linux 的引导过程

LINUX 引导过程

首先说明一下，这里讲的是 LINUX 引导经过的步骤，而不涉及 KERNEL 引导过程的内部细节。希望本文能对初学 LINUX 的朋友有所帮助。

一、从 BIOS 到 KERNEL

计算机在接通电源之后首先由 BIOS 进行自检，即进行所谓的 POST（Power On Self Test），然后依据 BIOS 内设置的引导顺序从硬盘、软盘或 CDROM 中读入“引导块”。如通常 BIOS 中设的引导顺序为 C 在最前面，那么就把 C 盘（第一个 IDE 硬盘）的第 0 柱面，第 0 头的第 1 个扇区读入内存，然后跳到那里开始执行。这个扇区有一个大家熟悉的名字——MBR（Main Boot Record）。换句话说，MBR 里面存放的是一小段程序以及分区表的数据。在使用 WIN9X 和 DOS 时，这里面放的代码就把分区表里标记为 Active 的分区的第一个扇区（一般存放着操作系统的引导代码）读入内存并跳转到那里开始执行。而在用 LILO 引导 LINUX 时，有两种选择：

(1) 把 LILO 安装在 MBR。这时就由 BIOS 直接把 LILO 代码调入内存，然后跳转执行 LILO。即

BIOS——>LILO（在 MBR 中）——>KERNEL

(2) 把 LILO 安装在 LINUX 分区，并把 LINUX 分区设为 Active。这时，BIOS 调入的是 WIN9X/DOS 下的 MBR 代码，然后由这段代码来调入 LILO 的代码（位于活动分区的第一个扇区）。即

BIOS——>MBR——>LILO（在活动分区的第一个扇区）——>KERNEL

因为在读入及执行 MBR 时，操作系统还没有起来，所以只能用 BIOS 提供的 INT13 来进行磁盘操作，而 INT13 只能读写硬盘 1024 柱面之前的数据，由此可知任何操作系统的引导代码必须在 1024 柱面之前。对于 LINUX 来说，不管你使用方式(1)还是方式(2)启动，都要保证 KERNEL 放在 1024 柱面之前。只有在 KERNEL 起来以后，才有读/写 1024 柱面以后数据的能力。因为 LINUX 不使用 INT13 来进行硬盘操作。从上面我们也可以看到，不存在什么“WIN95 可以，而 LINUX 不可以”的问题，作为操作系统要能被正确引导，在现有的 BIOS 下，它们的引导部分都必须在 1024 柱面之前。如果操作系统本身还是基于 INT13 来进行磁盘操作的话，那么它也只能读/写 1024 柱面之前的数据。

二、从 KERNEL 到 login prompt

在 KERNEL 起来之后，将生成第一个进程——init，实际上是执行了/sbin/init。init 的工作是根据/etc/inittab 来执行相应的脚本进行系统初始化，如设置键盘、字体，装载模块，设置网络，等等。

/etc/inittab 文件的每一行包括四个域：

id:runlevels:action:process

runlevel 是运行模式，通常为 0-6。模式 0 是 halt，模式 6 是 reboot，模式 1 是单用户，模式 2/3 是多用户，模式 5 是运行 xdm 以图形界面方式登录。id 为标识符，通常为两个字母。process 为需要执行的程序或脚本。action 包括有:

(1) defaultinit —— 指定缺省的运行模式（runlevel）

(2) sysinit —— 指定运行的第一个程序/脚本，此时 runlevels 域不起作用。

(3) boot —— 在 sysinit 之后执行，runlevels 域不起作用

(4) bootwait —— 同 boot，但 init 会等待该命令结束

(5) once —— 在进入有 runlevels 指定的运行模式时运行

(6) wait —— 同上，但 init 会等待该命令结束

(7) respawn —— 在进入相应 runlevel 时执行，并且若该进程结束，init 会再起一个进程执行同样的命令

(8) ctrlaltdel —— 指定在用户按下 Ctrl-Alt-Del 时执行的命令

对于 Redhat 来说，执行的顺序为：

```
/etc/rc.d/rc.sysinit            # 由 init 执行的第一个脚本
/etc/rc.d/rc $RUNLEVEL          # $RUNLEVEL 为缺省的运行模式
/sbin/mingetty                  # 等待用户登录
```

三、/etc/rc.d/rc.sysinit 及/etc/rc.d/rc

在 Redhat 中，/etc/rc.d/rc.sysinit 主要做在各个运行模式中相同的初始化工作，包括：

调入 keymap 以及系统字体

启动 swapping

设置主机名

设置 NIS 域名

检查（fsck）并 mount 文件系统

打开 quota

装载声卡模块

设置系统时钟

等等。

/etc/rc.d/rc 则根据其参数指定的运行模式来执行相应目录下的脚本。凡是以 Kxx 开头的，都以 stop 为参数来调用；凡是以 Sxx 开头的，都以 start 为参数来调用。调用的顺序按 xx 从小到大来执行。例如，假设缺省的运行模式是 3，/etc/rc.d/rc 就会按上述方式调用/etc/rc.d/rc3.d/下的脚本。

值得一提的是，Redhat 中的运行模式 2、3、5 都把/etc/rc.d/rc.local 做为初始化脚本中的最后一个，所以用户可以自己在这个文件中添加一些需要在其他初始化工作之后，登录之前执行的命令。

四、init 在等待/etc/rc.d/rc 执行完毕之后（因为在/etc/inittab 中/etc/rc.d/rc 的 action 是 wait），将在指定的各个虚拟终端上运行/sbin/mingetty，等待用户的登录。至此，LINUX 的启动结束。

五、对于 Slackware，作为 sysinit 的脚本是/etc/rc.d/rc.S，运行模式 1 的脚本是/etc/rc.d/rc.K，运行模式 2、3、4、5 的脚本是/etc/rc.d/rc.M。另外，装载模块的命令都集中在/etc/rc.d/rc.modules 中，/etc/rc.d/rc.local 为登录前执行的最后一个脚本。