



美国硅谷第二可用区开放

10余款云产品上线
全美双可用区尽享ECS88折，10.10~11.9 仅此一月

[查看详情](#)[🏠](#) » [技术](#) » [学习](#) » [查看内容](#)

RHCSA 系列（五）：RHEL7 中的进程管理：开机，关机

2015-9-18 14:36 评论: 1 收藏: 3

原文：<http://www.tecmint.com/rhcsa-exam-boot-process-and-process-management/>
译文：LCTT <https://linux.cn/article-6249-1.html>

作者：Gabriel Cánepa
译者：FSSlc

我们将概括和简要地复习从你按开机按钮来打开你的 RHEL 7 服务器到呈现出命令行界面的登录屏幕之间所发生的所有事情，以此来作为这篇文章的开始。

Linux 开机过程

请注意：

1. 相同的基本原则也可以应用到其他的 Linux 发行版本中，但可能需要较小的更改，并且
2. 下面的描述并不是旨在给出开机过程的一个详尽的解释，而只是介绍一些基础的东西

Linux 开机过程

1. 初始化 POST（加电自检）并执行硬件检查；
2. 当 POST 完成后，系统的控制权将移交给启动管理器的 first stage 第一阶段，它存储在一个硬盘的引导扇区（对于使用 BIOS 和 MBR 的旧式的系统而言）或存储在一个专门的 (U)EFI 分区上。
3. 启动管理器的第一阶段完成后，接着进入启动管理器的 second stage 第二阶段，通常大多数使用的是 GRUB（GRand Unified Boot Loader 的简称），它驻留在 `/boot` 中，然后开始加载内核和驻留在 RAM 中的初始化文件系统（被称为 initramfs，它包含执行必要操作所需要的程序和二进制文件，以此来最终挂载真实的根文件系统）。
4. 接着展示了 splash 闪屏 过后，呈现在我们眼前的是类似下图的画面，它允许我们选择一个操作系统和内核来启动：

启动菜单屏幕

5. 内核会对接入到系统的硬件进行设置，当根文件系统被挂载后，接着便启动 PID 为 1 的进程，这个进程将开始初始化其他的进程并最终呈现给我们一个登录提示符界面。

注意：假如我们想在启动后查看这些信息，我们可以使用 `dmesg` 命令 <<https://linux.cn/article-3587-1.html>>，并使用这个系列里的上一篇文章中介绍过的工具（注：即 `grep`）来过滤它的输出。

登录屏幕和进程的 PID

在上面的例子中，我们使用了大家熟知的 `ps` 命令来显示在系统启动过程中的一系列当前进程的信息，它们的父进程（或者换句话说，就是那个开启这些进程的进程）为 `systemd`（大多数现代的 Linux 发行版本已经切换到系统和服务器管理器）：

```
1. # ps -o ppid,pid,uname,comm --ppid=1
```

记住 **-o**（为 **-format** 的简写）选项允许你以一个自定义的格式来显示 **ps** 的输出，以此来满足你的需求；这个自定义格式使用 **man ps** 里 **STANDARD FORMAT SPECIFIERS** 一节中的特定关键词。

另一个你想自定义 **ps** 的输出而不是使用其默认输出的情形是：当你需要找到引起 CPU 或内存消耗过多的那些进程，并按照下列方式来对它们进行排序时：

```
1. # ps aux --sort=+pcpu      # 以 %CPU 来排序(增序)
2. # ps aux --sort=-pcpu     # 以 %CPU 来排序(降序)
3. # ps aux --sort=+pmem     # 以 %MEM 来排序(增序)
4. # ps aux --sort=-pmem     # 以 %MEM 来排序(降序)
5. # ps aux --sort=+pcpu,-pmem # 结合 %CPU (增序) 和 %MEM (降序)来排列
```

自定义 **ps** 命令的输出

systemd 的一个介绍

在 Linux 世界中，很少有能比在主流的 Linux 发行版本中采用 **systemd** 引起更多的争论的决定。**systemd** 的倡导者根据以下事实来表明其主要的优势：

1. 在系统启动期间，**systemd** 允许并发地启动更多的进程（相比于先前的 **SysVinit**，**SysVinit** 似乎总是表现得更慢，因为它一个接一个地启动进程，检查一个进程是否依赖于另一个进程，然后等待守护进程启动才可以启动的更多的服务），并且
2. 在一个运行着的系统中，它用作一个动态的资源管理器。这样在启动期间，当一个服务被需要时，才启动它（以此来避免消耗系统资源）而不是在没有一个合理的原因的情况下启动额外的服务。
3. 向后兼容 **sysvinit** 的脚本。

另外请阅读: '**init**' 和 '**systemd**' 背后的故事 <<http://www.tecmint.com/systemd-replaces-init-in-linux/>>

systemd 由 **systemctl** 工具控制，假如你了解 **SysVinit**，你将会对以下的内容感到熟悉：

- **service** 工具，在旧一点的系统中，它被用来管理 **SysVinit** 脚本，以及
- **chkconfig** 工具，为系统服务升级和查询运行级别信息
- **shutdown** 你一定使用过几次来重启或关闭一个运行的系统。

下面的表格展示了使用传统的工具和 **systemctl** 之间的相似之处：

旧式工具	Systemctl 等价命令	描述
service name start	systemctl start name	启动 name（这里 name 是一个服务）
service name stop	systemctl stop name	停止 name
service name condrestart	systemctl try-restart name	重启 name（如果它已经运行了）
service name restart	systemctl restart name	重启 name
service name reload	systemctl reload name	重载 name 的配置
service name status	systemctl status name	显示 name 的当前状态
service - status-all	systemctl	显示当前所有服务的状态
chkconfig name on	systemctl enable name	通过一个特定的单元文件，让 name 可以在系统启动时运行（这个文件是一个符号链接）。启用或禁用一个启动时的进程，实际上是增加或移除一个到 /etc/systemd/system 目录中的符号链接。
chkconfig name off	systemctl disable name	通过一个特定的单元文件，让 name 可以在系统启动时禁止运行（这个文件是一个符号链接）。
chkconfig -list name	systemctl is-enabled name	确定 name（一个特定的服务）当前是否启用。

chkconfig - list	systemctl - type=service	显示所有的服务及其是否启用或禁用。
shutdown -h now	systemctl poweroff	关机
shutdown -r now	systemctl reboot	重启系统

systemd 也引进了单元（unit）（它可能是一个服务，一个挂载点，一个设备或者一个网络套接字）和目标（target）（它们定义了 systemd 如何去管理和同时开启几个相关的进程，可以认为它们与在基于 SysVinit 的系统中的运行级别等价，尽管事实上它们并不等价）的概念。

总结归纳

其他与进程管理相关，但并不仅限于下面所列的功能的任务有：

1. 在考虑到系统资源的使用上，调整一个进程的执行优先级：

这是通过 **renice** 工具来完成的，它可以改变一个或多个正在运行着的进程的调度优先级。简单来说，调度优先级是一个允许内核（当前只支持 ≥ 2.6 的版本）根据某个给定进程被分配的执行优先级（即友善度（niceness），从 -20 到 19）来为其分配系统资源的功能。

renice 的基本语法如下：

```
1. | # renice [-n] priority [-gpu] identifier
```

在上面的通用命令中，第一个参数是即将使用的优先级数值，而另一个参数可以是进程 ID（这是默认的设置），进程组 ID，用户 ID 或者用户名。一个常规的用户（即除 root 以外的用户）只可以更改他或她所拥有的进程的调度优先级，并且只能增加友善度的层次（这意味着占用更少的系统资源）。

进程调度优先级

2. 按照需要杀死一个进程（或终止其正常执行）：

更精确地说，杀死一个进程指的是通过 **kill** 或 **pkill** <<https://linux.cn/article-2116-1.html>> 命令给该进程发送一个信号，让它优雅地（SIGTERM=15）或立即（SIGKILL=9）结束它的执行。

这两个工具的不同之处在于前一个被用来终止一个特定的进程或一个进程组，而后一个则允许你通过进程的名称和其他属性，执行相同的动作。

另外，pkill 与 pgrep 相捆绑，pgrep 提供将受符合的进程的 PID 给 pkill 来使用。例如，在运行下面的命令之前：

```
1. | # pkill -u gacanepa
```

查看一眼由 gacanepa 所拥有的 PID 或许会带来点帮助：

```
1. | # pgrep -l -u gacanepa
```

找到用户拥有的 PID

默认情况下，kill 和 pkill 都发送 SIGTERM 信号给进程，如我们上面提到的那样，这个信号可以被忽略（即该进程可能会终止其自身的执行，也可能不终止），所以当你因一个合理的理由要真正地停止一个运行着的进程，则你将需要在命令行中带上特定的 SIGKILL 信号：

```
1. | # kill -9 identifier          # 杀死一个进程或一个进程组
2. | # kill -s SIGNAL identifier  # 同上
3. | # pkill -s SIGNAL identifier  # 通过名称或其他属性来杀死一个进程
```

结论

在这篇文章中，我们解释了在 RHEL 7 系统中，有关开机启动过程的基本知识，并分析了一些可用的工具来帮助你通过使用一般的程序和

systemd 特有的命令来管理进程。

请注意，这个列表并不旨在涵盖有关这个话题的所有花哨的工具，请随意使用下面的评论栏来添加你自己钟爱的工具和命令。同时欢迎你的提问和其他的评论。

via: <http://www.tecmint.com/rhcsa-exam-boot-process-and-process-management/>
<<http://www.tecmint.com/rhcsa-exam-boot-process-and-process-management/>>

作者: [Gabriel Cánepa](http://www.tecmint.com/author/gacanepa/) <<http://www.tecmint.com/author/gacanepa/>> 译者: [FSSlc](https://github.com/FSSlc) <<https://github.com/FSSlc>> 校对: [wxy](https://github.com/wxy) <<https://github.com/wxy>>

本文由 [LCTT](https://github.com/LCTT/TranslateProject) <<https://github.com/LCTT/TranslateProject>> 原创翻译，[Linux中国](file:///root/github/my-notes/Manual/Linux.cn/RHCSA/RHCSA%E7%B3%BB%E5%88%975RHEL7%20%E4%B8%AD%E7%9A%84%E8%BF%9B%E7%A8%8B%E7%AE%A1%E7%90%86%EF%BC%9A%E5%BC%80%E6%9C%BA.html) <<file:///root/github/my-notes/Manual/Linux.cn/RHCSA/RHCSA%E7%B3%BB%E5%88%975RHEL7%20%E4%B8%AD%E7%9A%84%E8%BF%9B%E7%A8%8B%E7%AE%A1%E7%90%86%EF%BC%9A%E5%BC%80%E6%9C%BA.html>> 荣誉推出

原文: <http://www.tecmint.com/rhcsa-exam-boot-process-and-process-management/> <<http://www.tecmint.com/rhcsa-exam-boot-process-and-process-management/>>

作者: Gabriel Cánepa

译文: [LCTT](http://lctt.github.io/) <<http://lctt.github.io/>> <https://linux.cn/article-6249-1.html> <<https://linux.cn/article-6249-1.html>>

译者: FSSlc

发表评论

验证码 换一个



热点评论

netb2c [Chrome 45.0|GNU/Linux] 2015-9-20 12:49

又一个专题栏目了! :>

赞 2

体验环境



本文导航

- **Linux** 开机过程
- **systemd** 的一个介绍
- 总结归纳
- 结论

 RHCSA

• RHCSA 系列（四）：编辑文本文件及分析文本	2015-9-16
• RHCSA 系列（六）：使用 Parted 和 SSM 来配置和加密系统存储	2015-9-21
• RHCSA 系列（八）：加固 SSH，设定主机名及启用网络服务	2015-9-23
• RHCSA 系列（九）：安装、配置及加固一个 Web 和 FTP 服务器	2015-9-24
• RHCSA 系列（十）：Yum 包管理、Cron 自动任务计划和监控系统日志	2015-9-26
• RHCSA 系列（十一）：使用 firewalld 和 iptables 来控制网络流量	2015-9-29