Red Hat Enterprise Linux for Real Time 9

安装 RHEL 9 for Real Time

在 Red Hat Enterprise Linux 上安装 RHEL for Real Time 内核

Last Updated: 2023-12-08

Red Hat Enterprise Linux for Real Time 9 安装 RHEL 9 for Real Time

在 Red Hat Enterprise Linux 上安装 RHEL for Real Time 内核

法律通告

摘要

在 Red Hat Enterprise Linux 上安装 RHEL for Real Time 内核以获取低延迟，并使响应时间可预测。了解如何安装实时内核、执行安装后任务并配置内核变体，如实时、库存或调试内核进行引导。

目录

使开源包含更多

红帽承诺替换我们的代码、文档和网页属性中存在问题的语言。我们从这四个术语开始：master、slave、黑名单和白名单。由于此项工作十分艰巨，这些更改将在即将推出的几个发行版本中逐步实施。详情请查看 CTO Chris Wright 的信息。

对红帽文档提供反馈

我们感谢您对我们文档的反馈。让我们了解如何改进它。

通过 Jira 提交反馈（需要帐户）

1. 登录到 Jira 网站。
2. 在顶部导航栏中点 **Create**
3. 在 **Summary** 字段中输入描述性标题。
4. 在 **Description** 字段中输入您的改进建议。包括到文档相关部分的链接。
5. 点对话框底部的 **Create**。

第 1 章 安装 RHEL FOR REAL TIME

很多行业和机构需要非常高的计算性能，并需要低且可预测的延迟，特别是银行和电信业。延迟（或响应时间）定义为事件和系统响应之间的时间，通常以 microseconds (µs) 衡量。

对于在 Linux 环境中运行的大多数应用程序，基本的性能调优可以满足对性能提高的要求。但对于一些行业，延迟不仅需要非常低，并且需要可以被预测和管理。红帽开发了一个相应的替换内核来满足这个要求。RHEL for Real Time 作为 RHEL 9 的一部分发布，提供与 RHEL 9 的无缝集成。通过 RHEL for Real Time，用户可以在其机构中测量、配置和记录延迟。

警告

在安装 RHEL for Real Time 前，请确定正确调整基本平台，并调整系统 BIOS 参数。如果没有执行这些任务，则可能会阻止 RHEL Real Time 部署获得一致的性能。

1.1. 通过 RHEL FOR REAL TIME 优化延迟

RHEL for Real Time 的设计宗旨时，对应用程序提供精心调优的系统，以满足这些应用程序对极高确定性的要求。内核系统调优在确定性方面提供了大量改进。

例如，在很多工作负载中，通过全面的系统调优，可以获得大约 90% 的性能改进。因此，在使用 RHEL for Real Time 前，，我们建议客户首先执行标准的 RHEL 系统调整，看它是否可以满足对性能的要求。

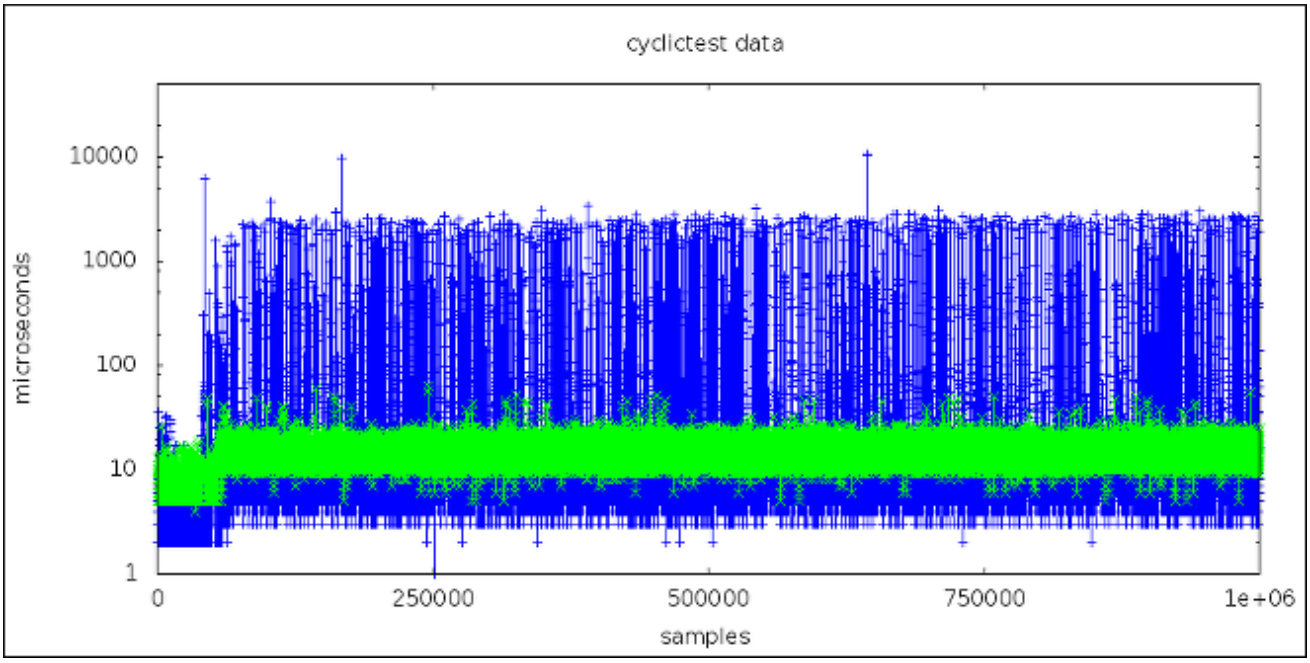
进行系统调优对于使用 Real Time kernel 的系统同样是非常重要的。在运行作为 RHEL 9 版本的一部分所提供的、没有进行过性能调优的标准内核系统上安装 Real Time 内核可能并不会获得显著的性能改进。调优标准内核将可能对延迟产生 90% 的改进。对于对延迟有非常高要求的工作负载，Real Time 内核提供了最后 10% 的延迟改进。

警告

在调整 Real Time 内核系统前，请确定已对基本系统进行了正确的调优，并调整了系统 BIOS 参数。如果没有执行这些任务，则可能会阻止 RHEL Real Time 部署获得一致的性能。

Real Time 内核的目的是，提供一致的、低延迟的、具有可预测的响应时间的内核。在系统中，会存在与实时内核关联的额外内核开销。这是因为在单独调度的线程中处理硬件中断。某些增加的工作负载开销会导致整个吞吐量下降。具体数量取决于工作负载，范围从 0% 到 30%。

对于典型的工作负载需要 millisecond (ms) 级别的延迟要求的环境，标准 RHEL 9 内核就足够了。但是，如果您的工作负载对核心内核功能（如中断处理）有严格的低延迟要求，需要满足 microsecond (µs) 级别的要求，则需要使用 Real Time 内核。



此图分别比较了一百万个使用 RHEL 9 和 RHEL for Real Time 内核的机器样本。

- 此图中的蓝色点代表已进行调优的 RHEL 9 内核的机器的系统响应时间（以 microseconds 为单位）。
- 图中的绿色点代表运行已调优的实时内核的机器的响应时间。

此图明确显示，Real Time 内核的响应时间非常一致，而标准内核的响应时间会根据不同的负载有很大的变化。

1.2. 使用 DNF 安装 RHEL FOR REAL TIME

除了使用 **dnf** 安装实时内核外，还可以从 Download Red Hat Enterprise Linux 门户下载包含 RHEL for Real Time 的 ISO 镜像。您可以使用此 ISO 镜像获取 RHEL for Real Time 中包含的所有 RPM 软件包。但是，由于这不是可引导 ISO 镜像，所以您无法使用它来创建可引导 USB 或者 CD 介质。

先决条件

- 最新版本的 RHEL 9 安装在 AMD64 或者 Intel64 系统中。实时内核在 AMD64 和 Intel 64 （也称为 x86_64）服务器平台上运行，这些平台经过认证可运行 Red Hat Enterprise Linux。
- 您的机器已注册，RHEL 附加到一个 RHEL for Real Time 订阅。
- 确定正确调整基本平台，并调整系统 BIOS 参数。

注意

在安装实时内核前无法执行预备任务，可能会阻止 RHEL for Real Time 内核部署提供一致的性能。

流程

1. 启用 RHEL for Real Time 仓库。

```
# subscription-manager repos --enable rhel-9-for-x86_64-rt-rpms
```

2. 安装 RHEL for Real Time 软件包组。

```
# dnf groupinstall RT
```

这个组会安装几个软件包：

- **kernel-rt** 包括 RHEL for Real Time 内核软件包。
- **kernel-rt-core** 包括核心 RHEL for Real Time 内核软件包。
- **kernel-rt-devel** 包括 RHEL for Real Time 内核开发软件包。
- **kernel-rt-modules** 包括 RHEL for Real Time 内核模块软件包。
- **kernel-rt-modules-core** 包括用于内核内核软件包的内核模块。
- **kernel-rt-modules-extra** 包括 RHEL for Real Time 内核额外模块软件包。
- **realtime-setup** 设置 RHEL for Real Time 所需的基本环境。
- **R** teval 评估系统是否适合 RHEL for Real Time。
- **rteval-loads** 包括 **rteval** 负载的源代码。
- **tuned-profiles-realtime** 包括额外的 **TuneD** 配置集，面向实时。

3. （可选） **tuna** 软件包包含一个工具，可帮助调整实时内核工作负载，从而大大从命令行或 GUI 自动执行 CPU 隔离和线程关联操作。这个软件包位于基本 RHEL 9 软件仓库中。

```
# dnf install tuna
```

注意

安装 RHEL for Real Time 内核时，它会自动设置为默认内核，并在下次引导时使用。您还可以将其他现有内核变体（如内核、 **kernel -debug** 或 **kernel-rt-debug** ）配置为默认的引导内核。如需更多信息，请参阅配置 kernel-rt 作为默认引导内核。

验证步骤

- 检查安装位置，并验证这些组件是否已成功安装。

```
# rpm -ql realtime-setup
/etc/security/limits.d/realtime.conf
/etc/sysconfig/realtime-setup
/etc/udev/rules.d/99-rhel-rt.rules
/usr/bin/realtime-setup
/usr/bin/rt-setup-kdump
/usr/bin/slub_cpu_partial_off
/usr/lib/.build-id
/usr/lib/.build-id/a4
/usr/lib/.build-id/a4/da77908aa4c6f048939f3267f1c552c456d117
/usr/lib/systemd/system/rt-entsk.service
/usr/lib/systemd/system/rt-setup.service
/usr/sbin/kernel-is-rt
/usr/sbin/rt-entsk
```

其他资源

- KVM 客户机能否实时(RT)内核运行？

1.3. RHEL FOR REAL TIME 软件仓库中的可用 RPM 软件包

RHEL for Real Time 存储库的 Red Hat Package Manager (RPM)包括以下软件包：

- **kernel-rt** 软件包，它是 RHEL for Real Time 内核软件包。
- RHEL for Real Time 内核测试软件包，其中包含实时内核的测试程序。
- RHEL for Real Time 调试软件包，用于调试和代码追踪。

表 1.1. 基本 RHEL for Real Time 内核软件包

RPM 软件包名称	描述	RT-specific	必填
kernel-rt	低延迟和抢占功能	是	是

表 1.2. RHEL for Real Time 内核测试软件包

RPM 软件包名称	描述	RT-specific	必填
kernel-rt-devel	用于内核开发的标头和库	是	否
kernel-rt-debug	带有编译调试功能的 RHEL for Real Time 内核 (slow)	是	否
kernel-rt-debug-devel	在 debug 内核中开发的标头和库	是	否
realtime-tests	用于衡量系统延迟以及提高优先级一致性 mutex 功能的工具	否	否

提供了调试软件包以用于 **perf**、**trace-cmd** 和 **crash** 工具来分析内核崩溃转储。调试软件包包括符号表，且非常大。因此，调试软件包与其他 RHEL for Real Time 软件包分开提供。您可以从 **RHEL for Real Time - Debug RPMs** 存储库下载调试软件包。

表 1.3. 基本 RHEL for Real Time 调试软件包

RPM 软件包名称	描述	RT-specific	必填
kernel-rt-debuginfo	分析和调试使用的符号，如 **perf** 或 **trace-cmd**	是	否
kernel-rt-debug-debuginfo	分析和追踪的符号	是	否

1.4. 安装后说明

安装实时内核后，请确保：

- 要实现最佳低延迟确定性，您需要执行 RHEL 进行特定实时系统调整。
- 您知道实时内核和标准内核的模块兼容性。
- 要启用 **kdump**，配置 RHEL for Real Time 启用 **kexec/kdump** 以提供崩溃转储信息。
- 验证 Real Time 内核是否为默认内核。

Real Time Kernel 和 Standard Kernel 的模块兼容性

实时内核与标准 Red Hat Enterprise Linux 9 内核有很大不同。因此，第三方内核模块与 RHEL for Real Time 不兼容。

内核模块本质上特定于其构建的内核。实时内核与标准内核有很大的不同，因此，模块也不同。因此，您无法从 Red Hat Enterprise Linux 9 获取第三方模块，并在实时内核中使用它们。如果您需要使用第三方模块，则必须将其与 RHEL for Real Time 标头文件进行重新编译，这些文件可在 RHEL for Real Time 开发和测试软件包中可用。为标准 Red Hat Enterprise Linux 9 提供的第三方驱动程序，但目前没有用于实时内核的自定义构建：

- EMC Powerpath
- Nvidia 图形
- 来自 Qlogic 的高级存储适配器配置工具

用户空间 **syscall** 接口与 RHEL for Real Time 兼容。

第 2 章 指定要运行的 RHEL 内核

您可以引导任何已安装的内核、标准或 Real Time。您可以在启动过程中在 GRUB 菜单中手动选择所需的内核。您还可以配置默认内核引导。

安装实时内核时，它会自动设置为默认内核，并在下次引导时使用。

2.1. 显示默认内核

您可以默认显示配置为引导的内核。

流程

- 查看默认内核：

```
~]# grubby --default-kernel
/boot/vmlinuz-4.18.0-80.rt9.138.el8.x86_64
```

命令的输出中的 **rt** 显示默认内核是一个实时内核。

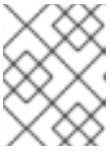
2.2. 显示正在运行的内核

您可以显示当前正在运行的内核

流程

- 显示系统当前运行的内核。

```
~]# uname -a
Linux rt-server.example.com 4.18.0-80.rt9.138.el8.x86_64 …
```



注意

当系统收到次要更新时，例如从 8.3 升级到 8.4 时，默认内核可能会自动从 Real Time 内核改为标准内核。

2.3. 将 KERNEL-RT 配置为默认的引导内核

在新安装的系统中，库存 RHEL **内核** 被设置为默认引导内核，并在下次引导时和后续系统更新时用作默认内核。您可以更改此配置，并将 **kernel-rt** 设为默认内核来引导，并将此配置在系统更新中持久保留。配置 **kernel-rt** 是一个一次性流程，您可以根据需要更改或恢复到另一个内核。您还可以将其他现有的内核变体（如 kernel、 **kernel -debug** ）或 **kernel-rt-debug** 配置为默认的引导内核。

流程

1. 要将 **kernel-rt** 配置为默认的引导内核，请输入以下命令：

```
# grubby --set-default=<RT_VMLINUZ>
```

RT_VMLINUZ 是与 **kernel-rt** 内核关联的 **vmlinux** 文件的名称。例如：

```
# grubby --set-default=/boot/vmlinuz-5.14.0-284.11.1.rt14.296.el9_2.x86_64+rt
```

2. 要将 **kernel-rt** 配置为系统更新中的默认引导内核，请输入以下命令：

```
# sed -i ‘s/UPDATEDEFAULT=.*/UPDATEDEFAULT=yes/g’/etc/sysconfig/kernel
# sed -i 's/DEFAULTKERNEL=.*/DEFAULTKERNEL=kernel-rt-core/g'/etc/sysconfig/kernel
```

当指定为 **yes** 时，**UPDATEDEFAULT** 变量将默认内核设置为使用系统更新更改。

在示例输出中，默认内核的路径特定于安装的 **kernel-rt-core** 软件包。您可以使用 **rpm -q kernel-rt-core** 命令从软件包确定内核的路径。

a. 可选：如果您需要从软件包确定内核的路径，首先列出安装的软件包：

```
# rpm -q kernel-rt-core
  kernel-rt-core-5.14.0-284.11.1.rt14.296.el9_2.x86_64
  kernel-rt-core-5.14.0-284.10.1.rt14.295.el9_2.x86_64
  kernel-rt-core-5.14.0-284.9.1.rt14.294.el9_2.x86_64
```

b. 要使用最新安装的软件包作为默认软件包，请输入以下命令从该软件包中查找引导镜像的路径：

```
# rpm -ql kernel-rt-core-5.14.0-284.11.1.rt14.296.el9_2.x86_64|grep‘^/boot/vmlinu’
/boot/vmlinuz-5.14.0-284.11.1.rt14.296.el9_2.x86_64.x86_64+rt
```

c. 要将 **kernel-rt** 配置为默认的引导内核，请输入以下命令：

```
# grubby --set-default=/boot/vmlinuz-5.14.0-284.11.1.rt14.296.el9_2.x86_64.x86_64+rt
```

验证

- 要验证 **kernel-rt** 是默认内核，请输入以下命令：

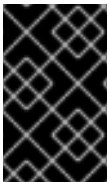
```
# grubby --default-kernel
/boot/vmlinuz-5.14.0-284.11.1.rt14.296.el9_2.x86_64.x86_64+rt
```

第 3 章 安装 KDUMP

在新的 Red Hat Enterprise Linux 安装中默认安装并激活 **kdump** 服务。了解 **kdump**，以及如何在默认未启用 时安装 **kdump**。

3.1. 什么是 KDUMP

kdump 是提供崩溃转储机制的服务。该服务可让您保存系统内存的内容以供分析。**kdump** 使用 **kexec** 系统调用在不重启的情况下引导至第二个内核（一个 *捕获内核*），然后捕获崩溃内核的内存的内容（*crash dump* 或 *vmcore*）并将其保存到一个文件中。这个第二个内核位于系统内存保留的一部分。



重要

内核崩溃转储可能会是系统失败时唯一可用的信息（关键错误）。因此，在关键任务环境中操作 **kdump** 非常重要。红帽建议系统管理员在正常内核更新周期内定期更新和测试 **kexec-tools**。这在部署了新内核功能时尤为重要。

您可以为机器上所有安装的内核或者只为指定的内核启用 **kdump**。当在机器上使用多个内核时，这非常有用，有些内核足够稳定，不必担心它们会崩溃。

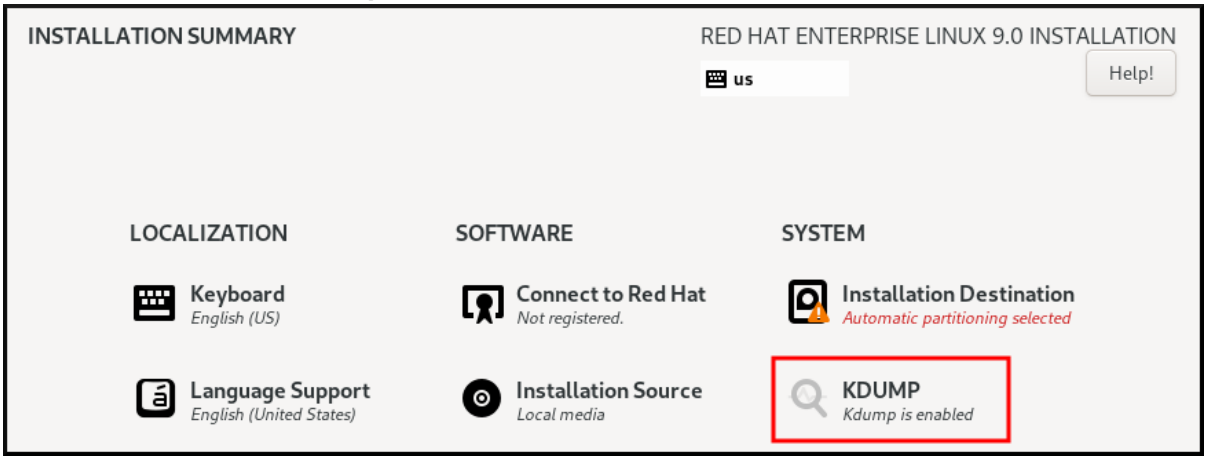
kdump 安装后，会创建一个默认的 **/etc/kdump.conf** 文件。文件包含默认的最小 **kdump** 配置。您可以编辑此文件来自定义 **kdump** 配置，但这不是必须的。

3.2. 使用 ANACONDA 安装 KDUMP

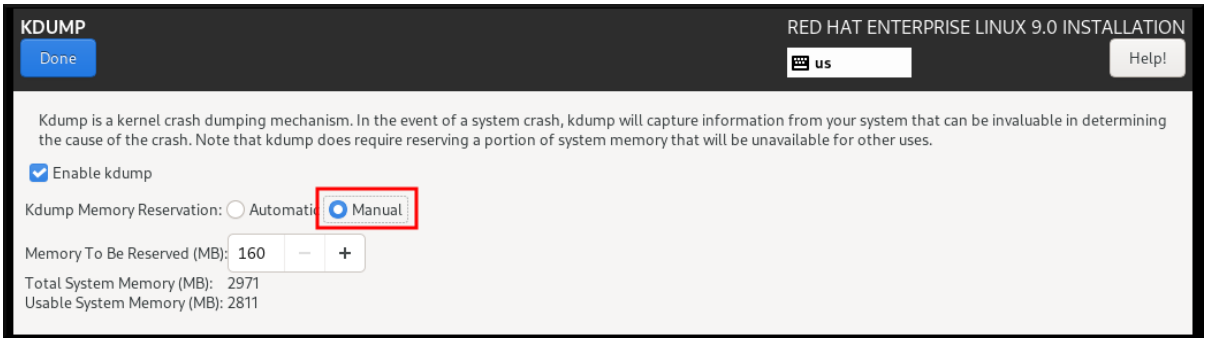
Anaconda 安装程序在交互安装过程中为 **kdump** 配置提供了一个图形界面屏幕。安装程序屏幕标题为 KDUMP，在主 Installation Summary 屏幕中提供。您可以启用 **kdump**，并保留所需的内存量。

流程

1. 进入 **Kdump** 字段。
2. 如果尚未启用，请启用 **kdump**。



3. 为 **kdump** 定义应保留多少内存。



3.3. 在命令行中安装 KDUMP

有些安装选项，比如自定义的 **Kickstart** 安装，在某些情况下，默认 **不** 安装或启用 **kdump**。如果是这种情况，请按照以下流程操作。

先决条件

- 有一个活跃的 RHEL 订阅
- **kexec-tools** 软件包
- 满足 **kdump** 配置和目标的要求。详情请查看 支持的 kdump 配置和目标。

流程

1. 检查是否在系统上安装了 **kdump** ：

```
# rpm -q kexec-tools
```

如果安装了该软件包，输出：

```
# kexec-tools-2.0.22-13.el9.x86_64
```

如果没有安装该软件包，输出：

```
package kexec-tools is not installed
```

2. 通过以下方法安装 **kdump** 和其他必要的软件包：

```
# dnf install kexec-tools
```

第 4 章 在命令行中配置 KDUMP

规划并构建 **kdump** 环境。

4.1. 估算 KDUMP 大小

在计划和构建 **kdump** 环境时，务必要知道崩溃转储文件需要多大的空间。

makedumpfile --mem-usage 命令估计崩溃转储文件需要的空间。它生成一个内存使用率报告。这个报告帮助您确定转储级别，以及可以安全排除哪些页面。

流程

- 执行以下命令以生成一个内存使用率报告：

```
# makedumpfile --mem-usage /proc/kcore


TYPE        PAGES    EXCLUDABLE    DESCRIPTION
----------------------------------------------------------
ZERO         501635      yes        Pages filled with zero
CACHE         51657      yes        Cache pages
CACHE_PRIVATE 5442       yes        Cache pages + private
USER          16301      yes        User process pages
FREE          77738211   yes        Free pages
KERN_DATA     1333192    no         Dumpable kernel data
```

重要

makedumpfile --mem-usage 命令会以页为单位报告所需的内存。这意味着您必须根据内核页面大小计算所使用的内存大小。

4.2. 配置 KDUMP 内存用量

系统引导过程中 **kdump** 的内存保留。内存大小在系统的 Grand Unified Bootloader (GRUB)配置中设置。内存大小取决于配置文件中指定的 **crashkernel=** 选项的值以及系统物理内存的大小。

您可以通过许多方式定义 **crashkernel=** 选项。您可以指定 **crashkernel=** 值或配置 **auto** 选项。**crashkernel=auto** 参数根据系统中的物理内存总量自动保留内存。配置后，内核会自动为捕获内核保留适当数量的所需内存。这有助于防止内存不足(OOM)错误。

注意

kdump 的自动内存分配因系统硬件架构和可用内存大小而异。

如果系统自动分配低于最小内存阈值，您可以手动配置保留内存量。

先决条件

- 您在系统上具有 root 权限。
- 满足 **kdump** 配置和目标的要求。详情请查看 支持的 kdump 配置和目标。

流程

1. 准备 **crashkernel=** 选项。
   - 例如：要保留 128 MB 内存，请使用：

     ```
     crashkernel=128M
     ```

   - 或者，您可以根据安装的内存总量将保留内存量设置为变量。变量中的内存保留语法为 **crashkernel=*<range1>*:*<size1>*,*<range2>*:*<size2>***。例如：

     ```
     crashkernel=512M-2G:64M,2G-:128M
     ```

     如果系统内存总量为 512 MB 和 2 GB，则命令保留 64 MB 内存。如果内存总量大于 2 GB，则内存保留为 128 MB。

   - 保留内存的偏移。
     有些系统需要保留内存并带有特定的固定偏移，因为 **crashkernel** 保留在早期发生，您可能需要为特殊用途保留更多内存。当您定义偏移时，保留内存会从那里开始。要偏移保留的内存，请使用以下语法：

     ```
     crashkernel=128M@16M
     ```

     在本例中，**kdump** 从 16 MB 开始保留 128 MB 内存（物理地址 **0x01000000**）。如果将 offset 参数设置为 0 或完全省略，**kdump** 会自动偏移保留内存。在设置变量内存保留时，也可以使用此语法。在这种情况下，偏移总是被最后指定。例如：

     ```
     crashkernel=512M-2G:64M,2G-:128M@16M
     ```

2. 将 **crashkernel=** 选项应用到引导装载程序配置：

   ```
   # grubby --update-kernel=ALL --args="crashkernel=<value>"
   ```

   将 ***<***value> 替换为您在上一步中准备的 **crashkernel=** 选项的值。

其他资源

- kdump 的内存要求
- 配置内核命令行参数
- 如何在系统引导前手动修改 GRUB 中的引导参数
- 如何在 Red Hat Enterprise Linux 8 中安装并引导自定义内核
- **grubby(8)** 手册页

4.3. 配置 KDUMP 目标

崩溃转储通常以一个文件形式存储在本地文件系统中，直接写入设备。或者，您可以为崩溃转储进行设置，以通过使用 **NFS** 或 **SSH** 协议的网络进行发送。一次只能设置其中一个选项来保留崩溃转储文件。默认行为是将其存储在本地文件系统的 **/var/crash/** 目录中。

先决条件

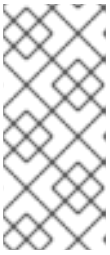
- **root** 权限。
- 满足 **kdump** 配置和目标的要求。详情请查看 支持的 kdump 配置和目标。

流程

- 要将崩溃转储文件保存在本地文件系统的 **/var/crash/** 目录中，请编辑 **/etc/kdump.conf** 文件并指定路径：

```
path /var/crash
```

选项 **path /var/crash** 代表 **kdump** 在其中保存崩溃转储文件的文件系统的路径。



注意

- 当您在 **/etc/kdump.conf** 文件中指定转储目标时，路径是 **相对** 于指定的转储目标。
- 当您没有在 **/etc/kdump.conf** 文件中指定转储目标时，该路径表示根目录的绝对 路径。

根据当前系统中挂载的内容，会自动采用转储目标和调整的转储路径。

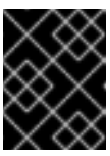
例 4.1. kdump 目标配置

```
# grep -v ^# /etc/kdump.conf | grep -v ^$
ext4 /dev/mapper/vg00-varcrashvol
path /var/crash
core_collector makedumpfile -c --message-level 1 -d 31
```

此处，转储目标被指定(**ext4 /dev/mapper/vg00-varcrashvol**)，因此在 **/var/crash** 处挂载。**path** 选项也被设置为 **/var/crash**，因此 **kdump** 会将 **vmcore** 文件保存在 **/var/crash/var/crash** 目录中。

- 要更改要保存崩溃转储的本地目录，请以 **root** 用户身份编辑 **/etc/kdump.conf** 配置文件：
  1. 从 **#path /var/crash** 行的开头删除 hash 符号("#")。
  2. 使用预期的目录路径替换该值。例如：

```
path /usr/local/cores
```



重要

在 RHEL 9 中，当 **kdump systemd** 服务启动时，使用 **path** 指令定义为 kdump 目标的目录必须存在，否则服务会失败。

- 要将文件写入不同的分区，请编辑 **/etc/kdump.conf** 配置文件：
  1. 根据您的选择，从 **#ext4** 行的开头删除 hash 符号("#")。
     - 设备名称（ **#ext4 /dev/vg/lv_kdump** 行）

- 文件系统标签（ **#ext4 LABEL=/boot** 行）
- UUID（ **#ext4 UUID=03138356-5e61-4ab3-b58e-27507ac41937** 行）

2. 将文件系统类型以及设备名称、标签或者 UUID 更改为所需值。例如：

```
ext4 UUID=03138356-5e61-4ab3-b58e-27507ac41937
```

注意

指定 UUID 值的正确语法是 **UUID="correct-uuid"** 和 **UUID=correct-uuid**。

重要

建议您使用 **LABEL=** 或 **UUID=** 指定存储设备。无法保证 **/dev/sda3** 等磁盘设备名称在重启后保持一致。

- 要将崩溃转储直接写入设备，请编辑 **/etc/kdump.conf** 配置文件：
  1. 删除 **#raw /dev/vg/lv_kdump** 行开头的哈希符号 ("#")。
  2. 使用预期的设备名称替换该值。例如：

```
raw /dev/sdb1
```

- 要使用 **NFS** 协议将崩溃转储保存到远程机器上：
  1. 删除 **#nfs my.server.com:/export/tmp** 行开头的哈希符号 ("#")。
  2. 使用有效的主机名和目录路径替换该值。例如：

```
nfs penguin.example.com:/export/cores
```

- 要使用 **SSH** 协议将崩溃转储保存到远程机器上：
  1. 从 **#ssh user@my.server.com** 行的开头删除 hash 符号("#")。
  2. 使用有效的用户名和密码替换该值。
  3. 在配置中包含 **SSH** 密钥。
     - 从 **#sshkey /root/.ssh/kdump_id_rsa** 行的开头删除哈希符号。
     - 将该值改为您要转储的服务器中有效密钥的位置。例如：

```
ssh john@penguin.example.com
sshkey /root/.ssh/mykey
```

4.4. 配置 KDUMP 核心收集器

kdump 服务使用 **core_collector** 程序捕获崩溃转储镜像。在 RHEL 中， **makedumpfile** 工具是默认的内核收集器。它通过以下方式帮助缩小转储文件：

- 压缩崩溃转储文件的大小，并只复制使用不同的转储级别所需的页面

- 排除不必要的崩溃转储页面
- 过滤崩溃转储中包含的页面类型。

语法

```
core_collector makedumpfile -l --message-level 1 -d 31
```

选项

- **-c**、**-l** 或 **-p** ：指定每个页的压缩 dump 文件的格式，使用 **zlib** 用于 **-c** 选项、使用 **lzo** 用于 **-l** 新选项，或 **snappy** 用于 **-p** 选项。
- **-d (dump_level)** ：排除页面，它们不会复制到转储文件中。
- **--message-level** ：指定消息类型。您可以通过使用这个选项指定 **message_level** 来限制打印的输出。例如，把 **message_level** 设置为 7 可打印常见消息和错误消息。 **message_level** 的最大值为 31

先决条件

- 您在系统上具有 root 权限。
- 满足 **kdump** 配置和目标的要求。详情请查看支持的 kdump 配置和目标。

流程

1. 以 **root** 用户身份，编辑 **/etc/kdump.conf** 配置文件并从 **#core_collector makedumpfile -l --message-level 1 -d 31** 的开头删除 hash 符号("#")。
2. 要启用崩溃转储文件压缩，请执行：

```
core_collector makedumpfile -l --message-level 1 -d 31
```

-l 选项指定 **转储** 压缩的文件格式。**-d** 选项将转储级别指定为 31。 **message-level** 选项将消息级别指定为 1。

另外，请考虑使用 **-c** 和 **-p** 选项的示例：

- 要使用 **-c** 压缩崩溃转储文件：

```
core_collector makedumpfile -c -d 31 --message-level 1
```

- 要使用 **-p** 压缩崩溃转储文件：

```
core_collector makedumpfile -p -d 31 --message-level 1
```

其他资源

- **makedumpfile (8)** 手册页
- kdump 的配置文件

4.5. 配置 KDUMP 默认失败响应

默认情况下，当 **kdump** 不能在配置的目标位置创建崩溃转储文件时，系统会重启，转储在此过程中会丢失。要更改此行为，请遵循以下步骤。

先决条件

- root 权限。
- 满足 **kdump** 配置和目标的要求。详情请查看 支持的 kdump 配置和目标。

流程

1. 以 **root** 用户身份，从 **/etc/kdump.conf** 配置文件的 **#failure_action** 行的开头删除 hash 符号 ("#")。
2. 将值替换为所需操作。

```
failure_action poweroff
```

其他资源

- 配置 kdump 目标

4.6. 测试 KDUMP 配置

测试 **kdump** 配置会验证配置，并记录崩溃转储使用指定工作负载完成的时间。

警告

测试 **kdump** 配置的命令将导致内核崩溃，并有数据丢失。请小心按照说明进行操作，不要使用活跃的生产系统来测试 **kdump** 配置。

流程

1. 在启用了 **kdump** 的情况下重启系统。
2. 检查 **kdump** 是否处于活跃状态。

```
# systemctl is-active kdump
active
```

3. 强制内核崩溃。

```
echo c > /proc/sysrq-trigger
```

警告

该命令可使内核崩溃，如果需要的话，会重启内核。

在内核重启时，***address-YYYY-MM-DD-HH:MM:SS*/vmcore** 文件会在您在 **/etc/kdump.conf** 文件中指定的位置（默认为 **/var/crash/**）创建。

其他资源

- 配置 kdump 目标

第 5 章 启用 KDUMP

通过使用这个流程，您可以为所有安装的内核或特定内核启用或禁用 **kdump** 服务。

5.1. 为所有安装的内核启用 KDUMP

您可以为在机器上安装的所有内核启用并启动 **kdump** 服务。

先决条件

- 管理员特权

流程

1. 向所有安装的内核添加 **crashkernel=auto** 命令行参数：

```
# grubby --update-kernel=ALL --args="crashkernel=auto"
```

2. 启用 **kdump** 服务。

```
# systemctl enable --now kdump.service
```

验证

- 检查 **kdump** 服务是否正在运行：

```
# systemctl status kdump.service

○ kdump.service - Crash recovery kernel arming
     Loaded: loaded (/usr/lib/systemd/system/kdump.service; enabled; vendor preset:
disabled)
     Active: active (live)
```

5.2. 为特定安装的内核启用 KDUMP

您可以为机器上的特定内核启用 **kdump** 服务。

先决条件

- 管理员特权

流程

1. 列出安装在机器上的内核。

```
# ls -a /boot/vmlinuz-*
/boot/vmlinuz-0-rescue-2930657cd0dc43c2b75db480e5e5b4a9 /boot/vmlinuz-4.18.0-
330.el8.x86_64 /boot/vmlinuz-4.18.0-330.rt7.111.el8.x86_64
```

2. 向系统的 Grand Unified Bootloader (GRUB)配置文件添加特定的 **kdump** 内核。
例如：

```
# grubby --update-kernel=vmlinuz-4.18.0-330.el8.x86_64 --args="crashkernel=auto"
```

3. 启用 **kdump** 服务。

```
# systemctl enable --now kdump.service
```

验证

- 检查 **kdump** 服务是否正在运行：

```
# systemctl status kdump.service

○ kdump.service - Crash recovery kernel arming
     Loaded: loaded (/usr/lib/systemd/system/kdump.service; enabled; vendor preset:
disabled)
     Active: active (live)
```

5.3. 禁用 KDUMP 服务

要在引导时禁用 **kdump** 服务，请按照以下流程操作。

先决条件

- 满足 **kdump** 配置和目标的要求。详情请查看支持的 kdump 配置和目标。
- 安装 **kdump** 的所有配置都是根据您的需要设置的。详情请参阅 安装 kdump。

流程

1. 要在当前会话中停止 **kdump** 服务：

```
# systemctl stop kdump.service
```

2. 要禁用 **kdump** 服务：

```
# systemctl disable kdump.service
```

警告

建议您设置 **kptr_restrict=1**。在这种情况下，**kdumpctl** 服务会加载崩溃内核，无论是否启用了 Kernel Address Space Layout(KASLR)。

故障排除步骤

当 **kptr_restrict** 没有设置为(1)时，如果启用了 KASLR，**/proc/kcore** 文件的内容将会生成全零。因此，**kdumpctl** 服务无法访问 **/proc/kcore** 并载入崩溃内核。

要临时解决这个问题，**/usr/share/doc/kexec-tools/kexec-kdump-howto.txt** 文件会显示一条警告信息，它推荐使用 **kptr_restrict=1** 设置。

要确保 **kdumpctl** 服务加载了崩溃内核，请验证 **kernel.kptr_restrict = 1** 是否已列在 **sysctl.conf** 文件中。

其他资源

- 管理 systemd

第 6 章 报告 RHEL FOR REAL TIME 错误

报告 RHEL for Real Time 错误的首选方法是提交 Red Hat Bugzilla 的错误报告。在造成错误前，识别发生问题的源（如标准内核或 RHEL for Real Time 内核）非常有用。

6.1. 诊断 RHEL FOR REAL TIME 错误

确定哪个内核(RHEL for Real Time 或标准内核)是问题的来源可能会加快您的 bug 修复的机会。按照以下步骤，您可以在提交错误报告前诊断问题的来源。

先决条件：

- 安装了 RHEL for Real Time 内核的最新版本。

流程：

1. 验证您具有 RHEL for Real Time 的最新版本。
2. 使用 **GRUB** 菜单引导进入 RHEL for Real Time 内核。
3. 如果出现问题，请向 RHEL for Real Time 报告错误。
4. 尝试使用标准内核重现问题。
   此故障排除步骤有助于识别问题的位置。

注意

如果标准内核没有出现这个问题，则程序错误可能是 RHEL for Real Time 特定增强中引入的变化造成的（Red Hat 在基准(4.18.0)内核之上应用）。

6.2. 使用 BUGZILLA 提交错误报告

在识别了特定于 RHEL for Real Time 的错误后，请使用以下步骤使用 Bugzilla 提交错误报告。

先决条件：

- 您有 Red Hat Bugzilla 帐户。

流程

1. 登录到 Bugzilla 帐户。
2. 点 **Enter A New Bug Report**。
3. 选择 **Red Hat classification**。
4. 选择 **Red Hat Enterprise Linux** 产品。
5. 输入 **Component**。
   例如，如果 kernel 问题是内核问题或受影响用户空间组件的名称，请使用 **kernel-rt**，如 **rteval**。
6. 提供有关 RHEL for Real Time 内核的错误问题的详细说明。
   在输入问题描述时，如果您能够在标准 RHEL 8 内核中重现问题，您还可以的状态。

其他资源

- Red Hat Bugzilla - 创建新的 Red Hat Bugzilla 帐户