  Bcast:192.168.122.255  Mask:255.255.255.0
          inet6 addr: fe80::5054:ff:fe26:9ef1/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:207 errors:0 dropped:0 overruns:0 frame:0
          TX packets:205 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:70374 (68.7 KiB)  TX bytes:25298 (24.7 KiB)
[output truncated]
```

アップになっていないボンドも含めてすべての既存のボンドを表示するには、以下を実行します。

```
~]# cat /sys/class/net/bonding_masters
bond0
```

/sys/class/net/bond <N> /bonding/ ディレクトリーにあるファイルを操作することで、各ボンディングを 個別に設定できます。まず、設定するボンドをダウンにします。

```
~]# ifconfig bond0 down
```

たとえば、1 秒間隔で bond0 で MII 監視を有効にするには、（root で）実行できます。

```
~]# echo 1000 > /sys/class/net/bond0/bonding/miimon
```

balance-alb モードで bond0 を設定するには、以下のいずれかを実行します。

```
~]# echo 6 > /sys/class/net/bond0/bonding/mode
```

またはモード名を使用します。

```
~]# echo balance-alb > /sys/class/net/bond0/bonding/mode
```

該当するボンディングのオプションを設定したら、ifconfig bond <N>upを実行してテストすることができ ます。オプションを変更する場合はインターフェースを停止して、sysfs を使用してそのパラメーターを修正後、有効に戻して再確認します。

ボンディングに最適なパラメーターセットを決定したら、これらのパラメーターを空白区切りの一覧として、設定するボンディングインターフェースの /etc/sysconfig/network-scripts/ifcfg-bond <N> ファイルの BONDING_OPTS= ディレクティブに追加します。ボンドが有効な場合 (たとえば、ONBOOT=yes ディレクティブが設定されていて、ブートシーケンス中にシステムがボンドを有効にする場合) はいつでも、BONDING_OPTS で指定されているボンディングオプションはそのボンドに対して有効となります。ボンディングインターフェース（および BONDING_OPTS）の設定に関する詳細は、「チャンネルボンディングインターフェース」 を参照してください。

以下の一覧は、より一般的なチャンネルボンディングパラメーターの多くの名前と、その機能を説明します。詳細は、modinfo ボンディング 出力の各 parm について簡単な説明、または kernel-doc パッケージの bonding.txt ファイルの詳細の説明を参照してください（「その他のリソース」を参照してください）。

ボンディングインターフェースパラメーター

arp_interval=<time_in_milliseconds>

ARP モニタリングが発生する頻度（ミリ秒単位）を指定します。この設定を設定する場合は、このパラメーターの開始点は 1000 になります。

必要なパラメーターをすべて指定するようにしてください。

arp_interval および arp_ip_target の両パラメーター、あるいは miimon パラメーターの指定は不可欠です。指定されないと、リンクが失敗した場合にネットワークパフォーマンスが低下する恐れがあります。

mode=0 または mode=2 (2 つの負荷分散モード) でこの設定を使用する場合、ネットワークスイッチは NIC 全体に均等にパケットを分散するよう設定する必要があります。これを実現する方法は、kernel-doc パッケージの bonding.txt ファイル（「その他のリソース」を参照）を参照してください。

デフォルトでは値は 0 に設定されており、ARP 監視を無効にします。

arp_ip_target=<ip_address>[,<ip_address_2>,...<ip_address_16>]

arp_interval パラメーターが有効になっていると、ARP 要求のターゲット IP アドレスを指定

します。最大 16 の IP アドレスは、コンマ区切りの一覧で指定できます。

arp_validate=<value>

ARP プローブのソース/ディストリビューションを検証します。デフォルトは none です。他の有効な値は、active、backup、および all です。

downdelay=<time_in_milliseconds>

リンクを無効にする前に、リンクの失敗後に待機する時間を指定します (ミリ秒単位)。値は、miimon パラメーターで指定される値の倍数でなければなりません。デフォルトでは値は 0 に設定されており、ARP 監視を無効にします。

lacp_rate=<value>

リンクパートナーが 802.3ad モードで LACPDU パケットを送信するレートを指定します。以下の値が使用できます。

- slow または 0: デフォルト設定。パートナーが 30 秒ごとに LACPDU を送信するよう指定します。
- fast または 1: パートナーが LACPDU を 1 秒ごとに送信するように指定します。

miimon=<time_in_milliseconds>

MII リンク監視が発生する頻度を指定します (ミリ秒単位)。MII は NIC がアクティブであることを検証するために使用されるため、これは高可用性が必要な場合に役立ちます。特定の NIC のドライバーが MII ツールに対応していることを確認するには、root で以下のコマンドを入力します。

```
~]# ethtool <interface_name> | grep "Link detected:"
```

このコマンドで、&lt ;interface_name> を、ボンドインターフェースではなく eth0 などのデバイスインターフェースの名前に置き換えます。MII が対応している場合は、コマンドは以下を返します。

```
Link detected: yes
```

高可用性のためにボンディングされたインターフェースを使用する場合、各 NIC のモジュールは MII に対応していなければなりません。値を 0 (デフォルト) に設定すると、この機能はオフになります。この設定を設定する際に、このパラメーターのスタート地点は 100 になります。

必要なパラメーターをすべて指定するようにしてください。

arp_interval および arp_ip_target の両パラメーター、あるいは miimon パラメーターの指定は不可欠です。指定されないと、リンクが失敗した場合にネットワークパフォーマンスが低下する恐れがあります。

mode=<value>

ボンディングポリシーの指定が可能になります。< value > は、以下のいずれかになります。

- balance-rr または 0: 耐障害性とロードバランシングにラウンドロビンポリシーを設定します。利用可能な最初のインターフェースからそれぞれのボンディングされたスレーブインターフェースで送受信が順次行われます。
- active-backup または 1: 耐障害性のためアクティブなバックアップポリシーを設定します。ボンディングインターフェースの中で最初に利用可能になったものから送受信が行われます。別のボンディングされたスレーブインターフェースは、アクティブなボンディングされたスレーブインターフェースが失敗した場合にのみ使用されます。
- balance-xor または 2: フォールトトレランスおよび負荷分散用の XOR（排他的な or）ポリシーを設定します。この方法を使用すると、インターフェースは受信要求の MAC アドレスとスレーブ NIC のいずれかの MAC アドレスが一致します。このリンクが確立されると、最初の利用可能なインターフェースから順番に送信が送信されます。
- broadcast または 3: 耐障害性にブロードキャストポリシーを設定します。すべての送信は、すべてのスレーブインターフェースで行われます。
- 802.3ad または 4: IEEE 802.3ad 動的リンクアグリゲーションのポリシーを設定します。同一の速度とデュプレックス設定を共有するアグリゲーショングループを作成します。アクティブなアグリゲーターのすべてのスレーブで送受信を行います。802.3ad に対応するスイッチが必要です。
- balance-tlb または 5: 耐障害性とロードバランシングのための送信ロードバランシング (TLB) ポリシーを設定します。発信トラフィックは、各スレーブインターフェースの現在の負荷に従って分散されます。受信トラフィックは、現在のスレーブにより受信されま

す。受信しているスレーブが失敗すると、別のスレーブが失敗したスレーブの MAC アドレスを引き継ぎます。このモードは、カーネルボンディングモジュールが認識しているローカルアドレスにのみ、適したものになります。このため、仮想マシンのブリッジの背後では使用できません。

- balance-alb または 6: 耐障害性とロードバランシングに適応型ロードバランシング (ALB) ポリシーを設定します。IPv4 トラフィック用の送受信ロードバランシングが含まれます。ARP ネゴシエーションにより、受信ロードバランシングが可能です。このモードは、カーネルボンディングモジュールが認識しているローカルアドレスにのみ、適したものになります。このため、仮想マシンのブリッジの背後では使用できません。

num_unsol_na=<number>

フェイルオーバーイベント後に発行される未諾の IPv6 Neighbor Advertisement の数を指定します。フェイルオーバーの直後に未要求の NA が発行されます。

有効な範囲は 0 - 255 です。デフォルト値は 1 です。このパラメーターは、active-backup モードにのみ影響します。

primary=<interface_name>

プライマリーデバイスのインターフェース名（例： eth0 ）を指定します。primary デバイスは、使用される最初のボンディングインターフェースであり、失敗しない限りは破棄されません。この設定が特に役立つのは、ボンディングインターフェースの NIC の 1 つが高速なため、大規模な負荷に対応できる場合です。

この設定は、ボンディングインターフェースが active-backup モードの場合にのみ有効です。kernel-doc パッケージの bonding.txt ファイル（「その他のリソース」を参照）を参照してください。

primary_reselect=<value>

プライマリースレーブに対して再選択ポリシーを指定します。これは、アクティブなスレーブの失敗やプライマリースレーブの回復が発生した場合に、どのようにプライマリースレーブが選択されてアクティブなスレーブになるかという点に影響します。このパラメーターは、プライマリースレーブと他のスレーブ間でのフリップフロップを防ぐように設計されています。以下の値が使用できます。

- always または 0: プライマリースレーブは有効になるといつでもアクティブなスレーブになります。

- *better または 1: プライマリースレーブの速度とデュプレックスが、現在のアクティブなスレーブの速度とデュプレックスと比べて良い場合は、プライマリースレーブは有効になるとアクティブなスレーブになります。*

- *failure または 2: 現在のアクティブなスレーブが失敗してプライマリースレーブが有効になる場合のみ、プライマリースレーブはアクティブなスレーブになります。*

primary_reselect の設定は、以下の 2 つの場合では無視されます。

- *アクティブなスレーブがない場合は、回復する最初のスレーブがアクティブなスレーブになります。*

- *初めにプライマリースレーブがスレーブにされた場合は、それは常にアクティブなスレーブになります。*

sysfs で primary_reselect ポリシーを変更すると、新しいポリシーに従って、アクティブなスレーブを即座に選択することができます。これにより、状況によってはアクティブなスレーブに変更が生じる場合があります。

updelay=<time_in_milliseconds>

リンクを有効にする前の待機時間を指定します (ミリ秒単位)。値は、miimon パラメーターで指定される値の倍数でなければなりません。デフォルトでは値は 0 に設定されており、ARP 監視を無効にします。

use_carrier=<number>

リンク状態を決定するために miimon が MII/ETHTOOL ioctls または netif_carrier_ok() を使用するかどうか指定します。netif_carrier_ok() 機能は、デバイスドライバーを使用して netif_carrier_on/off によりその状態を維持します。大半のデバイスドライバーはこの機能に対応しています。

MII/ETHROOL ioctls ツールは、カーネル内で非推奨の呼び出しシーケンスを使用します。ただし、これは使用しているデバイスドライバーが netif_carrier_on/off に対応しない場合でも設定可能です。

有効な値は以下のとおりです。

- 1: デフォルト設定。netif_carrier_ok() の使用を有効にします。
- 0: MII/ETHTOOL ioctls の使用を有効にします。

注記

リンクがアップになっているべきでない時にアップであるとボンディングインターフェースが主張した場合、使用しているネットワークデバイスドライバーは netif_carrier_on/off に対応しない可能性があります。

xmit_hash_policy=<value>

balance-xor および 802.3ad モードで、スレーブを選択する時に使用する送信ハッシュポリシーを選択します。以下の値が使用できます。

- 0 または layer2: デフォルト設定。このパラメーターは、ハードウェア MAC アドレスの XOR を使用してハッシュを生成します。使用する式は以下のとおりです。

```
(<source_MAC_address> XOR <destination_MAC>) MODULO <slave_count>
```

このアルゴリズムは、すべてのトラフィックを同じスレーブの特定のネットワークピアに割り振り、802.3ad に対応します。

- 1 または layer3+4: 上位レイヤープロトコルの情報を (利用可能な場合は) 使用して、ハッシュを生成します。これにより、特定のネットワークピアへのトラフィックが複数のスレーブに及ぶようにできますが、単一の接続では複数のスレーブに及びません。

断片化された TCP および UDP パケットに使用される公式は、以下のとおりです:

```
((<source_port> XOR <dest_port>) XOR
  ((<source_IP> XOR <dest_IP>) AND 0xffff)
    MODULO <slave_count>
```

断片化された TCP または UDP パケットおよび他のすべての IP プロトコルトラフィックの場合、送信元および宛先ポート情報は省略されます。IP 以外のトラフィックの場合、この式は layer2 がハッシュポリシーを送信するものと同じです。

このポリシーの目的は、特に PFC2 付きの Cisco スイッチや Foundry および IBM 製品など一部のスイッチの動作を真似ることです。

このポリシーで使用されるアルゴリズムは、802.3ad に対応していません。

- 2 または layer2+3: layer2 および layer3 プロトコル情報の組み合わせを使用して、ハッシュを生成します。

  XOR のハードウェアの MAC アドレスと IP アドレスを使用してハッシュを生成します。式は以下のとおりです。

  ```
  (((<source_IP> XOR <dest_IP>) AND 0xffff) XOR
    ( <source_MAC> XOR <destination_MAC> ))
      MODULO <slave_count>
  ```

  このアルゴリズムは、すべてのトラフィックを同じスレーブの特定のネットワークピアに割り振ります。IP 以外のトラフィックの場合、この式は layer2 送信ハッシュポリシーと同じです。

  このポリシーの目的は、特に layer3 ゲートウェイデバイスが大半の宛先に到達する必要がある環境において、layer2 単独の場合より分散されたトラフィックを提供することです。

  このアルゴリズムは、802.3ad に対応しています。

31.9. その他のリソース

カーネルモジュールとそのユーティリティーの詳細は、以下のリソースを参照してください。

インストールされているドキュメント

- *lsmod(8): lsmod コマンドの man ページです。*
- *modinfo(8): modinfo コマンドの man ページです。*
- *modprobe(8)>: modprobe コマンドの man ページです。*
- *rmmod(8): rmmod コマンドの man ページです。*
- *ethtool(8): ethtool コマンドの man ページです。*
- *mi-tool(8): mii-tool コマンドの man ページです。*

インストール可能なドキュメント

- */usr/share/doc/kernel-doc- <kernel_version> /Documentation/: kernel-doc パッケージが提供するこのディレクトリーには、カーネル、カーネルモジュール、およびそれぞれのパラメーターに関する情報が含まれます。カーネルのドキュメントにアクセスする前に、root で以下のコマンドを実行する必要があります。*

```
~]# yum install kernel-doc
```

オンラインドキュメント

- *- Red Hat ナレッジベースの記事「『Which bonding modes work when used with a bridge that virtual machine guests connect to?』」*

---

[5]
この例が暗黙的である可能性はありますが、e1000e ドライバーでは、デフォルトで Ethernet がオンになっています。

第32章 KDUMP クラッシュリカバリーサービス

kdump クラッシュダンプメカニズムを有効にすると、別のカーネルのコンテキストからシステムが起動します。この 2 番目のカーネルは、少ないメモリーを確保し、システムがクラッシュした場合にコアダンプイメージを取得することを目的としています。

コアダンプの分析は、システム障害の正確な原因を判断するのに非常に役立つため、この機能を有効にすることが強く推奨されます。本章では、Red Hat Enterprise Linuxnbsp;Hat Enterprise Red Hat Enterprise Linuxnbsp;Linux の kdump サービスの設定、テスト、および使用方法について説明します。また、crash ユーティリティーを使用して、生成されるコアダンプの分析方法の概要を説明します。

32.1. KDUMP サービスのインストール

システムで kdump サービスを使用するには、kexec-tools パッケージがインストールされていることを確認します。これを行うには、root で次のコマンドを実行します。

```
~]# yum install kexec-tools
```

Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux に新しいパッケージをインストールする方法の詳細は、「パッケージのインストール」 を参照してください。

32.2. KDUMP サービスの設定

kdump サービスを設定する一般的な方法は、初回ブート時に、カーネルダンプ設定グラフィカルユーティリティーを 使用し、コマンドラインで手動で行う 3 つの方法です。

INTEL チップセットでの IOMMU の無効化

Intel IOMMU ドライバーの現在の実装の制限により、kdump サービスがコアダンプイメージを取得できなくなることがあります。Intel アーキテクチャーで kdump を確実に使用するには、IOMMU サポートを無効にすることを推奨します。

一部の HW 設定で KDUMP が正常に動作しない

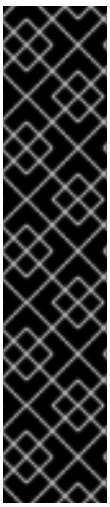
kdump サービスは、同じベンダーの HP Smart Array デバイスとシステムボードの特定の組み合わせで確実に機能しないことが知られています。そのため、実稼働環境で設定を使用する前に設定をテストすることを強く推奨します。必要な場合は、カーネルクラッシュダンプをネットワーク経由でリモートマシンに保存するように kdump を設定することを強く推奨します。kdump 設定のテスト方法に関する詳細は、「設定のテスト」 を参照してください。

32.2.1. 初回起動時での kdump の設定

システムを初めて起動すると、初回起動アプリケーションが起動し、新たにインストールしたシステムの初期設定でユーザーがガイドされます。kdump を設定するには、Kdump セクションに移動し、以下の手順に従います。

1. *kdump デーモンが起動時に起動するようにするには、Enable kdump? チェックボックスを選択します。これにより、ランレベル 2、3、4、および 5 のサービスが有効になり、現行セッションで起動します。同様に、チェックボックスの選択を解除すると、すべてのランレベルに対して無効になり、すぐにサービスを停止します。*

2. *Kdump メモリー フィールドの横にある上矢印ボタンおよび下矢印ボタンをクリックして値を増減し、kdump カーネル用に予約されるメモリー量を設定します。システムメモリーフィールドは 、システムで使用できる メモリーの残量に応じて変化します。*



システムに十分なメモリーがあることを確認します。

このセクションは、システムに十分なメモリーがある場合に限り利用できます。Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;LinuxRed Hat Enterprise Linux 6nbsp;6 システムのメモリー最小要件については、『Red Hat Enterprise Linux テクノロジーの機能と制限』 『の比較チャートで必要な最小数』 セクションを参照してください。kdump クラッシュリカバリーを有効にすると、最小メモリー要件が予約メモリーサイズで増加します。この値はユーザーによって決定され、物理メモリーが 1 TB のシステムの合計 192 MB（つまり、物理メモリーが 1 TB 件の 192 MB）にデフォルトで 128 MB を加算します。メモリーは、必要に応じて最大 896 MB の試行できます。特に論理ユニット番号(LUN)が多数あるシステムでは、大規模な環境で推奨されます。

32.2.2. カーネルダンプ設定ユーティリティーの使用

Kernel Dump Configuration ユーティリティーを起動するにはパネルから System →

Administration → Kernel crash dumps を選択するか、シェルプロンプトで system-config-kdump と入力します。図32.1「基本設定」 に示されるように、ウィンドウが表示されます。

このユーティリティーを使用すると kdump の設定のほか、システムの起動時にサービスを有効または無効にすることもできます。設定が完了したら 適用 をクリックして変更を保存します。システムの再起動が要求されますが、すでに認証されていない限り、スーパーユーザーのパスワードを入力するように求められます。

重要

SELinux が Enforcing モードで実行中の IBM System z または PowerPC システムでは、カーネルダンプ設定 ユーティリティーを起動する前に kdumpgui_run_bootloader のブール値を有効にする必要があります。このブール値により、system-config-kdump が bootloader_t SELinux ドメインでブートローダーを実行できます。ブール値を永続的に有効にするには、root で以下のコマンドを実行します。

```
~]# setsebool -P kdumpgui_run_bootloader 1
```

サービスの有効化

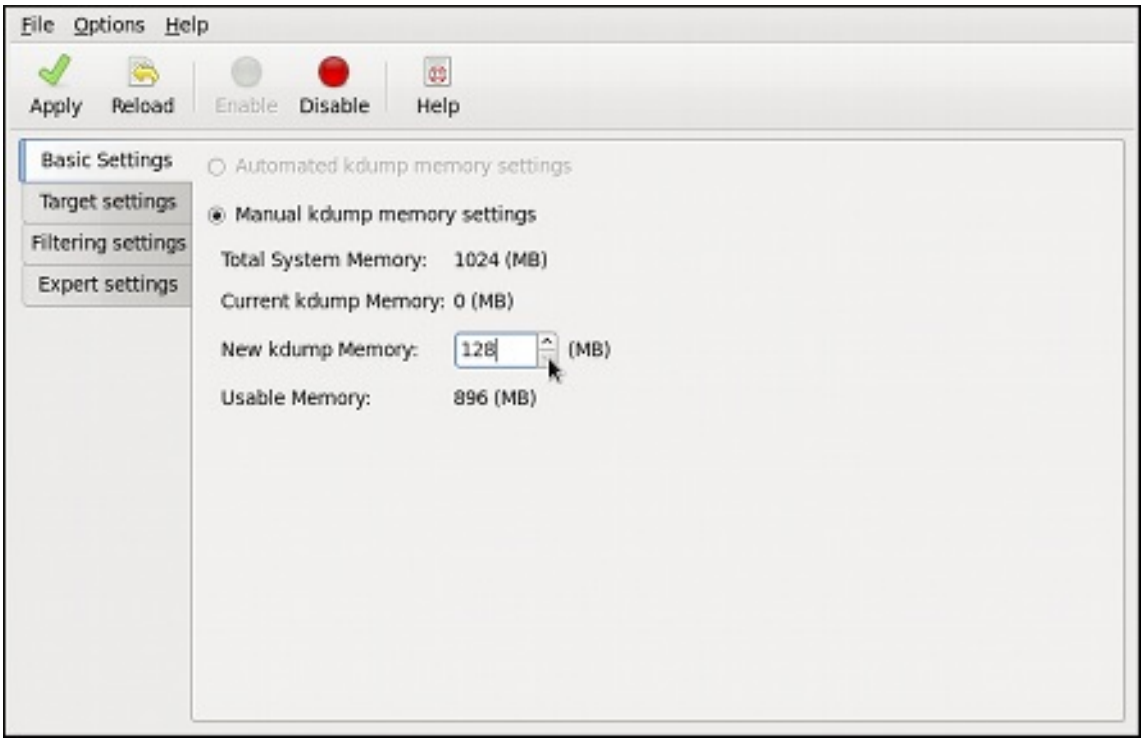
起動時に kdump デーモンを開始するには、ツールバーで Enable ボタンをクリックします。これにより、ランレベル 2、3、4、および 5 のサービスが有効になり、現行セッションで起動します。同様に、無効 ボタンをクリックすると、すべてのランレベルに対して無効になり、すぐにサービスを停止します。

ランレベルおよび一般的なサービスの設定に関する詳しい情報は、12章サービスおよびデーモン を参照してください。

基本設定タブ

基本設定 タブ では kdump カーネル用に予約されるメモリー量を設定できます。これを行うには、手動 kdump メモリー設定 ラジオボタンを選択し、新規の kdump メモリー フィールドの横にある上矢印ボタンを クリックして値を増減します。システムで 使用できるメモリーの 残量に応じて 使用可能なメモリー フィールドが変化します。

図32.1 基本設定



[D]

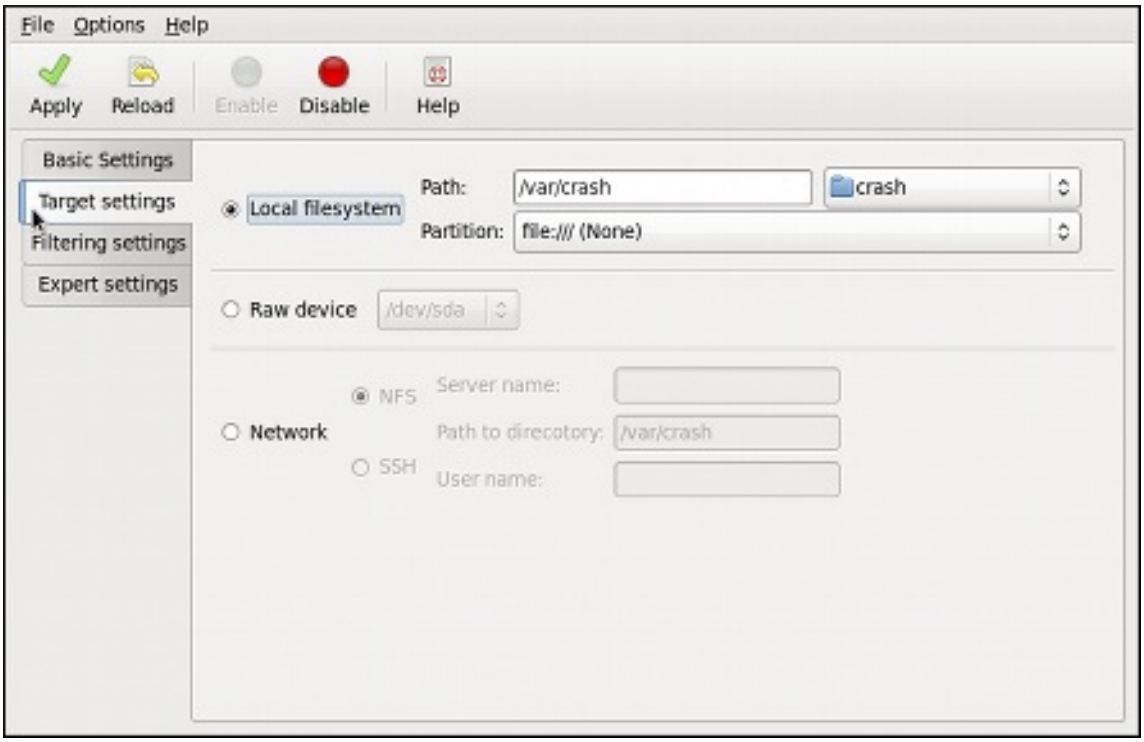
システムに十分なメモリーがあることを確認します。

このセクションは、システムに十分なメモリーがある場合に限り利用できます。Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;LinuxRed Hat Enterprise Linux 6nbsp;6 システムのメモリー最小要件については、『Red Hat Enterprise Linux テクノロジーの機能と制限』『の比較チャートで必要な最小数』セクションを参照してください。kdump クラッシュリカバリーを有効にすると、最小メモリー要件が予約メモリーサイズで増加します。この値はユーザーによって決定され、物理メモリーが 1 TB のシステムの合計 192 MB（つまり、物理メモリーが 1 TB 件の 192 MB）にデフォルトで 128 MB を加算します。メモリーは、必要に応じて最大 896 MB の試行できます。特に論理ユニット番号(LUN)が多数あるシステムでは、大規模な環境で推奨されます。

ターゲット設定タブ

ターゲット 設定 タブでは、vmcore ダンプのターゲットの場所を指定できます。これはローカルのファイルシステムにファイルとして保存するか、デバイスに直接書き込むか、または NFS (Network File System) や SSH (Secure Shell) などのプロトコルを使ってネットワーク経由で送信することができます。

図32.2 出力先

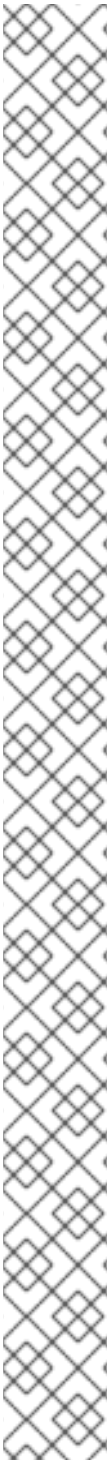


[D]

ローカルファイルシステムにダンプを保存するには、ローカルファイルシステムのラジオボタン を選択します。必要に応じて、パーティション とは別のパーティションを選択し、パス プル ダウン メニューからターゲットディレクトリーを選択して設定をカスタマイズできます。

デバイスに直接ダンプを書き込む場合は Raw デバイス ラジオボタンを選択し、目的のターゲットデバイスをその横にあるプルダウン一覧から選択します。

リモートマシンにダンプを保存するには、ネットワーク ラジオボタンを選択します。NFS プロトコルを使用するには、NFS ラジオボタンを選択して サーバー 名 と ディレクトリーへ のパス フィールドを入力します。SSH プロトコルを使用するには、SSH ラジオボタンを選択して サーバー 名、ディレクトリーへ のパス、ユーザー 名 フィールドにそれぞれリモートサーバーアドレス、ターゲットディレクトリー、有効なリモートユーザー名を入力します。SSH サーバーの設定方法と鍵ベースの認証の設定方法は 14章OpenSSH を参照してください。



注記

DASD ターゲットに DASD(Direct-Access Storage Devices)を使用する場合は、他の DASD を含む /etc/dasd.conf ファイルでデバイスを指定する必要があります。以下に例を示します。

```
0.0.2098
0.0.2198
0.0.2298
0.0.2398
```

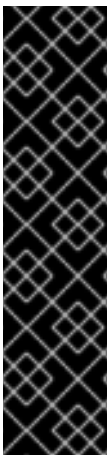
0.0.2298 および 0.0.2398 は、kdump ターゲットとして使用される DASD です。

同様に、SCSI（FCP 接続の Small Computer System Interface）ディスクを kdump ターゲットとして使用する場合は、その他の FCP 接続の SCSI ディスクを備えた /etc/zfcp.conf ファイルでディスクを指定する必要があります。以下に例を示します。

```
0.0.3d0c 0x500507630508c1ae 0x402424aa00000000
0.0.3d0c 0x500507630508c1ae 0x402424ab00000000
0.0.3d0c 0x500507630508c1ae 0x402424ac00000000
```

ここで、0.0.3d0c 0x500507630508c1ae 0x402424ab00000000 および 0. 0.3d0c 0x500507630508c1ae 0x402424ac00000000 は kdump ターゲットとして使用される FCP 接続の SCSI ディスクです。

『DASD および FCP 接続の SCSI ディスクの設定の詳細は、『『Installation Guide』 for Red Hat Enterprise Linux 6nbsp;Linux Red Hat Enterprise Linux 6nbsp;Linux Red Hat Enterprise Linux 6nbsp;6』の「Adding FCP-Attached Logical Units(LUN)』 」の章を参照してください。



VMCORE.FLAT ファイルを変換する必要があります。

SSH 経由でコアファイルをリモートターゲットに転送する場合は、転送のためにコアファイルをシリアライズする必要があります。これにより、ターゲットシステムの /var/crash/ ディレクトリーに vmcore.flat ファイルが作成されます。これは、crash ユーティリティーが読み取りできません。vmcore.flat を、クラッシュ で読み取り可能なダンプファイルに変換するには、ターゲットシステムで root で以下のコマンドを実行します。

```
~]# /usr/sbin/makedumpfile -R */tmp/vmcore-rearranged* < *vmcore.flat*
```

現在サポートされているターゲットの一覧は、表32.1「サポートしている kdump のダンプ出力先」 を参照してください。

表32.1 サポートしている kdump のダンプ出力先

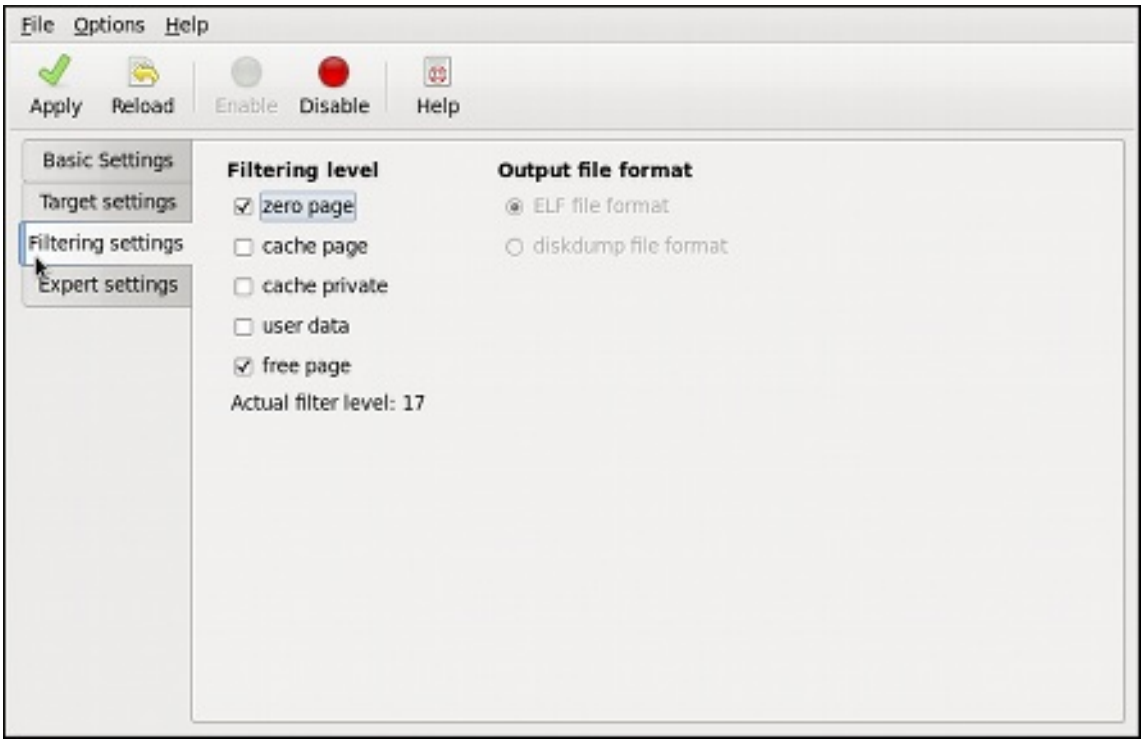
タイプ	対応しているダンプ出力先	対応していないダンプ出力先
Raw デバイス	ローカルで添付されたすべての raw ディスクとパーティション	—
ローカルファイルシステム	直接接続されているディスクドライブ、ハードウェア RAID 論理ドライブ、LVM デバイス、mdraid アレイ上の ext2、ext3、ext4、ext4、minix、btrfs、および xfs ファイルシステム。	auto タイプ（自動ファイルシステム検出）など、この表で明示的にサポート対象とされていないローカルファイルシステム。
リモートディレクトリー	IPv4 で NFS または SSH プロトコルを使用してアクセスするリモートディレクトリー。	NFS プロトコルを使用してアクセスする rootfs ファイルシステム上のリモートディレクトリー。
iSCSI	iBFT (iSCSI Boot Firmware Table) が使用されていない限り、ソフトウェアイニシエーター上で iSCSI プロトコルを使用してアクセスするリモートディレクトリー。	iBFT を使用して iSCSI プロトコルを使用してアクセスするリモートディレクトリー。
multipath	マルチパスベースのストレージ[a]	ハードウェアイニシエーター上で iSCSI プロトコルを使用してアクセスするリモートディレクトリー。
—	—	IPv6 でアクセスするリモートディレクトリー。
.	.	SMB/CIFS プロトコルを使用してアクセスするリモートディレクトリー。
.	.	FCoE (Fibre Channel over Ethernet)プロトコルを使用してアクセスするリモートディレクトリー。
.	.	ワイヤレスネットワークインターフェースを使ってアクセスするリモートディレクトリー

タイプ	対応しているダンプ出力先	対応していないダンプ出力先
[a] kexec-tools-2.0.0-245.el6 以降の Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;LinuxRed Hat Enterprise Linux 6nbsp;6 でサポートされています。		

フィルタリング設定タブ

Filtering Settings （フィルタリング設定）タブでは、vmcore ダンプのフィルターレベルを選択できます。

図32.3 フィルタリング



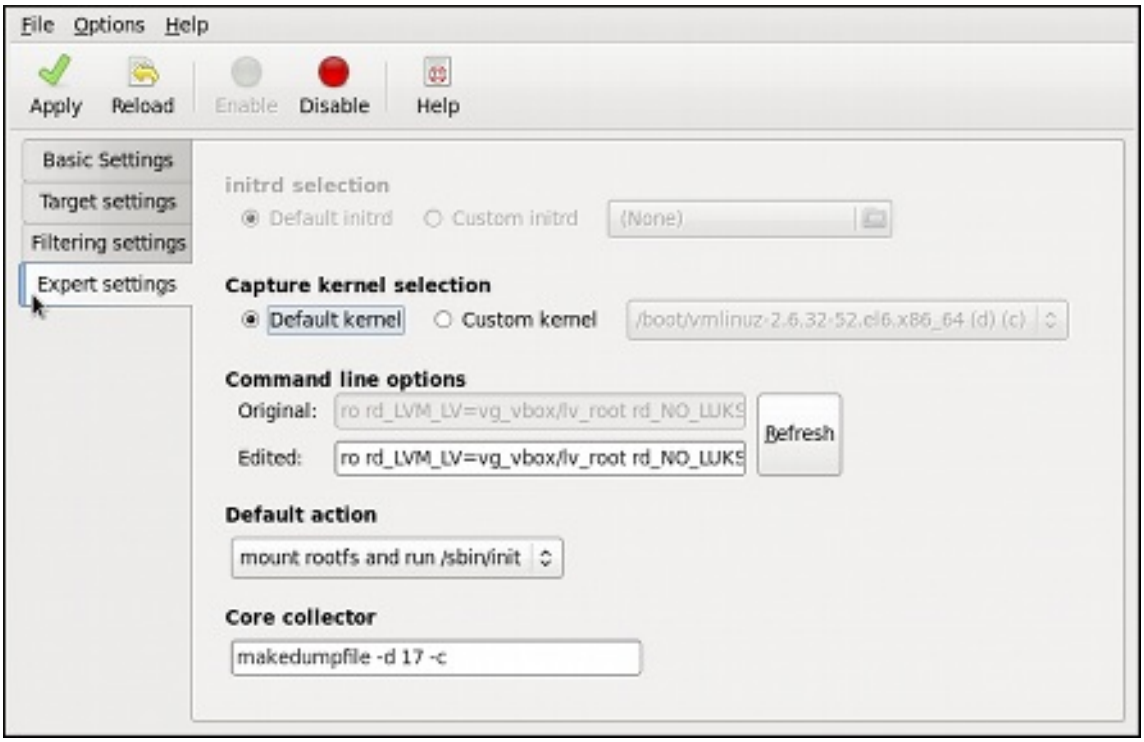
[D]

ダンプから ゼロページ、キャッシュページ、キャッシュ プライベート、ユーザーデータ、または 空きページ を除外するには、適切なラベルの横にあるチェックボックスを選択します。

エキスパート設定タブ

cent Settings タブでは、使用するカーネルおよび初期 RAM ディスクを選択したり、カーネルおよびコアコレクタープログラムに渡すオプションをカスタマイズできます。

図32.4 エキスパート設定



[D]

別の初期 RAM ディスクを使用するには、Custom initrd ラジオボタンを選択し、それの横にあるプルダウン一覧から必要な RAM ディスクを選択します。

別のカーネルをキャプチャーするには、カスタムカーネル ラジオボタンを選択し、右側のプルダウンの一覧から必要なカーネルイメージを選択します。

起動時にカーネルに渡されるオプションの一覧を調整するには、編集 済みテキストフィールドの内容を編集 します。リフレッシュ ボタンをクリックすると変更を常に元に戻すことができることに注意してください。

kdump がコアダンプを作成できない場合に実行するアクションを選択するには、Default action pulldown 一覧から適切なオプションを選択します。利用可能なオプションは、rootfs をマウントし、/sbin/init （デフォルトアクション）、再 起動（システムの再起動）、shell （対話式シェルプロンプトのあるユーザーに表示）、停止 （システムの停止）、および poweroff （システムの電源オフ）です。

makedumpfile コアコレクターに渡されるオプションをカスタマイズするには、Core collector のテキストフィールドを編集します。詳細は、「Core Collector の設定」 を参照してください。

32.2.3. コマンドラインで kdump の設定

メモリー使用量の設定

kdump カーネル用に予約されるメモリーは、システムの起動時に常に予約されます。つまり、メモリーのサイズはシステムのブートローダー設定で指定されます。本セクションでは、GRUB ブートローダーを使用して AMD64 システム、Intel 64 システム、および IBM Power Systems サーバーで予約メモリーの量を変更する方法と、zipl を使用して IBM System z で変更する方法を説明します。kdump カーネル用に予約するメモリー容量を設定するには、/boot/grub/grub.conf ファイルを編集し、例 32.1「/boot/grub/grub.conf ファイルのサンプル」 に示されるように kernel オプションの一覧に crashkernel= <size> M または crashkernel=auto を追加します。crashkernel=auto オプションは、システムの物理メモリーが以下の値以上である場合に限りメモリーを予約することに注意してください。

- 32 ビットおよび 64 ビット x86 アーキテクチャーでは 2 GB。
- ページサイズが 4 KB または 8 GB の場合は PowerPC の 2 GB。
- IBM S/390 上の 4 GB。

例32.1 /boot/grub/grub.conf ファイルのサンプル

```
# grub.conf generated by anaconda
#
# Note that you do not have to rerun grub after making changes to this file
# NOTICE:  You have a /boot partition.  This means that
#          all kernel and initrd paths are relative to /boot/, eg.
#          root (hd0,0)
#          kernel /vmlinuz-version ro root=/dev/sda3
#          initrd /initrd
#boot=/dev/sda
default=0
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
title Red Hat Enterprise Linux Server (2.6.32-220.el6.x86_64)
root (hd0,0)
kernel /vmlinuz-2.6.32-220.el6.x86_64 ro root=/dev/sda3 crashkernel=128M
initrd /initramfs-2.6.32-220.el6.x86_64.img
```

システムに十分なメモリーがあることを確認します。

このセクションは、システムに十分なメモリーがある場合に限り利用できます。Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;LinuxRed Hat Enterprise Linux 6nbsp;6 システムのメモリー最小要件については、『Red Hat Enterprise Linux テクノロジーの機能と制限』『の比較チャートで必要な最小数』セクションを参照してください。kdump クラッシュリカバリーを有効にすると、最小メモリー要件が予約メモリーサイズで増加します。この値はユーザーによって決定され、物理メモリーが 1 TB のシステムの合計 192 MB（つまり、物理メモリーが 1 TB 件の 192 MB）にデフォルトで 128 MB を加算します。メモリーは、必要に応じて最大 896 MB の試行できます。特に論理ユニット番号(LUN)が多数あるシステムでは、大規模な環境で推奨されます。

ターゲットタイプの設定

カーネルクラッシュがキャプチャーされると、コアダンプはローカルファイルシステムのファイルとして保存したり、デバイスに直接書き込みしたり、NFS(Network File System)または SSH(Secure Shell)プロトコルを使用してネットワーク上で送信したりすることができます。現時点で設定できるのはこれらのオプションの 1 つのみです。デフォルトのオプションは、vmcore ファイルをローカルファイルシステムの /var/crash/ ディレクトリーに保存することです。これを変更するには、root でテキストエディターで /etc/kdump.conf 設定ファイルを開き、以下のようにオプションを編集します。

コアダンプの保存先のローカルディレクトリーを変更するには、「#」path /var/crash の行頭にあるハッシュ記号(#)を取り除き、値を任意のディレクトリーパスに置き換えます。必要に応じて、ファイルを別のパーティションに書き込む場合は、#ext4 /dev/sda3 行と同じ手順を実行し、ファイルシステムタイプとデバイス（デバイス名、ファイルシステムラベル、UUID はすべてサポートされています）を変更します。以下に例を示します。

```
ext3 /dev/sda4
        path /usr/local/cores
```

ダンプをデバイスに直接書き込む場合は 「#」raw /dev/sda5 行の先頭にあるハッシュ記号(#)を取り除き、値を目的のデバイス名に置き換えます。以下に例を示します。

```
raw /dev/sdb1
```

NFS プロトコルを使用してリモートのマシンにダンプを保存する場合は 「#」net my.server.com:/export/tmp の行頭にあるハッシュ記号(#)を取り除き、値を有効なホスト名およびディレクトリーパスに置き換えます。以下に例を示します。

```
net penguin.example.com:/export/cores
```

SSH プロトコルを使用してリモートのマシンにダンプを保存する場合は 「#」net user@my.server.com の行頭にあるハッシュ記号(#)を取り除き、値を有効なユーザー名およびホスト

名に置き換えます。以下に例を示します。

```
net john@penguin.example.com
```

SSH サーバーの設定方法と鍵ベースの認証の設定方法は 14章OpenSSH を参照してください。

現在サポートされているターゲットの一覧は、表32.1「サポートしている kdump のダンプ出力先」 を参照してください。

注記

DASD ターゲットに DASD(Direct-Access Storage Devices)を使用する場合は、他の DASD を含む /etc/dasd.conf ファイルでデバイスを指定する必要があります。以下に例を示します。

```
0.0.2098
0.0.2198
0.0.2298
0.0.2398
```

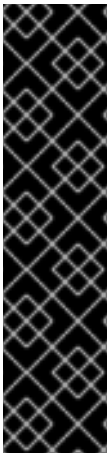
0.0.2298 および 0.0.2398 は、kdump ターゲットとして使用される DASD です。

同様に、SCSI（FCP 接続の Small Computer System Interface）ディスクを kdump ターゲットとして使用する場合は、その他の FCP 接続の SCSI ディスクを備えた /etc/zfcp.conf ファイルでディスクを指定する必要があります。以下に例を示します。

```
0.0.3d0c 0x500507630508c1ae 0x402424aa00000000
0.0.3d0c 0x500507630508c1ae 0x402424ab00000000
0.0.3d0c 0x500507630508c1ae 0x402424ac00000000
```

ここで、0.0.3d0c 0x500507630508c1ae 0x402424ab00000000 および 0. 0.3d0c 0x500507630508c1ae 0x402424ac00000000 は kdump ターゲットとして使用される FCP 接続の SCSI ディスクです。

『DASD および FCP 接続の SCSI ディスクの設定の詳細は、『『Installation Guide』 for Red Hat Enterprise Linux 6nbsp;Linux Red Hat Enterprise Linux 6nbsp;Linux Red Hat Enterprise Linux 6nbsp;6』の「Adding FCP-Attached Logical Units(LUN)』 」の章を参照してください。



VMCORE.FLAT ファイルを変換する必要があります。

SSH 経由でコアファイルをリモートターゲットに転送する場合は、転送のためにコアファイルをシリアライズする必要があります。これにより、ターゲットシステムの /var/crash/ ディレクトリーに vmcore.flat ファイルが作成されます。これは、crash ユーティリティーが読み取りできません。vmcore.flat を、クラッシュ で読み取り可能なダンプファイルに変換するには、ターゲットシステムで root で以下のコマンドを実行します。

```
~]# /usr/sbin/makedumpfile -R */tmp/vmcore-rearranged* < *vmcore.flat*
```

Core Collector の設定

vmcore ダンプファイルのサイズを小さくするために、kdump では外部アプリケーション（コアコレクター）を指定してデータを圧縮し、必要に応じてすべての関連性のない情報を除外できます。現在、完全にサポートされている唯一のコアコレクターは makedumpfile です。

コアコレクターを有効にするには、root で、テキストエディターで /etc/kdump.conf 設定ファイルを開き、「#」core_collector makedumpfile -c --message-level 1 -d 31 の行頭にあるハッシュ記号(#)を取り除き、以下のようにコマンドラインオプションを編集します。

ダンプファイルの圧縮を有効にするには、-c パラメーターを追加します。以下に例を示します。

```
core_collector makedumpfile -c
```

ダンプから特定のページを削除するには、-d valueパラメーターを追加します。value は、表 32.2「サポートされるフィルターレベル」 で説明されているように、省略するページの値の合計になります。ゼロと未使用ページを除外する場合は次のようになります。

```
core_collector makedumpfile -d 17 -c
```

利用可能なオプションの完全な一覧は、makedumpfile の man ページを参照してください。

表32.2 サポートされるフィルターレベル

オプション	説明
1	ゼロページ
2	キャッシュページ

オプション	説明
4	キャッシュプライベート
8	ユーザーページ
16	フリーページ

デフォルトアクションの変更

Red Hat Enterprise Linux 6.0 では、kdump がコアダンプの作成に失敗する場合のデフォルトの動作であるバージョン 6.2 までのデフォルトアクションにより、root ファイルシステムがマウントされ、/sbin/init が実行されます。

Red Hat Enterprise Linux 6.3 以降では、デフォルトの動作ではマシンを再起動します。この変更は、予約が少ないメモリーを使用して kdump が確実に動作するようにするために必要です。以前の動作を許可するために、mount_root_run_init オプションが 表32.3「サポートされるアクション」 に追加されました。

デフォルトの動作を変更するには、root で、テキストエディターで /etc/kdump.conf 設定ファイルを開き、「#」default shell の行頭にあるハッシュ記号(#)を取り除き、表32.3「サポートされるアクション」 の説明に従って値を希望のアクションに置き換えます。

表32.3 サポートされるアクション

オプション	説明
reboot	システムを再起動します。コアは失われます。
halt	システムを停止します。
poweroff	システムの電源を切ります。
shell	initramfs 内から msh セッションを実行して、ユーザーがコアを手動で記録できるようにします。
mount_root_run_init	Red Hat Enterprise Linux 6.2 以前からデフォルトのフェイルバック動作を有効にします。

以下に例を示します。

-

```
default halt
```

サービスの有効化

システムの起動時に kdump デーモンを起動するには、root で次のコマンドを実行します。

```
chkconfig kdump on
```

これにより、ランレベル 2、3、4、および 5 のサービスが有効になります。同様に、chkconfig kdump off を入力すると、すべてのランレベルで無効になります。現行セッションでサービスを起動するには、root で以下のコマンドを使用します。

```
service kdump start
```

ランレベルおよび一般的なサービスの設定に関する詳しい情報は、12章サービスおよびデーモン を参照してください。

32.2.4. 設定のテスト

これらのコマンドを使用する場合は注意してください。

以下のコマンドにより、カーネルがクラッシュします。以下の手順に従って、実稼働マシンでは使用しないでください。

設定をテストするには、kdump を有効にしてシステムを再起動し、サービスが実行中であることを確認します（Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux でサービスを実行する方法は 「サービスの実行」 を参照してください）。

```
~]# service kdump status
Kdump is operational
```

次に、シェルプロンプトで以下のコマンドを入力します。

```
echo 1 > /proc/sys/kernel/sysrq
echo c > /proc/sysrq-trigger
```

これにより、Linux カーネルがクラッシュし、アドレス-YYYY-MM-DD-HH:MM:SS/vmcore ファイ

ルが設定で選択した場所にコピーされます（デフォルトでは /var/crash/ ）。

32.3. コアダンプの分析

システムクラッシュの原因を特定するには、crash ユーティリティーを使用します。これにより、GDB(GNU Debugger)と非常に似た対話式プロンプトを利用できます。このユーティリティーを使用すると、実行中の Linux システムや netdump、diskdump、xendump、kdump で作成されたコアダンプを対話形式で分析できます。

関連するパッケージがインストールされていることを確認します。

vmcore ダンプファイルを分析するには、crash および kernel-debuginfo パッケージがインストールされている必要があります。システムに crash パッケージをインストールするには、root で次のコマンドを実行します。

```
yum install crash
```

kernel-debuginfo パッケージをインストールするには、yum-utils パッケージがインストールされていることを確認し、root で以下のコマンドを実行します。

```
debuginfo-install kernel
```

このコマンドを使用するには、デバッグパッケージを含むリポジトリーにアクセスできる必要があることに注意してください。システムが Red Hat Subscription Management に登録されている場合は、「現在の設定の表示」 の説明に従って rhel-6-variant-debug-rpms リポジトリーを有効にします。システムが RHN Classic に登録されている場合は、システムを rhel-architecture-variant-6-debuginfo チャンネルにサブスクライブし https://access.redhat.com/site/solutions/9907 ます。

32.3.1. crash ユーティリティーの実行

シェルプロンプトで次の形式のコマンドを入力してユーティリティーを起動します。

```
crash /usr/lib/debug/lib/modules/kernel/vmlinux /var/crash/timestamp/vmcore
```

カーネル のバージョンは、kdump が取得したものと同じである必要があります。現在実行中のカーネルを確認するには、uname -r コマンドを使用します。

例32.2 crash ユーティリティーの実行

```
~]# crash /usr/lib/debug/lib/modules/2.6.32-69.el6.i686/vmlinux \
/var/crash/127.0.0.1-2010-08-25-08:45:02/vmcore

crash 5.0.0-23.el6
Copyright (C) 2002-2010  Red Hat, Inc.
Copyright (C) 2004, 2005, 2006  IBM Corporation
Copyright (C) 1999-2006  Hewlett-Packard Co
Copyright (C) 2005, 2006  Fujitsu Limited
Copyright (C) 2006, 2007  VA Linux Systems Japan K.K.
Copyright (C) 2005  NEC Corporation
Copyright (C) 1999, 2002, 2007  Silicon Graphics, Inc.
Copyright (C) 1999, 2000, 2001, 2002  Mission Critical Linux, Inc.
This program is free software, covered by the GNU General Public License,
and you are welcome to change it and/or distribute copies of it under
certain conditions.  Enter "help copying" to see the conditions.
This program has absolutely no warranty.  Enter "help warranty" for details.

GNU gdb (GDB) 7.0
Copyright (C) 2009 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "i686-pc-linux-gnu"...

      KERNEL: /usr/lib/debug/lib/modules/2.6.32-69.el6.i686/vmlinux
    DUMPFILE: /var/crash/127.0.0.1-2010-08-25-08:45:02/vmcore  [PARTIAL DUMP]
        CPUS: 4
        DATE: Wed Aug 25 08:44:47 2010
      UPTIME: 00:09:02
LOAD AVERAGE: 0.00, 0.01, 0.00
       TASKS: 140
    NODENAME: hp-dl320g5-02.lab.bos.redhat.com
     RELEASE: 2.6.32-69.el6.i686
     VERSION: #1 SMP Tue Aug 24 10:31:45 EDT 2010
     MACHINE: i686  (2394 Mhz)
      MEMORY: 8 GB
       PANIC: "Oops: 0002 [#1] SMP " (check log for details)
         PID: 5591
     COMMAND: "bash"
        TASK: f196d560  [THREAD_INFO: ef4da000]
         CPU: 2
       STATE: TASK_RUNNING (PANIC)

crash>
```

32.3.2. メッセージバッファーの表示

カーネルメッセージバッファーを表示するには、対話式プロンプトで log コマンドを入力します。

例32.3 カーネルメッセージバッファーの表示

```
crash> log
... several lines omitted ...
EIP: 0060:[<c068124f>] EFLAGS: 00010096 CPU: 2
EIP is at sysrq_handle_crash+0xf/0x20
EAX: 00000063 EBX: 00000063 ECX: c09e1c8c EDX: 00000000
ESI: c0a09ca0 EDI: 00000286 EBP: 00000000 ESP: ef4dbf24
 DS: 007b ES: 007b FS: 00d8 GS: 00e0 SS: 0068
Process bash (pid: 5591, ti=ef4da000 task=f196d560 task.ti=ef4da000)
Stack:
 c068146b c0960891 c0968653 00000003 00000000 00000002 efade5c0 c06814d0
<0> ffffffb c068150f b7776000 f2600c40 c0569ec4 ef4dbf9c 00000002 b7776000
<0> efade5c0 00000002 b7776000 c0569e60 c051de50 ef4dbf9c f196d560 ef4dbfb4
Call Trace:
 [<c068146b>] ? __handle_sysrq+0xfb/0x160
 [<c06814d0>] ? write_sysrq_trigger+0x0/0x50
 [<c068150f>] ? write_sysrq_trigger+0x3f/0x50
 [<c0569ec4>] ? proc_reg_write+0x64/0xa0
 [<c0569e60>] ? proc_reg_write+0x0/0xa0
 [<c051de50>] ? vfs_write+0xa0/0x190
 [<c051e8d1>] ? sys_write+0x41/0x70
 [<c0409adc>] ? syscall_call+0x7/0xb
Code: a0 c0 01 0f b6 41 03 19 d2 f7 d2 83 e2 03 83 e0 cf c1 e2 04 09 d0 88 41 03 f3 c3 90 c7 05
c8 1b 9e c0 01 00 00 00 0f ae f8 89 f6 <c6> 05 00 00 00 00 01 c3 89 f6 8d bc 27 00 00 00 00 8d
50 d0 83
EIP: [<c068124f>] sysrq_handle_crash+0xf/0x20 SS:ESP 0068:ef4dbf24
CR2: 0000000000000000
```

このコマンドの使用方法についての詳しい情報を参照するには、help log と入力してください。

注記

カーネルメッセージバッファーには、システムクラッシュに関する最も重要な情報が含まれています。したがって、これは常に最初に vmcore-dmesg.txt ファイルにダンプされます。これは、たとえば、ターゲットの場所にスペースがないために vmcore ファイル全体の取得を試みる場合に便利です。デフォルトでは、vmcore-dmesg.txt は /var/crash/ ディレクトリーにあります。

32.3.3. バックトレースの表示

カーネルスタックトレースを表示するには、対話式プロンプトで bt コマンドを入力します。bt pid を使用して、選択したプロセスのバックトレースを表示できます。

例32.4 カーネルスタックトレースの表示

```
crash> bt
PID: 5591   TASK: f196d560  CPU: 2   COMMAND: "bash"
 #0 [ef4dbdcc] crash_kexec at c0494922
```

```
 #1 [ef4dbe20] oops_end at c080e402
 #2 [ef4dbe34] no_context at c043089d
 #3 [ef4dbe58] bad_area at c0430b26
 #4 [ef4dbe6c] do_page_fault at c080fb9b
 #5 [ef4dbee4] error_code (via page_fault) at c080d809
    EAX: 00000063  EBX: 00000063  ECX: c09e1c8c  EDX: 00000000  EBP: 00000000
    DS:  007b      ESI: c0a09ca0  ES:  007b      EDI: 00000286  GS:  00e0
    CS:  0060      EIP: c068124f  ERR: ffffffff  EFLAGS: 00010096
 #6 [ef4dbf18] sysrq_handle_crash at c068124f
 #7 [ef4dbf24] __handle_sysrq at c0681469
 #8 [ef4dbf48] write_sysrq_trigger at c068150a
 #9 [ef4dbf54] proc_reg_write at c0569ec2
#10 [ef4dbf74] vfs_write at c051de4e
#11 [ef4dbf94] sys_write at c051e8cc
#12 [ef4dbfb0] system_call at c0409ad5
    EAX: ffffffda  EBX: 00000001  ECX: b7776000  EDX: 00000002
    DS:  007b      ESI: 00000002  ES:  007b      EDI: b7776000
    SS:  007b      ESP: bfcb2088  EBP: bfcb20b4  GS:  0033
    CS:  0073      EIP: 00edc416  ERR: 00000004  EFLAGS: 00000246
```

このコマンドの使用方法についての詳しい情報を参照するには、help bt と入力してください。

32.3.4. プロセスステータスの表示

システム内のプロセスの状態を表示するには、対話式プロンプトで ps コマンドを入力します。ps pid を使用して、選択したプロセスの状態を表示できます。

例32.5 システム内のプロセスステータスの表示

```
crash> ps
   PID    PPID  CPU   TASK    ST  %MEM     VSZ    RSS  COMM
>     0      0   0  c09dc560  RU   0.0       0      0  [swapper]
>     0      0   1  f7072030  RU   0.0       0      0  [swapper]
      0      0   2  f70a3a90  RU   0.0       0      0  [swapper]
>     0      0   3  f70ac560  RU   0.0       0      0  [swapper]
      1      0   1  f705ba90  IN   0.0    2828   1424  init
... several lines omitted ...
   5566      1   1  f2592560  IN   0.0   12876    784  auditd
   5567      1   2  ef427560  IN   0.0   12876    784  auditd
   5587   5132   0  f196d030  IN   0.0   11064   3184  sshd
>  5591   5587   2  f196d560  RU   0.0    5084   1648  bash
```

このコマンドの使用方法についての詳しい情報を参照するには、help ps と入力してください。

32.3.5. 仮想メモリー情報の表示

基本的な仮想メモリー情報を表示するには、対話式プロンプトで vm コマンドを入力します。vm pid を使用して、選択したプロセスの情報を表示します。

例32.6 現在のコンテキストの仮想メモリー情報の表示

```
crash> vm
PID: 5591   TASK: f196d560  CPU: 2   COMMAND: "bash"
  MM       PGD      RSS    TOTAL_VM
f19b5900  ef9c6000  1648k    5084k
  VMA       START      END    FLAGS  FILE
f1bb0310    242000    260000 8000875  /lib/ld-2.12.so
f26af0b8    260000    261000 8100871  /lib/ld-2.12.so
efbc275c    261000    262000 8100873  /lib/ld-2.12.so
efbc2a18    268000    3ed000 8000075  /lib/libc-2.12.so
efbc23d8    3ed000    3ee000 8000070  /lib/libc-2.12.so
efbc2888    3ee000    3f0000 8100071  /lib/libc-2.12.so
efbc2cd4    3f0000    3f1000 8100073  /lib/libc-2.12.so
efbc243c    3f1000    3f4000 100073
efbc28ec    3f6000    3f9000 8000075  /lib/libdl-2.12.so
efbc2568    3f9000    3fa000 8100071  /lib/libdl-2.12.so
efbc2f2c    3fa000    3fb000 8100073  /lib/libdl-2.12.so
f26af888    7e6000    7fc000 8000075  /lib/libtinfo.so.5.7
f26aff2c    7fc000    7ff000 8100073  /lib/libtinfo.so.5.7
efbc211c    d83000    d8f000 8000075  /lib/libnss_files-2.12.so
efbc2504    d8f000    d90000 8100071  /lib/libnss_files-2.12.so
efbc2950    d90000    d91000 8100073  /lib/libnss_files-2.12.so
f26afe00    edc000    edd000 4040075
f1bb0a18   8047000   8118000 8001875  /bin/bash
f1bb01e4   8118000   811d000 8101873  /bin/bash
f1bb0c70   811d000   8122000 100073
f26afae0   9fd9000   9ffa000 100073
... several lines omitted ...
```

このコマンドの使用方法についての詳しい情報を参照するには、help vm と入力してください。

32.3.6. オープンファイルの表示

対話式プロンプトで files コマンドを入力してオープン ファイル に関する情報を表示します。files pid を使用して、選択したプロセスで開いているファイルを表示できます。

例32.7 現在のコンテキストのオープンファイルについての情報の表示

```
crash> files
PID: 5591   TASK: f196d560  CPU: 2   COMMAND: "bash"
ROOT: /    CWD: /root
 FD    FILE     DENTRY    INODE    TYPE  PATH
  0  f734f640  eedc2c6c  eecd6048  CHR   /pts/0
  1  efade5c0  eee14090  f00431d4  REG   /proc/sysrq-trigger
```

```
  2  f734f640  eedc2c6c  eecd6048  CHR   /pts/0
 10  f734f640  eedc2c6c  eecd6048  CHR   /pts/0
255  f734f640  eedc2c6c  eecd6048  CHR   /pts/0
```

このコマンドの使用方法についての詳しい情報を参照するには、help files と入力してください。

32.3.7. ユーティリティーの終了

対話式プロンプトを終了して crash を終了するには、exit または q を入力します。

例32.8 crash ユーティリティーの終了

```
crash> exit
~]#
```

32.4. IBM POWERPC ハードウェアにおける FADUMP の使用

Red Hat Enterprise Linux 6.8 以降では、kdump の代替ダンプメカニズムである ファームウェア支援ダンプ (fadump)が利用できるようになりました。fadump 機能は、IBM Power Systems でのみサポートされます。fadump の目的は、クラッシュしたシステムのダンプを有効にし、完全にリセットされたシステムからこれを実行し、システムが実稼働環境で復旧するまでの総経過時間を最小限に抑えることです。fadump 機能は、kdump と fadump メカニズムの間で無関係に切り替えるために、ユーザー空間にある kdump インフラストラクチャーと統合されます。

ファームウェア支援ダンプ(fadump)は、IBM PowerPC LPARS で利用可能な kexec-kdump に代わる信頼性の高い仕組みです。ファームウェア支援ダンプでは、PCI および I/O デバイスが再初期化され、完全にリセットされたシステムから、vmcore がキャプチャーされます。この仕組みでは、クラッシュ発生時にファームウェアを使用してメモリーを保存しますが、kdump ユーザー空間スクリプトを再利用して vmcore を保存します。

そのために、fadump はクラッシュ発生時にシステムファームウェアを使って保持する必要のあるメモリー領域を登録します。これらの領域には、ブートメモリー、システムレジスター、およびハードウェアのページテーブルエントリー (PTE) を除く、すべてのシステムメモリーコンテンツが含まれます。

注記

ブートメモリー と呼ばれる保持されないメモリー 領域は、クラッシュイベント後にカーネルを正常に起動するのに必要な RAM の容量です。デフォルトのブートメモリーサイズは、256 MB または全システム RAM の 5% のいずれか大きい方です。

kexec-initiated イベントとは異なり、fadump プロセスでは実稼働用のカーネルを使用してクラッシュダンプを復元します。PowerPC ハードウェアはクラッシュ後に起動すると、デバイスノード /proc/device-tree/rtas/ibm,kernel-dump が procfs で利用できるようにし、fadump 対応の kdump スクリプトは vmcore を保存するかどうかを確認します。この処理が完了すると、システムは正しく再起動されます。

fadump の有効化

1. 「kdump サービスのインストール」 および 「kdump サービスの設定」 で説明されているように kdump をインストールおよび設定します。

2. /etc/default/grub の GRUB_CMDLINE_LINUX の行に fadump=on を追加します。

```
GRUB_CMDLINE_LINUX="rd.lvm.lv=rhel/swap crashkernel=auto rd.lvm.lv=rhel/root
rhgb quiet fadump=on"
```

3. （オプション）デフォルトを許可する代わりに予約ブートメモリーを指定する場合は、/etc/default/grub の GRUB_CMDLINE_LINUX に fadump_reserve_mem=xxM を追加します。xx は必要なメモリー容量（メガバイト単位）に置き換えます。

```
GRUB_CMDLINE_LINUX="rd.lvm.lv=rhel/swap crashkernel=auto rd.lvm.lv=rhel/root
rhgb quiet fadump=on fadump_reserve_mem=xxM"
```

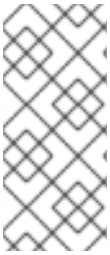
重要

すべてのブート設定オプションと同様に、必要になる前に設定をテストすることを強く推奨します。クラッシュカーネルから起動時に Out of Memory(OOM)エラーが発生する場合は、クラッシュカーネルが正常に起動できるまで、fadump_reserve_mem= で指定した値を増やします。この場合は、トライアンドエラーが必要になることがあります。

32.5. FUJITSU PRIMEQUEST システムにおける SADUMP の使用

Fujitsu PRIMEQUEST システムでは、ハードウェアが提供するスタンドアロンのダンプ(sadump)機能を有効にすることもできます。このユーティリティーは、Red Hat Enterprise Linux の標準部分で

ある kdump と Fujitsu が提供する追加の sadump BIOS ベースの機能を組み合わせて動作します。



注記

予期しない再起動が発生した場合にダンプが確実にキャプチャーされるようにするため、Fujitsu は PRIMEQUEST ハードウェアで常に sadump を有効にすることを推奨します。

通常、Red Hat Enterprise Linux が応答しなくなったため、kdump を処理できない場合に sadump ユーティリティーが呼び出されます。これらの条件には、以下が含まれます。

- kdump を起動する前に Red Hat Enterprise Linux のパニックまたはハングする
- kdump が機能している間にエラー

sadump の使用方法

sadump を使用するには、以下の手順を実施します。

1. 使用中のカーネルバージョンに応じて、以下のパッケージをインストールします。

```
# yum install kernel-debuginfo kernel-debuginfo-common
```

2. sadumpの UEFI の設定

詳細は、FUJITSU Server PRIMEQUEST 2000 Series Installation Manual を参照してください。

3. sadump向け Red Hat Enterprise Linux の設定

詳細は、「sadump 向け Red Hat Enterprise Linux の設定」 を参照してください。

4.
sadumpの開始

詳細は、FUJITSU Server PRIMEQUEST 2000 Series Installation Manual を参照してください。

5.
メモリーダンプを確認します。

詳細は、「メモリーダンプを確認します。」 を参照してください。

32.5.1. sadump 向け Red Hat Enterprise Linux の設定

1.
「kdump サービスのインストール」 および 「kdump サービスの設定」 で説明されているように kdump をインストールおよび設定します。

2.
sadump で kdump が想定どおりに起動されるようにします。

a.
カーネルパニック後に再起動しないように Red Hat Enterprise Linux を設定します。

デフォルトでは、Red Hat Enterprise Linux はカーネルパニック後に自動的に再起動されるため、sadump は起動しません。この動作を回避するには、/etc/sysctl.conf ファイルを以下のように設定します。

```
kernel.panic=0
```

b.
Red Hat Enterprise Linux がマスク不可割り込み(NMI)で kdump を開始するように設定します。

sadump を開始する手順では、最初に NMI で kdump を起動する必要があります。

/etc/sysctl.conf を以下のように設定します。

```
kernel.unknown_nmi_panic=1
```

3. sadump で kdump が正常に動作することを確認します。

a. kdump の後に停止するように Red Hat Enterprise Linux を設定します。

デフォルトでは、kdump が失敗すると、Red Hat Enterprise Linux は自動的に再起動するため、sadump が起動しないようになります。この動作を回避するには、/etc/kdump.conf ファイルを以下のように設定します。

```
default halt
```

または

```
default shell
```

b. sadump を開始するように Red Hat Enterprise Linux を設定します。

/etc/kdump.conf を設定して、System Management Interrupt(SMI)をブロックしないようにし、sadump が起動できるようにします。

```
blacklist kvm-intel
```

32.5.2. メモリーダンプを確認します。

メモリーダンプが終了し、Red Hat Enterprise Linux が再起動されると、特別なデバイスファイルを開くと、crash コマンドでメモリーダンプを確認できます。

以下の例では、/dev/sdb1 に保存されているメモリーダンプを確認する方法を示しています。

例32.9 ダンプ整合性の確認

```
crash /usr/lib/debug/lib/modules/2.6.32-358.el6.x86_64/vmlinux /dev/sdb1
```

32.6. その他のリソース

インストールされているドキュメント

- kdump.conf(5): 利用可能なオプションの詳細なドキュメントを含む /etc/kdump.conf 設定ファイルの man ページです。
- makedumpfile(8): makedumpfile コアコレクターの man ページです。
- kexec(8): kexec の man ページです。
- crash(8)- crash ユーティリティーの man ページです。
- /usr/share/doc/kexec-tools-version/kexec-kdump-howto.txt: kdump および kexec のインストールと使用の概要です。
- /usr/share/doc/kexec-tools-version/fadump-howto.txt: ハードウェアをリセットする方法など、fadump メカニズムの詳細。version を、お使いのシステムにインストールされている kexec-tools バージョンに置き換えます。

便利な Web サイト

https://access.redhat.com/kb/docs/DOC-6039

kexec および kdump 設定に関する Red Hat ナレッジベースアーティクルです。

https://access.redhat.com/kb/docs/DOC-45183

サポートしている kdump ターゲットに関する Red Hat ナレッジベースアーティクルです。

http://people.redhat.com/anderson/

crash ユーティリティーのホームページです。

パート IX. システムリカバリー

本パートでは、Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;LinuxRed Hat Enterprise Linux 6nbsp;6 のシステムのリカバリーモードを紹介し、特定の状況でシステムを修復する方法をユーザーに推奨しています。また、ReaR(Relax-and-Recover)障害復旧およびシステム移行ユーティリティーの使用方法も説明します。

第33章 システムリカバリー

Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;LinuxRed Hat Enterprise Linux 6nbsp;6 は、3 つのシステムリカバリーモード、レスキューモード、シングルユーザーモード、および緊急モード を提供し、誤作動システムを修復するために使用できます。本章では、各システムリカバリーモードで起動する方法を説明し、システムのリカバリーモードを使用してのみ解決できる特定の問題を解決するためのガイダンスを提供します。

以下は、システムリカバリーモードのいずれかで起動する必要がある、通常の理由です。

- Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux（ランレベル 3 または 5）で通常起動することはできません。
- システムが正常に実行中に解決できないハードウェアまたはソフトウェアの問題を解決する必要があります。または、ハードドライブから重要なファイルにアクセスしたい場合があります。
- root パスワードを忘れてしまった。

背後の問題の一部は、「システムリカバリーモードでの問題の解決」 で説明されています。

33.1. レスキューモード

レスキューモード は、システムのハードドライブではなく、CD-ROM や USB ドライブなどの外部メディアから完全に小規模な Red Hat Enterprise Linux 環境を起動する機能を提供します。さまざまな問題を修復するコマンドラインユーティリティーが含まれています。このモードでは、ファイルシステムを読み取り専用としてマウントしたり、ファイルシステムをマウントしたり、ドライバーディスクで提供されるドライバーをブラックリストに追加したり、システムパッケージをインストールまたはアップグレードしたり、パーティションを管理したりできます。

レスキューモード で起動するには、以下の手順に従います。

手順33.1 レスキューモードでの起動

1. 最小限の起動メディアまたは完全インストール DVD もしくは USB ドライブからシステムを起動し、ブートメニューが表示されるまで待ちます。選択したメディアからシステムを起動する方法は、『インストールガイド』 の該当する章を参照してください。

2. *起動メニューから、**rescue** キーワードをカーネルパラメーターとして起動コマンドラインに追加します。*

3. *システムの起動時にドライバー ディスクで提供されるサードパーティーのドライバー が必要な場合は、起動コマンドラインに **dd** オプションを追加して、そのドライバーを読み込みます。*

```
rescue dd
```

システムの起動時にディスクドライバーを使用する方法は、『インストールガイド』 の該当する章を参照してください。

4. ***Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;LinuxRed Hat Enterprise Linux 6nbsp;6** ディストリビューションに含まれるドライバーが原因でシステムが起動しないようにする場合は、起動コマンドラインに **rdblacklist** オプションを追加してそのドライバーをブラックリストに指定します。*

```
rescue rdblacklist=driver_name
```

5. *基本的な質問に回答し、プロンプトが表示されたら有効なレスキューイメージの場所を選択します。ローカル **CD-ROM**、ハードドライブ、**NFS** イメージ、**FTP**、または **HTTP** から関連するタイプを選択します。選択した場所には有効なインストールツリーが含まれ、インストールツリーは **Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux** のバージョンと同じバージョンの **Red Hat Enterprise Linuxnbsp;Linux** 用でなければなりません。これは、ブートしたディスクです。ハードドライブ、**NFS** サーバー、**FTP** サーバー、または **HTTP** サーバーにインストールツリーを設定する方法は、『インストール ガイド』 の該当する章を参照してください。*

*ネットワーク接続を必要としないレスキューイメージを選択すると、ネットワーク接続を確立するかどうかが尋ねられます。ネットワーク接続は、ファイルを別のコンピューターにバックアップする必要がある場合や、共有ネットワークの場所からいくつかの **RPM** パッケージをインストールする必要がある場合に役立ちます。*

6. *以下のメッセージが表示されます。*

```
The rescue environment will now attempt to find your Linux installation and mount it under the
directory /mnt/sysimage. You can then make any changes required to your system. If you
want to proceed with this step choose 'Continue'. You can also choose to mount your file
systems read-only instead of read-write by choosing 'Read-only'. If for some reason this
process fails you can choose 'Skip' and this step will be skipped and you will go directly to a
command shell.
```

Continue を選択すると、システムは /mnt/sysimage/ ディレクトリーに root パーティションをマウントしようとします。root パーティションには、通常、/home/、/boot/、/var/ などの複数のファイルシステムが含まれます。これは、正しい場所に自動的にマウントされます。パーティションのマウントに失敗すると、通知されます。Read-Only を選択すると、システムは /mnt/sysimage/ ディレクトリーにファイルシステムをマウントしようとしますが、読み取り専用モードになります。Skip を選択すると、ファイルシステムがマウントされません。ファイルシステムが破損していると思われる場合は、スキップ を選択します。

7.

レスキューモードでシステムを選択すると、仮想コンソール(VC)1 および VC 2 に以下のプロンプトが表示されます。Ctrl-Alt-F1 のキーの組み合わせを使用して VC 1 にアクセスし、Ctrl-Alt-F2 を使用して VC 2 にアクセスします。

```
sh-3.00b#
```

「Continue」を選択してパーティションを自動的にマウントし、正常にマウントされた場合は、シングルユーザーモード になります。

ファイルシステムがマウントされている場合でも、レスキューモード ではデフォルトの root パーティションは一時的な root パーティションで、通常のユーザーモードで使用されるファイルシステムの root パーティションではありません（runlevel 3 または 5）。ファイルシステムのマウントを選択し、正常にマウントされた場合は、次のコマンドを実行してレスキューモード環境の root パーティションを、ファイルシステムの root パーティションに変更できます。

```
sh-3.00b# chroot /mnt/sysimage
```

これは、root パーティションが / としてマウントされる必要がある rpm などのコマンドを実行する必要がある場合に便利です。chroot 環境を終了するには、exit と入力してプロンプトに戻ります。

Skip を選択した場合は、ディレクトリーを作成し、以下のコマンドを入力して、レスキューモード内でパーティションまたは LVM2 論理ボリュームを手動でマウントできます。

```
sh-3.00b# mkdir /directory
sh-3.00b# mount -t ext4 /dev/mapper/VolGroup00-LogVol02 /directory
```

/directory は作成したディレクトリーで、/dev/mapper/VolGroup00-LogVol02 はマウントする LVM2 論理ボリュームに置き換えます。パーティションが ext2 または ext3 タイプの場合は、ext4 をそれぞれ ext2 または ext3 に置き換えます。

すべての物理パーティションの名前が不明な場合は、次のコマンドを実行すると一覧が表示されます。

```
sh-3.00b# fdisk -l
```

LVM2 物理ボリュームやボリュームグループ、論理ボリュームの名前がすべて不明な場合はそれぞれ、pvdisplay、vgdisplay、lvdisplay のコマンドを使用します。

プロンプトから、以下のような多くの便利なコマンドが実行できます。

- ネットワークが開始されている場合、ssh、scp、ping
- テープドライブのユーザー用に dump と restore
- パーティションの管理に parted と fdisk
- ソフトウェア の インストールまたはアップグレードを行うための RPM
- テキストファイルの編集の vi

33.2. シングルユーザーモード

シングルユーザーモード は、1 人のユーザーの Linux 環境を提供し、ネットワーク化されたマルチユーザー環境で解決できない問題からシステムを復旧できます。シングルユーザーモード で起動できるようにするには、外部ブートデバイスは必要ありません。また、システムの実行中に直接切り替えることができます。実行中のシステムで シングルユーザーモード に切り替えるには、コマンドラインから以下のコマンドを実行します。

```
~]# init 1
```

シングルユーザーモード では、システムはマウントされたローカルファイルシステムで起動します。多くの重要なサービスが実行され、通常のシステムコマンドの多くは実行できるメンテナンスシェルになります。そのため、シングルユーザーモード は、システムの起動時に問題の解決に非常に便利ですが、正常に機能しない場合や、ログインすることができません。

ブート可能なファイルシステムでのみシングルユーザーモードで起動

シングルユーザーモード は、自動的にローカルファイルシステムのマウントを試みます。シングルユーザーモード で起動すると、ローカルファイルシステムが正常にマウントできない場合にデータが失われる可能性があります。

シングルユーザーモード で起動するには、以下の手順に従います。

手順33.2 シングルユーザーモードで起動

1. GRUB ブート画面で任意のキーを押すと GRUB 対話メニューに入ります。
2. 起動するカーネルのバージョンで Red Hat Enterprise Linux を選択し、a を押して行を追加します。
3. 行の最後にある別の単語として single と 入力 し、Enter を押して GRUB 編集モードを終了します。または、single の代わりに 1 を入力することもできます。

33.3. 緊急モード

緊急モード では、起動可能な最小限の環境が提供され、レスキューモード が利用できない場合でもシステムの修復が可能になります。緊急モード では、システムは root ファイルシステムのみをマウントし、読み取り専用としてマウントされます。また、システムはネットワークインターフェースをアクティブにせず、必要最小限のサービスのみが設定されます。システムは init スクリプトをロードしないため、init が破損しているか、または機能しない場合に、再インストール中に失われたデータを回復するためにファイルシステムをマウントできます。

緊急モード で起動するには、以下の手順に従います。

手順33.3 緊急モードでのブート

1. GRUB ブート画面で任意のキーを押すと GRUB 対話メニューに入ります。
2.

起動するカーネルのバージョンで Red Hat Enterprise Linux を選択し、a を押して行を追加します。

3. 行の最後にある別の単語として emergency と 入力 し、Enter を押して GRUB 編集モードを終了します。

33.4. システムリカバリーモードでの問題の解決

本セクションでは、システムの復旧モードの一部に対応する必要がある、最も一般的な問題のいくつかを解決する方法を説明します。

以下の手順は、root パスワードをリセットする方法を示しています。

手順33.4 root パスワードのリセット

1. 手順33.2「シングルユーザーモードで起動」 の説明に従って、シングルユーザーモード で起動します。

2. メンテナンスシェルコマンドラインから passwd コマンドを実行します。

起動不可能なシステムで最も一般的な原因の 1 つは、GRUB ブートローダーを含むマスターブートレコード(MBR)を上書きします。ブートローダーが上書きされても、レスキューモード でブートローダーを再設定しない限り、Red Hat Enterprise Linuxnbsp;Hat Enterprise Red Hat Enterprise Linuxnbsp;Linux を起動することはできません。

ハードドライブの MBR に GRUB を再インストールするには、次の手順に進みます。

手順33.5 GRUB ブートローダーの再インストール

1. 手順33.1「レスキューモードでの起動」 の説明に従って、レスキューモード で起動します。システムの root パーティションを読み書きモードでマウントしていることを確認します。

2. 以下のコマンドを実行して root パーティションを変更します。

```
sh-3.00b# chroot /mnt/sysimage
```

3. 以下のコマンドを実行して、GRUB ブートローダーを再インストールします。

```
sh-3.00b# /sbin/grub-install boot_part
```

boot_part は、ブートパーティション（通常は /dev/sda）に置き換えます。

4. GRUB が追加のオペレーティングシステムを制御するために追加エントリーが必要になる可能性があるため、/boot/grub/grub.conf ファイルを確認してください。

5. システムを再起動します。

システムが起動できなくなるもう 1 つの一般的な問題は、root パーティション番号の変更です。これは、通常、パーティションのサイズ変更やインストール後に新しいパーティションの作成時に発生する可能性があります。root パーティションのパーティション数が変わると、GRUB ブートローダーがそれをマウントできない可能性があります。この問題を修正するには、レスキューモード で起動し、/boot/grub/grub.conf ファイルを変更します。

誤作動またはドライバーがないと、システムが正常に起動できなくなる可能性があります。RPM パッケージマネージャーを使用して、誤作動ドライバーを削除したり、レスキューモード で更新されたドライバーや不足しているドライバーを追加したりすることができます。誤動作のあるドライバーを何らかの理由で削除できない場合は、代わりにドライバーをブラックリストに登録することで、起動時に読み込まれないようにすることができます。

DRIVER DISC がすべての定義済み INITRAMFS イメージを更新する

ドライバーディスクからドライバーをインストールする場合、ドライバーディスクはこのドライバーを使用するためにシステム上のすべての initramfs イメージを更新します。ドライバーが原因でシステムが起動できない場合は、別の initramfs イメージからシステムを起動する方法は使用できません。

システムの起動を妨げる誤作動ドライバーを削除するには、以下の手順に従います。

手順33.6 レスキューモードでのドライバーの削除

1. 手順33.1「レスキューモードでの起動」 の説明に従って、レスキューモード で起動します。システムの root パーティションを読み書きモードでマウントしていることを確認します。

2. *root ディレクトリーを /mnt/sysimage/ に変更します。*

```
sh-3.00b# chroot /mnt/sysimage
```

3. *以下のコマンドを実行してドライバーパッケージを削除します。*

```
sh-3.00b# rpm -e driver_name
```

4. *chroot 環境を終了します。*

```
sh-3.00b# exit
```

5. *システムを再起動します。*

システムが起動しないドライバーをインストールするには、以下の手順に従います。

手順33.7 レスキューモードでのドライバーのインストール

1. *手順33.1「レスキューモードでの起動」 の説明に従って、レスキューモード で起動します。システムの root パーティションを読み書きモードでマウントしていることを確認します。*

2. *ドライバーを含む RPM パッケージでメディアをマウントし、パッケージを /mnt/sysimage/ ディレクトリー配下の任意の場所にコピーします（例 : /mnt/sysimage/root/drivers/ ）。*

3. *root ディレクトリーを /mnt/sysimage/ に変更します。*

```
sh-3.00b# chroot /mnt/sysimage
```

4. *以下のコマンドを実行してドライバーパッケージをインストールします。*

```
sh-3.00b# rpm -ihv /root/drivers/package_name
```

この chroot 環境の /root/drivers/ は、元のレスキュー環境では

/mnt/sysimage/root/drivers/ であることに注意してください。

5. chroot 環境を終了します。

```
sh-3.00b# exit
```

6. システムを再起動します。

システムの起動を阻止し、root デバイスがマウントした後にこのドライバーを読み込めないようにするには、以下の手順に従います。

手順33.8 レスキューモードでのドライバーのブラックリスト登録

1. linux rescue rdblacklist=driver_nameコマンドを使用してレスキュー モード を起動します。driver_name はブラックリストに登録する必要のあるドライバーです。手順33.1「レスキューモードでの起動」 の手順に従って、システムの root パーティションを読み書きモードでマウントしていることを確認します。

2. vi エディターで /boot/grub/grub.conf ファイルを開きます。

```
sh-3.00b#  vi /boot/grub/grub.conf
```

3. システムの起動に使用するデフォルトのカーネルを特定します。各カーネルは grub.conf ファイルに、タイトル で始まる行のグループで指定します。デフォルトのカーネルは、ファイルの開始直後に default パラメーターで指定します。値が 0 の場合は、1 番目の行グループで説明されるカーネルを参照し、値 1 は 2 番目のグループで説明されるカーネルを指します。また、高い値は、後続のカーネルを順次参照します。

4. グループの カーネル 行を編集して、オプション rdblacklist=driver_name を追加します。driver_name はブラックリストに指定する必要のあるドライバーです。以下に例を示します。

```
kernel /vmlinuz-2.6.32-71.18-2.el6.i686 ro root=/dev/sda1 rhgb quiet
rdblacklist=driver_name
```

5. 以下を入力してファイルを保存し、vi エディターを終了します。

```
:wq
```

6. 以下のコマンドを実行して、root パーティションをマウントした後に ドライバーをブラックリストに登録する新しい /etc/modprobe.d/driver_name.conf を作成します。

```
echo "install driver_name" > /mnt/sysimage/etc/modprobe.d/driver_name.conf
```

7. システムを再起動します。

第34章 REAR (RELAX-AND-RECOVER)

ソフトウェアやハードウェア障害でシステムが破損した場合、システム管理者は新たなハードウェア環境上で完全に機能する状態にシステムを復元するために以下の 3 つのタスクを実行する必要があります。

1. 新規ハードウェア上でレスキューシステムを起動する
2. オリジナルのストレージレイアウトを複製する
3. ユーザーおよびシステムファイルの復元

ほとんどのバックアップソフトウェアは、3 番目の問題しか解決しません。最初の 2 番目の問題を解決するには、障害復旧およびシステム移行ユーティリティーである Relax-and-Recover(ReaR) を使用します。

バックアップソフトウェアはバックアップを作成します。ReaR は レスキューシステム を作成してバックアップソフトウェアを補完します。新しいハードウェアでレスキューシステムを起動すると、復元プロセスを開始する rear recover コマンドを実行できます。このプロセス中に ReaR はパーティションのレイアウトとファイルシステムを複製し、バックアップソフトウェアが作成したバックアップからのユーザーおよびシステムファイルの復元を促進し、最後にブートローダーをインストールします。デフォルトでは、ReaR が作成したレスキューシステムはストレージレイアウトとブートローダーのみを復元し、実際のユーザーおよびシステムファイルは復元しません。

本章では、ReaR の使用方法を説明します。

34.1. 基本的な REAR の使用方法

34.1.1. ReaR のインストール

root で以下のコマンドを実行して、rear パッケージをインストールします。

```
~]# yum install rear
```

34.1.2. ReaR の設定

ReaR は /etc/rear/local.conf ファイルで設定します。以下の行を追加してレスキューシステムの設

定を指定します。

```
OUTPUT=output format
OUTPUT_URL=output location
```

output format をレスキューシステムの形式に置き換えます。たとえば、ISO ディスクイメージの場合は ISO、起動可能な USB であれば USB などにします。

output location を、配置先に置き換えます。たとえば、ローカル ファイルシステムディレクトリーの file:///mnt/rescue_system/、SFTP ディレクトリーの場合は sftp://backup:.0.0/ などにします。

例34.1 レスキューシステムの形式および場所の設定

ReaR がレスキューシステムを ISO イメージとして /mnt/rescue_system/ ディレクトリーに出力するように設定するには、以下の行を /etc/rear/local.conf ファイルに追加します。

```
OUTPUT=ISO
OUTPUT_URL=file:///mnt/rescue_system/
```

オプション一覧は、man ページ rear(8) の「Rescue Image Configuration」のセクションを参照してください。

34.1.3. レスキューシステムの作成

以下の例では、出力結果が詳細モードとなるレスキューシステムを作成する方法を示しています。

```
~]# rear -v mkrescue
Relax-and-Recover 1.17.2 / Git
Using log file: /var/log/rear/rear-rhel68.log
mkdir: created directory `/var/lib/rear/output'
Creating disk layout
Creating root filesystem layout
TIP: To login as root via ssh you need to set up /root/.ssh/authorized_keys or
SSH_ROOT_PASSWORD in your configuration file
Copying files and directories
Copying binaries and libraries
Copying kernel modules
Creating initramfs
Making ISO image
Wrote ISO image: /var/lib/rear/output/rear-rhel68.iso (82M)
Copying resulting files to file location
```

例34.1「レスキューシステムの形式および場所の設定」 の設定で、ReaR は上記を出力します。最後の 2 行は、レスキューシステムが正常に作成され、設定されたバックアップの場所である /mnt/rescue_system/ にコピーされていることを確認します。システムのホスト名は rhel-68 であるため、バックアップの場所には、レスキューシステムと補助ファイルが含まれる rhel-68/ ディレクトリーが含まれるようになりました。

```
~]# ls -lh /mnt/rescue_system/rhel68/
total 82M
-rw-------. 1 root root  202 May  9 11:46 README
-rw-------. 1 root root 160K May  9 11:46 rear.log
-rw-------. 1 root root  82M May  9 11:46 rear-rhel68.iso
-rw-------. 1 root root  275 May  9 11:46 VERSION
```

レスキューシステムを外部メディアに移動して、障害の際になくならないようにします。

34.1.4. ReaR のスケジューリング

ReaR が cron ジョブスケジューラーを使用して定期的にレスキューシステムを作成するようにするには、以下の行を /etc/crontab ファイルに追加します。

```
minute hour day_of_month month day_of_week root /usr/sbin/rear mkrescue
```

上記のコマンドを cron 時間指定（「cron ジョブの設定」で詳細に説明）に置き換えます。

例34.2 ReaR のスケジューリング

ReaR が平日の 22:00 時にレスキューシステムを作成させるには、以下の行を /etc/crontab ファイルに追加します。

```
0 22 * * 1-5 root /usr/sbin/rear mkrescue
```

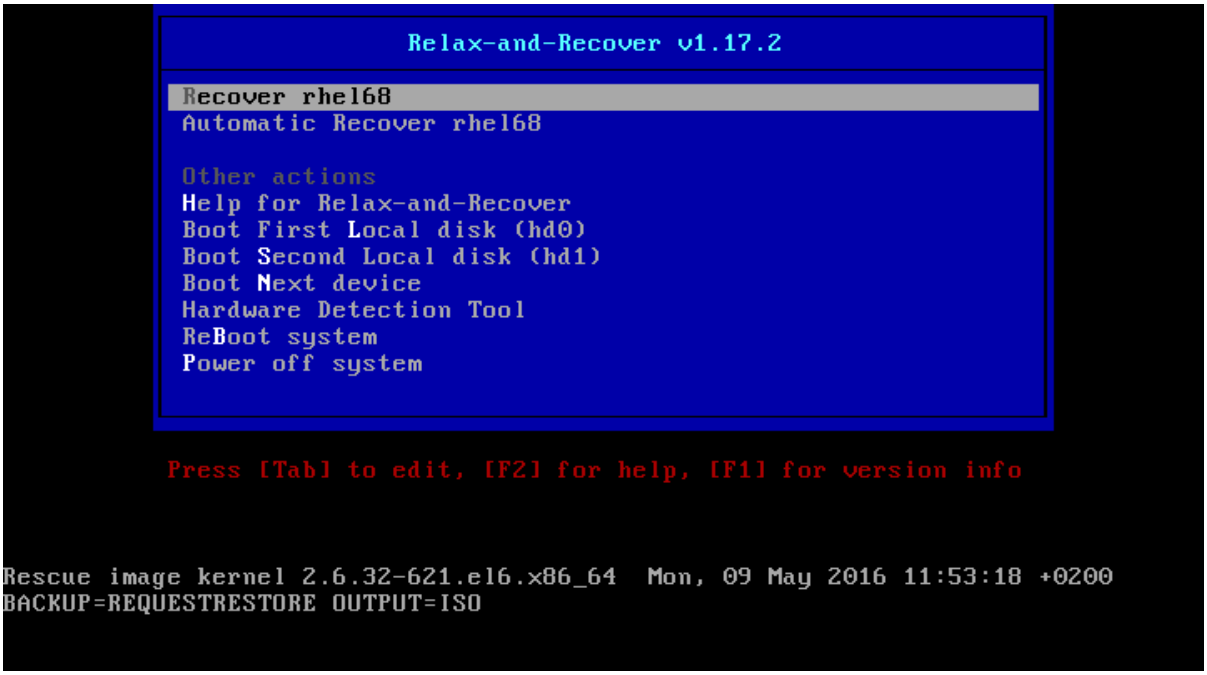
34.1.5. システムレスキューの実行

復旧または移行を実行するには、以下の手順を行います。

1. 新しいハードウェア上でレスキューシステムを起動します。たとえば、ISO イメージを DVD に書き込み、その DVD から起動します。

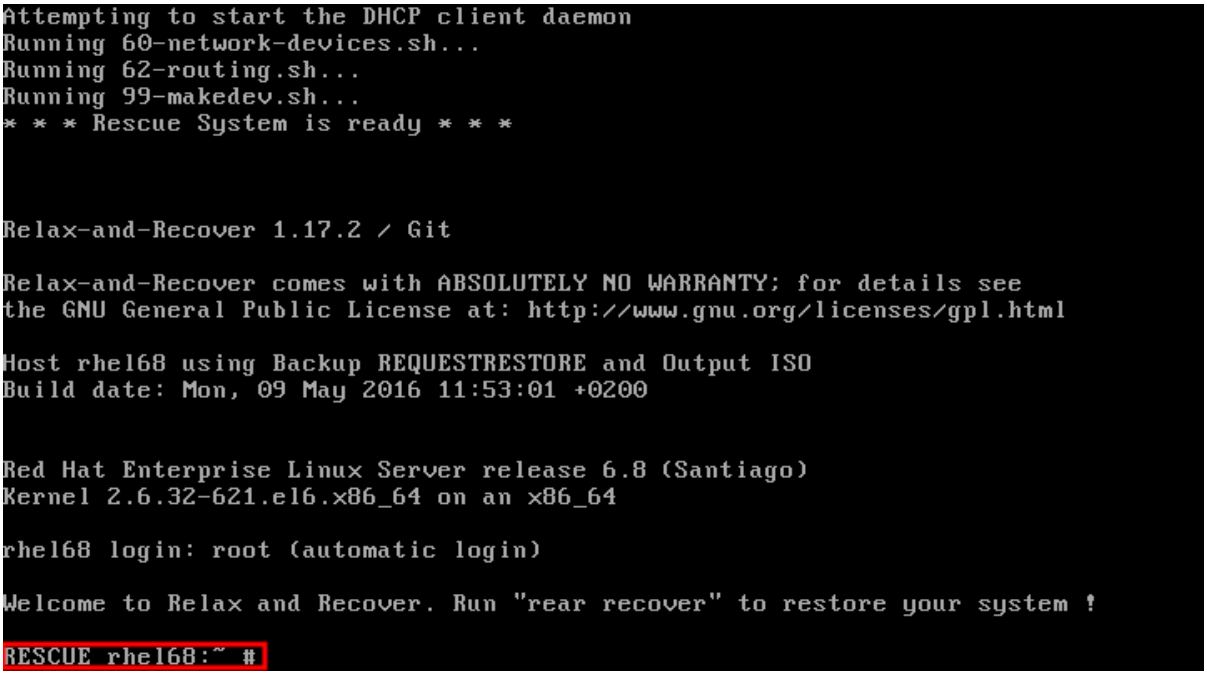
2. コンソールのインターフェースで "Recover" オプションを選択します。

図34.1 レスキューシステムのメニュー



3. 以下のプロンプトが表示されます。

図34.2 レスキューシステムのプロンプト



警告

次のステップでリカバリーを開始すると、元に戻すことができなくなり、システムの物理ディスクに保存されていたものが失われます。

4. **rear recover** コマンドを実行して復元または移行を実行します。するとレスキューシステムがパーティションレイアウトとファイルシステムを再作成します。

図34.3 レスキューシステム: "rear recover" の実行

```
RESCUE rhel68:~ # rear recover
Relax-and-Recover 1.17.2 / Git
Using log file: /var/log/rear/rear-rhel68.log
NOTICE: Will do driver migration
Comparing disks.
Disk configuration is identical, proceeding with restore.
Start system layout restoration.
Creating partitions for disk /dev/sda (msdos)
Creating LVM PV /dev/sda2
Restoring LVM VG VolGroup
Sleeping 3 seconds to let udev or systemd-udevd create their devices...
Creating ext4-filesystem / on /dev/mapper/VolGroup-lv_root
Mounting filesystem /
Creating ext4-filesystem /boot on /dev/sda1
Mounting filesystem /boot
Creating swap on /dev/mapper/VolGroup-lv_swap
Disk layout created.
Please start the restore process on your backup host.

Make sure that you restore the data into '/mnt/local' instead of '/' because the
hard disks of the recovered system are mounted there.

Please restore your backup in the provided shell and, when finished, type exit
in the shell to continue recovery.
rear> _
```

5. バックアップから **/mnt/local/** ディレクトリーにユーザーおよびシステムファイルを復元します。

例34.3 ユーザーおよびシステムファイルの復元

この例では、バックアップファイルは 「内部バックアップメソッドの設定」 の説明に従って作成された **tar** アーカイブになります。まず、アーカイブをストレージからコピーし、ファイルを **/mnt/local/** に展開し、アーカイブを削除します。

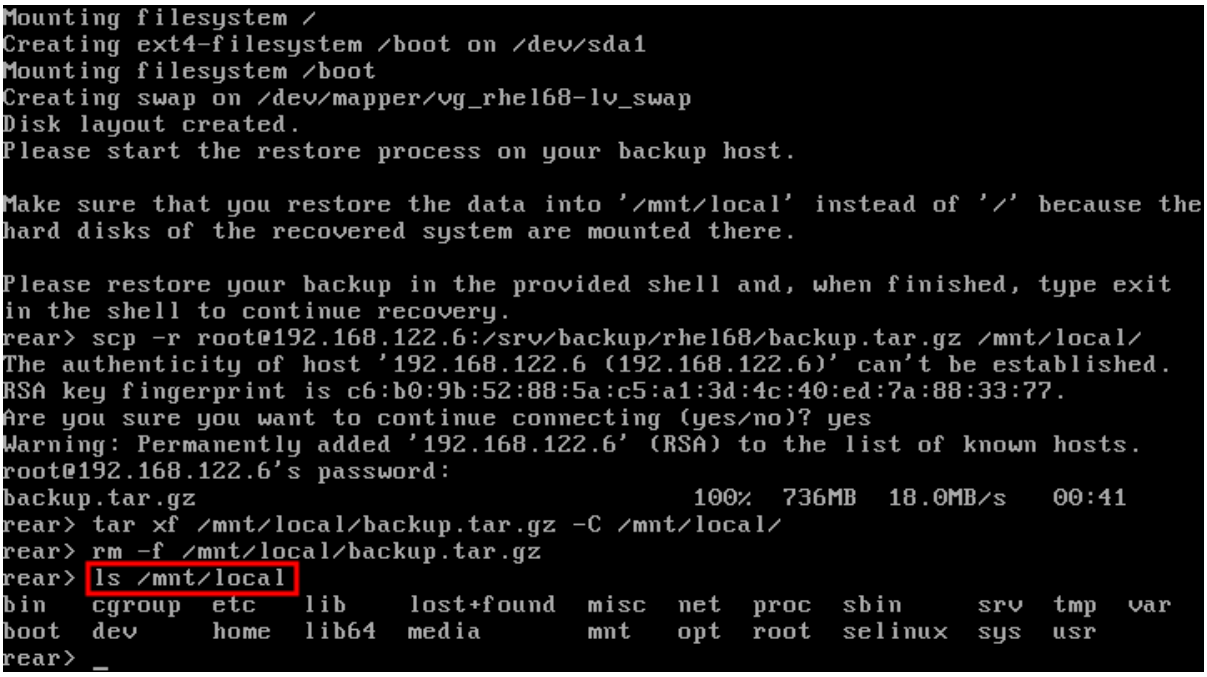
```
~]# scp root@192.168.122.6:/srv/backup/rhel68/backup.tar.gz /mnt/local/
~]# tar xf /mnt/local/backup.tar.gz -C /mnt/local/
~]# rm -f /mnt/local/backup.tar.gz
```

> 新規ストレージは、アーカイブと展開ファイルの両方を格納できるサイズである必要があります。

6. ファイルが復元されたことを確認します。

```
~]# ls /mnt/local/
```

図34.4 レスキューシステム: バックアップからのユーザーおよびシステムファイルの復元



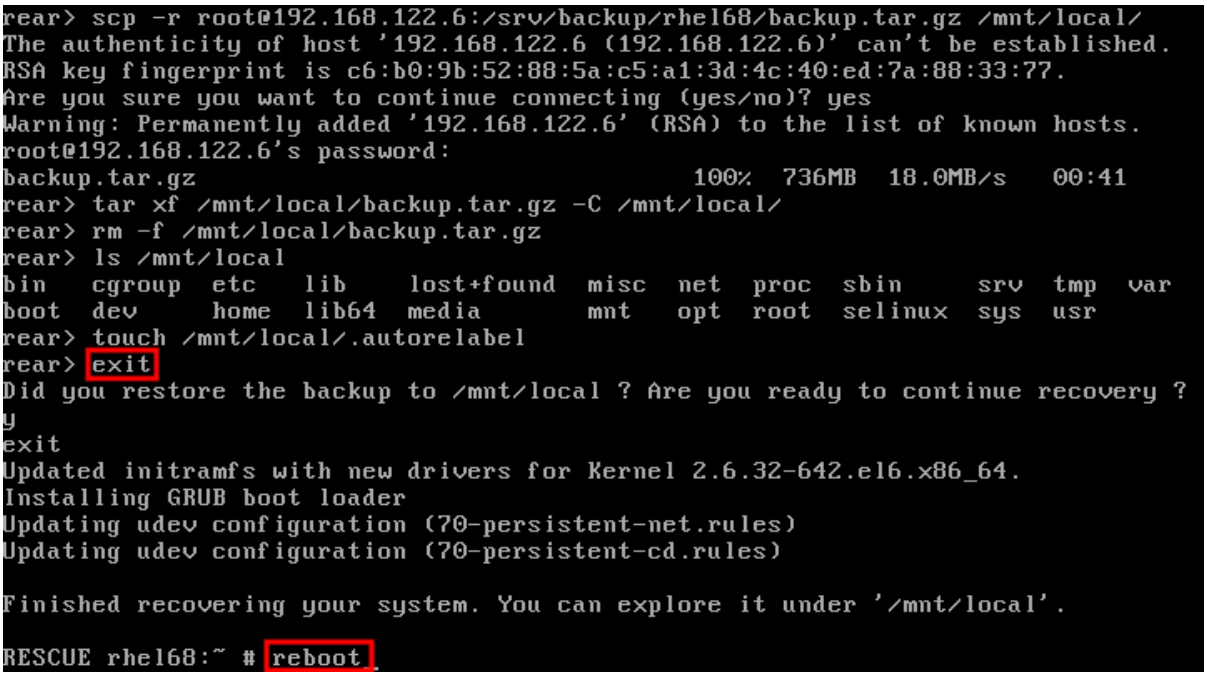
7. 次回の起動時に SELinux がファイルに再度ラベル付するようにします。

```
~]# touch /mnt/local/.autorelabel
```

> そうしないと、/etc/passwd ファイルの SELinux コンテキストが正しくない可能性があるため、システムにログインできなくなる可能性があります。

8. リカバリーを完了してシステムを再起動します。

図34.5 レスキューシステム: リカバリーの終了



その後、ReaRはブートローダーを再インストールします。再起動すると、SELinux がファイルシステム全体に再ラベル付けされます。

これでリカバリーしたシステムにログインできるようになります。

34.2. REAR をバックアップソフトウェアの統合

ReaR の主な目的はレスキューシステムを作成することですが、バックアップソフトウェアと統合することも可能です。統合は、ビルトイン、サポート対象、サポート対象外の各バックアップ方法で異なります。

34.2.1. ビルトインバックアップの場合

ReaR には、ビルトインまたは内部のバックアップメソッドが同梱されます。このメソッドは ReaR と完全に統合されており、以下の利点があります。

- *rear mkbackup コマンドを 1 つ使用してレスキューシステムと完全システムバックアップを作成できます。*
- *レスキューシステムが自動でバックアップからファイルを復元します。*

このため、ReaR はレスキューシステムと完全システムバックアップの両方の作成プロセスを処理できます。

34.2.1.1. 内部バックアップメソッドの設定

ReaR が内部バックアップメソッドを使用するようにするには、以下の行を /etc/rear/local.conf に追加します。

```
BACKUP=NETFS
BACKUP_URL=backup location
```

これらの行により、ReaR が tar コマンドを使用して完全システムバックアップのあるアーカイブを作成するようになります。backup location を、man ページの rear(8)の「Backup Software Integration」セクションのいずれかのオプションに置き換えます。バックアップの場所に十分な空き領域があるようにしてください。

例34.4 tar バックアップの追加

「基本的な ReaR の使用方法」 の例を拡張するには、ReaR が tar 完全システムバックアップを /srv/backup/ ディレクトリーに出力するようにします。

```
OUTPUT=ISO
OUTPUT_URL=file:///mnt/rescue_system/
BACKUP=NETFS
BACKUP_URL=file:///srv/backup/
```

内部バックアップメソッドでは、さらなる設定が可能です。

- 新規バックアップの作成時にこれまでのバックアップアーカイブを維持しておくようにするには、以下の行を追加します。

  ```
  NETFS_KEEP_OLD_BACKUP_COPY=y
  ```

- デフォルトでは、ReaR は実行時に毎回、完全バックアップを作成します。変更分のみをバックアップする増分にするには、以下の行を追加します。

  ```
  BACKUP_TYPE=incremental
  ```

これにより、NETFS_KEEP_OLD_BACKUP_COPY が自動的に y に設定されます。

- 増分バックアップに加えて、完全バックアップを定期的に実行するには、以下の行を追加します。

```
FULLBACKUPDAY="Day"
```

"Day" を "Mon"、"Tue"、"Wed"、"Thu" のいずれかに置き換えます。「金」、「土」、「日」。

- ReaR は、レスキューシステムとバックアップの両方を ISO イメージに含めることもできます。これを行うには、BACKUP_URL ディレクティブを iso:///backup/ に設定します。

```
BACKUP_URL=iso:///backup/
```

これはレスキューシステムがリカバリー中にバックアップをフェッチする必要がないことから、完全システムバックアップの一番簡単なメソッドになります。ただし、ストレージに十分なスペースが必要になります。また、単発の ISO バックアップは増分とすることができません。

注記

現在、ReaR は ISO イメージのコピーを 2 つ作成するため、ストレージをさらに 2 倍消費します。詳細は、『Red Hat Enterprise Linux 6 リリースノート』『の「ReaR が ISO イメージを作成』 する点」を参照してください。

例34.5 単一 ISO のレスキューシステムおよびバックアップの設定

以下の設定では、単一の ISO イメージとしてレスキューシステムとバックアップファイルが /srv/backup/ ディレクトリーに作成されます。

```
OUTPUT=ISO
OUTPUT_URL=file:///srv/backup/
BACKUP=NETFS
BACKUP_URL=iso:///backup/
```

- tar の代わりに rsync を使用するには、以下の行を追加します。

```
BACKUP_PROG=rsync
```

増分バックアップは tar 使用時にのみサポートされることに注意してください。

34.2.1.2. 内部バックアップメソッドを使用したバックアップの作成

BACKUP=NETFS を設定すると、ReaR はレスキューシステム、バックアップファイル、またはその両方を作成できます。

- **レスキューシステムのみ を作成するには、以下のコマンドを実行します。**

```
rear mkrescue
```

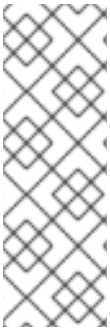
- **バックアップのみ を作成するには、以下のコマンドを実行します。**

```
rear mkbackuponly
```

- **レスキューシステムとバックアップ を作成するには、以下のコマンドを実行します。**

```
rear mkbackup
```

ReaR によるバックアップの作成は、NETFS メソッドの使用時のみ可能となります。ReaR は他のバックアップメソッドを開始することはできません。



注記

復元時には、BACKUP=NETFS 設定で作成したレスキューシステムは、rear recover の実行前にバックアップが存在することを前提としています。したがって、レスキューシステムが起動したら、BACKUP_URL で指定したディレクトリーにバックアップファイルをコピーします（単一 ISO イメージ使用時を除く）。その後、rear recover のみを実行します。

不要なレスキューシステムを再作成しないためには、最後にレスキューシステムが作成されてからストレージレイアウトが変更されたかどうかを確認します。以下のコマンドを実行します。

```
~]# rear checklayout
~]# echo $?
```

- ゼロ以外のステータスは、ディスクレイアウトに変更があったことを示します。また、ReaR 設定が変更した場合でも、ゼロ以外のステータスが返されます。

重要

rear checklayout コマンドはレスキューシステムが出力ロケーションに存在するかどうかを確認しないため、存在しない場合でも 0 を返すことができます。このため、レスキューシステムが利用可能であることを保証するのではなく、最後にレスキューシステムが作成されてからレイアウトに変更がないことのみが保証されます。

例34.6 rear checklayout の使用

レイアウトに変更があった場合にのみレスキューシステムを作成するようにするには、以下のコマンドを使用します。

```
~]# rear checklayout || rear mkrescue
```

34.2.2. サポート対象のバックアップメソッド

NETFS 内部バックアップメソッドのほかに、ReaR はいくつかの外部バックアップメソッドもサポートしています。この場合、レスキューシステムはバックアップから自動的にファイルを復元しますが、ReaR を使ってバックアップの作成を開始することはできません。

サポート対象の外部バックアップメソッドの一覧および設定オプションは、rear(8) の man ページの「Backup Software Integration」セクションを参照してください。

34.2.3. サポート対象外のバックアップメソッド

サポート対象外のバックアップメソッドでは、以下の 2 つのオプションが可能です。

1. レスキューシステムでは、ユーザーに手動でファイルを復元するようプロンプトが出ます。このシナリオは、「基本的な ReaR の使用方法」に記載されているものですが、バックアップファイルの形式が tar アーカイブとは異なる場合があります。

2. ユーザーが提供するカスタムコマンドを ReaR が実行します。これを設定するには、BACKUP ディレクティブを EXTERNAL に設定します。次に、EXTERNAL_BACKUP お

よび EXTERNAL_RESTORE ディレクティブを使用してバックアップおよび復元中に実行するコマンドを指定します。またオプションで、EXTERNAL_IGNORE_ERRORS と EXTERNAL_CHECK のディレクティブも指定します。設定例は、/usr/share/rear/conf/default.conf を参照してください。

付録A ネットワークデバイス命名における一貫性

Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;LinuxRed Hat Enterprise Linux 6nbsp;6 は、ネットワークインターフェースに対して一貫性のあるネットワークデバイス命名を提供します。この機能により、インターフェースの配置と区別が容易になるようにシステム上のネットワークインターフェース名が変更されます。

従来の Linux のネットワークインターフェースには次のように列挙されます。 eth[0123...]ただし、この名前がシャーシの実際のラベルに対応しているとは限りません。複数のネットワークアダプターを使用する最新のサーバープラットフォームでは、このインターフェースの非決定論的および反直感的な命名が行われています。これは、マザーボードに組み込まれたネットワークアダプター (Lan-on-Motherboard または LOM) とアドイン (シングルおよびマルチのポート) アダプターの両方に影響します。

新しい命名規則は、埋め込みまたは PCI スロットのいずれであっても、物理的な場所に基づいてネットワークインターフェースに名前を割り当てます。この命名規則に変換することで、システム管理者はネットワークポートの物理的な場所で推測したり、各システムを変更して、一貫した順序で名前を変更する必要がなくなります。

この機能は、biosdevname プログラムを介して実装され、すべての組み込みネットワークインターフェース、PCI カードネットワークインターフェース、および仮想機能ネットワークインターフェースの名前を既存のものから変更します。 eth[0123...] 表A.1「新しい命名規則」 に記載の新しい命名規則に対して、以下を行います。

表A.1 新しい命名規則

デバイス	旧式の名前	新しい名前
組み込み型ネットワークインターフェース (LOM)	**eth[0123...]**	**em[1234...]**[a]
PCI カードネットワークインターフェース	**eth[0123...]**	**p<*slot*>p<*ethernet port*>**[b]
仮想機能	**eth[0123...]**	**p<*slot*>p<*ethernet port*>_<*virtual interface*>**[c]

[a] 新しい列挙は **1** から始まります。

[b] 以下に例を示します。**p3p4**

[c] 以下に例を示します。**p3p4_1**

システム管理者は、デバイス名を任意の値に変更するために、引き続き /etc/udev/rules.d/70-

persistent-net.rules にルール を書き込むことができます。これは、この物理的な場所の命名規則よりも優先されます。

A.1. 影響を受けるシステム

一貫性のあるネットワークデバイスの命名は、Dell、C Series、および Precision Workstation システムのセットに対してデフォルトで有効になります。Dell システムへの影響についての詳細は、を参照してください https://access.redhat.com/kb/docs/DOC-47318。

その他のすべてのシステムでは、デフォルトでは無効になっています。詳細は、「システム要件」および「機能の有効化および無効化」 を参照してください。

システムの種類に関係なく、Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;LinuxRed Hat Enterprise Linux 6nbsp;6 ゲストを Red Hat Enterprise Linux 5 ホストで実行する場合、仮想マシン BIOS は SMBIOS 情報を提供しないため、デバイスの名前は変更されません。Red Hat Enterprise Linux 6.0 から Red Hat Enterprise Linux 6.1 へのアップグレードは影響を受けておらず、古いものとなります。 eth[0123...] 命名規則は引き続き使用されます。

A.2. システム要件

biosdevname プログラムはシステムの BIOS からの情報、特に SMBIOS 内に収納されている type 9 (システムスロット) フィールドと type 41 (オンボードデバイス拡張情報) フィールドからの情報を使用します。システムの BIOS に SMBIOS のバージョン 2.6 もしくはそれ以降がなければ、新しい命名規則は使用されません。ほとんどの旧型のハードウェアは、必要な SMBIOS バージョンとフィールド情報がある BIOS を欠いているため、この機能をサポートしていません。BIOS または SMBIOS バージョンの詳細については、ご使用のハードウェアの製造元にご連絡ください。

この機能を有効にするには、biosdevname パッケージもインストールする必要があります。biosdevname パッケージは、Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;Linux Red Hat Enterprise Linux 6nbsp;6 の ベース パッケージグループの一部です。最小インストール を 除くすべてのインストールオプションには、このパッケージが含まれます。これは、Red Hat Enterprise Linux 6.0 から RHEL 6.1 へのアップグレードにインストールされません。

A.3. 機能の有効化および無効化

通常デフォルトで設定されている Dell システムで一貫性のあるネットワークデバイスの命名を無効にするには、インストール中およびインストール後に、ブートコマンドラインに以下のオプションを渡します。

```
biosdevname=0
```

最小要件を満たす他のシステムタイプ（「システム要件」を参照）でこの機能を有効にするには、インストール中およびインストール後に、ブートコマンドラインに以下のオプションを渡します。

```
biosdevname=1
```

システムが最小要件を満たしている場合を除き、このオプションは無視され、システムは従来のネットワークインターフェース名の形式で起動します。

biosdevname インストールオプションが指定されている場合、システムの有効期間中にブートオプションとして留まる必要があります。

A.4. 管理者向け注意点

多くのシステムカスタム化ファイルはネットワークインターフェース名を含んでいる場合があるので、システムを旧式規則から新しい規則に移す際には、更新が必要となります。新しい命名規則を使用する場合は、カスタム **iptables** ルール、**irqbalance** を変更するスクリプト、および同様の設定ファイルなど、ネットワークインターフェース名をエリアで更新する必要があります。また、インストール用にこの変更を有効にするには、**ksdevice** パラメーターを介してデバイス名を使用する既存のキックスタートファイルを変更する必要があります。これらのキックスタートファイルは、ネットワークデバイスの **MAC** アドレスまたはネットワークデバイスの新しい名前を使用するように更新する必要があります。

Red Hatnbsp;Hat では、この機能をインストール時の選択とみなすことを強く推奨します。これを行うシステム管理者が、最小要件を満たすシステムで、**/etc/udev/rules.d/70-persistent-net.rules** ファイルと **HWADDR** 行をすべての **/etc/sysconfig/network-scripts/ifcfg-*** ファイルから削除します。さらに、この新しい命名規則を使用するように **ifcfg-*** ファイルの名前を変更します。再起動後に新しい名前が有効になります。ネットワークインターフェース名が含まれる可能性のあるカスタムスクリプト、**iptables** ルール、およびサービス設定ファイルを更新するのを忘れないようにしてください。

付録B RPM

RPM Package Manager (RPM)は、Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux やその他の Linux システムおよび UNIX システムで実行するオープンパッケージシステムです。Red Hat, Inc. および Fedora Project は、他のベンダーが独自の製品に RPM を使用することを推奨しています。RPM is distributed under the terms of the GPL (GNU General Public License).

RPM Package Manager は、RPM 形式 と連携するように構築されたパッケージでのみ機能します。RPM 自体は、事前にインストールされた rpm パッケージとして提供されます。RPM を使用すると、エンドユーザーはシステムの更新を簡素化します。RPM パッケージのインストール、アンインストール、およびアップグレードは、短いコマンドで実行できます。RPM はインストールされたパッケージとそのファイルのデータベースを維持するため、システムで強力なクエリーと検証を呼び出すことができます。

Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;LinuxRed Hat Enterprise Linux 6nbsp;6 の RPM パッケージ形式が改善されました。RPM パッケージは、XZ ロスレスデータ圧縮形式を使用して圧縮され、圧縮解除時の圧縮や CPU の使用量が減り、パッケージの署名や検証用に SHA-256 などの強力なハッシュアルゴリズムをサポートするようになりました。

RPM の代わりに YUM を使用（可能な場合）

ほとんどのパッケージ管理タスクでは、Yum パッケージマネージャーは RPM よりも同等で、大きな機能やユーティリティーを提供します。yum は、複雑なシステム依存関係の解決も実行および追跡し、RPM を使用してパッケージのインストールおよび削除を行う場合でもシステムの整合性チェックを行い、強制的にチェックを行います。このような理由から、パッケージ管理タスクを実行するときはいつでも RPM の代わりに Yum を使用することを強く推奨します。8章Yumを参照してください。

グラフィカルインターフェースを使用する場合は、Yum をバックエンドとして使用する PackageKit GUI アプリケーションを使用してシステムのパッケージを管理できます。詳しくは、9章PackageKit を参照してください。

正しいアーキテクチャーで RPM パッケージをインストールする！

パッケージをインストールする際には、オペレーティングシステムおよびプロセッサーアーキテクチャーと互換性があることを確認してください。これは通常、パッケージ名を確認して判断できます。以下の例の多くは、AMD64/Intel 64 コンピューターアーキテクチャー用にコンパイルされた RPM パッケージを示しています。そのため、RPM ファイル名は x86_64.rpm で終わります。

アップグレード中、RPM は設定ファイルを誤って処理するため、通常の .tar.gz ファイルで実行できないカスタマイズも失われないようにします。

開発者は、RPM を使用して、ソフトウェアソースコードを取り、エンドユーザー向けのソースパッケージとバイナリーパッケージにパッケージ化できます。このプロセスは非常にシンプルで、単一のファイルと作成するオプションのパッチから実行されます。ビルド命令とともに、元 のソースとパッチの区別が明確になり、新しいバージョンのソフトウェアがリリースされると、パッケージのメンテナンスが容易になります。

RPM コマンドの実行は ROOT として実行する必要があります。

RPM はシステムに変更を加えるため、RPM パッケージのインストール、削除、またはアップグレードを行うには root でログインする必要があります。

B.1. RPM 設計ゴール

RPM の使用方法を理解するには、RPM の設計目的を理解すると便利です。

アップグレード可能性

RPM を使用すると、完全に再インストールせずにシステムの個別コンポーネントをアップグレードできます。Red Hat Enterprise Linuxnbsp;Hat Enterprise Red Hat Enterprise Linuxnbsp;Linux などの RPM に基づいてオペレーティングシステムの新しいリリースを取得する場合は、マシンのオペレーティングシステムの新規コピーを再インストールする必要はありません（他のパッケージシステムに基づくオペレーティングシステムを使用する必要がある場合があります）。RPM は、システムのインテリジェントな完全自動インプレースアップグレードを可能にします。さらに、パッケージ内の設定ファイルはアップグレード後も保持されるため、カスタマイズは失われません。パッケージのアップグレードに必要な特別なアップグレードファイルはありません。システム上のパッケージのインストールとアップグレードの両方に同じ RPM ファイルが使用されるためです。

強力なクエリ

RPM は、強力なクエリーオプションを提供するように設計されています。パッケージや特定のファイルのみに対してデータベース全体の検索を実行できます。また、ファイルが属するパッケー

ジや、パッケージの出所を簡単に見つけることもできます。RPM パッケージに含まれるファイルは圧縮アーカイブにあり、カスタムバイナリーヘッダーにはパッケージとそのコンテンツに関する有用な情報が含まれており、個々のパッケージを素早く簡単にクエリーできます。

システムの検証

もう 1 つの強力な RPM 機能は、パッケージを検証する機能です。一部のパッケージで重要なファイルを削除した場合には、パッケージを確認することができます。その後、必要に応じてパッケージを再インストールできる時点で、異常が通知されます。変更した設定ファイルは再インストール時に保持されます。

純粋なソース

重要な設計目的は、ソフトウェアの元の作成者によって配布される Pristine ソフトウェアソースを使用できるようにしたことでした。RPM では、元のソースと、使用されたパッチ、完全なビルド命令があります。これは、いくつかの理由で重要な利点です。たとえば、新しいバージョンのプログラムがリリースされると、コンパイルを行うために必ずしもゼロから作業を開始する必要はありません。パッチを確認して、必要 な 内容を確認できます。コンパイルされたすべてのデフォルトと、ソフトウェアが適切にビルドできるように加えられたすべての変更は、この手法を使用して簡単に表示できます。

ソースを純粋に保つことの目的は開発者の場合にのみ重要だと思うかもしれませんが、エンドユーザーにとっても品質の高いソフトウェアになります。

B.2. RPM の使用

RPM には基本的な動作モードが 5 つあります（パッケージの構築をカウントしません）：インストール、アンインストール、アップグレード、クエリー、および検証を行います。このセクションでは、各モードの概要について説明します。詳細およびオプションの詳細は、rpm --help または man rpm を試してください。RPM の詳細は、「その他のリソース」 を参照してください。

B.2.1. RPM パッケージの検索

RPM パッケージを使用する前に、パッケージの検索場所を知っている必要があります。インターネット検索は多くの RPM リポジトリーを返しますが、Red Hat RPM パッケージを検索する場合は、以下の場所で確認できます。

- Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux インストールメディアには、インストール可能な RPM が多数含まれています。

- YUM パッケージマネージャーで提供される初期 RPM リポジトリー公式の Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux パッケージリポジトリーの使用方法は、8章Yum を参照してください。

- Extra Packages for Enterprise Linux(EPEL)は、Red Hat Enterprise Linux 用の高品質アドオンパッケージを提供するコミュニティーの作業です。EPEL RPM パッケージの詳細は を参照 http://fedoraproject.org/wiki/EPEL してください。

- Red Hatnbsp;Hat と互換性のない非公式のリポジトリーでは、RPM パッケージも提供されます。

サードパーティーのリポジトリーおよびパッケージの互換性

Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux システムで使用するサードパーティーのリポジトリーを検討する場合は、リポジトリーをパッケージソースとして追加する前に、リポジトリーの Web サイトに注意を払ってください。代替パッケージリポジトリーでは、Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux リポジトリーにすでに含まれているパッケージを含め、同じソフトウェアの異なる互換性のないバージョンが提供されることがあります。

- Red Hat エラータページ は から http://www.redhat.com/apps/support/errata/ 入手できます。

B.2.2. インストールおよび設定ガイド

RPM パッケージには、通常 tree-1.5.3-2.el6.x86_64.rpm などのファイル名があります。ファイル名には、パッケージ名（ツリー）、バージョン(1.5.3)、リリース(2)、オペレーティングシステムのメジャーバージョン(el6)、および CPU アーキテクチャー(x86_64)が含まれます。

rpm の-U オプションを使用すると、以下を行います。

- システムの既存の古いパッケージを、新しいバージョンにアップグレードします。

- 古いバージョンがインストールされていなくてもパッケージをインストールします。

つまり、rpm -U &lt ;rpm_file> は、パッケージに適した アップグレード または インストール の

いずれかの機能を実行できます。

tree-1.5.3-2.el6.x86_64.rpm パッケージが現在のディレクトリーにある場合は、root としてログインし、rpm の決定どおりに tree パッケージをアップグレードまたはインストールします。

```
rpm -Uvh tree-1.5.3-2.el6.x86_64.rpm
```

適切にフォーマットされた RPM のインストールに -UVH を使用

-v オプションおよび -h オプション（ -Uと組み合わせる）により、rpm はより詳細な出力を出力し、ハッシュ記号を使用して進捗メーターを表示します。

アップグレード/インストールに成功すると、以下の出力が表示されます。

```
Preparing...                ########################################### [100%]
   1:tree                   ########################################### [100%]
```

常に -I(INSTALL)オプションを使用して新しいカーネルパッケージをインストールします。

RPM では、パッケージのインストールには、前述した -U オプション（以前はアップグレードを示す -U オプション）と、-i オプション（インストール の必要あり）の 2 種類のオプションを利用できます。-U オプションはインストールおよびアップグレード機能の両方を使用するため、kernel パッケージ以外 のすべてのパッケージで rpm -Uvh を使用することが推奨されます。

-i オプションを使用して、アップグレードせずに新しいカーネルパッケージをインストール する必要があります。これは、-U オプションを使用してカーネルパッケージをアップグレードすると、以前の（以前の）カーネルパッケージが削除され、新しいカーネルに問題がある場合にシステムが起動できなくなる可能性があるためです。したがって、rpm -i < kernel_package> コマンドを 使用して、古い kernel パッケージを置き換え ず に新しいカーネルをインストールします。kernel パッケージのインストールの詳細は、30章カーネルの手動によるアップグレードを参照してください。

パッケージのインストールまたはアップグレード時に、パッケージの署名が自動的にチェックされます。署名は、パッケージが承認されたパーティーによって署名されていることを確認します。たとえ

ば、署名の検証に失敗した場合、以下のようなエラーメッセージが表示されます。

```
error: tree-1.5.3-2.el6.x86_64.rpm: Header V3 RSA/SHA256 signature: BAD, key ID
d22e77f2
```

新しいヘッダーのみの署名の場合は、以下のようなエラーメッセージが表示されます。

```
error: tree-1.5.3-2.el6.x86_64.rpm: Header V3 RSA/SHA256 signature: BAD,
key ID d22e77f2
```

署名を検証するために適切なキーがインストールされていない場合、メッセージには NOKEY という単語が含まれます。

```
warning: tree-1.5.3-2.el6.x86_64.rpm: Header V3 RSA/SHA1 signature: NOKEY, key ID
57bbccba
```

パッケージの署名の確認に関する詳細は、「パッケージの署名の確認」 を参照してください。

B.2.2.1. インストールされているパッケージ

同じ名前のパッケージがすでにインストールされている場合は、以下の出力が表示されます。

```
Preparing...                ########################################### [100%]
  package tree-1.5.3-2.el6.x86_64 is already installed
```

ただし、パッケージをインストールする場合は、--replacepkgs オプションを使用できます。これにより、RPM にエラーを無視するように指示します。

```
rpm -Uvh --replacepkgs tree-1.5.3-2.el6.x86_64.rpm
```

このオプションは、RPM からインストールされたファイルが削除された場合や、RPM から元の設定ファイルをインストールする場合に便利です。

B.2.2.2. 競合するファイル

別のパッケージですでにインストールされているファイルを含むパッケージのインストールを試みると、以下が表示されます。

```
Preparing... ##########################################################
 file /usr/bin/foobar from install of foo-1.0-1.el6.x86_64 conflicts
with file from package bar-3.1.1.el6.x86_64
```

RPM がこのエラーを無視するようにするには、--replacefiles オプションを使用します。

```
rpm -Uvh --replacefiles foo-1.0-1.el6.x86_64.rpm
```

B.2.2.3. 解決できない依存関係

RPM パッケージは、他のパッケージに依存する場合があります。つまり、適切に実行するために他のパッケージをインストールする必要があります。未解決の依存関係があるパッケージをインストールしようとすると、以下のような出力が表示されます。

```
error: Failed dependencies:
  bar.so.3()(64bit) is needed by foo-1.0-1.el6.x86_64
```

CD-ROM や DVD などの Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux インストールメディアからパッケージをインストールする場合は、依存関係が利用できる可能性があります。Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux インストールメディア、またはアクティブな Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux ミラーリングのいずれかで推奨されるパッケージを見つけ、これをコマンドに追加します。

```
rpm -Uvh foo-1.0-1.el6.x86_64.rpm    bar-3.1.1.el6.x86_64.rpm
```

両方のパッケージのインストールに成功すると、以下のような出力が表示されます。

```
Preparing...          ########################################### [100%]
   1:foo              ########################################### [ 50%]
   2:bar              ########################################### [100%]
```

--whatprovides オプションを試して、必要なファイルを含むパッケージを特定します。

```
rpm -q --whatprovides "bar.so.3"
```

bar.so.3 が含まれるパッケージが RPM データベースにある場合は、パッケージ名が表示されます。

```
bar-3.1.1.el6.i586.rpm
```

警告： パッケージのインストールの禁止

rpm で Failed dependencies エラーが表示されるパッケージをインストールするように強制 することができますが（ --nodeps オプションを使用）、これは推奨されず、通常はインストールされたパッケージの実行に失敗します。rpm --nodeps でパッケージのインストールまたは削除により、アプリケーションが誤作動したり、クラッシュしたりしてしまう可能性があり、重要なパッケージ管理の問題やシステム障害などの問題が生じる可能性があります。このような警告は、RPM、Yum、PackageKit（ RPM、Yum、PackageKit）が認識しているので、これらの警告を確認して、依存関係のアカウンティングが重要であるため、これらの警告を提案してください。Yum パッケージマネージャーは、依存関係の解決と、オンラインリポジトリーから依存関係をフェッチすることができるため、依存関係の解決を考慮せずに rpm によるアクションの実行よりも安全で、より容易かつスマートになります。

B.2.3. 設定ファイルの変更

RPM は設定ファイルを使ってパッケージのインテリジェントなアップグレードを実行するので、以下のメッセージが表示されることがあります。

```
saving /etc/foo.conf as /etc/foo.conf.rpmsave
```

このメッセージは、設定ファイルに加えた変更が、パッケージ内の新しい設定ファイルと前方互換性 がないことを意味します。そのため、RPM は元のファイルを保存し、新しいファイルをインストールしてください。システムが適切に機能し続けるようにするには、2 つの設定ファイル間の違いを調べ、できるだけ早く解決する必要があります。

また、RPM は、foo.conf.rpmnew などのようにパッケージの新しい 設定ファイルを保存し、変更されなかった設定ファイルを残すこともできます。通常は、変更した設定ファイルから新しい設定ファイルへの変更をマージすることで、変更した設定ファイルと新しい設定ファイル間で競合を解決する必要が あり ます。

古い バージョン番号（つまり、より新しいバージョンのパッケージがすでにインストールされている場合）のパッケージへのアップグレードを試みると、出力は以下のようになります。

```
package foo-2.0-1.el6.x86_64.rpm (which is newer than foo-1.0-1) is already installed
```

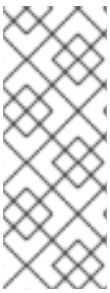
RPM を強制的にアップグレードするには、--oldpackage オプションを使用します。

```
rpm -Uvh --oldpackage foo-1.0-1.el6.x86_64.rpm
```

B.2.4. アンインストール

パッケージのアンインストールは、パッケージのインストールと同様に簡単です。シェルプロンプトで以下のコマンドを入力します。

```
rpm -e foo
```



RPM -E およびパッケージ名エラー

元のパッケージファイル **foo -1.0-1.el6.x86_64** の名前ではなく、パッケージ名 **foo** を使用していたことに注意してください。**rpm -e** コマンドおよび元のフルファイル名を使用してパッケージをアンインストールしようとすると、パッケージ名エラーが発生します。

別のインストール済みパッケージが削除しようとしているパッケージに依存すると、パッケージをアンインストールする際に依存関係エラーが発生する可能性があります。以下に例を示します。

```
rpm -e ghostscript
error: Failed dependencies:
        libgs.so.8()(64bit) is needed by (installed) libspectre-0.2.2-3.el6.x86_64
        libgs.so.8()(64bit) is needed by (installed) foomatic-4.0.3-1.el6.x86_64
        libijs-0.35.so()(64bit) is needed by (installed) gutenprint-5.2.4-5.el6.x86_64
        ghostscript is needed by (installed) printer-filters-1.1-4.el6.noarch
```

「解決できない依存関係」 で共有オブジェクトライブラリー（例： **< library_name > .so. <number >** ファイル）を検索する方法と同様に、この正確な構文を使用して **64** ビットの共有オブジェクトライブラリーを検索できます（ファイル名を引用符で囲むようにしてください）。

```
~]# rpm -q --whatprovides "libgs.so.8()(64bit)"
ghostscript-8.70-1.el6.x86_64
```

警告： パッケージのインストールの禁止

rpm で Failed dependencies エラーが表示されるパッケージを削除するように強制 することができますが（ --nodeps オプションを使用）、これは推奨されず、他のインストール済みアプリケーションの影響を引き起こす可能性があります。rpm --nodeps でパッケージのインストールまたは削除により、アプリケーションが誤作動したり、クラッシュしたりしてしまう可能性があり、重要なパッケージ管理の問題やシステム障害などの問題が生じる可能性があります。このような警告は、RPM、Yum、PackageKit（ RPM、Yum、PackageKit）が認識しているので、これらの警告を確認して、依存関係のアカウンティングが重要であるため、これらの警告を提案してください。Yum パッケージマネージャーは、依存関係の解決と、オンラインリポジトリーから依存関係をフェッチすることができるため、依存関係の解決を考慮せずに rpm によるアクションの実行よりも安全で、より容易かつスマートになります。

B.2.5. Freshening

Freshening はアップグレードに似ていますが、存在しないパッケージのみがアップグレードされます。シェルプロンプトで以下のコマンドを入力します。

```
rpm -Fvh foo-2.0-1.el6.x86_64.rpm
```

RPM の新規オプションは、コマンドラインで指定したパッケージのバージョンと、システムにインストールされているパッケージのバージョンを確認します。すでにインストールされているパッケージの新規バージョンが RPM の新規オプションで処理されると、そのパッケージが新しいバージョンにアップグレードされます。ただし、同じ名前のパッケージが存在しない場合、RPM のnewen オプションはパッケージをインストールしません。これ は、古いバージョンのパッケージがすでにインストールされているかどうかにかかわらず、アップグレードによってパッケージがインストールされるため、RPM のアップグレードオプションとは異なります。

フラッシュは、単一のパッケージまたはパッケージグループで機能します。多数の異なるパッケージをダウンロードし、システムにすでにインストールされているパッケージのみをアップグレードする場合は、新たにジョブを行います。したがって、RPM を使用する前にダウンロードしたグループから不要なパッケージを削除する必要はありません。

この場合は、.rpm glob で以下を実行します。*

```
rpm -Fvh *.rpm
```

その後、RPM はすでにインストールされているパッケージのみを自動的にアップグレードします。

B.2.6. クエリ

RPM データベースは、システムにインストールされているすべての RPM パッケージに関する情報を保存します。これは /var/lib/rpm/ ディレクトリーに保存され、インストールされているパッケージ、各パッケージのバージョン、およびインストール以降のパッケージ内のファイルへの変更の算出に使用されます。

このデータベースをクエリーするには、-q オプションを使用します。rpm -q package name コマンドは、インストールされているパッケージのパッケージ名、バージョン、リリース番号を表示します。& lt;package_name>たとえば、rpm -q ツリーを使用してインストールされたパッケージ ツリーをクエリーすると、以下の出力が生成されます。

```
tree-1.5.2.2-4.el6.x86_64
```

RPM man ページのサブ見出しである以下の Package Selection オプション （詳細は man rpm を参照してください）を使用して、クエリーをさらに改良したり、フィルタリングしたりできます。

- -a: 現在インストールされているパッケージをすべてクエリーします。
- -f & lt;file_name> - パッケージを所有する RPM データベースに < file_name> をクエリーし ます。ファイルの絶対パスを指定します（例： rpm -qf ls ではなく rpm -qf /bin/ls）。
- -p &lt ;package_file> - アンインストールしたパッケージ < package_file> をクエリーします。

クエリーされたパッケージに関する情報を表示する方法は複数あります。以下のオプションは、検索する情報のタイプを選択するために使用されます。これらは パッケージクエリー オプション と呼ばれます。

- -i は、名前、説明、リリース、サイズ、ビルド日、インストール日、ベンダー、その他のその他の情報など、パッケージ情報を表示します。
- -l は、パッケージに含まれるファイルの一覧を表示します。

- *-s は、パッケージ内の全ファイルの状態を表示します。*

- *-d は、パッケージにドキュメントとしてマークされているファイルの一覧を表示します（man ページ、情報ページ、README など）。*

- *-c は、設定ファイルとしてマークされているファイルの一覧を表示します。インストール後に編集したファイルで、パッケージをお使いのシステムに適用およびカスタマイズします（例： sendmail.cf、passwd、inittab など）。*

ファイルの一覧を表示するオプションについては、コマンドに -v を追加して、一般的な ls -l 形式でリストを表示します。

B.2.7. 検証

パッケージの検証は、パッケージからインストールしたファイルに関する情報と、元のパッケージからの同じ情報を比較します。また、検証により、各ファイルのファイルサイズ、MD5 合計、パーミッション、タイプ、所有者、およびグループを比較します。

コマンド rpm -V はパッケージを検証します。クエリーに一覧表示される 検証オプション のいずれかを使用して、検証するパッケージを指定できます。検証の簡単な使用方法は rpm -V ツリー です。これは、tree パッケージ内のすべてのファイルが、最初にインストールした時と同じであることを検証します。以下に例を示します。

- *特定のファイルを含むパッケージを確認するには、次のコマンドを実行します。*

```
rpm -Vf /usr/bin/tree
```

この例では、/usr/bin/tree は、パッケージのクエリーに使用されるファイルへの絶対パスです。

- *システム全体でインストールされたすべてのパッケージ（これには多少時間がかかります）を確認するには、以下のコマンドを実行します。*

```
rpm -Va
```

- *RPM パッケージファイルに対してインストール済みのパッケージを確認するには、次のコ*

マンドを実行します。

```
rpm -Vp tree-1.5.3-2.el6.x86_64.rpm
```

このコマンドは、RPM データベースが破損していると思われる場合に便利です。

すべてが適切に検証された場合、出力はありません。不一致がある場合は、それらが表示されます。出力の形式は 8 文字の文字列（"cは設定ファイルを示します）、ファイル名です。8 文字はそれぞれ、ファイルの 1 つの属性と RPM データベースに記録された属性の値を比較した結果を示します。1 回の期間(.)は、テストに合格したことを意味します。以下の文字は、特定の不一致を示しています。

- 5 - MD5 checksum
- S - ファイルサイズ
- L - シンボリックリンク
- T - ファイル変更時間
- D - device
- U - user
- G - group
- M - モード（パーミッションおよびファイルタイプを含む）
- ? - 読み取りできないファイル（ファイルパーミッションエラーなど）

出力が表示された場合は、最善の判断を使用して、パッケージを削除したり、再インストールするか、別の方法で問題を修正する必要があるかどうかを判断します。

B.3. パッケージの署名の確認

パッケージが破損していないか、改ざんされていないことを確認する場合は、シェルプロンプトで以下のコマンドを入力して md5sum のみを確認します（< rpm_file > は RPM パッケージのファイル名です）。

```
rpm -K --nosignature <rpm_file>
```

<rpm_file>: rsa sha1 (md5) pgp md5 OK メッセージ（特に OK 部分）が表示されます。この簡単なメッセージは、ダウンロード中にファイルが破損しなかったことを意味します。より詳細なメッセージを表示するには、コマンドの -K を -Kvv に置き換えます。

一方、パッケージを作成した開発者だけに信頼する方法。パッケージが開発者の GnuPG キー で 署名 されている場合、開発者がそれらが本人であることがわかっていることが分かります。

RPM パッケージは GNU Privacy Guard （または GnuPG）を使用して署名し、ダウンロードしたパッケージに信頼されているパッケージを明確にするのに役立てることができます。

GnuPG は、安全な通信を行うためのツールです。これは、電子プライバシープログラムである PGP の暗号化テクノロジーの完全な置換です。GnuPG を使用すると、ドキュメントの有効性を認証し、他の受信者との間でデータを暗号化/復号化できます。GnuPG は PGP 5.x ファイルを復号および検証することもできます。

インストール時に、ART はデフォルトでインストールされます。これにより、Red Hat から受け取るパッケージを確認できるようにするために GnuPG の使用をすぐに開始できます。これを実行する前に、まず Red Hat の公開鍵をインポートする必要があります。

B.3.1. キーのインポート

Red Hat パッケージを検証するには、Red Hat GnuPG キーをインポートする必要があります。これを行うには、シェルプロンプトで以下のコマンドを実行します。

```
rpm --import /usr/share/rhn/RPM-GPG-KEY
```

RPM 検証用にインストールされた鍵の一覧を表示するには、以下のコマンドを実行します。

```
rpm -qa gpg-pubkey*
```

Red Hat キーの出力には、以下が含まれます。

```
gpg-pubkey-db42a60e-37ea5438
```

特定のキーの詳細を表示するには、rpm -qi の後に直前のコマンドの出力を使用します。

```
rpm -qi gpg-pubkey-db42a60e-37ea5438
```

B.3.2. パッケージの署名の確認

ビルダーの GnuPG キーのインポート後に RPM ファイルの GnuPG 署名を確認するには、以下のコマンドを使用します（< rpm-file > は RPM パッケージのファイル名に置き換えてください）。

```
rpm -K <rpm-file>
```

すべてが正常に行われると、md5 gpg OK というメッセージが表示されます。つまり、パッケージの署名が破損しないため、安全にインストールして使用することができます。

B.4. RPM の使用方法に関する実例および一般的な例

RPM は、システムの管理と問題の診断と修正の両方に役立ちます。すべてのオプションを把握する最適な方法は、いくつかの例を参照することです。

- 場合によっては、誤って一部のファイルを削除しましたが、削除内容を把握しているわけではありません。システム全体を検証し、足りないものを確認するには、以下のコマンドを実行します。

  ```
  rpm -Va
  ```

  一部のファイルが欠落しているか、破損している場合、パッケージを再インストールするか、またはアンインストールしてからパッケージを再インストールします。

- 場合によっては、認識しないファイルが表示される場合があります。パッケージを所有するパッケージを確認するには、次のコマンドを実行します。

  ```
  rpm -qf /usr/bin/ghostscript
  ```

出力は以下のようになります。

```
ghostscript-8.70-1.el6.x86_64
```

- 以下のシナリオにおいて、上記の 2 つの例を組み合わせることができます。/usr/bin/paste に問題があるとします。そのプログラムを所有するパッケージを確認しますが、どのパッケージが 貼り付ける かは分からません。以下のコマンドを入力します。

```
rpm -Vf /usr/bin/paste
```

また、適切なパッケージが検証されています。

- 特定のプログラムに関する詳細情報を見つけてもよろしいですか？以下のコマンドを試し、そのプログラムを所有するパッケージに含まれるドキュメントを確認できます。

```
rpm -qdf /usr/bin/free
```

出力は以下のようになります。

```
/usr/share/doc/procps-3.2.8/BUGS
/usr/share/doc/procps-3.2.8/FAQ
/usr/share/doc/procps-3.2.8/NEWS
/usr/share/doc/procps-3.2.8/TODO
/usr/share/man/man1/free.1.gz
/usr/share/man/man1/pgrep.1.gz
/usr/share/man/man1/pkill.1.gz
/usr/share/man/man1/pmap.1.gz
/usr/share/man/man1/ps.1.gz
/usr/share/man/man1/pwdx.1.gz
/usr/share/man/man1/skill.1.gz
/usr/share/man/man1/slabtop.1.gz
/usr/share/man/man1/snice.1.gz
/usr/share/man/man1/tload.1.gz
/usr/share/man/man1/top.1.gz
/usr/share/man/man1/uptime.1.gz
/usr/share/man/man1/w.1.gz
/usr/share/man/man1/watch.1.gz
/usr/share/man/man5/sysctl.conf.5.gz
/usr/share/man/man8/sysctl.8.gz
/usr/share/man/man8/vmstat.8.gz
```

- 新しい RPM が見つかるかもしれませんが、何ができるのかは分かないでしょう。情報を確認するには、以下のコマンドを使用します。

```
rpm -qip crontabs-1.10-32.1.el6.noarch.rpm
```

出力は以下のようになります。

```
Name        : crontabs                     Relocations: (not relocatable)
Version     : 1.10                          Vendor: Red Hat, Inc.
Release     : 32.1.el6                     Build Date: Thu 03 Dec 2009 02:17:44 AM CET
Install Date: (not installed)                Build Host: js20-bc1-11.build.redhat.com
Group       : System Environment/Base       Source RPM: crontabs-1.10-32.1.el6.src.rpm
Size        : 2486                         License: Public Domain and GPLv2
Signature   : RSA/8, Wed 24 Feb 2010 08:46:13 PM CET, Key ID 938a80caf21541eb
Packager    : Red Hat, Inc. <http://bugzilla.redhat.com/bugzilla>
Summary     : Root crontab files used to schedule the execution of programs
Description :
The crontabs package contains root crontab files and directories.
You will need to install cron daemon to run the jobs from the crontabs.
The cron daemon such as cronie or fcron checks the crontab files to
see when particular commands are scheduled to be executed.  If commands
are scheduled, it executes them.
Crontabs handles a basic system function, so it should be installed on
your system.
```

- 場合によっては、crontabs RPM パッケージがどのファイルをインストールするかを見てみましょう。以下を入力します。

```
rpm -qlp crontabs-1.10-32.1.el6.noarch.rpm
```

出力は以下の例のようになります。

```
/etc/cron.daily
/etc/cron.hourly
/etc/cron.monthly
/etc/cron.weekly
/etc/crontab
/usr/bin/run-parts
/usr/share/man/man4/crontabs.4.gz
```

これらはいくつかの例です。RPM を使用する際には、その使用がさらに分かっている可能性があります。

B.5. その他のリソース

RPM は、パッケージのクエリー、インストール、アップグレード、および削除を行う数多くのオプ

ションと、非常に複雑なユーティリティーです。RPM の詳細は、以下の資料を参照してください。

B.5.1. インストールされているドキュメント

- rpm --help - このコマンドは、RPM パラメーターのクイックリファレンスを表示します。

- man rpm - RPM の man ページには、rpm --help コマンドよりも RPM パラメーターに関する詳細が記載されています。

B.5.2. 便利な Web サイト

- RPM Web サイト - http://www.rpm.org/

- RPM メーリングリストはサブスクライブでき、そのアーカイブはここに読み取られます。https://lists.rpm.org/mailman/listinfo/rpm-list

付録C X ウィンドウシステム

Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux の中核となるのは、多くのユーザーにとってカーネルですが、オペレーティングシステムは X Window System (X)が提供するグラフィカル環境です。

他のウィンドウ環境は UNIX 世界に存在していました。これには、X Window System のリリースを 6 月 1984 年にプレリリースするものも含まれます。ただし、X は Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux など、多くの UNIX と同様のオペレーティングシステム向けのデフォルトのグラフィカル環境でした。

Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux のグラフィカル環境は、X.Org Foundation によって提供されます。これは、X Window System および関連テクノロジーの開発およびストラテジーを管理するために作成されたオープンソース組織です。X.Org は、グローバルで数百の開発者で構成される、大規模な迅速な開発プロジェクトです。さまざまなハードウェアデバイスやアーキテクチャーに対する幅広いサポートを特長とし、myriad オペレーティングシステムおよびプラットフォーム上で稼働します。

X Window System は、クライアントサーバーアーキテクチャーを使用します。その主な目的は、さまざまなコンピューティングマシンおよびグラフィックマシンで実行されるネットワーク透過ウィンドウシステムを提供することです。X サーバー（ Xorg バイナリー）は、ネットワークまたはローカルループバックインターフェースを介して X クライアント アプリケーションからの接続をリッスンします。サーバーは、ビデオカード、モニター、キーボード、マウスなどのハードウェアと通信します。X クライアントアプリケーションがユーザー空間に存在し、ユーザー用の グラフィカルユーザーインターフェース (GUI)を作成し、X サーバーにユーザー要求を渡します。

C.1. X サーバー

Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;LinuxRed Hat Enterprise Linux 6nbsp;6 は、X サーバーのバージョンを使用します。これには、以前のリリース以上のビデオドライバー、EXA、およびプラットフォームのサポート拡張機能が含まれます。さらに、本リリースには X サーバーの自動設定機能が複数含まれ、一般的な入力ドライバー evdev が、カーネルが認識するすべての入力デバイス（ほとんどのマウスやキーボードを含む）に対応しています。

X11R7.1 は、X Window System モジュラー化に特定の利点を活用するための最初のリリースでした。本リリースでは、X を論理的に個別のモジュールに分割し、オープンソース開発者がシステムにコードを簡単に提供できるようになりました。

現在のリリースでは、すべてのライブラリー、ヘッダー、およびバイナリーが /usr/ ディレクトリー下にあります。/etc/X11/ ディレクトリーには、X クライアントおよびサーバーアプリケーションの設定ファイルが含まれています。これには、X サーバー自体、X ディスプレイマネージャー、その他の多くのベースコンポーネント用の設定ファイルが含まれています。

新しい Fontconfig ベースのフォントアーキテクチャーの設定ファイルは /etc/fonts/fonts.conf のままです。フォントの設定および追加の詳細は、「fonts」 を参照してください。

X サーバーはさまざまなハードウェアで高度なタスクを実行するため、動作するハードウェアに関する詳細情報が必要です。X サーバーは、それを実行するほとんどのハードウェアを自動的に検出して、それに応じて独自のハードウェアを設定できます。または、設定ファイルでハードウェアを手動で指定することもできます。

インストールに X パッケージが選択されていない限り、Red Hat Enterprise Linux システムインストーラー Anaconda、Anaconda は X を自動的にインストールおよび設定します。X サーバーが管理するモニター、ビデオカード、またはその他のデバイスに変更が加えられた場合は、ほとんどの場合、X がこれらの変更を自動的に検出して再設定します。まれに、X を手動で再設定する必要があります。

C.2. デスクトップ環境およびウィンドウマネージャー

X サーバーが実行されたら、X クライアントアプリケーションがそれに接続してユーザーの GUI を作成できます。GUI の範囲は、Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux で利用できます。これは、Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux に精通している Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux（基本的な Tab Window Manager (twm)）から、Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp; KDEなどの高度な開発およびインタラクティブなデスクトップ環境まで利用できます。

後者の方が包括的な GUI を作成するには、X クライアントアプリケーションの 2 つのメインクラス（ ウィンドウマネージャー と デスクトップ環境 ）を X サーバーに接続する必要があります。

C.2.1. 同時 GUI セッションの最大数

異なるユーザーの複数の GUI セッションを同じマシン上で同時に実行できます。同時 GUI セッションの最大数は、ハードウェア（特にメモリーサイズ）や実行中のアプリケーションの ワークロード要求によって制限されます。一般的な PC では、以前説明した状況によって、同時 GUI セッションの最大数は 10 から 15 を超えません。一部のアプリケーションが予期せずに終了する可能性があるため、同じマシン上で同じユーザーを GNOME を複数回ログインすることはサポートされません。

C.2.2. デスクトップ環境

デスクトップ環境では、さまざまな X クライアントを統合して、共通のグラフィカルユーザー環境と開発環境を作成します。

デスクトップ環境では、X クライアントや他の実行中のプロセスが相互に通信できるように高度な機能があります。また、その環境で書き込まれたすべてのアプリケーションが、ドラッグアンドドロップ

操作などの高度なタスクを実行できるようにします。

Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux は、2 つのデスクトップ環境を提供します。

- GNOME - GTK+ 2 グラフィカルツールキットに基づく Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux のデフォルトデスクトップ環境。
- kde - Qt 4 グラフィカルツールキットに基づく代替デスクトップ環境。

GNOME と KDE の両方に、単語プロセッサー、スプレッドシート、Web ブラウザーなどの高度な製品ビリティーアプリケーションがあります。これらはどちらも GUI の外観および操作をカスタマイズするツールも提供します。また、GTK+ 2 ライブラリーと Qt ライブラリーの両方が存在する場合、KRE アプリケーションは GNOME で実行し、その逆も同様です。

C.2.3. ウィンドウマネージャー

ウィンドウマネージャー は、デスクトップ環境の一部である X クライアントプログラム、または場合によってはスタンドアロンである X クライアントプログラムです。その主な目的は、グラフィカルウィンドウの位置、サイズ変更、移動の方法を制御することです。ウィンドウマネージャーはタイトルバー、ウィンドウフォーカスの動作、ユーザー指定のキーとマウスボタンバインディングも制御します。

Red Hat Enterprise Linux リポジトリーは、5 つの異なるウィンドウマネージャーを提供します。

metacity

Metacity ウィンドウマネージャーは、GNOME のデフォルトのウィンドウマネージャーです。カスタムテーマをサポートするシンプルで効率的なウィンドウマネージャーです。このウィンドウマネージャーは、GNOME デスクトップをインストールすると、依存関係として自動的にプルされます。

kwin

KWin ウィンドウマネージャーは KDE のデフォルトのウィンドウマネージャーです。カスタムテーマをサポートする効率的なウィンドウマネージャーです。このウィンドウマネージャーは、KDE デスクトップのインストール時に自動的に依存関係としてプルされます。

compiz

Compiz compositing ウィンドウマネージャーは OpenGL をベースとしており、3D グラフィックハードウェアを使用して、ウィンドウ管理にデスクトップ効果をすばやく作成できます。ワークスペースなどの高度な機能はロード可能なプラグインとして実装されます。このウィンドウマネージャーを実行するには、compiz パッケージをインストールする必要があります。

mwm

Motif Window Manager (mwm)は、基本的なスタンドアロンのウィンドウマネージャーです。これはスタンドアロンとして設計されているため、GNOME または KDE と併用しないでください。このウィンドウマネージャーを実行するには、openmotif パッケージをインストールする必要があります。

twm

選択可能なウィンドウマネージャー間で最も基本的なツールセットを提供する minimalist Tab Window Manager (twm)は、スタンドアロンまたはデスクトップ環境で使用できます。このウィンドウマネージャーを実行するには、xorg-x11-twm パッケージをインストールする必要があります。

C.3. X サーバー設定ファイル

X サーバーは単一のバイナリー実行ファイル /usr/bin/Xorg です。このファイルを参照するシンボリックリンク X も提供しています。関連する設定ファイルは、/etc/X11/ ディレクトリーおよび /usr/share/X11/ ディレクトリーに保存されます。

X Window System は、2 つの異なる設定スキームをサポートします。xorg.conf.d ディレクトリーの設定ファイルには、ベンダーおよびディストリビューションからの事前設定済みの設定が含まれ、これらのファイルは手動で編集しないでください。一方、xorg.conf ファイルの設定は完全に行われますが、ほとんどのシナリオでは必要ありません。

XORG.CONF ファイルが必要な場合

ディスプレイと周辺(peripheral)に必要なパラメーターはすべて、インストール時に自動的に検出され設定されます。X サーバー(/etc/X11/xorg.conf)の設定ファイルは、X Window System の現在のリリースには提供されていません。ファイルを手動で作成して新規ハードウェアを設定したり、複数のビデオカードで環境を設定したり、デバッグを行う場合にも便利です。

/usr/lib/xorg/modules/（または / usr/lib64/xorg/modules/）ディレクトリーには、実行時に動的に

ロードできる X サーバーモジュールが含まれます。デフォルトでは、/usr/lib/xorg/modules/ の一部のモジュールのみが、X サーバーにより自動的に読み込まれます。

Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;LinuxRed Hat Enterprise Linux 6nbsp;6 がインストールされると、HAL（ハードウェア Abstraction Layer）設定バックエンドによりインストールプロセス中に収集される情報を使用して X の設定ファイルが作成されます。X サーバーが起動するたびに、HAL に入力デバイスの一覧が尋ねられ、各サーバーはそれに対応するドライバーとともに追加します。新しい入力デバイスがプラグインされるたびに、または既存の入力デバイスが削除されると、HAL は X サーバーにその変更について通知します。この通知システムにより、xorg.conf ファイルに設定されている マウス、kbd ドライバー、または vmmouse ドライバーを使用するデバイスは、X サーバーでは無視されます。詳細は、「ServerFlags セクション」 を参照してください。追加の設定は /etc/X11/xorg.conf.d/ ディレクトリーに提供され、HAL で取得した設定をオーバーライドまたは拡張することができます。

C.3.1. 設定構造

X 設定ファイルの形式は、システムハードウェアの特定の側面に対応するさまざまなセクションで構成されています。各セクションは、セクション 「section-name" 行」 で始まります。ここで、「section-name" はセクションのタイトルであり、EndSection 行で終わります。各セクションには、オプション名と 1 つ以上のオプション値を含む行が含まれます。これらの一部は、二重引用符(")で囲まれている場合があります。

/etc/X11/xorg.conf ファイル内の一部のオプションはブール値スイッチを受け入れ、機能をオンまたはオフにします。使用できる値は次のとおりです。

- 1、、true、または yes - オプションをオンにします。
- 0、off、false、または no - オプションをオフにします。

キーボードの典型的な設定ファイルを以下に示します。ハッシュ記号(#)で始まる行は X サーバーによって読み取られず、人間が判読できるコメントに使用されます。

```
# This file is autogenerated by system-setup-keyboard. Any
# modifications will be lost.

Section "InputClass"
  Identifier  "system-setup-keyboard"
  MatchIsKeyboard "on"
  Option    "XkbModel"  "pc105"
  Option    "XkbLayout" "cz,us"
# Option    "XkbVariant"  "(null)"
  Option    "XkbOptions"  "terminate:ctrl_alt_bksp,grp:shifts_toggle,grp_led:scroll"
EndSection
```

C.3.2. xorg.conf.d ディレクトリー

X サーバーは、2 つの設定ディレクトリーをサポートします。/usr/share/X11/xorg.conf.d/ はベンダーまたはサードパーティーパッケージとは別の設定ファイルを提供します。このディレクトリーのファイルへの変更は、/etc/X11/xorg.conf ファイルで指定した設定で上書きできます。/etc/X11/xorg.conf.d/ ディレクトリーには、ユーザー固有の設定が保存されます。

設定ディレクトリーの接尾辞 .conf が含まれるファイルは、起動時に X サーバーが解析し、従来の xorg.conf 設定ファイルの一部として処理されます。これらのファイルには 1 つ以上のセクションが含まれる可能性があります。セクションのオプションの説明と設定ファイルの一般的なレイアウトについては、「xorg.conf ファイル」 または xorg.conf(5) の man ページを参照してください。X サーバーは基本的に、設定ファイルのコレクションを、最後に xorg.conf のエントリーを持つ 1 つの大きなファイルとして扱います。カスタムの設定を /etc/xorg.conf に配置し、ディストリビューションが提供する設定スニペットのディレクトリーを残しておくことを推奨します。

C.3.3. xorg.conf ファイル

以前のリリースの X Window System では、/etc/X11/xorg.conf ファイルを使用して X の初期設定を保存することができました。X サーバーが管理するモニター、ビデオカード、または他のデバイスで変更が発生した場合は、手動でファイルを編集する必要がありました。Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux では、/etc/X11/xorg.conf ファイルを手動で作成および編集する必要はほとんどありません。ただし、特に異常なハードウェア構成をトラブルシューティングまたは設定する場合など、利用可能なさまざまなセクションとオプションのパラメーターを理解すると便利です。

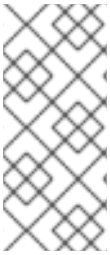
以下の重要なセクションは、典型的な /etc/X11/xorg.conf ファイルに表示される順序で説明されています。X サーバー設定ファイルの詳細は、xorg.conf(5) man ページを参照してください。このセクションは、一般的な設定シナリオで以下のほとんどの設定オプションが不要であるため、上級ユーザー向けのものです。

C.3.3.1. InputClass セクション

InputClass は、ホットプラグされたデバイスを含むデバイスのクラスには適用されません。InputClass セクションのスコープは指定された一致によって制限されます。入力デバイスに適用するには、以下の例にあるようにすべての一致をデバイスに適用する必要があります。

```
Section "InputClass"
  Identifier      "touchpad catchall"
  MatchIsTouchpad "on"
  Driver          "synaptics"
EndSection
```

このスニペットが xorg.conf ファイルまたは xorg. conf.d ディレクトリーにある場合、システムに存在するすべてのタッチパッドに synaptics ドライバーが割り当てられます。



XORG.CONF.Dの英数字のソート

xorg.conf.d ディレクトリーの設定ファイルの英数字のソートにより、上記の例の Driver 設定は、以前に設定したドライバーオプションを上書きすることに注意してください。より一般的なクラス（以前の）が一覧表示されるはずです。

match オプションは、セクションが適用されるデバイスを指定します。デバイスに一致するには、すべての一致オプションに対応している必要があります。以下のオプションは、InputClass セクションで一般的に使用されます。

- *MatchIsPointer、MatchIsKeyboard、MatchIsTouchpad、MatchIsTouchscreen、MatchIsJoystick - デバイスのタイプを指定するブール値オプション。*
- *MatchProduct "product_name" - このオプションは、デバイスの製品名で product_name サブ文字列が発生した場合に一致します。*
- *MatchVendor "vendor_name" - このオプションは、vendor_name サブ文字列がデバイスのベンダー名で発生する場合に一致します。*
- *MatchDevicePath "/path/to/device" - このオプションは、デバイスパスが「/path/to/device " テンプレート」で指定されたパターンに対応している場合は、任意のデバイスに一致します（例： /dev/input/event* ）。詳細は、man ページの fnmatch(3) を参照してください。*
- *MatchTag "tag_pattern" - このオプションは、HAL 設定バックエンドに割り当てられたタグが 1 つ以上 tag_pattern パターンと一致する場合に一致します。*

設定ファイルに複数の InputClass セクションが含まれる可能性があります。これらのセクションはオプションで、自動的に追加されるときに入力デバイスのクラスを設定するために使用されます。入力デバイスは、複数の InputClass セクションを照合できます。これらのセクションを指定する場合は、各入力クラスは重複が発生した場合に以前の設定を上書きする可能性があるため、一般的な一致を特定の上に配置することが推奨されます。

C.3.3.2. InputDevice セクション

各 InputDevice セクションは、X サーバーの 1 つの入力デバイスを設定します。以前のバージョンでは、システムは通常キーボードに InputDevice セクションが少なくとも 1 つあり、ほとんどのマウス

設定は自動的に検出されていました。

Red Hat Enterprise Linux 6nbsp;Hat Enterprise Linux 6nbsp;LinuxRed Hat Enterprise Linux 6nbsp;6 では、ほとんどのセットアップには InputDevice 設定が必要で、xorg-x11-drv-* 入力ドライバーパッケージは HAL を使用して自動設定を提供します。キーボードとマウスの両方のデフォルトのドライバーは evdev です。

以下の例は、キーボードの通常の InputDevice セクションを示しています。

```
Section "InputDevice"
  Identifier "Keyboard0"
  Driver "kbd"
  Option "XkbModel" "pc105"
  Option "XkbLayout" "us"
EndSection
```

以下のエントリーは、InputDevice セクションで一般的に使用されます。

- identifier: この InputDevice セクションの一意の名前を指定します。これは必須エントリーです。
- driver: デバイスのデバイスドライバー X を読み込む必要がある名前を指定します。AutoAddDevices オプションが有効にされている場合（デフォルト設定）、Driver "mouse" または Driver "kbd" のある入力デバイスセクションは無視されます。これは、従来のマウスドライバーとキーボードドライバーと新しい evdev 汎用ドライバーの競合により必要になります。代わりに、サーバーは入力デバイスにバックエンドからの情報を使用します。xorg.conf のカスタム入力デバイス設定は、バックエンドに移動する必要があります。多くの場合、バックエンドは HAL で、設定場所は /etc/X11/xorg.conf.d ディレクトリーになります。
- オプション: デバイスに必要なオプションを指定します。

  デバイスの自動検出された値を上書きする場合には、マウスを指定することもできます。通常は、xorg.conf ファイルにマウスを追加する場合には、以下のオプションが含まれます。

  - protocol: IMPS/2 などのマウスで使用されるプロトコルを指定します。

- *device - 物理デバイスの場所を指定します。*

- *Emulate3Buttons: 両方のマウスボタンを同時に押す場合に、ボタンを 3 つのマウスのように動作させるかどうかを指定します。*

このセクションの有効なオプションの一覧は、xorg.conf(5) man ページを参照してください。

C.3.3.3. ServerFlags セクション

任意の ServerFlags セクションには、その他のグローバル X サーバーの設定が含まれます。このセクションの設定は、ServerLayout セクションに配置されているオプションで上書きできます（詳細は「ServerLayout セクション」 を参照してください）。

ServerFlags セクション内の各エントリーは単一行を使用し、オプション の後に二重引用符(")で囲まれているオプションで始まります。

以下は、ServerFlags セクションの例です。

```
Section "ServerFlags"
  Option "DontZap" "true"
EndSection
```

以下は、最も役立つオプションの一部を示しています。

- *"DontZap" "boolean" - < boolean > の値が true に設定されている場合、この設定により Ctrl+Alt+Backspace キーの組み合わせが X サーバーをすぐに終了できなくなります。*

X キーボード拡張

このオプションが有効になっている場合でも、キーの組み合わせは XKB(XKB)マップで設定してから使用する必要があります。マップにキーの組み合わせを追加する方法の 1 つとして、以下のコマンドを実行します。

```
setxkbmap -option "terminate:ctrl_alt_bksp"
```

- *"DontZoom" "boolean" - < boolean > の値が true に設定されている場合、この設定は Ctrl+Alt+Keypad-Plus+Ctrl+Alt+Keypad- キーの組み合わせを使用して、設定されたビデオ解決を循環させます。*

- *"AutoAddDevices" "boolean" - < boolean > の値が false に設定されている場合、サーバーはホットプラグ入力デバイスではなく、xorg.conf ファイルで設定したデバイスのみに依存します。入力デバイスの詳細は、「InputDevice セクション」 を参照してください。このオプションはデフォルトで有効にされており、HAL（ハードウェア抽象化レイヤー）はデバイス検出のバックエンドとして使用されます。*

C.3.3.4. ServerLayout セクション

ServerLayout セクションは、X サーバーが制御する入出力デバイスをバインドします。少なくとも、1 つの入力デバイスと 1 つの出力デバイスを指定する必要があります。デフォルトでは、モニター（出力デバイス）およびキーボード（入力デバイス）が指定されます。

以下の例は、一般的な ServerLayout セクションを示しています。

```
Section "ServerLayout"
  Identifier "Default Layout"
  Screen 0 "Screen0" 0 0
  InputDevice "Mouse0" "CorePointer"
  InputDevice "Keyboard0" "CoreKeyboard"
EndSection
```

ServerLayout セクションでは、以下のエントリーが使用されます。

- *identifier: この ServerLayout セクションの一意の名前を指定します。*

-

screen: X サーバーで使用する 画面 セクションの名前を指定します。複数の スクリーン オプションが存在する可能性があります。

以下は、通常の Screen エントリーの例です。

```
Screen 0 "Screen0" 0 0
```

この例の Screen エントリー(0)の最初の番号は、ビデオカードの最初のモニターコネクターまたは ヘッド が、識別子が「Screen0 」の Screen セクションで指定されている設定を使用することを示しています。

識別子 「 Screen 0」 を含む Screen セクションの例は、「Screen セクション」 を参照してください。

ビデオカードに複数のヘッドがある場合は、別の番号と異なる Screen セクション識別子を持つ別の Screen エントリーが必要です。

「Screen0」 の右側にある数字は、画面の左上隅に絶対 X と Y 座標を指定します（デフォルトでは0 ）。

- *InputDevice: X サーバーで使用する InputDevice セクションの名前を指定します。*

  デフォルトのマウスで、もう 1 つはデフォルトのキーボード用の InputDevice エントリーが 2 つ以上あることが推奨されます。CorePointer および CoreKeyboard オプションは、これらが主要なマウスとキーボードであることを示しています。AutoAddDevices オプションが有効になっている場合は、ServerLayout セクションにこのエントリーを指定する必要はありません。AutoAddDevices オプションが無効になっていると、マウスとキーボードの両方がデフォルト値で自動検出されます。

- *オプション "option-name" - セクションの追加パラメーターを指定する任意のエントリー。ここに記載のオプションは、ServerFlags セクションに記載されているオプションを上書きします。*

  &lt ;option-name> を、man ページの xorg.conf(5) のこのセクションに記載されている有効なオプションに置き換えます。

/etc/X11/xorg.conf ファイルに複数の ServerLayout セクションを配置することができます。ただし、デフォルトでは、サーバーは最初に発生した最初のサーバーのみを読み取ります。別の ServerLayout セクションがある場合は、X セッションの開始時にコマンドラインの引数として指定できます（ Xorg -layout <layoutname> コマンドと同様 ）。

C.3.3.5. Files セクション

Files セクションは、フォントのパスなど、X サーバーへの重要なサービスパスを設定します。これらのパスは通常は自動検出されるため、これは任意のセクションです。このセクションを使用して、自動検出した値を上書きできます。

以下の例は、典型的な ファイル セクションを示しています。

```
Section "Files"
  RgbPath "/usr/share/X11/rgb.txt"
  FontPath "unix/:7100"
EndSection
```

以下のエントリーは、Files セクションで一般的に使用されます。

- modulePath: X サーバーモジュールを格納する代替ディレクトリーを指定する任意のパラメーターです。

C.3.3.6. Monitor セクション

各 Monitor セクションは、システムによって使用されるモニターのタイプを設定します。ほとんどのモニターは自動的に検出されるようになりました。

以下の例は、モニターの通常の Monitor セクションを示しています。

```
Section "Monitor"
  Identifier "Monitor0"
  VendorName "Monitor Vendor"
  ModelName "DDC Probed Monitor - ViewSonic G773-2"
  DisplaySize 320 240
  HorizSync 30.0 - 70.0
  VertRefresh 50.0 - 180.0
EndSection
```

Monitor セクションでは、以下のエントリーが使用されます。

- *identifier: この Monitor セクションの一意の名前を指定します。これは必須エントリーです。*

- *vendorname: モニターのベンダーを指定する任意のパラメーター。*

- *modelName: モニターのモデル名を指定する任意のパラメーター。*

- *DisplaySize: モニターの図領域の物理サイズを指定するオプションのパラメーターです。*

- *HorizSync: kHz でモニターと互換性のある水平同期の周波数の範囲を指定します。これらの値は、X サーバーがモニターの組み込みまたは指定の Modeline エントリーの有効性を判断するのに役立ちます。*

- *VertRefresh - kHz で、モニターでサポートされる垂直更新頻度の範囲を指定します。これらの値は、X サーバーがモニターの組み込みまたは指定の Modeline エントリーの有効性を判断するのに役立ちます。*

- *modeline: 特定 の解像度でのモニターの追加ビデオモードを指定する任意のパラメーターで、特定の水平同期および垂直更新の解像度がある。Modeline エントリーの詳細は、xorg.conf(5) man ページを参照してください。*

- *オプション "option-name" - セクションの追加パラメーターを指定する任意のエントリー。&lt ;option-name> を、man ページの xorg.conf(5) のこのセクションに記載されている有効なオプションに置き換えます。*

C.3.3.7. デバイス セクション

各 Device セクションには、システムにビデオカードを 1 つ設定します。デバイス セクションは最小限ですが、マシンにインストールされるビデオカードごとに追加のインスタンスが発生する可能性があります。

以下の例は、ビデオカードの一般的な Device セクションを示しています。

```
Section "Device"
  Identifier "Videocard0"
  Driver "mga"
```

```
  VendorName "Videocard vendor"
  BoardName "Matrox Millennium G200"
  VideoRam 8192
  Option "dpms"
EndSection
```

以下のエントリーは、Device セクションで一般的に使用されます。

- identifier: この デバイス セクションの一意の名前を指定します。これは必須エントリーです。

- Driver: ビデオカードを使用するために X サーバーが読み込む必要があるドライバーを指定します。ドライバーの一覧は、/usr/share/hwdata/videodrivers を参照してください。これは、hwdata パッケージでインストールされます。

- vendorname - ビデオカードのベンダーを指定する任意のパラメーター。

- BoardName: ビデオカードの名前を指定する任意のパラメーター。

- video ram: ビデオ カードで利用可能な RAM 容量をキロバイト単位で指定するオプションのパラメーターです。この設定は、X サーバーがビデオ RAM の量を検出できるようにプローブできないビデオカードにのみ必要です。

- BusID: ビデオカードのバスの場所を指定するエントリー。ビデオカードが 1 つしかないシステムでは BusID エントリーが任意となり、デフォルトの /etc/X11/xorg.conf ファイルにない場合があります。ただし、複数のビデオカードが搭載されているシステムでは、BusID エントリーが必要です。

- screen: Device セクションが設定するビデオカード上のモニターコネクターまたはヘッドを指定する任意のエントリー。このオプションは、複数のヘッドを持つビデオカードにのみ有用です。

  複数のモニターが同じビデオカード上の異なるヘッドに接続されている場合は、個別の Device セクションが存在し、本セクションのそれぞれに異なる Screen 値を指定する必要があります。

  Screen エントリーの値は整数である必要があります。ビデオカードの最初のヘッドの値は 0 です。それぞれの追加ヘッドの値は、この値を 1 つ増やします。

- オプション "option-name" - セクションの追加パラメーターを指定する任意のエントリー。&lt ;option-name> を、man ページの xorg.conf(5) のこのセクションに記載されている有効なオプションに置き換えます。

  より一般的なオプションの 1 つに、「dpms」（ディスプレイ Power Management Signaling、VESA 標準用）があり、モニター用の「スターバルコンプライアンス」設定を有効にします。

C.3.3.8. Screen セクション

各 スクリーン セクションは、Device セクションと Monitor セクションを参照して、1 つのビデオカード（またはビデオカードヘッド）を 1 つのモニターにバインドします。スクリーン セクションは最小限ですが、マシン上にある各ビデオカードとモニターの組み合わせに追加インスタンスが発生する場合があります。

以下の例は、一般的な Screen セクションを示しています。

```
Section "Screen"
  Identifier "Screen0"
  Device "Videocard0"
  Monitor "Monitor0"
  DefaultDepth 16

  SubSection "Display"
    Depth 24
    Modes "1280x1024" "1280x960" "1152x864" "1024x768" "800x600" "640x480"
  EndSubSection

  SubSection "Display"
    Depth 16
    Modes "1152x864" "1024x768" "800x600" "640x480"
  EndSubSection
EndSection
```

以下のエントリーは、Screen セクションで一般的に使用されます。

- identifier: この スクリーン セクションに一意の名前を指定します。これは必須エントリーです。

- device - デバイス セクションの一意の名前を指定します。これは必須エントリーです。

- *monitor: Monitor セクションの一意の名前を指定します。これは、特定の Monitor セクションが xorg.conf ファイルに定義されている場合のみ必要です。通常、モニターは自動的に検出されます。*

- *DefaultDepth: デフォルトの色深度をビット単位で指定します。上記の例では、16 （数千の色を提供する）がデフォルトです。DefaultDepth エントリーは 1 つだけ許可されますが、Xorg コマンドラインオプション -depth < n > で上書きできます。ここで、< n > は追加の深度を指定します。*

- *subsection "Display" - 特定の色深度で利用可能な画面モードを指定します。Screen セクションには複数の Display サブセクションを使用できます。これは、画面モードが自動的に検出されるため、完全にオプションになります。*

  通常、このサブセクションは自動検出されたモードを上書きするのに使用します。

- *オプション "option-name" - セクションの追加パラメーターを指定する任意のエントリー。&lt ;option-name> を、man ページの xorg.conf(5) のこのセクションに記載されている有効なオプションに置き換えます。*

C.3.3.9. DRI セクション

オプションの DRI セクション は、Direct Rendering Infrastructure (DRI)のパラメーターを指定します。DRI は、3D ソフトウェアアプリケーションが最新のビデオハードウェアに組み込まれた 3D ハードウェアアクセラレーション機能を活用できるようにするインターフェースです。さらに、DRI はビデオカードドライバーでサポートされている場合に、ハードウェアアクセラレーションで 2D パフォーマンスを向上できます。

DRI グループおよびモードは自動的にデフォルト値に初期化されるため、このセクションはほとんど使用されません。別のグループまたはモードが必要な場合は、このセクションを xorg.conf ファイルに追加すると、デフォルト値が上書きされます。

以下の例は、典型的な DRI セクションを示しています。

```
Section "DRI"
  Group 0
  Mode 0666
EndSection
```

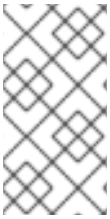
さまざまなビデオカードは異なる方法で DRI を使用しているため、http://dri.freedesktop.org/wiki/

を最初に参照せずにこのセクションに追加しないでください。

C.4. FONTS

Red Hat Enterprise Linuxnbsp;Hat Enterprise Red Hat Enterprise Linuxnbsp;Linux は Fontconfig サブシステムを使用して、X Window System でフォントを管理および表示します。フォント管理を簡素化し、アンチエイリアスなどの高度な表示機能を提供します。このシステムは、Qt 3 または GTK+ 2 のグラフィカルツールキットまたは新しいバージョンを使用してプログラムされるアプリケーションに使用されます。

Fontconfig font サブシステムを使用すると、アプリケーションはシステム上のフォントに直接アクセスし、X FreeType インターフェースライブラリー (Xft)または他のレンダリングメカニズムを使用して、アンチエイリアスなどの高度な機能で Fontconfig フォントをレンダリングできます。グラフィカルアプリケーションは、Fontconfig で Xft ライブラリーを使用して、画面にテキストを描画できます。



フォント設定

fontconfig は、/etc/fonts/fonts.conf 設定ファイルを使用しますが、これは手動で編集しないでください。

フォントグループ

ユーザーがリモート X アプリケーションの実行を想定しているシステムには、フォント グループをインストールする必要があります。これは、インストーラーでグループを選択し、インストール後に yum groupinstall fonts コマンドを実行して実行できます。

C.4.1. Fonts の Fontconfig への追加

Fontconfig サブシステムに新しいフォントを追加するのは簡単なプロセスです。

1.

個々のユーザーにフォントを追加するには、新しいフォントをユーザーのホームディレクトリーの .fonts/ ディレクトリーにコピーします。

システム全体でフォントを追加するには、新しいフォントを /usr/share/fonts/ ディレクトリーにコピーします。ユーザーインストールしたデフォルトのフォントを区別するため

に、local/ や similar などの新しいサブディレクトリーを作成することが推奨されます。

2. fc-cache コマンドを root として実行し、フォント情報キャッシュを更新します。

```
fc-cache <path-to-font-directory>
```

このコマンドで、< path-to-font-directory > を新しいフォントを含むディレクトリーに置き換えます（ /usr/share/fonts/local/ または / home/ <user> /.fonts/）。

対話型フォントのインストール

個々のユーザーは、fonts : /// をnautilus アドレスバーに入力して、そこで新しいフォントファイルをドラッグすることで、フォントを対話的にインストールすることもできます。

C.5. ランレベルおよび X

多くの場合、Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux インストーラーは、ランレベル 5 と呼ばれるグラフィカルログイン環境で起動するように設定します。ただし、ランレベル 3 と呼ばれるテキストのみのマルチユーザーモードで起動し、そこから X セッションを開始することは可能です。

以下のサブセクションでは、ランレベル 3 とランレベル 5 の両方で X がどのように起動するかについて説明します。ランレベルの詳細は、「デフォルトのランレベルの設定」 を参照してください。

C.5.1. ランレベル 3

ランレベル 3 では、X セッションを開始する最善の方法は、startx と入力してから startx と入力します。startx コマンドは、xinit コマンドのフロントエンドで、X サーバー(Xorg)を起動し、X クライアントアプリケーションをこれに接続します。ユーザーがランレベル 3 でシステムにログインしているため、startx はディスプレイマネージャーを起動したり、ユーザーを認証したりしません。ディスプレイマネージャーの詳細は、「ランレベル 5」 を参照してください。

1. startx コマンドを実行すると、ユーザーのホームディレクトリーで .xinitrc ファイルを検索し、デスクトップ環境と、実行するその他の X クライアントアプリケーションを定義します。.xinitrc ファイルが存在しない場合は、代わりにシステムのデフォルト /etc/X11/xinit/xinitrc ファイルを使用します。

2. 次に、デフォルトの xinitrc スクリプトは、ユーザーのホームディレクトリー内の

.Xresources、.Xmodmap、.Xkbmap、および /etc/X11/ ディレクトリーの Xresources、Xmodmap、Xkbmap などのユーザー定義のファイルおよびデフォルトのシステムファイルを検索します。Xmodmap ファイルおよび Xkbmap ファイルが存在する場合は、キーボードを設定する xmodmap ユーティリティーにより使用されます。Xresources ファイルは、特定の設定値をアプリケーションに割り当てるために読み取られます。

3. *上記のオプションを設定したら、xinitrc スクリプトは /etc/X11/xinit/xinitrc.d/ ディレクトリーにあるすべてのスクリプトを実行します。このディレクトリーにある重要なスクリプトの 1 つは xinput.sh で、デフォルト言語などの設定を行います。*

4. *xinitrc スクリプトは、ユーザーのホームディレクトリーで .Xclients の実行を試行し、見つからない場合は /etc/X11/xinit/Xclients に変更します。Xclients ファイルの目的は、デスクトップ環境を開始するか、基本ウィンドウマネージャーを起動することです。ユーザーのホームディレクトリーの .Xclients スクリプトは、.Xclients-default ファイルでユーザー指定のデスクトップ環境を起動します。ユーザーのホームディレクトリーに .Xclients が存在しない場合は、標準の /etc/X11/xinit/Xclients スクリプトは、GNOME を試行して、最初に KDE、その後に twm が続きます。*

ランレベル 3 では、X セッションを終了してから、ユーザーはテキストモードのユーザーセッションに戻ります。

C.5.2. ランレベル 5

システムがランレベル 5 で起動すると、ディスプレイマネージャー と呼ばれる特別な X クライアントアプリケーションが起動します。ユーザーは、デスクトップ環境またはウィンドウマネージャーを起動する前にディスプレイマネージャーを使用して認証する必要があります。

システムにインストールされたデスクトップ環境によっては、ユーザー認証の処理に 3 つの異なるディスプレイマネージャーを使用できます。

- *GDM (GNOME Display Manager): Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux のデフォルトディスプレイマネージャー。GNOME を使用すると、ユーザーは言語の設定、シャットダウン、再起動、またはシステムへのログインを行うことができます。*
- *KDM: ユーザーがシステムのシャットダウン、再起動、またはログインを可能にする KDE のディスプレイマネージャー。*
- *XDM(X Window Display Manager): ユーザーがシステムへのログインのみを許可する非常に基本的なディスプレイマネージャー。*

ランレベル 5 で起動すると、/etc/X11/prefdm スクリプトは、/etc/sysconfig/desktop ファイルを参照して推奨されるディスプレイマネージャーを決定します。このファイルでは、このファイルのオプションの一覧を利用できます。

```
/usr/share/doc/initscripts-<version-number>/sysconfig.txt
```

&lt ;version-number> は、 initscripts パッケージのバージョン番号に置き換えます。

各ディスプレイマネージャーは /etc/X11/xdm/Xsetup_0 ファイルを参照してログイン画面を設定します。ユーザーがシステムにログインすると、/etc/X11/xdm/GiveConsole スクリプトを実行して、コンソールの所有権をユーザーに割り当てます。次に、/etc/X11/xdm/Xsession スクリプトを実行すると、通常、ランレベル 3 から X を起動し、/etc/X11/xinit/ xinitrc.d/ ディレクトリーでスクリプトを実行する多くのタスクが実行されます。

ユーザーは、GNOME または KDE ディスプレイマネージャーを使用して認証する際に使用するデスクトップ環境を指定できます。そのためには、System → Preferences → More Preferences → Sessions の順に選択します。ディスプレイマネージャーでデスクトップ環境を指定しないと、/etc/X11/xdm/Xsession スクリプトは、ユーザーのホームディレクトリーの .xsession ファイルおよび .Xclients ファイルをチェックして、読み込むデスクトップ環境を決定します。最後の手段として、ランレベル 3 と同じ方法で使用するデスクトップ環境またはウィンドウマネージャーを選択する /etc/X11/xinit/Xclients ファイルが使用されます。

ユーザーがデフォルトの表示(:0)で X セッションを終了し、ログアウトすると、/etc/X11/xdm/TakeConsole スクリプトが実行され、コンソールの所有権を root ユーザーに再割り当てされます。ユーザーがログインした後も実行を継続する元のディスプレイマネージャーが、新しいディスプレイマネージャーを起動して制御します。これにより、X サーバーが再起動され、新しいログインウィンドウが表示され、プロセス全体が再び開始します。

ランレベル 5 から X からログアウトした後に、ユーザーはディスプレイマネージャーに返されます。

ディスプレイマネージャーによるユーザー認証の制御方法に関する詳細は、/usr/share/doc/gdm-<version-number> /README を参照してください。< version-number > は、gdm パッケージのインストールまたは xdm の man ページです。

C.6. リモートでのグラフィカルアプリケーションへのアクセス

以下の方法を使用して、リモートサーバーのグラフィカルアプリケーションにアクセスできます。

- ローカル X サーバーで、別のアプリケーションを SSH セッションから直接起動できます。これには、X11 転送を有効にする必要があります。詳しくは、「X11 転送」 を参照してください。

- VNC を使用してネットワーク上で X セッション全体を実行できます。この方法は、特に Linux 以外のシステムなど、X サーバーなしでワークステーションを使用している場合に役立ちます。詳しくは、15章TigerVNC を参照してください。

C.7. その他のリソース

X サーバー、それに接続するクライアント、および分類されたデスクトップ環境とウィンドウマネージャーに関する多くの詳細情報があります。

C.7.1. インストールされているドキュメント

- /usr/share/X11/doc/: X Window System アーキテクチャーに関する詳細情報と、Xorg プロジェクトに関する追加情報を新規ユーザーとして取得する方法が説明されています。

- /usr/share/doc/gdm- <version-number> /README - ディスプレイマネージャーによるユーザー認証の制御方法に関する情報が含まれています。

- man xorg.conf: xorg.conf 設定ファイルに関する情報が含まれています。これには、ファイル内のさまざまなセクションの意味と構文が含まれます。

- man Xorg: Xorg ディスプレイサーバーを説明しています。

C.7.2. 便利な Web サイト

- http://www.X.org/: X.Org Foundation のホームページです。Red Hat Enterprise Linuxnbsp;Hat Enterprise Red Hat Enterprise Linuxnbsp;Linux にバンドルされた X Window System のメジャーリリースを生成し、必要なハードウェアを制御し、GUI 環境を提供します。

- http://dri.sourceforge.net/: DRI(Direct Rendering Infrastructure)プロジェクトのホームページです。DRI は、X のコアハードウェア 3D アクセラレーションコンポーネントです。

- *http://www.gnome.org/: GNOME プロジェクトのホーム*

- *http://www.kde.org/: KDE デスクトップ環境のホーム*

付録D SYSCONFIG ディレクトリー

この付録では、/etc/sysconfig/ ディレクトリーにあるファイルおよびディレクトリー、それらの機能、およびそれらのコンテンツの概要を説明します。これらのファイルの多くには非常に特殊な状況やまれな状況でのみ使用されるオプションが多数あるため、この付録の情報は完了することを目的としていません。

/ETC/SYSCONFIG/ ディレクトリーのコンテンツ

/etc/sysconfig/ ディレクトリーの実際の内容は、マシンにインストールされているプログラムによって異なります。設定ファイルが属するパッケージ名を検索するには、シェルプロンプトで以下を入力します。

```
~]$ yum provides /etc/sysconfig/filename
```

Red Hat Enterprise Linuxnbsp;Hat Enterprise Red Hat Enterprise Linuxnbsp;Linux に新しいパッケージをインストールする方法の詳細は、「パッケージのインストール」 を参照してください。

D.1. /ETC/SYSCONFIG/ ディレクトリーのファイル

以下のセクションでは、通常 /etc/sysconfig/ ディレクトリーにあるファイルを説明します。

D.1.1. /etc/sysconfig/arpwatch

/etc/sysconfig/arpwatch ファイルは、システムの起動時に arpwatch デーモンに引数を渡すために使用されます。デフォルトでは、以下のオプションが含まれます。

OPTIONS=value

arpwatch デーモンに渡す追加のオプション。以下に例を示します。

```
OPTIONS="-u arpwatch -e root -s 'root (Arpwatch)'"
```

D.1.2. /etc/sysconfig/authconfig

/etc/sysconfig/authconfig ファイルは、ホストで使用する認証を設定します。デフォルトでは、以下のオプションが含まれます。

USEMKHOMEDIR=boolean

初回ログイン時にユーザーのホームディレクトリーを作成(yes)または無効(no)するブール値。以下に例を示します。

```
USEMKHOMEDIR=no
```

USEPAMACCESS=boolean

PAM 認証を有効(yes)または無効(no)するブール値。以下に例を示します。

```
USEPAMACCESS=no
```

USESSSDAUTH=boolean

SSSD 認証を有効(yes)または無効(no)するブール値。以下に例を示します。

```
USESSSDAUTH=no
```

USESHADOW=ブール値

シャドウパスワードを有効(yes)または無効(no)するブール値。以下に例を示します。

```
USESHADOW=yes
```

USEWINBIND=boolean

ユーザーアカウント設定の Winbind を使用して有効化(yes)または無効(no)するブール値。以下に例を示します。

```
USEWINBIND=no
```

USEDB=boolean

FAS 認証を有効(yes)または無効(no)するブール値。以下に例を示します。

```
USEDB=no
```

USEFPRINTD=boolean

フィンガープリント認証を有効(yes)または無効(no)するブール値。以下に例を示します。

```
USEFPRINTD=yes
```

FORCESMARTCARD=boolean

スマートカード認証を強制するブール値(yes)または無効(no)以下に例を示します。

```
FORCESMARTCARD=no
```

PASSWDALGORITHM=value

パスワードアルゴリズム。値 は、bigcrypt、descrypt、md5、sha256、または sha512 です。以下に例を示します。

```
PASSWDALGORITHM=sha512
```

USELDAPAUTH=boolean

LDAP 認証を有効(yes)または無効(no)するブール値。以下に例を示します。

```
USELDAPAUTH=no
```

USELOCAUTHORIZE=boolean

ローカルユーザーのローカル認可を有効(yes)または無効(no)するブール値。以下に例を示します。

```
USELOCAUTHORIZE=yes
```

USECRACKLIB=ブール値

CrackLib を使用して有効化(yes)または無効(no)するブール値。以下に例を示します。

```
USECRACKLIB=yes
```

USEWINBINDAUTH=boolean

Winbind 認証を有効(yes)または無効(no)するブール値。以下に例を示します。

```
USEWINBINDAUTH=no
```

USESMARTCARD=boolean

スマートカード認証を有効(yes)または無効(no)するブール値。以下に例を示します。

```
USESMARTCARD=no
```

USELDAP=boolean

ユーザーアカウント設定に LDAP を使用して有効化(yes)または無効(no)するブール値。以下に例を示します。

```
USELDAP=no
```

USENIS=ブール値

ユーザーアカウント設定に NIS を使用して有効化(yes)または無効(no)するブール値。以下に例を示します。

```
USENIS=no
```

USEKERBEROS=ブール値

Kerberos 認証を有効(yes)または無効(no)するブール値。以下に例を示します。

```
USEKERBEROS=no
```

USESYSNETAUTH=boolean

ネットワークサービスでシステムアカウントを認証するブール値(yes)または無効(no)以下に例を示します。

```
USESYSNETAUTH=no
```

USESMBAUTH=ブール値

SMB 認証を有効(yes)または無効(no)するブール値。以下に例を示します。

```
USESMBAUTH=no
```

USESSSD=ブール値

SSSD を使用してユーザー情報を取得するブール値(yes)または無効(no)以下に例を示します。

```
USESSSD=no
```

USEHESIOD=boolean

Hesoid name サービスを使用して有効化(yes)または無効(no)のブール値。以下に例を示します。

```
USEHESIOD=no
```

このトピックの詳細については、13章認証の設定 を参照してください。

D.1.3. /etc/sysconfig/autofs

/etc/sysconfig/autofs ファイルは、デバイスの自動マウント用のカスタムオプションを定義します。このファイルは、自動マウントデーモンの操作を制御します。これは、ファイルシステムを使用する際に自動的にマウントして、アクティブでないとアンマウントします。ファイルシステムには、ネットワークファイルシステム、CD-ROM ドライブ、ディスクチップなどのメディアを含めることができます。

デフォルトでは、以下のオプションが含まれます。

MASTER_MAP_NAME=value

マスターマップのデフォルト名。以下に例を示します。

```
MASTER_MAP_NAME="auto.master"
```

TIMEOUT=value

デフォルトのマウントタイムアウトです。以下に例を示します。

```
TIMEOUT=300
```

NEGATIVE_TIMEOUT=value

マウント試行に失敗した場合のデフォルトの負のタイムアウトです。以下に例を示します。

```
NEGATIVE_TIMEOUT=60
```

MOUNT_WAIT=value

マウント からの応答を待つ時間。以下に例を示します。

```
MOUNT_WAIT=-1
```

UMOUNT_WAIT=value

umount からの応答を待つ時間。以下に例を示します。

```
UMOUNT_WAIT=12
```

BROWSE_MODE=boolean

マップを参照するブール値(yes)または disable(no)以下に例を示します。

```
BROWSE_MODE="no"
```

MOUNT_NFS_DEFAULT_PROTOCOL=value

mount.nfs が使用するデフォルトのプロトコル。以下に例を示します。

```
MOUNT_NFS_DEFAULT_PROTOCOL=4
```

APPEND_OPTIONS=boolean

有効にするブール値(yes)または disable(no)は、グローバルオプションを置き換える代わりに追加します。以下に例を示します。

```
APPEND_OPTIONS="yes"
```

LOGGING=value

デフォルトのログレベル。値 は none、verbose、または debug のいずれかである必要があります。以下に例を示します。

```
LOGGING="none"
```

LDAP_URI=value

プロトコル:// サーバーの形式におけるサーバー URI のスペース区切りリスト。以下に例を示します。

```
LDAP_URI="ldaps://ldap.example.com/"
```

LDAP_TIMEOUT=value

同期 API 呼び出しのタイムアウト。以下に例を示します。

```
LDAP_TIMEOUT=-1
```

LDAP_NETWORK_TIMEOUT=value

ネットワーク応答のタイムアウト。以下に例を示します。

```
LDAP_NETWORK_TIMEOUT=8
```

SEARCH_BASE=value

マップ検索のベース識別名(DN)です。以下に例を示します。

```
SEARCH_BASE=""
```

AUTH_CONF_FILE=value

SASL 認証設定ファイルのデフォルトの場所。以下に例を示します。

```
AUTH_CONF_FILE="/etc/autofs_ldap_auth.conf"
```

MAP_HASH_TABLE_SIZE=value

マップキャッシュのハッシュテーブルサイズ。以下に例を示します。

```
MAP_HASH_TABLE_SIZE=1024
```

USE_MISC_DEVICE=boolean

autofs のその他のデバイスを使用して有効化(yes)または無効(no)するブール値。以下に例を示します。

```
USE_MISC_DEVICE="yes"
```

OPTIONS=value

LDAP デーモンに渡される追加のオプション。以下に例を示します。

```
OPTIONS=""
```

D.1.4. /etc/sysconfig/clock

/etc/sysconfig/clock ファイルは、システムクロックから読み込む値の解釈を制御します。これは Date/Time Properties ツールで使用するため、手動で編集しないでください。デフォルトでは、以下のオプションが含まれます。

ZONE=value

/usr/share/zoneinfo （ /etc/localtime はコピー）の下にあるタイムゾーンファイル。以下に例を示します。

```
ZONE="Europe/Prague"
```

Date/Time Properties ツールおよびその使用方法は、「日付/時刻のプロパティのツール」 を参照してください。

D.1.5. /etc/sysconfig/dhcpd

/etc/sysconfig/dhcpd ファイルを使用して、システムの起動時に dhcpd デーモンに引数を渡します。デフォルトでは、以下のオプションが含まれます。

DHCPDARGS=value

dhcpd デーモンに渡す追加のオプション。以下に例を示します。

```
DHCPDARGS=
```

DHCP およびその使用方法についての詳細は、16章DHCP サーバー を参照してください。

D.1.6. /etc/sysconfig/firstboot

/etc/sysconfig/firstboot ファイルは、firstboot ユーティリティーを実行するかどうかを定義します。デフォルトでは、以下のオプションが含まれます。

RUN_FIRSTBOOT=boolean

firstboot プログラムを実行するブール値(YES)または無効(NO)以下に例を示します。

```
RUN_FIRSTBOOT=NO
```

システムの初回起動時に、init プログラムは /etc/rc.d/init.d/firstboot スクリプトを呼び出します。これは、/etc/sysconfig/firstboot ファイルを検索します。このファイルに RUN_FIRSTBOOT=NO オプションが含まれていない場合は、firstboot プログラムが実行され、システムの初期設定でユーザーが提示されます。

FIRSTBOOT プログラムを再び実行できます。

次回システム起動時に firstboot プログラムを起動するには、RUN_FIRSTBOOT オプションの値を YES に変更し、シェルプロンプトで以下を入力します。

```
~]# chkconfig firstboot on
```

D.1.7. /etc/sysconfig/i18n

/etc/sysconfig/i18n 設定ファイルは、デフォルトの言語、サポートされる言語、およびデフォルトのシステムフォントを定義します。デフォルトでは、以下のオプションが含まれます。

pidgin=value

デフォルトの言語。以下に例を示します。

```
LANG="en_US.UTF-8"
```

SUPPORTED=値

サポートされる言語のコロン区切りリスト。以下に例を示します。

```
SUPPORTED="en_US.UTF-8:en_US:en"
```

SYSFONT=value

デフォルトのシステムフォントです。以下に例を示します。

```
SYSFONT="latarcyrheb-sun16"
```

D.1.8. /etc/sysconfig/init

/etc/sysconfig/init ファイルは、システムの起動プロセス中にどのように表示され、機能するかを制御します。デフォルトでは、以下のオプションが含まれます。

BOOTUP=値

起動スタイル。値は色（標準の 色 ブート表示）、verbose （より多くの情報を提供する古いスタイル表示）、または ANSI フォーマットなしの新しいスタイルの表示にはその他のものである必

要があります。以下に例を示します。

```
BOOTUP=color
```

RES_COL=value

ステータスラベルが開始される列の数。以下に例を示します。

```
RES_COL=60
```

MOVE_TO_COL=value

カーソルを RES_COL で指定された列に移動するターミナルシーケンス（上記を参照）。以下に例を示します。

```
MOVE_TO_COL="echo -en \\033[${RES_COL}G"
```

SETCOLOR_SUCCESS=value

成功の色を設定するターミナルシーケンス。以下に例を示します。

```
SETCOLOR_SUCCESS="echo -en \\033[0;32m"
```

SETCOLOR_FAILURE=value

失敗の色を設定するターミナルシーケンス。以下に例を示します。

```
SETCOLOR_FAILURE="echo -en \\033[0;31m"
```

SETCOLOR_WARNING=value

警告の色を設定するターミナルシーケンス。以下に例を示します。

```
SETCOLOR_WARNING="echo -en \\033[0;33m"
```

SETCOLOR_NORMAL=value

デフォルトの色を設定するターミナルシーケンス。以下に例を示します。

```
SETCOLOR_NORMAL="echo -en \\033[0;39m"
```

LOGLEVEL=value

コンソールの初期のロギングレベル。値 は 1 （カーネルパニックのみ）から 8 （デバッグ情報を含むすべて）の範囲にある必要があります。以下に例を示します。

```
LOGLEVEL=3
```

PROMPT=boolean

ホットキーの起動を有効(yes)または無効(no)するブール値。以下に例を示します。

```
PROMPT=yes
```

AUTOSWAP=boolean

スワップ署名のあるデバイスを有効(yes)または無効(no)するブール値。以下に例を示します。

```
AUTOSWAP=no
```

ACTIVE_CONSOLES=value

アクティブなコンソールの一覧。以下に例を示します。

```
ACTIVE_CONSOLES=/dev/tty[1-6]
```

SINGLE=value

シングルユーザーモードのタイプ。値 は /sbin/sulogin （ログインするパスワードの入力を求めるプロンプトが出されます）または /sbin/sushell （ユーザーを直接ログイン）のいずれかにする必要があります。以下に例を示します。

```
SINGLE=/sbin/sushell
```

D.1.9. /etc/sysconfig/ip6tables-config

/etc/sysconfig/ip6tables-config ファイルには、システムの起動時に IPv6 パケットフィルタリングを設定する、または ip6tables サービスが開始されるたびに、カーネルが使用する情報を保存します。ip6tables ルールに精通していない限り、変更しないでください。デフォルトでは、以下のオプションが含まれます。

IP6TABLES_MODULES=value

ファイアウォールルールの適用後に読み込まれるヘルパーのスペース区切りリスト。以下に例を示します。

```
IP6TABLES_MODULES="ip_nat_ftp ip_nat_irc"
```

IP6TABLES_MODULES_UNLOAD=boolean

ファイアウォールが停止または再起動時にアンロードされる(yes)モジュールまたは無効化(no)モジュールのアンロードを行うブール値。以下に例を示します。

```
IP6TABLES_MODULES_UNLOAD="yes"
```

IP6TABLES_SAVE_ON_STOP=boolean

ファイアウォールが停止したときに、現在のファイアウォールルールを有効化または無効化(no)するブール値。以下に例を示します。

```
IP6TABLES_SAVE_ON_STOP="no"
```

IP6TABLES_SAVE_ON_RESTART=boolean

ファイアウォールが再起動すると、現在のファイアウォールルールを有効化または無効化(no)するブール値。以下に例を示します。

```
IP6TABLES_SAVE_ON_RESTART="no"
```

IP6TABLES_SAVE_COUNTER=boolean

ルールカウンターとチェーンカウンターを保存するブール値(yes)または無効(no)以下に例を示します。

```
IP6TABLES_SAVE_COUNTER="no"
```

IP6TABLES_STATUS_NUMERIC=boolean

ステータス出力の IP アドレスおよびポート番号を有効化または無効化(no)するブール値。以下に例を示します。

```
IP6TABLES_STATUS_NUMERIC="yes"
```

IP6TABLES_STATUS_VERBOSE=boolean

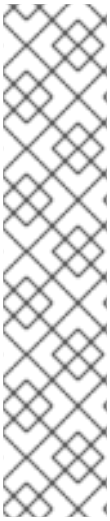
ステータスの出力でパケット数およびバイト数に関する情報を出力するブール値(yes)または無効(no)以下に例を示します。

```
IP6TABLES_STATUS_VERBOSE="no"
```

IP6TABLES_STATUS_LINENUMBERS=boolean

ステータス出力の行番号を表示(yes)または無効(no)するブール値。以下に例を示します。

```
IP6TABLES_STATUS_LINENUMBERS="yes"
```



IP6TABLES コマンドを使用してルールを作成します。

ip6tables コマンドを使用して、手動でルールを作成できます。作成したら、シェルプロンプトで以下を入力します。

```
~]# service ip6tables save
```

これにより、ルールが /etc/sysconfig/ip6tables に追加されます。このファイルが存在しても、それに保存されているファイアウォールルールは、システムの再起動やサービスの再起動で維持されます。

D.1.10. /etc/sysconfig/kernel

/etc/sysconfig/kernel 設定ファイルは、以下の 2 つのオプションを使用して、システムの起動時にカーネルの選択を制御します。

UPDATEDEFAULT=yes

このオプションでは、ブートエントリーの選択で、新たにインストールしたカーネルがデフォルトになります。

DEFAULTKERNEL=kernel

このオプションは、デフォルトとして使用するパッケージタイプを指定します。

D.1.10.1. 古いカーネルバージョンをデフォルトとして維持

ブートエントリーの選択で、古いカーネルバージョンをデフォルトとして維持するには、以下を実行します。

- *以下のように /etc/sysconfig/kernel の UPDATEDEFAULT オプションをコメントアウトします。*

```
# UPDATEDEFAULT=yes
```

D.1.10.2. カーネルデバッガーをデフォルトカーネルとして設定

ブートエントリーの選択で、カーネルデバッガーをデフォルトのカーネルとして設定するには、以下を行います。

- *以下のように /etc/sysconfig/kernel 設定ファイルを編集します。*

```
DEFAULTKERNEL=kernel-debug
```

D.1.11. /etc/sysconfig/keyboard

/etc/sysconfig/keyboard ファイルは、キーボードの動作を制御します。デフォルトでは、以下のオプションが含まれます。

KEYTABLE=value

キーテーブルファイルの名前。キーテーブルとして使用できるファイルは /lib/kbd/keymaps/i386/ ディレクトリーで開始され、そこから異なるキーボードレイアウトにブランチされ ます。KEYTABLE 設定に一致する最初のファイル名が使用されます。以下に例を示します。

```
KEYTABLE="us"
```

MODEL=value

キーボードモデル。以下に例を示します。

```
MODEL="pc105+inet"
```

LAYOUT=value

キーボードレイアウト。以下に例を示します。

```
LAYOUT="us"
```

KEYBOARDTYPE=value

キーボードタイプ。使用できる値は pc （PS/2 キーボード）または sun （Sun キーボード）です。以下に例を示します。

```
KEYBOARDTYPE="pc"
```

D.1.12. /etc/sysconfig/ldap

/etc/sysconfig/ldap ファイルは、LDAP サーバーの基本設定を保持します。デフォルトでは、以下のオプションが含まれます。

SLAPD_OPTIONS=value

slapd デーモンに渡される追加のオプション。以下に例を示します。

```
SLAPD_OPTIONS="-4"
```

SLURPD_OPTIONS=value

***slurpd* デーモンに渡す追加のオプション。以下に例を示します。**

```
SLURPD_OPTIONS=""
```

SLAPD_LDAP=boolean

LDAP over TCP(ldap:///)を使用して有効化(yes)または無効(no)するブール値。以下に例を示します。

```
SLAPD_LDAP="yes"
```

SLAPD_LDAPI=boolean

LDAP over IPC(ldapi:///)を使用して有効または無効にするブール値（つまり ldapi:///）。以下に例を示します。

```
SLAPD_LDAPI="no"
```

SLAPD_LDAPS=boolean

LDAP over TLS(ldaps:///)を使用して有効または無効にするブール値（つまり ldaps:///）。以下に例を示します。

```
SLAPD_LDAPS="no"
```

SLAPD_URLS=value

URL のスペース区切りの一覧。以下に例を示します。

```
SLAPD_URLS="ldapi:///var/lib/ldap_root/ldapi ldapi:/// ldaps:///"
```

SLAPD_SHUTDOWN_TIMEOUT=value

***slapd* がシャットダウンするのを待つ時間。以下に例を示します。**

```
SLAPD_SHUTDOWN_TIMEOUT=3
```

SLAPD_ULIMIT_SETTINGS=value

slapd デーモンを起動する前に ulimit に渡されるパラメーター。以下に例を示します。

```
SLAPD_ULIMIT_SETTINGS=""
```

LDAP およびその設定の詳細は、「OpenLDAP」 を参照してください。

D.1.13. /etc/sysconfig/named

/etc/sysconfig/named ファイルは、システムの起動時に名前 付き デーモンに引数を渡すために使用されます。デフォルトでは、以下のオプションが含まれます。

ROOTDIR=value

名前付き デーモンが実行される chroot 環境。値 は完全なディレクトリーパスでなければなりません。以下に例を示します。

```
ROOTDIR="/var/named/chroot"
```

chroot 環境を先に設定する必要があります（詳細は、シェルプロンプトで info chroot と入力します）。

OPTIONS=value

名前付き に渡す追加のオプション。以下に例を示します。

```
OPTIONS="-6"
```

-t オプションは使用しないでください。代わりに、上記のように ROOTDIR を使用します。

KEYTAB_FILE=value

キータブファイル名。以下に例を示します。

```
KEYTAB_FILE="/etc/named.keytab"
```

BIND DNS サーバーとその設定の詳細は、「BIND」 を参照してください。

D.1.14. /etc/sysconfig/network

/etc/sysconfig/network ファイルは、必要なネットワーク設定に関する情報を指定するために使用されます。デフォルトでは、以下のオプションが含まれます。

network networkING=boolean

ネットワークを有効化または無効化するブール値(yes)または無効(no)です。以下に例を示します。

```
NETWORKING=yes
```

HOSTNAME=value

マシンのホスト名。以下に例を示します。

```
HOSTNAME=penguin.example.com
```

このファイルには、以下のオプションの一部を含めることもできます。

GATEWAY=値

ネットワークのゲートウェイの IP アドレス。以下に例を示します。

```
GATEWAY=192.168.1.1
```

これは、インターフェースの ifcfg ファイルに GATEWAY ディレクティブがない場合に、デフォルトゲートウェイとして使用されます。

NM_BOND_VLAN_ENABLED=boolean

ボンディング、ブリッジ、VLAN インターフェースの検出および管理を NetworkManager アプリケーションが許可(yes)または禁止(no)するブール値。以下に例を示します。

```
NM_BOND_VLAN_ENABLED=yes
```

NM_CONTROLLED ディレクティブは、このオプションによって異なります。

注記

IPv6 を完全に無効にする場合は、以下の行を /etc/sysctl.conf に追加する必要があります。

```
net.ipv6.conf.all.disable_ipv6=1
```

```
net.ipv6.conf.default.disable_ipv6=1
```

さらに、カーネルコマンドラインに ipv6.disable=1 を追加すると、IPv6 を実装するカーネルモジュール net-pf-10 が無効になります。

カスタム INIT スクリプトの使用を回避

ネットワーク設定には、カスタムの init スクリプトを使用しないでください。ブート後ネットワークサービスの再起動を実行する際に、ネットワーク init スクリプト外で実行されるネットワーク設定をカスタム init スクリプトを設定すると、予期しない結果が得られます。

D.1.15. /etc/sysconfig/ntpd

/etc/sysconfig/ntpd ファイルは、システムの起動時に ntpd デーモンに引数を渡すために使用されます。デフォルトでは、以下のオプションが含まれます。

OPTIONS=value

ntpd に渡される追加のオプション。以下に例を示します。

```
OPTIONS="-u ntp:ntp -p /var/run/ntpd.pid -g"
```

ntpd デーモンの設定方法に関する詳細は、「ネットワーク時刻プロトコルのプロパティ」 または「ネットワーク時刻プロトコルの設定」 を参照してください。

D.1.16. /etc/sysconfig/quagga

/etc/sysconfig/quagga ファイルは、Quagga デーモンの基本設定を保持します。デフォルトでは、以下のオプションが含まれます。

QCONFDIR=value

Quagga デーモンの設定ファイルが含まれるディレクトリー。以下に例を示します。

```
QCONFDIR="/etc/quagga"
```

BGPD_OPTS=value

bgpd デーモンに渡すその他のオプション。以下に例を示します。

```
BGPD_OPTS="-A 127.0.0.1 -f ${QCONFDIR}/bgpd.conf"
```

OSPF6D_OPTS=value

ospf6d デーモンに渡される追加のオプション。以下に例を示します。

```
OSPF6D_OPTS="-A ::1 -f ${QCONFDIR}/ospf6d.conf"
```

OSPFD_OPTS=value

ospfd デーモンに渡される追加のオプション。以下に例を示します。

```
OSPFD_OPTS="-A 127.0.0.1 -f ${QCONFDIR}/ospfd.conf"
```

RIPD_OPTS=value

ripd デーモンに渡す追加のオプション。以下に例を示します。

```
RIPD_OPTS="-A 127.0.0.1 -f ${QCONFDIR}/ripd.conf"
```

RIPNGD_OPTS=value

ripngd デーモンに渡される追加のオプション。以下に例を示します。

```
RIPNGD_OPTS="-A ::1 -f ${QCONFDIR}/ripngd.conf"
```

ZEBRA_OPTS=value

zebra デーモンに渡される追加オプション。以下に例を示します。

```
ZEBRA_OPTS="-A 127.0.0.1 -f ${QCONFDIR}/zebra.conf"
```

ISISD_OPTS=value

isisd デーモンに渡される追加オプション。以下に例を示します。

```
ISISD_OPTS="-A ::1 -f ${QCONFDIR}/isisd.conf"
```

WATCH_OPTS=value

watchquagga デーモンに渡される追加オプション。以下に例を示します。

```
WATCH_OPTS="-Az -b_ -r/sbin/service_%s_restart -s/sbin/service_%s_start -
k/sbin/service_%s_stop"
```

WATCH_DAEMONS=value

監視されるデーモンのスペース区切りの一覧。以下に例を示します。

```
WATCH_DAEMONS="zebra bgpd ospfd ospf6d ripd ripngd"
```

D.1.17. /etc/sysconfig/radvd

/etc/sysconfig/radvd ファイルを使用して、システムの起動時に radvd デーモンに引数を渡します。デフォルトでは、以下のオプションが含まれます。

OPTIONS=value

radvd デーモンに渡す追加のオプション。以下に例を示します。

```
OPTIONS="-u radvd"
```

D.1.18. /etc/sysconfig/samba

/etc/sysconfig/samba ファイルを使用して、システムの起動時に Samba デーモンに引数を渡します。デフォルトでは、以下のオプションが含まれます。

SMBDOPTIONS=value

smbd に渡される追加のオプション。以下に例を示します。

```
SMBDOPTIONS="-D"
```

NMBDOPTIONS=値

nmbd に渡される追加のオプション。以下に例を示します。

```
NMBDOPTIONS="-D"
```

WINBINDOPTIONS=値

winbindd に渡す追加のオプション。以下に例を示します。

```
WINBINDOPTIONS=""
```

Samba およびその設定の詳細は、「Samba」 を参照してください。

D.1.19. /etc/sysconfig/saslauthd

/etc/sysconfig/saslauthd ファイルは、SASL 認証サーバーである saslauthd に渡される引数を制御するために使用されます。デフォルトでは、以下のオプションが含まれます。

SOCKETDIR=value

saslauthd 's listening socket のディレクトリー。以下に例を示します。

```
SOCKETDIR=/var/run/saslauthd
```

MECH=value

ユーザーパスワードの検証に使用する認証メカニズム。以下に例を示します。

```
MECH=pam
```

DAEMONOPTS=value

saslauthd サービスを起動するために /etc/rc.d/init.d/saslauthd init スクリプトによって使用される daemon() 関数に渡されるオプション。以下に例を示します。

```
DAEMONOPTS="--user saslauth"
```

FLAGS=value

saslauthd サービスに渡される追加のオプション。以下に例を示します。

```
FLAGS=
```

D.1.20. /etc/sysconfig/selinux

/etc/sysconfig/selinux ファイルには、SELinux の基本設定オプションが含まれます。これは /etc/selinux/config へのシンボリックリンクで、デフォルトでは以下のオプションが含まれます。

SELINUX=value

セキュリティーポリシー。この 値 には、Enforcing （セキュリティーポリシーは常に強制される）、Permissive （ポリシーの強制、適切な警告が表示される）、または disabled （ポリシーが使用されない）のいずれかを使用できます。以下に例を示します。

```
SELINUX=enforcing
```

SELINUXTYPE=値

保護タイプ。この 値 は、ターゲット（対象 のプロセスが保護されています）または ml（複数レベルの セキュリティー保護）のいずれかにすることができます。以下に例を示します。

```
SELINUXTYPE=targeted
```

D.1.21. /etc/sysconfig/sendmail

/etc/sysconfig/sendmail は、Sendmail アプリケーションのデフォルト値を設定するのに使用されます。デフォルトでは、以下の値が含まれます。

DAEMON=boolean

sendmail をデーモンとして実行させるブール値(yes)または無効(no)以下に例を示します。

```
DAEMON=yes
```

QUEUE=value

メッセージを処理する間隔。以下に例を示します。

```
QUEUE=1h
```

Sendmail およびその設定の詳細は、 「Sendmail」 を参照してください。

D.1.22. /etc/sysconfig/spamassassin

/etc/sysconfig/spamassassin ファイルは、システムの起動時に、pam d デーモン（デーモン化さ

れたバージョンの Spamassassin）に引数を渡すために使用されます。デフォルトでは、以下のオプションが含まれます。

SPAMDOPTIONS=value

spamd デーモンに渡される追加のオプション。以下に例を示します。

```
SPAMDOPTIONS="-d -c -m5 -H"
```

Spamassassin およびその設定の詳細は、「spam フィルター」 を参照してください。

D.1.23. /etc/sysconfig/squid

/etc/sysconfig/squid ファイルを使用して、システムの起動時に squid デーモンに引数を渡します。デフォルトでは、以下のオプションが含まれます。

SQUID_OPTS=value

squid デーモンに渡す追加のオプション。以下に例を示します。

```
SQUID_OPTS=""
```

SQUID_SHUTDOWN_TIMEOUT=value

squid デーモンがシャットダウンするのを待つ時間。以下に例を示します。

```
SQUID_SHUTDOWN_TIMEOUT=100
```

SQUID_CONF=value

デフォルトの設定ファイルです。以下に例を示します。

```
SQUID_CONF="/etc/squid/squid.conf"
```

D.1.24. /etc/sysconfig/system-config-users

/etc/sysconfig/system-config-users ファイルは User Manager ユーティリティーの設定ファイルであるため、手動で編集しないでください。デフォルトでは、以下のオプションが含まれます。

FILTER=ブール値

システムユーザーの有効(true)または無効(false)のフィルターを有効にするブール値。以下に例を示します。

```
FILTER=true
```

ASSIGN_HIGHEST_UID=boolean

新しく追加したユーザーに、利用可能な UID の最大数を割り当てるブール値(true)または無効(false)です。以下に例を示します。

```
ASSIGN_HIGHEST_UID=true
```

ASSIGN_HIGHEST_GID=boolean

新しく追加したグループに利用可能な最大 GID を割り当てる有効化(true)または無効(false)のブール値。以下に例を示します。

```
ASSIGN_HIGHEST_GID=true
```

PREFER_SAME_UID_GID=boolean

可能な場合は、新たに追加したユーザーに同じ UID と GID を使用して有効化(true)または無効(false)するブール値。以下に例を示します。

```
PREFER_SAME_UID_GID=true
```

User Manager およびその使用方法についての詳細は、「ユーザーマネージャーアプリケーションを使用したユーザーの管理」 を参照してください。

D.1.25. /etc/sysconfig/vncservers

/etc/sysconfig/vncservers ファイルは、仮想ネットワークコンピューティング (VNC)サーバーの起動方法を設定します。デフォルトでは、以下のオプションが含まれます。

VNCSERVERS=value

スペースで区切られた 表示:ユーザー名 のペア以下に例を示します。

```
VNCSERVERS="2:myusername"
```

VNCSERVERARGS[display]=value

指定された ディスプレイ で実行している VNC サーバーに渡される追加の引数。以下に例を示します。

```
VNCSERVERARGS[2]="-geometry 800x600 -nolisten tcp -localhost"
```

D.1.26. /etc/sysconfig/xinetd

/etc/sysconfig/xinetd ファイルは、システムの起動時に xinetd デーモンに引数を渡すために使用されます。デフォルトでは、以下のオプションが含まれます。

EXTRAOPTIONS=値

xinetd に渡される追加オプション。以下に例を示します。

```
EXTRAOPTIONS=""
```

XINETD_LANG=value

xinetd が開始するすべてのサービスに渡すロケール情報。xinetd 環境からロケール情報を削除するには、空の文字列("")または none を使用できます。以下に例を示します。

```
XINETD_LANG="en_US"
```

xinetd サービスの設定方法は 12章サービスおよびデーモン を参照してください。

D.2. /ETC/SYSCONFIG/ ディレクトリーのディレクトリー

以下のディレクトリーは、通常 /etc/sysconfig/ にあります。

/etc/sysconfig/cbq/

このディレクトリーには、ネットワークインターフェースの帯域幅管理にクラス ベースの Queuing を 実行するために必要な設定ファイルが含まれています。CBQ は、IP アドレス、プロトコル、およびアプリケーションタイプの組み合わせに基づいて、ユーザートラフィックをクラスの階層に分割します。

/etc/sysconfig/networking/

このディレクトリーは、非推奨になった Network Administration Tool (system-config-network)により使用され、その内容は手動で編集しないでください。グラフィカル設定ツールを使用したネットワークインターフェースの設定に関する詳細は、10章NetworkManager を参照してください。

/etc/sysconfig/network-scripts/

このディレクトリーには、以下のネットワーク関連の設定ファイルが含まれます。

- *eth0 イーサネットインターフェースの ifcfg-eth0 など、設定済みの各ネットワークインターフェースのネットワーク設定ファイル。*
- *ifup や ifdown などのネットワークインターフェースの稼働/ダウンに使用されるスクリプト。*
- *ifup-isdn や ifdown-isdn など、ISDN インターフェースを稼働/ダウンさせるのに使用するスクリプト。*
- *直接編集すべきでないさまざまな共有ネットワーク機能スクリプト。*

/etc/sysconfig/network-scripts/ ディレクトリーの詳細は、11章Network Interfaces を参照してください。

/etc/sysconfig/rhn/

このディレクトリーには、Red Hat Network の設定ファイルと GPG キーが含まれます。このディレクトリー内のファイルは手動で編集する必要はありません。Red Hat Network の詳細は、Red Hat Network の Web サイト(https://rhn.redhat.com/)を参照してください。

D.3. その他のリソース

本章では、/etc/sysconfig/ ディレクトリーのファイルの概要としてのみ説明します。以下のソースにはより包括的な情報が記載されています。

D.3.1. インストールされているドキュメント

/usr/share/doc/initscripts-version/sysconfig.txt

/etc/sysconfig/ ディレクトリーにあるファイルのさらに権威一覧と、それらで利用できる設定オプション。

付録E PROC ファイルシステム

Linux カーネルには、コンピューター上の物理デバイスへのアクセスを制御し、それらのデバイスとプロセスがいつ、どのような方法で情報のやりとりを行うかをスケジュールするという、2 つの主要な機能があります。/proc/ ディレクトリー（ proc ファイルシステムとも呼ばれます）には、カーネルの現在の状態を表す特殊なファイルの階層が含まれており、アプリケーションとユーザーがシステムのカーネルビューにピア接続できるようにします。

/proc/ ディレクトリーには、システムハードウェアと実行中のプロセスの詳細情報が記載されています。さらに、/proc/ 内のファイルの一部は、ユーザーおよびアプリケーションが設定変更をカーネルと通信できるように操作できます。

/PROC/IDE/ ディレクトリーおよび /PROC/PCI/ ディレクトリー

2.6 カーネルの後のバージョンでは、/proc/ide/ ディレクトリーおよび /proc/pci/ ディレクトリーが廃止されました。/proc/ide/ ファイルシステムは sysfs のファイルに置き換えられました。PCI デバイスに関する情報を取得するには、代わりに lspci を使用してください。sysfs または lspci の詳細は、それぞれの man ページを参照してください。

E.1. 仮想ファイルシステム

Linux システムでは、すべてのデータが ファイル として保管されます。大半のユーザーは、主要な 2 つのファイルタイプ (テキストとバイナリ) について精通していますただし、/proc/ ディレクトリーには、仮想 ファイル と呼ばれる別のタイプのファイルが含まれています。したがって、/proc/ は多くの場合、仮想ファイルシステム と呼ばれます。

仮想ファイルには、固有の性質があります。大半の仮想ファイルは、サイズがゼロバイトで表示されていても、表示すると大量の情報が格納されている場合があります。又、仮想ファイルの時刻及び日付のスタンプの大半は、現在の時刻と日付を反映します。これは、常に更新されていることを示しています。

/proc/interrupts、/proc/meminfo、/proc/mounts、および /proc/partitions などの仮想ファイルは、システムのハードウェアの最新の状態を保ちます。/proc/filesystems ファイルや /proc/sys/ ディレクトリーは、システム設定情報とインターフェースを提供します。

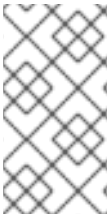
情報を体系化するために、同様のトピックに関する内容が記載されたファイルは、仮想ディレクトリ/サブディレクトリにグループ化されます。プロセスディレクトリには、システムで実行中の各プロセスに関する情報が格納されます。

E.1.1. 仮想ファイルの表示

/proc/ ファイル内のほとんどのファイルは、テキストファイルと同様に動作し、有用なシステムおよびハードウェアデータを人間が判読できるテキスト形式で保存します。そのため、cat、より 、以下を使用してそれらを表示できます。たとえば、システムの CPU に関する情報を表示するには、cat /proc/cpuinfo を実行します。これにより、以下のような出力が返されます。

```
processor : 0
vendor_id : AuthenticAMD
cpu family : 5
model  : 9
model name : AMD-K6(tm) 3D+
Processor stepping : 1 cpu
MHz  : 400.919
cache size : 256 KB
fdiv_bug : no
hlt_bug  : no
f00f_bug : no
coma_bug : no
fpu  : yes
fpu_exception : yes
cpuid level : 1
wp  : yes
flags  : fpu vme de pse tsc msr mce cx8 pge mmx syscall 3dnow k6_mtrr
bogomips : 799.53
```

/proc/ の一部のファイルには、人間が判読できない情報が含まれます。このようなファイルから情報を取得するには、lspci、apm、free、top などのツールを使用します。



特定のファイルは、ROOT 権限でしかアクセスできない

/proc/ ディレクトリーにある仮想ファイルの一部は、root ユーザーのみが読み取りできます。

E.1.2. 仮想ファイルの変更

一般的なルールとして、/proc/ ディレクトリー内のほとんどの仮想ファイルは読み取り専用です。ただし、カーネルの設定の調整にも使用できます。これは特に /proc/sys/ サブディレクトリー内のファイルに該当します。

仮想ファイルの値を変更するには、以下のコマンドを使用します。

```
echo value > /proc/file
```

たとえば、ホスト名をすぐに変更するには、次のコマンドを実行します。

```
echo www.example.com > /proc/sys/kernel/hostname
```

他のファイルはバイナリースイッチまたはブール値スイッチとして機能します。cat /proc/sys/net/ipv4/ip_forward を入力すると、0 （off または false）または 1 （on または true）を返します。0 は、カーネルがネットワークパケットを転送していないことを示します。パケット転送を有効にするには、echo 1 > /proc/sys/net/ipv4/ip_forward を実行します。

SYSCTL コマンド

/proc/sys/ サブディレクトリーの設定を変更するのに使用される別のコマンドは、/sbin/sysctl です。このコマンドについての詳しい情報は、を参照してください。「sysctl コマンドの使用」

/proc/sys/ サブディレクトリーで利用可能なカーネル設定ファイルの一覧は、「/proc/sys/」 を参照してください。

E.2. PROC ファイルシステム内のトップレベルのファイル

以下は、/proc/ ディレクトリーの最上位のより有用な仮想ファイルの一覧です。

ファイルの内容が異なる場合があります。

ほとんどの場合、本セクションに記載するファイルの内容は、マシンにインストールされているファイルと同じではありません。これは、Red Hat Enterprise Linuxnbsp;Hat Enterprise Red Hat Enterprise Linuxnbsp;Linux が本書の作業用に実行しているハードウェアに固有の情報の多くであるためです。

E.2.1. /proc/buddyinfo

/proc/buddyinfo ファイルは、主にメモリーの断片化問題の診断に使用されます。この出力は、使用するメモリーレイアウトによって異なります。これはアーキテクチャー固有のものです。32 ビットシステムの例を以下に示します。

```
Node 0, zone      DMA     90      6      2      1      1      ...
Node 0, zone   Normal   1650    310      5      0      0      ...
Node 0, zone  HighMem      2      0      0      1      1      ...
```

*buddy アルゴリズムを使用すると、各列は、特定の時点で利用可能な特定の順序のメモリーページ数（特定のサイズ）の数を表します。上記の例では、ゾーン DMA の場合は 2^0*PAGE_SIZE バイトの大きいメモリーチャンクがあります。同様に、2 つの^1*PAGE_SIZE チャンクと 2 の^2*PAGE_SIZE チャンクが利用できるメモリーがあります。*

DMA 行は、システムの最初の 16 MB のメモリーを参照し、HighMem 行はシステムの 896 MB を超えるすべてのメモリーを参照し、Normal 行はメモリーを参照します。

64 ビットシステムでは、出力は以下のようになります。

```
Node 0, zone      DMA      0      3      1      2      4      3   1   2   3   3   1
Node 0, zone    DMA32    295   25850   7065   1645   835   220  78   6   0   1   0
Node 0, zone   Normal   3824    3359    736    159    31     3   1   1   1   1   0
```

DMA 行は、システムの最初の 16 MB のメモリーを参照します。DMA32 行は、4 GB を超えるメモリーに対応できないデバイスに割り当てられるすべてのメモリーを参照します。また、Normal 行は、システム上に 4 GB を超えるすべてのメモリーが含まれます。DMA32

E.2.2. /proc/cmdline

このファイルは、起動時にカーネルに渡されるパラメーターを示しています。/proc/cmdline ファイルのサンプルは以下のようになります。

```
ro root=/dev/VolGroup00/LogVol00 rhgb quiet 3
```

これは、カーネルが最初のボリュームグループ((ro))にある読み取り専用（LogVol00で署名）にマウントされていることを示しています。/dev/VolGroup00LogVol00 は、LVM 以外のシステム（論理ボリューム管理）のディスクパーティションと同等です。/dev/VolGroup00 は、/dev/hda1 の概念と同様ですが、はるかに拡張可能です。

Red Hat Enterprise Linuxnbsp;Hat Enterprise Linuxnbsp;Linux で使用される LVM の詳細は、http://www.tldp.org/HOWTO/LVM-HOWTO/index.html を参照してください。

次に、rhgb は rhgb パッケージがインストールされ、グラフィカルブートがサポートされていることを伝えます。ここでは、/etc/inittab に id:5:initdefault: に設定されたデフォルト のランレベルが表示されます。

最後に、quiet は、すべての詳細なカーネルメッセージが起動時に抑制されていることを示します。

E.2.3. /proc/cpuinfo

この仮想ファイルは、システムで使用されるプロセッサーの種別を特定します。以下は、/proc/cpuinfo の典型的な出力の例です。

```
processor : 0
vendor_id : GenuineIntel
cpu family : 15
model  : 2
model name : Intel(R) Xeon(TM) CPU 2.40GHz
stepping : 7 cpu
MHz  : 2392.371
cache size : 512 KB
physical id : 0
siblings : 2
runqueue : 0
fdiv_bug : no
hlt_bug  : no
f00f_bug : no
coma_bug : no
fpu  : yes
fpu_exception : yes
cpuid level : 2
wp  : yes
flags  : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca  cmov pat pse36 clflush
dts acpi mmx fxsr sse sse2 ss ht tm
bogomips : 4771.02
```

- processor - 各プロセッサーに識別番号を提供します。プロセッサーが 1 つあるシステムでは、0 のみが存在します。
- cpu family - 権威的に、システム内のプロセッサーの種類を特定します。Intel ベースのシステムの場合は、「86」の前に数字を配置して値を決定します。これは、586、486、386 などの古いシステムのアーキテクチャーを特定しようとする場合に役立ちます。これらの特定のアーキテクチャーごとに RPM パッケージがコンパイルされているため、この値は、インストールするパッケージを判別するのに役立ちます。
- model name - プロセッサーの共通名（プロジェクト名を含む）を表示します。
- cpu MHz - プロセッサーから 10 進数の間の正確な速度を megahertz に示します。
- cache size - プロセッサーで利用可能なレベル 2 メモリーキャッシュの量を表示します。
-

siblings - ハイパースレッディングを使用するアーキテクチャーについて、同じ物理 CPU 上のシブリング CPU の合計数を表示します。

- *flags - 浮動小数点単位(FPU)の存在や MMX 命令の処理機能など、プロセッサーにさまざまな特性を定義します。*

E.2.4. /proc/crypto

このファイルは、Linux カーネルが使用するインストール済みの暗号をすべて一覧表示します。これには、それぞれの追加情報も含まれます。/proc/crypto ファイルのサンプルは以下のようになります。

```
name         : sha1
module       : kernel
type         : digest
blocksize    : 64
digestsize   : 20
name         : md5
module       : md5
type         : digest
blocksize    : 64
digestsize   : 16
```

E.2.5. /proc/devices

このファイルは、現在設定されているさまざまな文字およびブロックデバイスを表示します（モジュールが読み込まれていないデバイスを含めません）。以下は、このファイルからの出力例です。

```
Character devices:
  1 mem
  4 /dev/vc/0
  4 tty
  4 ttyS
  5 /dev/tty
  5 /dev/console
  5 /dev/ptmx
  7 vcs
  10 misc
  13 input
  29 fb
  36 netlink
  128 ptm
  136 pts
  180 usb

Block devices:
  1 ramdisk
  3 ide0
  9 md
```

```
22 ide1
253 device-mapper
254 mdp
```

/proc/devices からの出力には、デバイスのメジャー番号と名前が含まれており、Character devices と Block devices の 2 つの主要なセクションに分かれています。

文字デバイス は ブロックデバイス と似ていますが、2 つの基本的な違いを除きます。

1. 文字デバイスにはバッファーは必要ありません。ブロックデバイスはバッファーを利用できるため、それに対応する前に要求を順序付けることができます。これは、ハードドライブなどの情報を格納するように設計されたデバイスで重要になります。デバイスに書き込む前に情報を順序付ける機能により、より効率的な順序で配置できるためです。

2. 文字デバイスは、事前に設定されたサイズでデータを送信します。ブロックデバイスは、デバイスごとに設定されたサイズのブロックの情報を送信および受信できます。

デバイスの詳細は、kernel-doc パッケージの devices.txt ファイル（「その他のリソース」を参照）を参照してください。

E.2.6. /proc/dma

このファイルには、使用中の登録済みの ISA DMA チャンネルの一覧が含まれます。/proc/dma ファイルのサンプルは以下のようになります。

```
4: cascade
```

E.2.7. /proc/execdomains

このファイルは、Linux カーネルで 現在対応している実行ドメイン と、それがサポートする個人機能の範囲を一覧表示します。

```
0-0   Linux           [kernel]
```

実行ドメインは、オペレーティングシステムの「権限」とみなすことができます。Solaris、UnixWare、FreeBSD などの他のバイナリー形式を Linux と併用できるため、プログラマーはタスクのパーソナリティーを変更することで、オペレーティングシステムがこれらのバイナリーからシステムコールを処理する方法を変更できます。PER_LINUX 実行ドメインを除き、動的にロード可能なモジュールとして異なるパーソナリティーを実装できます。

E.2.8. /proc/fb

このファイルには、フレームバッファーデバイス番号とそれを制御するドライバーが含まれます。フレームバッファーデバイスを含むシステムの /proc/fb の典型的な出力は以下のようになります。

```
0 VESA VGA
```

E.2.9. /proc/filesystems

このファイルは、カーネルが現在対応しているファイルシステムタイプの一覧を表示します。一般的な /proc/filesystems ファイルからの出力例を以下に示します。

```
nodev   sysfs
nodev   rootfs
nodev   bdev
nodev   proc
nodev   sockfs
nodev   binfmt_misc
nodev   usbfs
nodev   usbdevfs
nodev   futexfs
nodev   tmpfs
nodev   pipefs
nodev   eventpollfs
nodev   devpts
 ext2
nodev   ramfs
nodev   hugetlbfs
 iso9660
nodev   mqueue
 ext3
nodev   rpc_pipefs
nodev   autofs
```

最初の列は、ファイルシステムがブロックデバイスにマウントされているかどうかを示します。nodev で始まるものは、デバイスにマウントされません。2 列目には、サポートされるファイルシステムの名前が一覧表示されます。

mount コマンドは、引数として指定されていない場合にここに一覧表示されるファイルシステムを循環します。

E.2.10. /proc/interrupts

このファイルは、x86 アーキテクチャー上の IRQ ごとの割り込みの数を記録します。標準の /proc/interrupts は以下のようになります。

```
  CPU0
  0:   80448940        XT-PIC  timer
  1:     174412        XT-PIC  keyboard
  2:          0        XT-PIC  cascade
  8:          1        XT-PIC  rtc
 10:     410964        XT-PIC  eth0
 12:      60330        XT-PIC  PS/2 Mouse
 14:    1314121        XT-PIC  ide0
 15:    5195422        XT-PIC  ide1
NMI:          0
ERR:          0
```

マルチプロセッサーマシンの場合、このファイルは若干異なる場合があります。

```
   CPU0       CPU1
  0: 1366814704          0        XT-PIC  timer
  1:        128        340    IO-APIC-edge  keyboard
  2:          0          0        XT-PIC  cascade
  8:          0          1    IO-APIC-edge  rtc
 12:       5323       5793    IO-APIC-edge  PS/2 Mouse
 13:          1          0        XT-PIC  fpu
 16:   11184294   15940594   IO-APIC-level  Intel EtherExpress Pro 10/100 Ethernet
 20:    8450043   11120093   IO-APIC-level  megaraid
 30:      10432      10722   IO-APIC-level  aic7xxx
 31:         23         22   IO-APIC-level  aic7xxx
NMI:          0
ERR:          0
```

最初の列は IRQ 番号を参照します。システムの各 CPU には、独自の列と IRQ ごとの独自の割り込みの数があります。次のコラムは割り込みのタイプを報告し、最後の列には IRQ にあるデバイスの名前が含まれます。

このファイルに表示される割り込みのタイプはそれぞれ、アーキテクチャー固有のものであり、異なることを意味します。x86 マシンでは、以下の値が一般的です。

- XT-PIC - これは、古い AT コンピューター割り込みです。
- IO-APIC-edge - この割り込みの voltage シグナルは low から high に移行し、割り込みが発生し、1 回のみ通知される エッジ を作成します。この種の割り込みや IO-APIC-level 割り込みは、586 ファミリーのプロセッサーが搭載されたシステムでのみ表示されます。
- IO-APIC-level - シグナルが再び低いまで voltage シグナルが高いと割り込みを生成します。

E.2.11. /proc/iomem

このファイルには、各物理デバイスのシステムメモリーの現在のマップが表示されます。

```
00000000-0009fbff : System RAM
0009fc00-0009ffff : reserved
000a0000-000bffff : Video RAM area
000c0000-000c7fff : Video ROM
000f0000-000fffff : System ROM
00100000-07ffffff : System RAM
00100000-00291ba8 : Kernel code
00291ba9-002e09cb : Kernel data
e0000000-e3ffffff : VIA Technologies, Inc. VT82C597 [Apollo VP3] e4000000-e7ffffff : PCI Bus
#01
e4000000-e4003fff : Matrox Graphics, Inc. MGA G200 AGP
e5000000-e57fffff : Matrox Graphics, Inc. MGA G200 AGP
e8000000-e8ffffff : PCI Bus #01
e8000000-e8ffffff : Matrox Graphics, Inc. MGA G200 AGP
ea000000-ea00007f : Digital Equipment Corporation DECchip 21140 [FasterNet]
ea000000-ea00007f : tulip ffff0000-ffffffff : reserved
```

最初の列には、異なるタイプのメモリーによって使用されるメモリーレジスターが表示されます。2 列目には、それらのレジスター内にあるメモリーの種類が記載され、どのメモリーレジスターがシステム RAM 内のカーネルで使用されているか、またはネットワークインターフェースカードに複数のイーサネットポートがある場合は、メモリーレジスターがポートごとに割り当てられます。

E.2.12. /proc/ioports

/proc/ioports の出力には、デバイスの入力または出力通信に使用される現在登録されているポートリージョンの一覧が示されます。このファイルは非常に長い場合があります。以下は部分的なリストです。

```
0000-001f : dma1
0020-003f : pic1
0040-005f : timer
0060-006f : keyboard
0070-007f : rtc
0080-008f : dma page reg
00a0-00bf : pic2
00c0-00df : dma2
00f0-00ff : fpu
0170-0177 : ide1
01f0-01f7 : ide0
02f8-02ff : serial(auto)
0376-0376 : ide1
03c0-03df : vga+
03f6-03f6 : ide0
03f8-03ff : serial(auto)
0cf8-0cff : PCI conf1
```

```
d000-dfff : PCI Bus #01
e000-e00f : VIA Technologies, Inc. Bus Master IDE
e000-e007 : ide0
e008-e00f : ide1
e800-e87f : Digital Equipment Corporation DECchip 21140 [FasterNet]
e800-e87f : tulip
```

最初の列は、2 番目の列に一覧表示されているデバイス用に予約される I/O ポートアドレス範囲を示します。

E.2.13. /proc/kcore

このファイルは、システムの物理メモリーを表し、コアファイル形式で保存されます。ほとんどの /proc/ ファイルとは異なり、kcore はサイズを表示します。この値はバイト単位で指定され、使用される物理メモリー(RAM)のサイズと 4 KB を加えたサイズと同じになります。

このファイルのコンテンツは、gdb などのデバッガーで検証されるため、人間が判読できる訳ではありません。

/PROC/KCORE の内容の表示を試行しないでください。

/proc/kcore 仮想ファイルは表示しないでください。ターミナルでファイルスクラブルテキスト出力の内容。このファイルが誤って表示されていた場合は、Ctrl+C を押してプロセスを停止し、reset と入力してコマンドラインプロンプトに戻ります。

E.2.14. /proc/kmsg

このファイルは、カーネルによって生成されたメッセージを保持するために使用されます。これらのメッセージは、/sbin/klogd や /bin/dmesg などの他のプログラムで選択されます。

E.2.15. /proc/loadavg

このファイルは、CPU と IO の両方に関して負荷平均と、アップタイム およびその他のコマンドで使用される追加のデータについて見つかります。/proc/loadavg ファイルのサンプルは以下のようになります。

```
0.20 0.18 0.12 1/80 11206
```

最初の 3 列は、最後の 1、5、15 分間の CPU および IO の使用状況を測定します。4 列目には、現在実行中のプロセス数とプロセス合計数が表示されます。最後の列には、最後に使用したプロセス ID が表示されます。

さらに、負荷平均は、実行する準備ができているプロセスの数も参照します（例： 実行キューで、CPU 共有を待機します）。

E.2.16. /proc/locks

このファイルは、カーネルが現在ロックされているファイルを表示します。このファイルには内部カーネルのデバッグデータが含まれ、システムの使用により大きく異なる可能性があります。負荷の少ないシステムの /proc/locks ファイルの例を以下に示します。

```
1: POSIX  ADVISORY  WRITE 3568 fd:00:2531452 0 EOF
2: FLOCK  ADVISORY  WRITE 3517 fd:00:2531448 0 EOF
3: POSIX  ADVISORY  WRITE 3452 fd:00:2531442 0 EOF
4: POSIX  ADVISORY  WRITE 3443 fd:00:2531440 0 EOF
5: POSIX  ADVISORY  WRITE 3326 fd:00:2531430 0 EOF
6: POSIX  ADVISORY  WRITE 3175 fd:00:2531425 0 EOF
7: POSIX  ADVISORY  WRITE 3056 fd:00:2548663 0 EOF
```

各ロックには、一意の番号で始まる独自の行があります。2 番目の列は、使用されるロックのクラスを指し、FLOCK は flock システムコールから古いスタイルの UNIX ファイルロックを表し、POSIX はロックのシステムコールからの新しい POSIX ロックを表します。

3 番目の列には、ADVISORY または MANDATORY の 2 つの値を指定することができます。ADVISORY つまり、ロックは他のユーザーがデータにアクセスしないようにし、他のユーザーがロックを試みることのみを防ぎます。MANDATORY ロックが保持される間にデータへの他のアクセスが許可されないことを意味します。4 番目のコラムは、ロックがホルダー READ または WRITE のファイルへのアクセスを許可するかどうかを示します。5 列目には、ロックを保持するプロセスの ID が表示されます。6 番目のコラムは、MAJOR-DEVICE:MINOR-DEVICE:INODE-NUMBER の形式で、ロックされるファイルの ID を表示します。7 番目と 8h 列は、ファイルのロックされたリージョンの開始と終了を示しています。

E.2.17. /proc/mdstat

このファイルには、マルチディスク、RAID 設定の現在の情報が含まれます。システムにこのような設定が含まれていない場合は、/proc/mdstat は以下のようになります。

```
Personalities :  read_ahead not set unused devices: <none>
```

このファイルは、ソフトウェア RAID または md デバイスが存在しない限り、上記と同じ状態にな

ります。その場合は、/proc/mdstat を表示して、mdX RAID デバイスの現在の状態を見つけます。

以下の /proc/mdstat ファイルは、現在ディスクを再同期している間に、md0 が RAID 1 デバイスとして設定されたシステムを示しています。

```
Personalities : [linear] [raid1] read_ahead 1024 sectors
md0: active raid1 sda2[1] sdb2[0] 9940 blocks [2/2] [UU] resync=1% finish=12.3min algorithm 2 [3/3] [UUU]
unused devices: <none>
```

E.2.18. /proc/meminfo

これは、システムの RAM 使用率に関する多くの有用な情報を報告するため、/proc/ ディレクトリー内で一般的に使用されるファイルの 1 つです。

以下の /proc/meminfo 仮想ファイルのサンプルは、2 GB の RAM と 1 GB のスワップ領域が搭載されているシステムです。

```
MemTotal:       1921988 kB
MemFree:        1374408 kB
Buffers:          32688 kB
Cached:          370540 kB
SwapCached:           0 kB
Active:          344604 kB
Inactive:         80800 kB
Active(anon):     22364 kB
Inactive(anon):       4 kB
Active(file):    322240 kB
Inactive(file):   80796 kB
Unevictable:          0 kB
Mlocked:              0 kB
SwapTotal:      1048572 kB
SwapFree:       1048572 kB
Dirty:               48 kB
Writeback:            0 kB
AnonPages:        22260 kB
Mapped:           13628 kB
Shmem:              196 kB
Slab:             91648 kB
SReclaimable:     34024 kB
SUnreclaim:       57624 kB
KernelStack:       2880 kB
PageTables:        3620 kB
NFS_Unstable:         0 kB
Bounce:               0 kB
WritebackTmp:         0 kB
CommitLimit:    2009564 kB
Committed_AS:    134216 kB
```

```
VmallocTotal:   34359738367 kB
VmallocUsed:       12276 kB
VmallocChunk:   34359712840 kB
HardwareCorrupted:     0 kB
AnonHugePages:         0 kB
HugePages_Total:       0
HugePages_Free:        0
HugePages_Rsvd:        0
HugePages_Surp:        0
Hugepagesize:       2048 kB
DirectMap4k:        8064 kB
DirectMap2M:     2088960 kB
```

ファイルにはキロバイト（kB、1 kB は 1000 B）が表示されますが、実際には kibibytes になります（1 KiB は 1024 B になります）。/proc/meminfo におけるこの意図は知られていますが、従来の懸念のために修正されていません。プログラムは /proc/meminfo に依存して "kB" 文字列でサイズを指定します。

/proc/meminfo の情報の多くは、free コマンド、top コマンド、および ps コマンドで使用されます。実際、free コマンドの出力は、/proc/meminfo の内容と構造と似ています。ただし、/proc/meminfo 自体には詳細が記載されています。

- MemTotal - kibibytes 内の使用可能な RAM の合計量。これは、予約済みビットとカーネルバイナリーコードの数を引いた物理 RAM です。
- MemFree - kibibytes 内の物理 RAM の容量は、システムが未使用のままです。
- Buffers - raw ディスクブロックの一時ストレージの量(kibibytes)。
- Cached - キャッシュメモリーとして使用される物理メモリーの容量(kibibytes)。
- SwapCached - kibibytes では、swap に移動した後、メインメモリーに戻りますが、引き続き swap ファイル内に残ります。メモリーを再度スワップに移動する必要がないため、I/O が保存されます。
- Active - キビバイトで、より最近使用され、絶対に必要でない限り、通常は回収されません。
- Inactive - 最近使用されたメモリーの量（kibibytes では使用されず、他の目的で回収可能）。

- *Active(anon) - 匿名メモリーおよび tmpfs/shmem メモリーの量（kibibytes ではアクティブ、システムが最後に swap に移動された後にアクティブ使用であった）。*

- *Inactive(anon) - エビクションの候補である kibibytes の匿名および tmpfs/shmem メモリーの量。*

- *Active(file) - ファイルキャッシュメモリーの量（kibibytes 内）、またはシステムが最後に回収された後にアクティブ使用であったり、アクティブ使用であったりします。*

- *Inactive(file) - ディスクから新たに読み込まれるファイルキャッシュメモリー、kibibytes の量、または回収を行う候補です。*

- *Unevictable - ユーザープログラムによってメモリーにロックされているため、ページアウトコードによって検出されるメモリーの量をページアウトコードで検出します。*

- *Mlocked - ユーザープログラムによってメモリーにロックされているためにエビクトできないメモリーの合計量(kibibytes)。*

- *SwapTotal - kibibytes 内で利用可能な swap の合計量。*

- *SwapFree - kibibytes 内の空きスワップの合計量。*

- *Dirty - kibibytes 内のメモリーの合計量。ディスクへの書き込みを待機しています。*

- *Writeback - kibibytes 内のメモリーの合計量。これは、ディスクにアクティブに書き戻されます。*

- *AnonPages - ファイルでサポートされておらず、ユーザー空間ページのテーブルにマップされるページで使用されるメモリーの合計量(kibibytes)。*

- *Mapped - ライブラリーなど、mmaped ファイルに使用されるメモリー（kibibytes 単位）。*

- *Shmem - 共有メモリー(shmem)および tmpfs によって使用されるメモリーの合計量 (kibibytes)。*

- *Slab - カーネルが独自の使用のためにデータ構造をキャッシュするために使用されるメモリーの合計量（kibibytes 単位）。*

- *SReclaimable - キャッシュなど、回収できる Slab の一部です。*

- *SUnreclaim - メモリーがない場合でも、Slab の一部で回収できません。*

- *KernelStack - システムの各タスクに対して実行されるカーネルスタックの割り当てで使用されるメモリーの量(kibibytes)。*

- *PageTables - 最低ページテーブルレベル専用のメモリーの合計量(kibibytes)。*

- *NFS_Unstable - サーバーに送信される NFS ページの量(kibibytes)は、まだ安定したストレージにコミットされていない。*

- *Bounce - ブロックデバイス「bounce buffer」に使用されるメモリーの量(kibibytes)。*

- *WritebackTmp - FUSE が一時的なライトバックバッファーに使用するメモリーの量 (kibibytes)。*

- *CommitLimit - オーバーコミット比(vm.overcommit_ratio)に基づいてシステムで現在利用可能なメモリーの合計量。この制限は、厳密なオーバーコミットアカウンティングが有効にされている場合にのみ準拠されます（ vm.overcommit_memoryのモード 2）。CommitLimit 以下の式で計算されます。*

$$\frac{([\text{total RAM pages}] - [\text{total huge TLB pages}]) * \text{overcommit\_ratio}}{100} + [\text{total swap pages}]$$

たとえば、1 GB の物理メモリーと vm.overcommit_ratio が 30 の 7 GB の swap を搭載したシステムでは、7.3 GB が CommitLimit になります。

- *Committed_AS - ワークロードを完了すると予測されるメモリーの合計量(kibibytes)。この値は、最悪の場合のシナリオ値を表し、スワップメモリーも含まれます。*

- *VMallocTotal - 割り当てられた仮想アドレス空間の合計メモリー容量（kibibytes 単位）。*

- *VMallocUsed - 使用されている仮想アドレス空間のメモリー合計量(kibibytes)。*

- *VMallocChunk - 利用可能な仮想アドレス空間の最大連続したメモリーブロック(kibibytes)。*

- *HardwareCorrupted - キビバイトのメモリー量(kibibytes)では、ハードウェアによって識別され、カーネルによって外部に設定されたメモリーの量により使用されません。*

- *AnonHugePages - ファイルでサポートされておらず、ユーザー空間ページのテーブルにマップされるヒュージページによって使用されるメモリーの合計量(kibibytes)。*

- *HugePages_Total - システムの Huge Page の合計数。この数字は、/proc/sys/vm/hugetlb_pool で指定される hugepages とは別のメガバイトセットで Hugepagesize を除算することで派生します。この統計は、x86、gitops、および AMD64 のアーキテクチャーにのみ表示されます。*

- *HugePages_Free - システムで利用可能なヒュージページの合計数この統計は、x86、gitops、および AMD64 のアーキテクチャーにのみ表示されます。*

- *HugePages_Rsvd - hugetlbfs に予約されている未使用の Huge Page の数。*

- *HugePages_Surp : 余分のヒュージページ数。*

- *Hugepagesize - kibibytes の各 hugepages ユニットのサイズ。デフォルトでは、32 ビットアーキテクチャー用の uniprocessor カーネルの 4096 KB です。SMP、Hugemem カーネル、および AMD64 の場合、デフォルトは 2048 KB です。gitops アーキテクチャーの場合、デフォルトは 262144 KB です。この統計は、x86、gitops、および AMD64 のアーキテクチャーにのみ表示されます。*

- *DirectMap4k - 4 kB ページマッピングのあるカーネルアドレス空間にマップされるメモリーの量(kibibytes)。*

- *DirectMap2M - 2 MB ページマッピングを使用してカーネルアドレス空間にマップされるメモリーの量(kibibytes)。*

E.2.19. /proc/misc

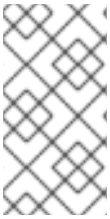
このファイルは、デバイス番号 10 など、その他の主要なデバイス（その他の主要なデバイス）に登録されているその他のドライバーを一覧表示します。

```
63 device-mapper 175 agpgart 135 rtc 134 apm_bios
```

最初の列は各デバイスのマイナー番号で、2 番目の列には、使用するドライバーが表示されます。

E.2.20. /proc/modules

このファイルは、カーネルに読み込まれているすべてのモジュールの一覧を表示します。そのコンテンツはシステムの設定や使用によって異なりますが、/proc/modules ファイル出力のサンプルと同様の方法で整理する必要があります。



/PROC/MODULES の内容

この例は、読み取り可能な形式で再フォーマットされています。この情報は、/sbin/lsmod コマンドでも表示できます。

```
nfs     170109  0 -         Live 0x129b0000
lockd    51593   1 nfs,      Live 0x128b0000
nls_utf8 1729    0 -         Live 0x12830000
vfat     12097   0 -         Live 0x12823000
fat      38881   1 vfat,     Live 0x1287b000
autofs4  20293   2 -         Live 0x1284f000
sunrpc   140453  3 nfs,lockd, Live 0x12954000
3c59x    33257   0 -         Live 0x12871000
```

```
uhci_hcd 28377   0 -         Live 0x12869000
md5      3777    1 -         Live 0x1282c000
ipv6     211845 16 -         Live 0x128de000
ext3     92585   2 -         Live 0x12886000
jbd      65625   1 ext3,     Live 0x12857000
dm_mod   46677   3 -         Live 0x12833000
```

最初の列には、モジュール名が含まれます。

2 列目は、モジュールのメモリーサイズ（バイト単位）を指します。

3 列目には、現在読み込まれているモジュールのインスタンス数が記載されています。ゼロの値は、アンロードされたモジュールを表します。

4 列目は、モジュールが別のモジュールに依存して機能させるかどうかを示し、他のモジュールを一覧表示します。

5 番目のコラムには、モジュールが置かれる負荷状態が表示されます。ライブ、読み込み、または リロード のみの 値のみになります。

6 番目のコラムには、読み込んだモジュールの現在のカーネルメモリーオフセットが一覧表示されます。この情報はデバッグの目的で、または oprofile などのプロファイリングツールに役立ちます。

E.2.21. /proc/mounts

このファイルは、システムで使用されるすべてのマウントの一覧を提供します。

```
rootfs / rootfs rw 0 0
/proc /proc proc rw,nodiratime 0 0 none
/dev ramfs rw 0 0
/dev/mapper/VolGroup00-LogVol00 / ext3 rw 0 0
none /dev ramfs rw 0 0
/proc /proc proc rw,nodiratime 0 0
/sys /sys sysfs rw 0 0
none /dev/pts devpts rw 0 0
usbdevfs /proc/bus/usb usbdevfs rw 0 0
/dev/hda1 /boot ext3 rw 0 0
none /dev/shm tmpfs rw 0 0
none /proc/sys/fs/binfmt_misc binfmt_misc rw 0 0
sunrpc /var/lib/nfs/rpc_pipefs rpc_pipefs rw 0 0
```

この出力は /etc/mtab の内容と似ていますが、/proc/mounts が最新のものである点が異なります。

最初の列はマウントするデバイスを指定し、2 列目はマウントポイントを示し、3 列目はファイルシステムのタイプを示し、4 番目の列は読み取り専用(ro)または読み書き(rw)をマウントするかどうかを示します。5 列目と 6 列目は、/etc/mtab で使用される形式に一致するように設計されたダミーの値です。

E.2.22. /proc/mtrr

このファイルは、システムで使用する現在の Memory Type Range Registers(MTRR)を参照します。システムアーキテクチャーが MTRR に対応している場合は、/proc/mtrr ファイルは以下のようになります。

```
reg00: base=0x00000000 (   0MB), size= 256MB: write-back, count=1
reg01: base=0xe8000000 (3712MB), size=  32MB: write-combining, count=1
```

MTRR は、プロセッサー（Pentium II 以降）の Intel P6 ファミリーと使用し、メモリー範囲へのプロセッサーアクセスを制御します。PCI バスまたは AGP バスでビデオカードを使用する場合は、適切に設定された /proc/mtrr ファイルは、150% を超えるパフォーマンスを向上させることができます。

多くの場合、この値はデフォルトで適切に設定されます。このファイルを手動で設定する方法は、以下の場所を参照してください。

```
/usr/share/doc/kernel-doc-<kernel_version>/Documentation/<arch>/mtrr.txt
```

E.2.23. /proc/partitions

このファイルには、パーティションブロック割り当て情報が含まれます。基本的なシステムからこのファイルのサンプリングは以下のようになります。

```
major minor  #blocks  name
  3     0   19531250 hda
  3     1     104391 hda1
  3     2   19422585 hda2
253     0   22708224 dm-0
253     1     524288 dm-1
```

ここでの情報のほとんどは、以下の列を除き、ユーザーにほとんど重要ではありません。

-

major - このパーティションを持つデバイスのメジャー番号。/proc/partitions,(3)のメジャー番号は、/proc/devices のブロックデバイス ide0 に対応します。

- *minor - このパーティションを持つデバイスのマイナー番号。これは、パーティションを異なる物理デバイスに分割し、パーティション名の最後にある番号に関連付けられます。*
- *#blocks - 特定のパーティションに含まれる物理ディスクブロックの数を一覧表示します。*
- *name - パーティションの名前。*

E.2.24. /proc/slabinfo

このファイルは、スラブ レベルでのメモリー使用量に関する完全な情報を提供します。バージョン 2.2 を超える Linux カーネルは、スラブプール を使用してページレベルでメモリーを管理します。一般的に使用されるオブジェクトには独自のスラブプールがあります。

詳細な /proc/slabinfo ファイルを手動で解析する代わりに、/usr/bin/slabtop プログラムは、カーネルスラブキャッシュ情報をリアルタイムで表示します。このプログラムでは、列のソートや画面の更新など、カスタム設定を行うことができます。

/usr/bin/slabtop のスクリーンショットの例は通常、以下の例のようになります。

```
Active / Total Objects (% used)    : 133629 / 147300 (90.7%)
Active / Total Slabs (% used)      : 11492 / 11493 (100.0%)
Active / Total Caches (% used)     : 77 / 121 (63.6%)
Active / Total Size (% used)       : 41739.83K / 44081.89K (94.7%)
Minimum / Average / Maximum Object : 0.01K / 0.30K / 128.00K
OBJS   ACTIVE USE      OBJ   SIZE     SLABS OBJ/SLAB CACHE SIZE NAME
44814  43159  96%    0.62K   7469     6     29876K ext3_inode_cache
36900  34614  93%    0.05K    492     75     1968K buffer_head
35213  33124  94%    0.16K   1531     23     6124K dentry_cache
7364   6463  87%    0.27K    526     14     2104K radix_tree_node
2585   1781  68%    0.08K     55     47      220K vm_area_struct
2263   2116  93%    0.12K     73     31      292K size-128
1904   1125  59%    0.03K     16     119      64K size-32
1666    768  46%    0.03K     14     119      56K anon_vma
1512   1482  98%    0.44K    168      9      672K inode_cache
1464   1040  71%    0.06K     24     61       96K size-64
1320    820  62%    0.19K     66     20      264K filp
678    587  86%    0.02K      3     226       12K dm_io
678    587  86%    0.02K      3     226       12K dm_tio
576    574  99%    0.47K     72      8      288K proc_inode_cache
528    514  97%    0.50K     66      8      264K size-512
```

```
492    372  75%    0.09K     12       41        48K bio
465    314  67%    0.25K     31       15       124K size-256
452    331  73%    0.02K      2      226         8K biovec-1
420    420 100%    0.19K     21       20        84K skbuff_head_cache
305    256  83%    0.06K      5       61        20K biovec-4
290      4   1%    0.01K      1      290         4K revoke_table
264    264 100%    4.00K    264        1      1056K size-4096
260    256  98%    0.19K     13       20        52K biovec-16
260    256  98%    0.75K     52        5       208K biovec-64
```

/usr/bin/slabtop に含まれる /proc/slabinfo で一般的に使用されるいくつかの統計には以下が含まれます。

- OBJS - 使用中のオブジェクト（割り当て済み）および使用されていないスペアを含むオブジェクト（メモリーブロック）の合計数。
- ACTIVE - 使用中のオブジェクト（メモリーブロック）の数（割り当て済み）
- USE - アクティブなオブジェクトの合計(ACTIVE/OBJS)(100)
- OBJ SIZE - オブジェクトのサイズ
- SLABS - スラブの合計数。
- OBJ/SLAB - スラブに適合するオブジェクトの数。
- CACHE SIZE - スラブのキャッシュサイズ。
- NAME - スラブの名前。

/usr/bin/slabtop プログラムの詳細は、slabtop man ページを参照してください。

E.2.25. /proc/stat

このファイルは、最後に再起動してからシステムに関するさまざまな統計を追跡します。/proc/stat

の内容は非常に長くなりますが、通常は以下の例のようになります。

```
cpu  259246 7001 60190 34250993 137517 772 0
cpu0 259246 7001 60190 34250993 137517 772 0
intr 354133732 347209999 2272 0 4 4 0 0 3 1 1249247 0 0 80143 0 422626 5169433
ctxt 12547729
btime 1093631447
processes 130523
procs_running 1
procs_blocked 0
preempt 5651840
cpu  209841 1554 21720 118519346 72939 154 27168
cpu0 42536 798 4841 14790880 14778 124 3117
cpu1 24184 569 3875 14794524 30209 29 3130
cpu2 28616 11 2182 14818198 4020 1 3493
cpu3 35350 6 2942 14811519 3045 0 3659
cpu4 18209 135 2263 14820076 12465 0 3373
cpu5 20795 35 1866 14825701 4508 0 3615
cpu6 21607 0 2201 14827053 2325 0 3334
cpu7 18544 0 1550 14831395 1589 0 3447
intr 15239682 14857833 6 0 6 6 0 5 0 1 0 0 0 29 0 2 0 0 0 0 0 0 0 94982 0 286812
ctxt 4209609
btime 1078711415
processes 21905
procs_running 1
procs_blocked 0
```

一般的に使用される統計には、以下のものがあります。

- *cpu - システムがユーザーモードにあり、優先度が低いユーザーモード(nice)、システムモード、アイドルタスク、I/O wait、IRQ(hardirq)、および softirq をそれぞれ持つ jiffies（x86 システムの 1 秒の1/100）を測定します。IRQ(hardirq)は、ハードウェアイベントへのダイレクト応答です。IRQ は、softirq が実行する場合の「heavy」作業のキューに対する最小限の作業を取ります。softirq は IRQ よりも優先順位が低いため、頻繁に中断される可能性があります。すべての CPU の合計は一番上にありますが、各 CPU には独自の統計がリストされます。以下の例は、マルチスレッドが有効になっている 4 方向の Intel Pentium Warehouse 設定であるため、合計 4 つの物理プロセッサーと合計 8 つの仮想プロセッサー 4 つの仮想プロセッサーを示しています。*

- *page - システムが書き込んだメモリーページ数。*

- *swap - システムがオンとアウトしたスワップページの数。*

- *intr - システムが発生した割り込みの数。*

- btime - ブート時間（1970 年 1 月 1 日以降の秒数）で測定され、epoch はエポックとして知られています。

E.2.26. /proc/swaps

このファイルは、スワップ領域とその使用状況を測定します。swap パーティションが 1 つしかない場合には、/proc/swaps の出力は以下のようになります。

```
Filename                        Type        Size    Used    Priority
/dev/mapper/VolGroup00-LogVol01 partition   524280  0       -1
```

この情報の一部は /proc/ ディレクトリーの他のファイルにありますが、/proc/swap はすべてのスワップファイル名のスナップショット、スワップ領域の種類、合計サイズ、使用領域のサイズ（キロバイト単位）を表します。優先度の列は、複数のスワップファイルが使用されている場合に役に立ちます。優先度が低いほど、スワップファイルが使用される可能性が高くなります。

E.2.27. /proc/sysrq-trigger

echo コマンドを使用してこのファイルに書き込むと、リモートの root ユーザーは、ローカルターミナルのようにほとんどの System Request Key コマンドをリモートで実行できます。このファイルに値を echo するには、/proc/sys/kernel/ gitops を 0 以外の値に設定する必要があります。System Request Key の詳細は、「/proc/sys/kernel/」 を参照してください。

このファイルへの書き込みは可能ですが、root ユーザーであっても読み取ることはできません。

E.2.28. /proc/uptime

このファイルは、最後に再起動してからシステムがオンになっている期間を詳細に説明します。/proc/uptime の出力は非常に最小限です。

```
350735.47 234388.90
```

最初の値は、システムが起動している合計秒数を表します。2 つ目の値は、各コアがアイドル状態で費やした時間（秒単位）の合計です。したがって、2 番目の値は、複数のコアを持つシステムの全体のアップタイムよりも大きくなる可能性があります。

E.2.29. /proc/version

このファイルは、Linux カーネルのバージョン、カーネルのコンパイルに使用される gcc のバージョン、およびカーネルコンパイルの時間を指定します。また、カーネルコンパイラーのユーザー名

（括弧内）も含まれます。

```
Linux version 2.6.8-1.523 (user@foo.redhat.com) (gcc version 3.4.1 20040714 \ (Red Hat
Enterprise Linux 3.4.1-7)) #1 Mon Aug 16 13:27:03 EDT 2004
```

この情報は、ユーザーのログイン時に表示されるバージョンデータなど、さまざまな目的で使用されます。

E.3. /PROC/ 内のディレクトリー

カーネルに関する一般的な情報は、/proc/ ディレクトリー内のディレクトリーおよびサブディレクトリーにグループ化されます。

E.3.1. プロセスディレクトリー

すべての /proc/ ディレクトリーには、数値名のディレクトリーが多数含まれます。これらの一覧は以下のようになります。

```
dr-xr-xr-x    3 root     root           0 Feb 13 01:28 1
dr-xr-xr-x    3 root     root           0 Feb 13 01:28 1010
dr-xr-xr-x    3 xfs      xfs            0 Feb 13 01:28 1087
dr-xr-xr-x    3 daemon   daemon         0 Feb 13 01:28 1123
dr-xr-xr-x    3 root     root           0 Feb 13 01:28 11307
dr-xr-xr-x    3 apache   apache         0 Feb 13 01:28 13660
dr-xr-xr-x    3 rpc      rpc            0 Feb 13 01:28 637
dr-xr-xr-x    3 rpcuser  rpcuser        0 Feb 13 01:28 666
```

これらのディレクトリーは、プログラムの プロセス ID に基づいて名前が付けられ、そのプロセスに固有の情報が含まれているため、プロセスディレクトリーと呼ばれます。各プロセスディレクトリーの所有者とグループは、プロセスを実行するユーザーに設定されます。プロセスが終了すると、/proc/ プロセスディレクトリー vanishes になります。

各プロセスディレクトリーには、以下のファイルが含まれます。

- cmdline - プロセスの起動時に実行したコマンドが含まれます。
- cwd - プロセスの現在の作業ディレクトリーへのシンボリックリンク。

- *environ - プロセスの環境変数一覧。環境変数は大文字で指定され、値は小文字です。*

- *exe - このプロセスの実行可能ファイルへのシンボリックリンク。*

- *fd - 特定のプロセスのすべてのファイル記述子を含むディレクトリー。これらは番号付きのリンクで指定されます。*

```
total 0
lrwx------ 1 root root 64 May 8 11:31 0 -> /dev/null
lrwx------ 1 root root 64 May 8 11:31 1 -> /dev/null
lrwx------ 1 root root 64 May 8 11:31 2 -> /dev/null
lrwx------ 1 root root 64 May 8 11:31 3 -> /dev/ptmx
lrwx------ 1 root root 64 May 8 11:31 4 -> socket:[7774817]
lrwx------ 1 root root 64 May 8 11:31 5 -> /dev/ptmx
lrwx------ 1 root root 64 May 8 11:31 6 -> socket:[7774829]
lrwx------ 1 root root 64 May 8 11:31 7 -> /dev/ptmx
```

- *maps - このプロセスに関連するさまざまな実行ファイルやライブラリーファイルに対するメモリーマップの一覧。このファイルはプロセスの複雑さに応じて長くなりますが、sshd プロセスからの出力サンプルは以下のようになります。*

```
08048000-08086000 r-xp 00000000 03:03 391479 /usr/sbin/sshd
08086000-08088000 rw-p 0003e000 03:03 391479 /usr/sbin/sshd
08088000-08095000 rwxp 00000000 00:00 0
40000000-40013000 r-xp 0000000 03:03 293205 /lib/ld-2.2.5.so
40013000-40014000 rw-p 00013000 03:03 293205 /lib/ld-2.2.5.so
40031000-40038000 r-xp 00000000 03:03 293282 /lib/libpam.so.0.75
40038000-40039000 rw-p 00006000 03:03 293282 /lib/libpam.so.0.75
40039000-4003a000 rw-p 00000000 00:00 0
4003a000-4003c000 r-xp 00000000 03:03 293218 /lib/libdl-2.2.5.so
4003c000-4003d000 rw-p 00001000 03:03 293218 /lib/libdl-2.2.5.so
```

- *mem - プロセスが保持するメモリー。このファイルはユーザーが読み取ることはできません。*

- *root - プロセスのルートディレクトリーへのリンク*

- *stat - プロセスのステータス*

- *statm - プロセスが使用しているメモリーのステータス以下は、/proc/statm ファイルのサ*

ンプルです。

```
263 210 210 5 0 205 0
```

7 列は、プロセスのさまざまなメモリー統計に関連します。左から右に、使用されるメモリーの以下の側面を報告します。

1. *プログラム合計サイズ（キロバイト単位）。*
2. *メモリー部分のサイズ（キロバイト単位）。*
3. *共有されるページ数。*
4. *コードであるページ数。*
5. *データ/スタックのページ数。*
6. *ライブラリーページ数。*
7. *ダーティーページ数。*

- *status - stat または statm よりも読み取り可能な形式でプロセスのステータスsshd の出力例を以下に示します。*

```
Name: sshd
State: S (sleeping)
Tgid: 797
Pid: 797
PPid: 1
TracerPid: 0
Uid: 0 0 0 0
Gid: 0 0 0 0
FDSize: 32
Groups:
VmSize:     3072 kB
VmLck:         0 kB
VmRSS:       840 kB
```

```
VmData:     104 kB
VmStk:       12 kB
VmExe:      300 kB
VmLib:     2528 kB
SigPnd: 0000000000000000
SigBlk: 0000000000000000
SigIgn: 8000000000001000
SigCgt: 0000000000014005
CapInh: 0000000000000000
CapPrm: 00000000fffffeff
CapEff: 00000000fffffeff
```

この出力の情報には、プロセス名と ID、状態（S (sleeping)、R (running)など）、プロセスを実行するユーザー/グループ ID、メモリー使用量に関する詳細情報が含まれます。

E.3.1.1. /proc/self/

/proc/self/ ディレクトリーは、現在実行中のプロセスへのリンクです。これにより、プロセス ID を把握せずにプロセスが自身を確認できます。

シェル環境内では、/proc/self/ ディレクトリーの一覧が、そのプロセスのプロセスディレクトリーの一覧表示と同じ内容を生成します。

E.3.2. /proc/bus/

このディレクトリーには、システムで利用可能なさまざまなバスに固有の情報が含まれています。たとえば、PCI および USB バスを含む標準システムでは、各バスの現在のデータは /proc/bus/ 内のサブディレクトリー（ /proc/bus/ pci/ など）内のサブディレクトリー内で利用できます。

/proc/bus/ 内のサブディレクトリーとファイルは、システムに接続されているデバイスによって異なります。ただし、各バスタイプには少なくとも 1 つのディレクトリーがあります。これらのバスディレクトリー内では、通常、バイナリーファイルを含む 001 などの数字名のサブディレクトリーがあります。

たとえば、/proc/bus/usb/ サブディレクトリーには、USB バス上の各種デバイスと、そのデバイスに必要なドライバーを追跡するファイルが含まれます。以下は、/proc/bus/usb/ ディレクトリーの一覧の例です。

```
total 0 dr-xr-xr-x    1 root     root           0 May  3 16:25 001
-r--r--r--    1 root     root           0 May  3 16:25 devices
-r--r--r--    1 root     root           0 May  3 16:25 drivers
```

/proc/bus/usb/001/ ディレクトリーには、最初の USB バス上のすべてのデバイスが含まれ、デバイス ファイルはマザーボードの USB ルートハブを識別します。

以下は、/proc/bus/usb/devices ファイルの例です。

```
T:  Bus=01 Lev=00 Prnt=00 Port=00 Cnt=00 Dev#=  1 Spd=12  MxCh= 2
B:  Alloc=  0/900 us ( 0%), #Int=  0, #Iso=  0
D:  Ver= 1.00 Cls=09(hub  ) Sub=00 Prot=00 MxPS= 8 #Cfgs=  1
P:  Vendor=0000 ProdID=0000 Rev= 0.00
S:  Product=USB UHCI Root Hub
S:  SerialNumber=d400
C:* #Ifs= 1 Cfg#= 1 Atr=40 MxPwr=  0mA
I:  If#= 0 Alt= 0 #EPs= 1 Cls=09(hub  ) Sub=00 Prot=00 Driver=hub
E:  Ad=81(I) Atr=03(Int.) MxPS=   8 Ivl=255ms
```

E.3.3. /proc/bus/pci

2.6 Linux カーネルの後のバージョンでは、/proc/bus/pci ディレクトリーが優先されるため、/proc/pci ディレクトリーが廃止されました。cat /proc/bus/pci/devices コマンドを使用して、システムに存在する PCI デバイスの一覧を取得することはできますが、出力の読み取りと解釈は困難です。

PCI デバイスの人間が判読できる一覧を表示するには、以下のコマンドを実行します。

```
~]# /sbin/lspci -vb
00:00.0 Host bridge: Intel Corporation 82X38/X48 Express DRAM Controller
        Subsystem: Hewlett-Packard Company Device 1308
        Flags: bus master, fast devsel, latency 0
        Capabilities: [e0] Vendor Specific Information <?>
        Kernel driver in use: x38_edac
        Kernel modules: x38_edac

00:01.0 PCI bridge: Intel Corporation 82X38/X48 Express Host-Primary PCI Express Bridge
(prog-if 00 [Normal decode])
        Flags: bus master, fast devsel, latency 0
        Bus: primary=00, secondary=01, subordinate=01, sec-latency=0
        I/O behind bridge: 00001000-00001fff
        Memory behind bridge: f0000000-f2ffffff
        Capabilities: [88] Subsystem: Hewlett-Packard Company Device 1308
        Capabilities: [80] Power Management version 3
        Capabilities: [90] MSI: Enable+ Count=1/1 Maskable- 64bit-
        Capabilities: [a0] Express Root Port (Slot+), MSI 00
        Capabilities: [100] Virtual Channel <?>
        Capabilities: [140] Root Complex Link <?>
        Kernel driver in use: pcieport
        Kernel modules: shpchp

00:1a.0 USB Controller: Intel Corporation 82801I (ICH9 Family) USB UHCI Controller #4 (rev
```

```
02) (prog-if 00 [UHCI])
        Subsystem: Hewlett-Packard Company Device 1308
        Flags: bus master, medium devsel, latency 0, IRQ 5
        I/O ports at 2100
        Capabilities: [50] PCI Advanced Features
        Kernel driver in use: uhci_hcd
[output truncated]
```

この出力は、カーネルで表示されるのではなく、PCI バス上のカードに表示されるすべての IRQ 番号およびアドレスのソートリストです。デバイスの名前とバージョンを提供する以外に、この一覧には詳細な IRQ 情報も示され、管理者が競合を素早く検索できるようになります。

E.3.4. /proc/driver/

このディレクトリーには、カーネルにより使用されている特定のドライバーの情報が含まれます。

ここで見つかった共通ファイルは、システムの リアルタイムクロック(RTC) のドライバーからの出力を提供する rtc です。/proc/driver/rtc からの出力例を以下に示します。

```
rtc_time        : 16:21:00
rtc_date        : 2004-08-31
rtc_epoch       : 1900
alarm           : 21:16:27
DST_enable      : no
BCD             : yes
24hr            : yes
square_wave     : no
alarm_IRQ       : no
update_IRQ      : no
periodic_IRQ    : no
periodic_freq   : 1024
batt_status     : okay
```

RTC の詳細は、以下のインストール済みドキュメントを参照してください。

/usr/share/doc/kernel-doc-<kernel_version>/Documentation/rtc.txt.

E.3.5. /proc/fs

このディレクトリーには、エクスポートするファイルシステムが表示されます。NFS サーバーを実行している場合は、cat /proc/fs/nfsd/exports を実行すると、共有されているファイルシステムと、そのファイルシステムに付与されているパーミッションが表示されます。NFS でのファイルシステム共有

の詳細は、『『ストレージ管理ガイド』』 の 『「ネットワークファイルシステム(NFS)』 」の章を参照してください。

E.3.6. /proc/irq/

このディレクトリーは、IRQ を CPU アフィニティーに設定するために使用されます。これにより、システムは特定の IRQ を 1 つの CPU のみに接続できます。または、CPU が IRQ を処理しないように除外できます。

それぞれの IRQ には独自のディレクトリーがあり、各 IRQ の個々の設定を許可します。/proc/irq/prof_cpu_mask ファイルは、IRQ ディレクトリーの smp_affinity ファイルのデフォルト値が含まれるビットマスクです。smp_affinity の値は、特定の IRQ を処理する CPU を指定します。

/proc/irq/ ディレクトリーの詳細は、以下のインストール済みドキュメントを参照してください。

> */usr/share/doc/kernel-doc-*kernel_version*/Documentation/filesystems/proc.txt*

E.3.7. /proc/net/

このディレクトリーは、さまざまなネットワークパラメーターと統計を全体的に見ていきます。このディレクトリー内の各ディレクトリーおよび仮想ファイルは、システムのネットワーク設定の特徴を説明します。以下は、/proc/net/ ディレクトリーの一部の一覧です。

- arp: カーネルの ARP テーブルを一覧表示します。このファイルは、ハードウェアアドレスをシステム上の IP アドレスに接続するのに特に便利です。
- ATM / ディレクトリー： このディレクトリー内のファイルには、ATM(Asynchronous Transfer Mode) 設定および統計が含まれます。このディレクトリーは、主に ATM ネットワークと ADSL カードで使用されます。
- dev - システムに設定したさまざまなネットワークデバイスの一覧を表示し、送受信の統計で完了します。このファイルは、各インターフェースが送受信したバイト数、パケットのインバウンドおよびアウトバウンドの数、発生したエラー数、破棄されたパケット数などを表示します。
- dev_mcast - 各デバイスがリッスンしているレイヤー2マルチキャストグループを一覧表示します。

- *IGMP: このシステムが参加する IP マルチキャストアドレスを一覧表示します。*

- *ip_conntrack: IP 接続の転送を行うマシンの追跡済みネットワーク接続を一覧表示します。*

- *ip_tables_names: 使用中の iptables のタイプを一覧表示します。このファイルは、iptables がシステムでアクティブな場合にのみ存在し、filter、mangle、または nat のいずれかの値が含まれます。*

- *ip_mr_cache: マルチキャストルーティングキャッシュを一覧表示します。*

- *ip_mr_vif - マルチキャスト仮想インターフェースを一覧表示します。*

- *netstat: TCP タイムアウト、SYN クッキーの送受信など、ネットワーク統計の幅広い詳細なコレクションが含まれます。*

- *psched - グローバルパケットスケジューラーパラメーターを一覧表示します。*

- *raw: raw デバイスの統計を一覧表示します。*

- *route: カーネルのルーティングテーブルを一覧表示します。*

- *rt_cache: 現在のルーティングキャッシュが含まれます。*

- *SNMP - 使用中のさまざまなネットワークプロトコル用の Simple Network Management Protocol(SNMP)データの一覧。*

- *sockstat: ソケット統計を提供します。*

- *tcp: 詳細な TCP ソケット情報が含まれます。*

- *tr_rif: トークンリング RIF ルーティングテーブルを一覧表示します。*
- *udp: 詳細な UDP ソケット情報が含まれます。*
- *UNIX - 現在使用中の UNIX ドメインソケットを一覧表示します。*
- *ワイヤレス - ワイヤレス インターフェースデータを一覧表示します。*

E.3.8. /proc/scsi/

このディレクトリーのプライマリーファイルは /proc/scsi/scsi で、認識されているすべての SCSI デバイスの一覧が含まれています。この一覧から、デバイスの種類、およびモデル名、ベンダー、SCSI チャネル、および ID データを利用できます。

たとえば、システムに SCSI CD-ROM、テープドライブ、ハードドライブ、および RAID コントローラーが含まれている場合、このファイルは以下のようになります。

```
Attached devices:
Host: scsi1
Channel: 00
Id: 05
Lun: 00
Vendor: NEC
Model: CD-ROM DRIVE:466
Rev: 1.06
Type:   CD-ROM
ANSI SCSI revision: 02
Host: scsi1
Channel: 00
Id: 06
Lun: 00
Vendor: ARCHIVE
Model: Python 04106-XXX
Rev: 7350
Type:   Sequential-Access
ANSI SCSI revision: 02
Host: scsi2
Channel: 00
Id: 06
Lun: 00
Vendor: DELL
Model: 1x6 U2W SCSI BP
Rev: 5.35
Type:   Processor
```

```
ANSI SCSI revision: 02
Host: scsi2
Channel: 02
Id: 00
Lun: 00
Vendor: MegaRAID
Model: LD0 RAID5 34556R
Rev: 1.01
Type:   Direct-Access
ANSI SCSI revision: 02
```

システムが使用する各 SCSI ドライバーには、/proc/scsi/ 内に独自のディレクトリーがあり、このドライバーを使用する各 SCSI コントローラーに固有のファイルが含まれます。上記の例では、2 つのドライバーが使用されているため、aic7xxx/ ディレクトリーおよび megaraid/ ディレクトリーが存在します。各ディレクトリーのファイルには、通常、そのドライバーを使用する SCSI コントローラーの I/O アドレス範囲、IRQ 情報、および統計が含まれます。各コントローラーは、異なるタイプおよび情報量を報告できます。このシステムの例に含まれる Adaptec AIC-7880 Ultra SCSI ホストアダプターは、以下の出力を生成します。

```
Adaptec AIC7xxx driver version: 5.1.20/3.2.4
Compile Options:
TCQ Enabled By Default : Disabled
AIC7XXX_PROC_STATS     : Enabled
AIC7XXX_RESET_DELAY    : 5
Adapter Configuration:
SCSI Adapter: Adaptec AIC-7880 Ultra SCSI host adapter
Ultra Narrow Controller     PCI MMAPed
I/O Base: 0xfcffe000
Adapter SEEPROM Config: SEEPROM found and used.
Adaptec SCSI BIOS: Enabled
IRQ: 30
SCBs: Active 0, Max Active 1, Allocated 15, HW 16, Page 255
Interrupts: 33726
BIOS Control Word: 0x18a6
Adapter Control Word: 0x1c5f
Extended Translation: Enabled
Disconnect Enable Flags: 0x00ff
Ultra Enable Flags: 0x0020
Tag Queue Enable Flags: 0x0000
Ordered Queue Tag Flags: 0x0000
Default Tag Queue Depth: 8
Tagged Queue By Device array for aic7xxx
host instance 1:       {255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255}
Actual queue depth per device for aic7xxx host instance 1:       {1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1}
Statistics:

(scsi1:0:5:0) Device using Narrow/Sync transfers at 20.0 MByte/sec, offset 15
Transinfo settings: current(12/15/0/0), goal(12/15/0/0), user(12/15/0/0)
Total transfers 0 (0 reads and 0 writes)
   < 2K      2K+     4K+     8K+    16K+    32K+    64K+   128K+
Reads:        0       0       0       0       0       0       0       0
Writes:       0       0       0       0       0       0       0       0

(scsi1:0:6:0) Device using Narrow/Sync transfers at 10.0 MByte/sec, offset 15
```

```
Transinfo settings: current(25/15/0/0), goal(12/15/0/0), user(12/15/0/0)
Total transfers 132 (0 reads and 132 writes)
 < 2K     2K+     4K+     8K+    16K+    32K+    64K+   128K+
Reads:       0       0       0       0       0       0       0       0
Writes:      0       0       0       1     131       0       0       0
```

この出力は、チャンネル ID に基づいてコントローラーに接続された SCSI デバイスへの転送速度と、そのデバイスの読み取りまたは書き込みを行うファイルの量やサイズに関する詳細な統計を示しています。たとえば、このコントローラーは 1 秒あたり 20 メガバイトの CD-ROM と通信しますが、テープドライブは 1 秒あたり 10 メガバイトのみと通信します。

E.3.9. /proc/sys/

/proc/ sys/ ディレクトリーは、システムに関する情報だけでなく、システム管理者がカーネル機能をすぐに有効または無効にすることができるため、/proc/ の他のディレクトリーとは異なります。

/PROC/SYS/ の内容を変更する場合は注意してください。

/proc/sys/ ディレクトリー内のさまざまなファイルを使用して、実稼働システムの設定を変更する際には注意してください。誤った設定を変更すると、カーネルが不安定になり、システムの再起動が必要になります。

このため、/proc/sys/ の値を変更する前に、このファイルでオプションが有効であることを確認してください。

特定のファイルを設定することができるかどうか、または情報提供のみ用に設計されている場合は、シェルプロンプトで -l オプションを使用して一覧表示できます。ファイルが書き込み可能である場合は、カーネルの設定に使用できます。たとえば、/proc/sys/fs の一部の一覧は以下のようになります。

```
-r--r--r--  1 root  root  0 May 10 16:14 dentry-state
-rw-r--r--  1 root  root  0 May 10 16:14 dir-notify-enable
-rw-r--r--  1 root  root  0 May 10 16:14 file-max
-r--r--r--  1 root  root  0 May 10 16:14 file-nr
```

このリストでは、dir-notify-enable ファイルおよび file-max ファイルを書き込みできるため、カーネルの設定に使用できます。その他のファイルは、現在の設定に関するフィードバックのみを提供します。

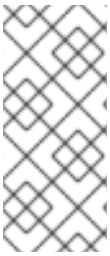
/proc/sys/ ファイル内の値を変更するには、ファイルに新しい値をエコーして実行します。たとえば、実行中のカーネルでシステム要求キーを有効にするには、以下のコマンドを入力します。

```
echo 1 > /proc/sys/kernel/sysrq
```

これにより、pidgin の値 は 0 (off)から 1 (on)に変更します。

いくつかの /proc/sys/ 設定ファイルには、複数の値が含まれます。新しい値を正しく送信するには、以下の例のように、echo コマンドで渡された各値の間にスペース文字を置きます。

```
echo 4 2 45 > /proc/sys/kernel/acct
```



ECHO コマンドを使用して行った変更は永続的ではありません

システムの再起動時に echo コマンドを使用して行われた設定変更は消えます。システムの再起動後に設定の変更を有効にするには、「sysctl コマンドの使用」 を参照してください。

/proc/sys/ ディレクトリーには、実行中のカーネルのさまざまな側面を制御するサブディレクトリーが複数含まれています。

E.3.9.1. /proc/sys/dev/

このディレクトリーは、システム上の特定のデバイスのパラメーターを提供します。ほとんどのシステムには、cdrom/ ディレクトリーと raid/ という 2 つ以上のディレクトリーがあります。カスタマイズしたカーネルには、parport/ などの他のディレクトリーを持たせることができます。これにより、複数のデバイスドライバー間で 1 つの並列ポートを共有できます。

cdrom/ ディレクトリーには info と呼ばれるファイルが含まれており、重要な CD-ROM パラメーターが数多く表示されます。

```
CD-ROM information, Id: cdrom.c 3.20 2003/12/17
drive name:             hdc
drive speed:            48
drive # of slots:       1
Can close tray:         1
Can open tray:          1
Can lock tray:          1
Can change speed:       1
Can select disk:        0
```

```
Can read multisession:  1
Can read MCN:           1
Reports media changed:  1
Can play audio:         1
Can write CD-R:         0
Can write CD-RW:        0
Can read DVD:           0
Can write DVD-R:        0
Can write DVD-RAM:      0
Can read MRW:           0
Can write MRW:          0
Can write RAM:          0
```

このファイルは迅速にスキャンして、不明な CD-ROM の特性を検出することができます。複数の CD-ROM がシステムで利用可能な場合、各デバイスには情報の独自の列が付与されます。

autoclose や checkmedia など、/proc/sys/dev/cdrom のさまざまなファイルを使用して、システムの CD-ROM を制御できます。これらの機能を有効または無効にするには、echo コマンドを使用します。

RAID サポートがカーネルにコンパイルされている場合は、/proc/sys/dev/raid/ ディレクトリーは、少なくとも 2 つのファイルで speed_limit_min および speed_limit_max で利用できます。この設定により、ディスクの再同期など、I/O 集約型タスク用の RAID デバイスのアクセラレーションが決定されます。

E.3.9.2. /proc/sys/fs/

このディレクトリーには、クォータ、ファイルハンドル、inode、dentry 情報など、ファイルシステムのさまざまな側面に関するオプションおよび情報が含まれています。

binfmt_misc/ ディレクトリーは、その他のバイナリー形式のカーネルサポートを提供するために使用されます。

/proc/sys/fs/ の重要なファイルには以下が含まれます。

- dentry-state: ディレクトリーキャッシュのステータスを表示します。ファイルは以下の例のようになります。

  ```
  57411 52939 45 0 0 0
  ```

  最初の番号はディレクトリーキャッシュエントリーの合計数を示し、2 番目の番号には未

使用のエントリーの数が表示されます。3 番目の数字は、ディレクトリーが解放され、回収されるタイミングの間の秒数を示し、4 つ目はシステムで現在要求されているページを測定します。最後の 2 つの数字は使用されず、ゼロのみが表示されます。

- file-max: カーネルが割り当てるファイルハンドルの最大数を表示します。このファイルに値を増やすと、利用可能なファイルハンドルがないためにエラーを解決できます。

- file-nr: 割り当てられたファイルハンドルの数、使用されるファイルハンドル数、およびファイルハンドルの最大数を表示します。

- overflowgid および overflowuid: 16 ビットのグループ ID とユーザー ID のみをサポートするファイルシステムで使用する固定グループ ID とユーザー ID を定義します。

E.3.9.3. /proc/sys/kernel/

このディレクトリーには、カーネルの操作に直接影響を与えるさまざまな設定ファイルが含まれています。最も重要なファイルには以下が含まれます。

- Acc t - ログを含むファイルシステムで利用可能な空き領域の割合に基づいて、プロセスアカウンティングの一時停止を制御します。デフォルトでは、ファイルは以下のようになります。

  ```
  4 2 30
  ```

  最初の値はロギングを再開するのに必要な空き領域の割合を定め、2 番目の値はロギングが一時停止される際に空き領域のしきい値を設定します。3 番目の値で、カーネルがファイルシステムをポーリングして、ロギングを一時停止または再開するかどうかを確認する間隔を秒単位で設定します。

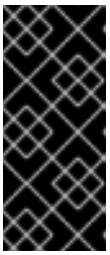
- Ctrl-alt-del - Ctrl+Alt+Delete が init を使用してコンピューターを正常に再起動するか (0)、ダーティーバッファーをディスクに同期せずに即時に再起動するかを制御します(1)。

- domainName: example.com などのシステムドメイン名を設定します。

- exec-shield: カーネルの Exec Shield 機能を設定します。exec Shield は、特定タイプのバッファーオーバーフロー攻撃に対する保護を提供します。

この仮想ファイルには、以下の 2 つの値を使用できます。

- *0 - Exec Shield を無効にします。*
- *1 - Exec Shield を有効にします。これはデフォルト値になります。*



EXEC SHIELD の使用

Exec Shield が無効になっている間に起動したセキュリティーの影響を受けるアプリケーションを実行している場合は、Exec Shield を有効にするには、これらのアプリケーションを再起動する必要があります。

- *hostname - www.example.com などのシステムのホスト名を設定します。*
- *hotplug: 設定の変更がシステムによって検出される際に使用されるユーティリティーを設定します。これは主に USB および Cardbus PCI で使用されます。このロールを満たす新しいプログラムをテストしない限り、/sbin/hotplug のデフォルト値は変更しないでください。*
- *modprobe - カーネルモジュールの読み込みに使用するプログラムの場所を設定します。デフォルト値は /sbin/modprobe です。これは、カーネルスレッドが kmod を呼び出す際にモジュールをロードするために kmod が呼び出しを行うことを意味します。*
- *msgmax: あるプロセスから別のプロセスに送信されるメッセージの最大サイズを設定し、デフォルトで 8192 バイトに設定されます。プロセス間でキューに置かれたメッセージがスワップ不可能なカーネルメモリーに保存されるため、この値を引き上げる際には注意してください。msgmax を増やすと、システムの RAM 要件が増えます。*
- *msgmnb - 単一のメッセージキューに最大バイト数を設定します。デフォルトは 16384 です。*
- *msgmni - メッセージキュー識別子の最大数を設定します。デフォルトは 4008 です。*
- *osrelease - Linux カーネルリリース番号を一覧表示します。このファイルは、カーネルソースを変更して再コンパイルするとのみ変更できます。*

- *OSType: オペレーティングシステムのタイプを表示します。デフォルトでは、このファイルは Linux に設定されています。この値は、カーネルソースを変更して再コンパイルするだけでのみ変更できます。*

- *overflowgid および overflowuid: 16 ビットグループとユーザー ID のみをサポートするアーキテクチャー上のシステムコールで使用する固定グループ ID とユーザー ID を定義します。*

- *panic: システムでカーネルパニックが発生したときにカーネルの再起動を延期する秒数を定義します。デフォルトでは、値は 0 に設定されています。これは、パニック後の自動再起動を無効にします。*

- *printk: このファイルは、エラーメッセージの印刷やロギングに関連するさまざまな設定を制御します。カーネルが報告する各エラーメッセージには、メッセージの重要性を定義するログレベルが関連付けられています。loglevel 値は、以下の順序で分類されます。*

  - *0 - カーネルの緊急事態。システムが利用できません。*
  - *1 - カーネルアラート。すぐに対処する必要があります。*
  - *2 - 重大な問題があると見なされるカーネルの状態。*
  - *3 - 一般的なカーネルエラー状態。*
  - *4 - 一般的なカーネルの警告状態。*
  - *5 - 正常だが重大な状態のカーネル通知。*
  - *6 - カーネル情報メッセージ。*
  - *7 - カーネルのデバッグレベルメッセージ。*

printk ファイルには、以下の 4 つの値があります。

```
6   4   1   7
```

これらの各値は、エラーメッセージを処理するさまざまなルールを定義します。コンソールログレベル と呼ばれる最初の値は 、コンソール に出力されるメッセージの最も低い優先度を定義します。（優先度が小さいほど、ログレベル番号が高いことに注意してください。） 2 番目の値は、明示的なログレベルが適用されないメッセージのデフォルトのログレベルを設定します。3 つ目の値は、コンソールログレベルの可能な限り低いログレベル設定を設定します。最後の値は、コンソールログレベルのデフォルト値を設定します。

- random/ ディレクトリー： カーネルの乱数の生成に関連する多くの値を一覧表示します。
- sem - カーネル内での semaphore 設定を構成します。semaphore は、特定のプロセスの使用を制御するために使用される System V IPC オブジェクトです。
- shmall: システムの一度に使用可能な共有メモリーの合計量をバイト単位で設定します。デフォルトでは、この値は 2097152 です。
- shmmax: カーネルで許可される最大共有メモリーセグメントサイズを設定します。デフォルトでは、この値は 33554432 です。ただし、カーネルはこれよりも大きな値をサポートします。
- SHMMN I: システム全体で共有メモリーセグメントの最大数を設定します。デフォルトでは、この値は 4096 です。
- Sys Rq: この値がゼロ(0)以外の値に設定されている場合、システム要求キーを有効にします。

  System Request Key は、単純なキーの組み合わせでカーネルへの即時入力を可能にします。たとえば、System Request Key を使用すると、システムをすぐにシャットダウンまたは再起動したり、マウントされたすべてのファイルシステムを同期したり、重要な情報をコンソールにダンプしたりできます。System Request Key を開始するには、Alt+SysRq+システム要求コード を入力します。システム要求コード を、以下のシステム要求コードのいずれかに置き換えます。

  - r - キーボードの raw モードを無効にし、XLATE に設定します（すべてのキーに対し

て Alt、Ctrl、Shift などの修飾子を認識しない制限されたキーボードモード）。

- k - 仮想コンソールでアクティブなプロセスをすべて強制終了します。Secure Access Key (SAK)とも呼ばれます。通常、ログインプロンプトが init から起動され、ユーザー名とパスワードをキャプチャーするように設計された trojan コピーではなく、init から起動されたことを確認するために使用されます。

- b - ファイルシステムのマウントを解除したり、システムに接続されているディスクを同期したりせずにカーネルを再起動します。

- c - ファイルシステムのマウントを解除したり、システムに接続されているディスクを同期したりせずに、システムがクラッシュします。

- O - システムをオフにします。

- s: システムに接続されているディスクの同期を試行します。

- u - すべてのファイルシステムのマウントを解除して再マウントしようとすると、すべてのファイルシステムを読み取り専用として再マウントします。

- p - すべてのフラグを出力し、コンソールに登録します。

- T - プロセス一覧をコンソールに出力します。

- m: メモリー統計をコンソールに出力します。

- 0 から 9 - コンソールのログレベルを設定します。

- e: SIGTERM を使用して init 以外のすべてのプロセスを強制終了します。

- i: SIGKILL を使用して init 以外のすべてのプロセスを強制終了します。

- *i: SIGKILL（initを含む）を使用してすべてのプロセスを強制終了します。この System Request Key コードを発行した後には、システムが使用できなくなります。*

- *h: ヘルプテキストを表示します。*

この機能は、開発カーネルを使用する場合や、システムのフリーズが発生した場合に最も有益です。

SYSTEM REQUEST KEY 機能を有効にする場合は注意してください。

System Request Key 機能は、無人コンソールがシステムへのアクセスを持つ攻撃者を提供するため、セキュリティー上のリスクと見なされます。このため、デフォルトでは無効になっています。

システム要求キーの詳細は、/usr/share/doc/kernel-doc-kernel_version/Documentation/gitops.txt を参照してください。

- *tainted: 非GPL モジュールが読み込まれているかどうかを示します。*

  - *0 - GPL 以外のモジュールはロードされません。*

  - *1 - GPL ライセンスのない少なくとも 1 つのモジュール（ライセンスのないモジュールを含む）が読み込まれます。*

  - *2 - 1 つ以上のモジュールは、 smod -f コマンドで 強制的にロードされました。*

- *threads-max: カーネルによって使用される最大スレッド数を設定します。デフォルト値は 2048 です。*

- *version: カーネルの最終コンパイル日時を表示します。このファイルの最初のフィールド（例： #3 ）は、カーネルがソースベースから構築される回数に関連します。*

E.3.9.4. /proc/sys/net/

このディレクトリーには、さまざまなネットワークトピックに関するサブディレクトリーが含まれています。カーネルのコンパイル時にさまざまな設定により、ethernet/、ipv4/、ipx/、 ipv6/ など、ここに異なるディレクトリーが利用可能になります。これらのディレクトリー内のファイルを変更すると、システム管理者は実行中のシステムのネットワーク設定を変更できます。

Linux で利用可能な様々なネットワークオプションがある場合、最も一般的な /proc/sys/net/ ディレクトリーのみを説明します。

/proc/sys/net/core/ ディレクトリーには、カーネルとネットワーク層との間の相互作用を制御するさまざまな設定が含まれています。これらのファイルの最も重要なのは、以下のとおりです。

- message_burst: message_cost で定義される期間のカーネルログに書き込まれる新しい警告メッセージの最大数を設定します。このファイルのデフォルト値は 10 です。

  message_cost と組み合わせて、この設定は、ネットワークコードからカーネルログに書き込まれた警告メッセージのレート制限を強制し、サービス 拒否(DoS)攻撃を軽減するために使用されます。DoS 攻撃の考え方は、エラーを生成し、ログファイルでディスクパーティションをいっぱいにするか、エラーロギングを処理するためにすべてのシステムのリソースを必要とするリクエストでターゲットシステムを提供することです。

  message_burst および message_cost の設定は、システムの許容リスクと包括的なロギングのニーズに基づいて変更されるように設計されています。たとえば、message_burst を 10 に設定し、message_cost を 5 に設定すると、システムは 5 秒ごとに最大 10 個のメッセージ数を書き込むことができます。

- message_cost: message_burst の期間を定義して、すべての警告メッセージにコストを設定します。値が大きいほど、警告メッセージが無視されます。このファイルのデフォルト値は 5 です。

- netdev_max_backlog - 特定のインターフェースがパケットを処理できるよりも早く受信した場合に許可されるパケットの最大数を設定します。このファイルのデフォルト値は 1000 です。

- optmem_max: ソケットごとに許容される最大バッファーサイズを設定します。

- *rmem_default - 受信ソケットバッファーのデフォルトサイズをバイト単位で設定します。*

- *rmem_max - 受信ソケットバッファーの最大サイズをバイト単位で設定します。*

- *wmem_default: 送信ソケットバッファーのデフォルトサイズをバイト単位で設定します。*

- *wmem_max: 送信ソケットバッファーの最大サイズをバイト単位で設定します。*

/proc/sys/net/ipv4/ ディレクトリーには、追加のネットワーク設定が含まれます。これらの設定は、相互と併用されるものの多くは、システムに対する攻撃を防止したり、システムをルーターとして機能させたりする際に役立ちます。

これらのファイルを変更する場合には注意が必要です。

これらのファイルが誤って変更すると、システムへのリモート接続に影響する可能性があります。

以下は、/proc/sys/net/ipv4/ ディレクトリーにある重要なファイルの一部です。

- *icmp_echo_ignore_all と icmp_echo_ignore_broadcasts: カーネルが、すべてのホストからの ICMP ECHO パケットを無視すること、またはブロードキャストアドレスとマルチキャストアドレスから発信元のものだけを無視できるようにします。0 の値はカーネルが応答できますが、1 の値はパケットを無視します。*

- *ip_default_ttl: デフォルトの Time To Live(TTL) を設定します。これにより、宛先に到達する前にパケットが実行できるホップ数が制限されます。この値を増やすと、システムパフォーマンスが低下する可能性があります。*

- *ip_forward: システム上のインターフェースからパケットを転送するのを許可します。デフォルトでは、このファイルは 0 に設定されます。このファイルを 1 に設定すると、ネット*

ワークパケットの転送が可能になります。

- ip_local_port_range: ローカルポートが必要な場合に TCP または UDP が使用するポートの範囲を指定します。最初の番号は使用される最小ポートで、2 番目の番号は最も高いポートを指定します。デフォルト 1024 から 4999 よりも多くのポートを必要とするシステムは、32768 から 61000 の範囲を使用する必要があります。

- tcp_syn_retries: 接続の試行時にシステムが SYN パケットを再転送する回数に制限を指定します。

- tcp_retries1 - 受信接続への応答が許可される再送信の数を設定します。デフォルトは 3 です。

- tcp_retries2: TCP パケットの許可される再送信の数を設定します。デフォルトは 15 です。

/usr/share/doc/kernel-doc-kernel_version/Documentation/networking/ip-sysctl.txt ファイルには、/proc/sys/net/ipv 4/ ディレクトリーおよび /proc/sys/net/ipv6/ ディレクトリーで利用可能なファイルとオプションの一覧が含まれています。sysctl -a コマンドを使用して、sysctl キー形式でパラメーターを一覧表示します。

/proc/sys/net/ipv4/ ディレクトリーには、ネットワークスタックのさまざまな側面をカバーします。/proc/sys/net/ipv4/conf/ ディレクトリーを使用すると、未設定のデバイス（/proc/sys/net/ipv4/conf/default / サブディレクトリー内）のデフォルト設定の使用や、（/proc/sys/net/ipv4/conf /all / サブディレクトリー内）の上書きの設定など、各システムインターフェースを異なる方法で設定できます。

重要

Red Hat Enterprise Linux 6 はデフォルトで 厳密 な 逆方向パス転送 です。rp_filter ファイルで設定を変更する前に、『Red Hat Enterprise Linux 6nbsp; Hat Enterprise Linux 6nbsp;LinuxRed Hat Enterprise Linux 6nbsp;6 Security Guide』の「Reverse Path Forwarding」および「『Red Hat』 ナレッジベースアーティクル「rp_filter」を参照 してください。

/proc/sys/net/ipv4/neigh/ ディレクトリーには、システムに直接接続されたホスト（ネットワーク近い と呼ばれる）と通信する設定が含まれており、複数のホップ離れているシステム用の異なる設定も含まれます。

IPV4 でのルーティングには、独自のディレクトリー /proc/sys/net/ipv4/route/ もあります。conf/ および neigh/ とは異なり、/ proc/sys/net/ipv4/route/ ディレクトリーには、システム上の任意のインターフェースとのルーティングに適用される仕様が含まれます。max_size、max_delay、min_delay などの設定の多くは、ルーティングキャッシュのサイズ制御に関連します。ルーティングキャッシュを削除するには、すべての値を フラッシュ ファイルに書き込みます。

これらのディレクトリーおよびそれらの設定ファイルに設定できる値は、以下を参照してください。

> /usr/share/doc/kernel-doc-*kernel_version*/Documentation/filesystems/proc.txt

E.3.9.5. /proc/sys/vm/

このディレクトリーは、Linux カーネルの仮想メモリー(VM)サブシステムの設定を容易にします。カーネルは仮想メモリーを広範囲に使用し、一般的にスワップ領域と呼ばれます。

以下のファイルは、一般的に /proc/sys/vm/ ディレクトリーにあります。

- block_dump: 有効な場合はブロック I/O デバッグを設定します。ファイルに実行された読み取り/書き込みおよびブロックのダーティー操作はすべて、それに応じてログに記録されます。これは、ラップトップのバッテリーコンサービス向けに、ディスクのスピンアップとスピンダウンを診断する場合に便利です。block_dump が有効になっている場合のすべての出力は、dmesg で取得できます。デフォルト値は 0 です。

  KLOGD デーモンの停止

  block_dump がカーネルのデバッグと同時に有効になっている場合は、block_dump によって発生する間違ったディスクアクティビティーが生成されるため、klogd デーモンを停止するのは困難です。

- dirty_background_ratio: pdflush デーモンを介して、合計メモリーの割合でダーティーデータのバックグラウンドライトバックを開始します。デフォルト値は 10 です。

- dirty_expire_centisecs: ダーティーインメモリーデータが書き込みアウトの対象となるのに十分な古いタイミングを定義します。この間隔よりもダーティーなインメモリーであったデータは、次に pdflush デーモンがウェイクアップしたときに書き出されます。デフォルト値は 3000 で、1 秒の 100 分の 1 で表されます。

-

dirty_ratio: pdflush でダーティーデータのジェネレーターの合計メモリーの割合で、ダーティーデータのアクティブなライトバックを開始します。デフォルト値は 20 です。

- *dirty_writeback_centisecs: pdflush デーモンのウェイクアップの間隔を定義します。これは、ダーティーなインメモリーデータをディスクに定期的に書き込みます。デフォルト値は 500 で、1 秒の 100 分の 1 で表されます。*

- *laptop_mode: できるだけディスクスペースを下げてハードディスクを起動する必要がある回数を最小限に抑えるため、ラップトップでバッテリー電源を確保します。これにより、将来のすべての I/O プロセスを組み合わせてスピンアップする頻度を削減することで、効率性が向上します。デフォルト値は 0 ですが、ラップトップ上のバッテリーが使用されている場合に自動的に有効になります。*

  この値は、ユーザーにバッテリー電源が有効になっていると、auid デーモンによって自動的に制御されます。ラップトップが ACPI(Advanced Configuration and Power Interface)仕様をサポートする場合は、ユーザーの変更や対話は必要ありません。

  詳細は、以下のインストール済みドキュメントを参照してください。

  /usr/share/doc/kernel-doc-kernel_version/Documentation/laptop-mode.txt

- *max_map_count: プロセスが持つことのできるメモリーマップエリアの最大数を設定します。多くの場合、65536 のデフォルト値が適切です。*

- *min_free_kbytes - Linux 仮想マシン（仮想メモリーマネージャー）を強制して、最小キロバイト数をそのまま保持します。仮想マシンはこの数値を使用して、システムの lowmem ゾーンごとに pages_min 値を計算します。デフォルト値は、マシンの合計メモリーに対応します。*

- *nr_hugepages: カーネルに設定された hugetlb ページの現在の数を示します。*

  詳細は、以下のインストール済みドキュメントを参照してください。

  /usr/share/doc/kernel-doc-kernel_version/Documentation/vm/hugetlbpage.txt

-

nr_pdflush_threads: 現在実行している pdflush デーモンの数を指定します。このファイルは読み取り専用であるため、ユーザーが変更することはできません。I/O 負荷が大きい場合、カーネルによってデフォルト値が 2 に増えます。

- *overcommit_memory: 大規模なメモリー要求が許可または拒否される条件を設定します。以下の 3 つのモードを使用できます。*

  - *0 - カーネルは、利用可能なメモリー量と、フラットな無効なリクエストに見積もって、コミット処理でヒューリスティックなメモリーを実行します。ただし、メモリーは正確なアルゴリズムではなくヒューリスティックで割り当てられるため、この設定により、システムで利用可能なメモリーをオーバーロードできることがあります。これはデフォルトの設定です。*

  - *1 - カーネルは、コミット処理にメモリーを実行しません。この設定では、メモリーのオーバーロードの可能性は増大するため、メモリー集約されたタスク（一部の標準ソフトウェアによって実行されるタスクなど）に対してパフォーマンスが向上します。*

  - *2 - カーネルは、割り当てられたスワップ領域の合計と、/proc/sys/vm/overcommit_ratio で指定された物理 RAM のパーセンテージを超過するメモリーの要求に障害が発生します。この設定は、メモリーのオーバーコミットのリスクが少なくなるべく最適です。*

    この設定の使用

    この設定は、物理メモリーよりも大きい swap 領域を持つシステムにのみ推奨されます。

- *overcommit_ratio: /proc/sys/vm/overcommit_memory が 2 に設定されている場合に考慮される物理 RAM のパーセンテージを指定します。デフォルト値は 50 です。*

- *page-cluster: 1 回試行で読み込まれたページ数を設定します。16 ページに実際に関連する 3 のデフォルト値は、ほとんどのシステムに適しています。*

- *swappiness: スワップするマシンのサイズを決定します。値が大きいほどスワップが増えます。パーセンテージでデフォルト値は 60 に設定されます。*

カーネルベースの全ドキュメントは、ローカルにインストールされている場所にあります。

/usr/share/doc/kernel-doc-kernel_version/Documentation/。これには追加情報が含まれています。

E.3.10. /proc/sysvipc/

このディレクトリーには、System V IPC リソースに関する情報が含まれます。このディレクトリーのファイルは、メッセージ(msg)、セmaphores(sem)、および共有メモリー(shm)の System V IPC 呼び出しに関連します。

E.3.11. /proc/tty/

このディレクトリーには、システムで利用可能な、現在使用している tty デバイス に関する情報が含まれています。元々、テレタイプデバイス と呼ばれる文字ベースのデータターミナルは tty デバイス と呼ばれます。

Linux には、3 種類の tty デバイスがあります。シリアルデバイス は、モデム経由やシリアルケーブルなど、シリアル接続で使用されます。仮想ターミナル は、システムコンソールで Alt+F-key を押す際に利用可能な仮想コンソールなど、一般的なコンソール接続を作成します。擬似ターミナル は、XFree86 などの高レベルのアプリケーションで使用される双方向通信を作成します。ドライバー ファイルは、以下の例のように、現在使用中の tty デバイスの一覧です。

```
serial               /dev/cua        5  64-127 serial:callout
serial               /dev/ttyS       4  64-127 serial
pty_slave            /dev/pts      136   0-255 pty:slave
pty_master           /dev/ptm      128   0-255 pty:master
pty_slave            /dev/ttyp       3   0-255 pty:slave
pty_master           /dev/pty        2   0-255 pty:master
/dev/vc/0            /dev/vc/0       4       0 system:vtmaster
/dev/ptmx            /dev/ptmx       5       2 system
/dev/console         /dev/console    5       1 system:console
/dev/tty             /dev/tty        5       0 system:/dev/tty
unknown              /dev/vc/%d      4    1-63 console
```

/proc/tty/driver/serial ファイルには、各シリアル tty 行の使用状況の統計とステータスが表示されます。

tty デバイスをネットワークデバイスとして使用するために、Linux カーネルは、デバイスの ラインディスク を強制します。これにより、ドライバーはデバイス上で送信されるすべてのデータのブロックに特定のタイプのヘッダーを配置することが可能となり、接続のリモートエンドがデータブロックのストリームの 1 つとして処理できるようにします。SLIP と PPP は一般的なライン規則で、それぞれがシリアルリンクでシステムを相互に接続するのに一般的に使用されます。

E.3.12. /proc/PID/

OOM (Out of Memory) は、スワップ領域を含む利用可能なメモリーがすべて割り当てられているコンピューティング状態を指します。この状況が発生すると、システムがパニックになり、期待通りに機能しなくなります。/proc/sys/vm/panic_on_oom で OOM の動作を制御するスイッチがあります。1 に設定すると、カーネルは OOM でパニックが生じます。0 の設定は、カーネルに対し、OOM で oom_killer という名前の関数を呼び出すように指示します。通常、oom_killer は不正なプロセスを強制終了し、システムは存続します。

これを変更する最も簡単な方法は、新しい値を /proc/sys/vm/panic_on_oom にエコーすることです。

```
# cat /proc/sys/vm/panic_on_oom
1

# echo 0 > /proc/sys/vm/panic_on_oom

# cat /proc/sys/vm/panic_on_oom
0
```

oom_killer スコアを調整して、プロセスの強制終了を優先することもできます。/proc/PID/ には、oom_adj と oom_score という 2 つのツールのラベルがあります。oom_adj の有効なスコアは、-16 から +15 の範囲にあります。現在の oom_killer スコアを表示するには、プロセスの oom_score を表示します。oom_killer は、スコアが最も高いプロセスを強制終了します。

この例では、PID が 12465 のプロセスの oom_score を調整し、oom_killer が強制終了する可能性が低くなります。

```
# cat /proc/12465/oom_score
79872

# echo -5 > /proc/12465/oom_adj

# cat /proc/12465/oom_score
78
```

また、-17 に特殊な値もあり、これにより、そのプロセスの oom_killer が無効になります。以下の例では、oom_score は 0 の値を返し、このプロセスは強制終了されないことを示します。

```
# cat /proc/12465/oom_score
78

# echo -17 > /proc/12465/oom_adj

# cat /proc/12465/oom_score
0
```

badness（） と呼ばれる関数を使用して、各プロセスの実際のスコアを決定します。これは、検査された各プロセスに「ポイント」を追加して行います。プロセススコアは以下の方法で行われます。

1. *各プロセスのスコアのベースは、メモリーサイズです。*

2. *プロセスの子（カーネルスレッドを含む）のメモリーサイズもスコアに追加されます。*

3. *プロセスのスコアが「niced」プロセスのスコアが引き上げられ、長時間実行中のプロセスの場合は減少します。*

4. *CAP_SYS_ADMIN および CAP_SYS_RAWIO 機能を持つプロセスにより、スコアが減少します。*

5. *最終的なスコアは、oom_adj ファイルに保存されている値によってビットシフトされます。*

そのため、oom_score 値が最も高いプロセスは、ほとんどの場合権限がなく、その子とともに大量のメモリーを使用し、生の I/O を処理しない可能性があります。

E.4. SYSCTL コマンドの使用

/sbin/sysctl コマンドを使用して、/proc/sys/ ディレクトリーのカーネル設定を表示、設定、および自動化します。

/proc/sys/ ディレクトリーで設定可能なすべての設定の概要は、root で /sbin/sysctl -a コマンドを入力します。これにより、以下のような小規模な一覧である、大規模で包括的な一覧が作成されます。

```
net.ipv4.route.min_pmtu = 552
kernel.sysrq = 0
kernel.sem = 250     32000     32     128
```

これは、各ファイルが個別に表示されていた場合と同じ情報になります。唯一の違いはファイルの場所です。たとえば、/proc/sys/net/ipv4/route/min_pmtu ファイルは net.ipv4.route.min_pmtu として一覧表示され、ディレクトリースラッシュはドットと proc.sys の部分に置き換えられます。

sysctl コマンドを echo の代わりに使用して、/proc/sys/ ディレクトリーの書き込み可能なファイルに値を割り当てることができます。たとえば、コマンドを使用する代わりに使用されます。

```
echo 1 > /proc/sys/kernel/sysrq
```

以下のように同等の sysctl コマンドを使用します。

```
sysctl -w kernel.sysrq="1"
kernel.sysrq = 1
```

/proc/sys/ でこのような単一の値を設定すると、テスト中に便利ですが、/proc/sys/ 内の特別な設定はマシンの再起動時に失われるため、この方法は実稼働システムでは機能しません。カスタム設定を保持するには、/etc/sysctl.conf ファイルに追加します。

/etc/sysctl.conf ファイルは、initscripts パッケージでインストールして、一部のカーネルのデフォルト値を上書きするため、使用できるパラメーターは一部だけ含まれます。sysctl -a コマンドを使用して、sysctl キー形式でパラメーターを一覧表示します。設定可能な設定の詳細は、/usr/share/doc/kernel-doc-kernel_version/Documentation/networking/ip-sysctl.txt ファイルを参照してください。

システムを起動するたびに、init プログラムは /etc/rc.d/rc.sysinit スクリプトを実行します。このスクリプトには、/etc/ sysctl.conf を使用して sysctl を実行し、カーネルに渡される値を決定するコマンドが含まれます。したがって、/etc/sysctl.conf に追加した値は、システムが起動するたびに有効になります。sysctl の解析後に読み込まれるモジュールは、この設定を上書きする可能性があることに注意してください。

E.5. その他のリソース

以下は、proc ファイルシステムに関する追加情報の追加ソースです。

インストール可能なドキュメント

- /usr/share/doc/kernel-doc-kernel_version/Documentation/: kernel-doc パッケージが提供するこのディレクトリーには、proc ファイルシステムに関するドキュメントが含まれています。カーネルのドキュメントにアクセスする前に、root で以下のコマンドを実行する必要があります。

  ```
  ~]# yum install kernel-doc
  ```

- /usr/share/doc/kernel-doc-kernel_version/Documentation/filesystems/proc.txt: /proc/ ディレクトリーのあらゆる側面に関する情報が含まれます。

- */usr/share/doc/kernel-doc-kernel_version/Documentation/gitops.txt: System Request Key オプションの概要*

- */usr/share/doc/kernel-doc-kernel_version/Documentation/sysctl/: カーネル (kernel.txt)、ファイルシステムへのアクセス(fs.txt)、および仮想メモリー使用量(vm.txt)など、さまざまな sysctl ヒントを含むディレクトリー。*

- */usr/share/doc/kernel-doc-kernel_version/Documentation/networking/ip-sysctl.txt: IP ネットワークオプションの詳細な概要*

付録F 改訂履歴

改訂 9-3 Wed Mar 15 2017 Mirek Jahoda
6.9 GA 公開用バージョン

改訂 8-3 Mon May 30 2016 Maxim Svistunov
/proc/meminfo の付録セクションが更新され、若干改良されました。

改訂 8-2 Wed May 25 2016 Maxim Svistunov
『ReaR(Relax-and-Recover)』 が追加され、若干の改善されました。

改訂 8-1 Thu May 10 2016 Maxim Svistunov
Red Hat Enterprise Linux 6.8 GA リリースの導入ガイド

改訂 7-1 Tue Jul 14 2015 Barbora Ančincová
Red Hat Enterprise Linux 6.7 GA リリースの導入ガイド

改訂 7-0 Fri Apr 17 2015 Barbora Ančincová
Red Hat Enterprise Linux 6.7 Beta リリースの導入ガイド

改訂 6-3 Thu Apr 2 2015 Barbora Ančincová
『TigerVNC』 の更新、『ログファイルの表示および管理』、『システム登録およびサブスクリプション管理』、および 『kdump クラッシュリカバリーサービスの更新』。

改訂 6-2 Fri Oct 14 2014 Barbora Ančincová
Red Hat Enterprise Linux 6.6 GA リリースの導入ガイド

改訂 6-1 Fri Aug 22 2014 Jaromír Hradílek
『NetworkManager』、『ネットワークインターフェース』、『認証の設定』、『kdump クラッシュリカバリーサービス』、および 『proc ファイルシステムの』 更新

改訂 6-0 Mon Aug 11 2014 Jaromír Hradílek
Red Hat Enterprise Linux 6.6 Beta リリースの導入ガイド

改訂 5-1 Thu Nov 21 2013 Jaromír Hradílek
Red Hat Enterprise Linux 6.5 GA リリースの導入ガイド

改訂 5-0 Thu Oct 3 2013 Jaromír Hradílek
Red Hat Enterprise Linux 6.5 Beta リリースの導入ガイド

改訂 4-1 Thu Feb 21 2013 Jaromír Hradílek
Red Hat Enterprise Linux 6.4 GA リリースの導入ガイド

改訂 4-0 Thu Dec 6 2012 Jaromír Hradílek
Red Hat Enterprise Linux 6.4 Beta リリースの導入ガイド

改訂 3-1 Wed Jun 20 2012 Jaromír Hradílek
Red Hat Enterprise Linux 6.3 GA リリースの導入ガイド

改訂 3-0 Tue Apr 24 2012 Jaromír Hradílek
Red Hat Enterprise Linux 6.3 Beta リリースの導入ガイド

改訂 2-1 Tue Dec 6 2011 Jaromír Hradílek

Red Hat Enterprise Linux 6.2 GA リリースの導入ガイド

改訂 2-0 Mon Oct 3 2011 Jaromír Hradílek
Red Hat Enterprise Linux 6.2 Beta リリースの導入ガイド

改訂 1-1 Wed May 19 2011 Jaromír Hradílek
Red Hat Enterprise Linux 6.1 GA リリースの導入ガイド

改訂 1-0 Tue Mar 22 2011 Jaromír Hradílek
Red Hat Enterprise Linux 6.1 Beta リリースの導入ガイド

改訂 0-1 Tue Nov 9 2010 Douglas Silas
Red Hat Enterprise Linux 6.0 GA リリースの導入ガイド

改訂 0-0 Mon Nov 16 2009 Douglas Silas
初版作成。Red Hat Enterprise Linux 6 導入ガイド

索引