

- [developerWorks 中国](#)
- [技术主题](#)
- [Linux](#)
- [文档库](#)

浅析 Linux 初始化 init 系统，第 1 部分: sysvinit

从 sysvinit 到 systemd

近年来，Linux 系统的 init 进程经历了两次重大的演进，传统的 sysvinit 已经淡出历史舞台，新的 init 系统 UpStart 和 systemd 各有特点，而越来越多的 Linux 发行版采纳了 systemd。本文简要介绍了这三种 init 系统的使用和原理，每个 Linux 系统管理员和系统软件开发者都应该了解它们，以便更好地管理系统和开发应用。本文是系列的第一部分，主要讲述 sysvinit 的特点和使用。

刘明，老程序员一名，喜欢使用和研究 Linux，目前从事数据仓库开发和应用。

2014 年 7 月 02 日

什么是 Init 系统,init 系统的历史和现状

Linux 操作系统的启动首先从 BIOS 开始，接下来进入 boot loader，由 boot loader 载入内核，进行内核初始化。内核初始化的最后一步就是启动 pid 为 1 的 init 进程。这个进程是系统的第一个进程。它负责产生其他所有用户进程。

init 以守护进程方式存在，是所有其他进程的祖先。init 进程非常独特，能够完成其他进程无法完成的任务。

Init 系统能够定义、管理和控制 init 进程的行为。它负责组织和运行许多独立的或相关的始化工作(因此被称为 init 系统)，从而让计算机系统进入某种用户预订的运行模式。

仅仅将内核运行起来是毫无实际用途的，必须由 init 系统将系统代入可操作状态。比如启动外壳 shell 后，便有了人机交互，这样就可以让计算机执行一些预订程序完成有实际意义的任务。或者启动 X 图形系统以便提供更佳的人机界面，更加高效的完成任务。这里，字符界面的 shell 或者 X 系统都是一种预设的运行模式。

大多数 Linux 发行版的 init 系统是和 System V 相兼容的，被称为 sysvinit。这是人们最熟悉的 init 系统。一些发行版如 Slackware 采用的是 BSD 风格 Init 系统，这种风格使用较少，本文不再涉及。其他的发行版如 Gentoo 是自己定制的。Ubuntu 和 RHEL 采用 upstart 替代了传统的 sysvinit。而 Fedora 从版本 15 开始使用了一个被称为 systemd 的新 init 系统。

可以看到不同的发行版采用了不同的 init 实现，本系列文章就是打算讲述三个主要的 Init 系统：sysvinit，UpStart 和 systemd。了解它们各自的设计特点，并简要介绍它们的使用。

在 Linux 主要应用于服务器和 PC 机的时代，SysVinit 运行非常良好，概念简单清晰。它主要依赖于 Shell 脚本，这就决定了它的最大弱点：启动太慢。在很少重新启动的 Server 上，这个缺点并不重要。而当 Linux 被应用到移动终端设备的时候，启动慢就成了一个大问题。为了更快地启动，人们开始改进 sysvinit，先后出现了 upstart 和 systemd 这两个主要的新一代 init 系统。Upstart 已经开发了 8 年多，在不少系统中已经替换 sysvinit。Systemd 出现较晚，但发展更快，大有取代 upstart 的趋势。

本文的第一部分先简要介绍 sysvinit。

Sysvinit 概况

sysvinit 就是 system V 风格的 init 系统，顾名思义，它源于 System V 系列 UNIX。它提供了比 BSD 风格 init 系统更高的灵活性。是已经风行了几十年的 UNIX init 系统，一直被各类 Linux 发行版所采用。

运行级别

Sysvinit 用术语 runlevel 来定义"预订的运行模式"。Sysvinit 检查 '/etc/inittab' 文件中是否含有 'initdefault' 项。这告诉 init 系统是否有一个默认运行模式。如果没有默认的运行模式，那么用户将进入系统控制台，手动决定进入何种运行模式。

sysvinit 中运行模式描述了系统各种预订的运行模式。通常会有 8 种运行模式，即运行模式 0 到 6 和 S 或者 s。

每种 Linux 发行版对运行模式的定义都不太一样。但 0，1，6 却得到了大家的一致赞同：

- 0 关机
- 1 单用户模式
- 6 重启

通常在 /etc/inittab 文件中定义了各种运行模式的工作范围。比如 RedHat 定义了 runlevel 3 和 5。运行模式 3 将系统初始化为字符界面的 shell 模式；运行模式 5 将系统初始化为 GUI 模式。无论是命令行界面还是 GUI，运行模式 3 和 5 相对于其他运行模式而言都是完整的正式的运行状态，计算机可以完成用户需要的任务。而模式 1，S 等往往用于系统故障之后的排错和恢复。

很显然，这些不同的运行模式下系统需要初始化运行的进程和需要进行的初始化准备都是不同的。比如运行模式 3 不需要启动 X 系统。用户只需要指定需要进入哪种模式，sysvinit 将负责执行所有该模式所必须的初始化工作。

sysvinit 运行顺序

Sysvinit 巧妙地用脚本，文件命名规则和软链接来实现不同的 runlevel。首先，sysvinit 需要读取/etc/inittab 文件。分析这个文件的内容，它获得以下一些配置信息：

- 系统需要进入的 runlevel
- 捕获组合键的定义
- 定义电源 fail/restore 脚本
- 启动 getty 和虚拟控制台

得到配置信息后，sysvinit 顺序地执行以下这些步骤，从而将系统初始化为预订的 runlevel X。

- /etc/rc.d/rc.sysinit
- /etc/rc.d/rc 和/etc/rc.d/rcX.d/ (X 代表运行级别 0-6)
- /etc/rc.d/rc.local
- X Display Manager（如果需要的话）

首先，运行 rc.sysinit 以便执行一些重要的系统初始化任务。在 RedHat 公司的 RHEL5 中(RHEL6 已经使用 upstart 了)，rc.sysinit 主要完成以下这些工作。

- 激活 udev 和 selinux
- 设置定义在/etc/sysctl.conf 中的内核参数
- 设置系统时钟
- 加载 keymaps
- 使能交换分区
- 设置主机名(hostname)
- 根分区检查和 remount
- 激活 RAID 和 LVM 设备
- 开启磁盘配额
- 检查并挂载所有文件系统
- 清除过期的 locks 和 PID 文件

完成了以上这些工作之后，sysvinit 开始运行/etc/rc.d/rc 脚本。根据不同的 runlevel，rc 脚本将打开对应该 runlevel 的 rcX.d 目录(X 就是 runlevel)，找到并运行存放在该目录下的所有启动脚本。每个 runlevel X 都有一个这样的目录，目录名为/etc/rc.d/rcX.d。

在这些目录下存放着很多不同的脚本。文件名以 S 开头的脚本就是启动时应该运行的脚本，S 后面跟的数字定义了这些脚本的执行顺序。在/etc/rc.d/rcX.d 目录下的脚本其实都是一些软链接文件，真实的脚本文件存放在/etc/init.d 目录下。如下所示：

清单 1.rc5.d 目录下的脚本

```
[root@www ~]# ll /etc/rc5.d/
lrwxrwxrwx 1 root root 16 Sep  4  2008 K02dhcdd -> ../init.d/dhcdd
....(中间省略)....
lrwxrwxrwx 1 root root 14 Sep  4  2008 K91capi -> ../init.d/capi
lrwxrwxrwx 1 root root 23 Sep  4  2008 S00microcode_ctl -> ../init.d/microcode_ctl
lrwxrwxrwx 1 root root 22 Sep  4  2008 S02lvm2-monitor -> ../init.d/lvm2-monitor
....(中间省略)....
lrwxrwxrwx 1 root root 17 Sep  4  2008 S10network -> ../init.d/network
....(中间省略)....
lrwxrwxrwx 1 root root 11 Sep  4  2008 S99local -> ../rc.local
lrwxrwxrwx 1 root root 16 Sep  4  2008 S99smartd -> ../init.d/smartd
....(底下省略)....
```

当所有的初始化脚本执行完毕。Sysvinit 运行/etc/rc.d/rc.local 脚本。

rc.local 是 Linux 留给用户进行个性化设置的地方。您可以把自己私人想设置和启动的东西放到这里，一台 Linux Server 的用户一般不止一个，所以才有这样的考虑。

Sysvinit 和系统关闭

Sysvinit 不仅需要负责初始化系统，还需要负责关闭系统。在系统关闭时，为了保证数据的一致性，需要小心地按顺序进行结束和清理工作。

比如应该先停止对文件系统有读写操作的服务，然后再 umount 文件系统。否则数据就会丢失。

这种顺序的控制这也是依靠/etc/rc.d/rcX.d/目录下所有脚本的命名规则来控制的，在该目录下所有以 K 开头的脚本都将在关闭系统时调用，字母 K 之后的数字定义了它们的执行顺序。

这些脚本负责安全地停止服务或者其他的关闭工作。

Sysvinit 的管理和控制功能

此外，在系统启动之后，管理员还需要对已经启动的进程进行管理和控制。原始的 sysvinit 软件包包含了一系列的控制启动，运行和关闭所有其他程序的工具。

halt

停止系统。

init

这个就是 sysvinit 本身的 init 进程实体，以 pid1 身份运行，是所有用户进程的父进程。最主要的作用是在启动过程中使用/etc/inittab 文件创建进程。

killall5

就是 SystemV 的 killall 命令。向除自己的会话(session)进程之外的其它进程发出信号，所以不能杀死当前使用的 shell。

last

回溯/var/log/wtmp 文件(或者-f 选项指定的文件)，显示自从这个文件建立以来，所有用户的登录情况。

lastb

作用和 last 差不多，默认情况下使用/var/log/btmp 文件，显示所有失败登录企图。

mesg

控制其它用户对用户终端的访问。

pidof

找出程序的进程识别号(pid)，输出到标准输出设备。

poweroff

等于 shutdown -h -p，或者 telinit 0。关闭系统并切断电源。

reboot

等于 shutdown -r 或者 telinit 6。重启系统。

runlevel

读取系统的登录记录文件(一般是/var/run/utmp)把以前和当前的系统运行级输出到标准输出设备。

shutdown

以一种安全的方式终止系统，所有正在登录的用户都会收到系统将要终止通知，并且不准新的登录。

sulogin

当系统进入单用户模式时，被 init 调用。当接收到启动加载程序传递的-b 选项时，init 也会调用 sulogin。

telinit

实际是 init 的一个连接，用来向 init 传送单字符参数和信号。

utmpdump

以一种用户友好的格式向标准输出设备显示/var/run/utmp 文件的内容。

wall

向所有有信息权限的登录用户发送消息。

不同的 Linux 发行版在这些 sysvinit 的基本工具基础上又开发了一些辅助工具用来简化 init 系统的管理工作。比如 RedHat 的 RHEL 在 sysvinit 的基础上开发了 initscripts 软件包，包含了大量的启动脚本 (如 rc.sysinit)，还提供了 service，chkconfig 等命令行工具，甚至一套图形化界面来管理 init 系统。其他的 Linux 发行版也有各自的 initscript 或其他名字的 init 软件包来简化 sysvinit 的管理。

只要您理解了 sysvinit 的机制，在一个最简的仅有 sysvinit 的系统下，您也可以直接调用脚本启动和停止服务，手动创建 inittab 和创建软连接来完成这些任务。因此理解 sysvinit 的基本原理和命令是最重要的。您甚至也可以开发自己的一套管理工具。

Sysvinit 的小结

Sysvinit 的优点是概念简单。Service 开发人员只需要编写启动和停止脚本，概念非常清楚；将 service 添加/删除到某个 runlevel 时，只需要执行一些创建/删除软连接文件的基本操作；这些都不需要学习额外的知识或特殊的定义语法(UpStart 和 Systemd 都需要用户学习新的定义系统初始化行为的语言)。

其次，sysvinit 的另一个重要优点是确定的执行顺序：脚本严格按照启动数字的大小顺序执行，一个执行完毕再执行下一个，这非常有益于错误排查。UpStart 和 systemd 支持并发启动，导致没有人可以确定地了解具体的启动顺序，排错不易。

但是串行地执行脚本导致 sysvinit 运行效率较慢，在新的 IT 环境下，启动快慢成为一个重要问题。此外动态设备加载等 Linux 新特性也暴露出 sysvinit 设计的一些问题。针对这些问题，人们开始想办法改进 sysvinit，以便加快启动时间，并解决 sysvinit 自身的设计问题。

Upstart 是第一个被广泛应用的新一代 init 系统。我们在接下来的第二部分介绍 UpStart。

参考资料

学习

- Linux From Scratch 项目关于 [sysvinit](#) 的材料，对 sysvinit 有比较全面和简练的介绍。
- 维基百科上 [init 系统](#) 的条目对 init 进程有简略却精确的介绍。
- 经典文章 [鸟哥的 Linux 私房菜](#) 中有对 init, rc.d 等概念的精彩而详细的介绍。
- 在 [developerWorks Linux 专区](#) 寻找为 Linux 开发人员（包括 [Linux 新手入门](#)）准备的更多参考资料。

讨论

- 加入 [developerWorks 中文社区](#)。查看开发人员推动的博客、论坛、组和维基，并与其他 developerWorks 用户交流。



IBM Bluemix 资源中心

文章、教程、演示，帮助您构建、部署和管理云应用。



developerWorks 中文社区

立即加入来自 IBM 的专业 IT 社交网络。



IBM 软件资源中心

免费下载、试用软件产品，构建应用并提升技能。

static.content.url=http://www.ibm.com/developerworks/js/artrating/
SITE_ID=10
Zone=Linux
ArticleID=976052
ArticleTitle=浅析 Linux 初始化 init 系统，第 1 部分: sysvinit
publish-date=07022014