第28章　运行级别脚本

如果希望在系统启动时自动运行某些应用程序、服务或脚本，或者在系统重启动时能够正确地关闭这些程序，那么需要创建运行级别脚本。除一种 LINUX变体外，所有的LINUX版本都含有这种基于系统V的运行级别配置目录，就像其他 UNIX版本那样。

既然所有的系统都含有这种类型的配置，我们在本章中将会对它加以介绍，但如果你的系统不含有这种目录，也不要紧。还可以通过其他方法在系统启动时自动运行程序；本章的后半部分也将介绍这些方法。

本章包含下列内容：

- 运行级别。
- 如何创建rc.scripts。
- 如何在不同的运行级别实现相应的 rc.scripts。
- 如何从inittab中启动应用程序。

能够创建运行级别脚本就意味着能够更灵活地控制系统。如果需要在某一个特定的运行级别启动或停止程序，就得创建运行级别脚本（它们通常被称为 rc.script）。

任何用关键字 start或stop调用的、能够启动或停止程序运行的脚本都可以看作是一个 rc.script。注意，应当由用户来保证他或她所提交的脚本是一个有效的脚本，能够正确地启动或停止某一服务。

运行级别配置目录的机制使得 rc.script只在系统切换运行级别时有效。它不负责检查某一运行级别中所有的特定服务是否都已经被启动或停止。这是 shell编程者的事。

还可以按照希望运行的服务来控制系统的运行级别，但是这已经超出了本书的讨论范围。

28.1　怎么知道系统中是否含有运行级别目录

rc.scripts一般保存在 (实际上是个链接，这一点我们将在后面讲述　)/etc/rcN.d或/etc/rc.d/rcN.d目录下，

其中，N是一个数字。通常是7个，因为rcN.d目录的序号是从0到6，不过在系统上可能会有另外几个目录，如rcS.d。这并不重要，这里我们只关心带有数字的目录。

```
$ pwd
/etc
drwxr-xr-x   2 root     sys          1024 Dec 22  1996 rc0.d
drwxr-xr-x   2 root     sys          1024 Dec 22  1996 rc1.d
drwxr-xr-x   2 root     sys          1024 Dec 22  1996 rc2.d
drwxr-xr-x   2 root     sys          1024 Dec 22  1996 rc3.d
drwxr-xr-x   2 root     sys          1024 Dec 22  1996 rc4.d
drwxr-xr-x   2 root     sys          1024 Dec 22  1996 rc5.d
drwxr-xr-x   2 root     sys          1024 Dec 22  1996 rc6.d
drwxr-xr-x   2 root     sys          1024 Dec 22  1996 rcS.d
```

如果是LINUX系统，那么……

```
$ pwd
```

（续）

```
/etc/rc.d
$ ls
init.d      rc.local    rc0.d       rc2.d       rc4.d       rc6.d
rc          rc.sysinit  rc1.d       rc3.d       rc5.d
```

如果我们使用cd命令进入这些rcN.d目录，会发现这些目录中的rc.scripts实际上是一些链接。

```
$ pwd
/etc/rc.d/rc2.d
$ ls -l
lrwxrwxrwx   1 root root 16 Dec 3 15:16 K87ypbind -> ../init.d/yd
lrwxrwxrwx   1 root root 17 Dec 3 15:10 K89portmap -> ../init.d/p
lrwxrwxrwx   1 root root 17 Dec 3 15:07 S01kerneld -> ../init.d/d
...
...
```

28.2 确定当前的运行级别

本章不是针对系统管理员的，但是作为shell编程者，应当了解rc.scripts是什么，它们是被怎样放置到运行级别配置目录中的。顺便说一下，如果想知道当前的运行级别，可以用下面的命令：

```
$ who -r
   .       run-level 4  Apr 22 13:26    4    0    3
```

在‘run level’后面的数字就是当前的运行级别。后面的时间是系统最近一次重启动的时间。

如果是LINUX系统，那么……

$ runlevel

2 3

第一列表示系统的前一个运行级别，第二列表示系统当前的运行级别，在这里是3。

28.3 快速熟悉inittab

运行级别目录中含有一系列启动服务的脚本。这里的“服务”可以是守护进程、应用程序、服务器、子系统或脚本进程。在系统启动的过程中，将会启动一个名为init的进程（它是系统中所有进程的祖先）。它所要完成的一部分工作就是看看需要启动哪些服务，应当缺省地进入哪一个运行级别。它通过查看一个名为inittab的配置文件来获得上述信息，该配置文件位于/etc目录下。init进程还按照该文件中的设置加载特定的进程。如果需要编辑这个配置文件，一定要先做一个备份。如果该文件被破坏或出现“降级”错误，系统将无法正常启动，到那时，将不得不进入单用户模式并修正该文件。

inittab文件所包含的域具有严格的格式。该文件中每个条目的格式为：

```
id:rstart:action:process
```

其中，id域是相应进程的唯一标识。

rstart域所包含的数字表示运行该进程的级别。

action域告诉init进程如何对待process所对应的进程。这里可以有很多种动作，但是最常

见的是wait和respawn。wait意味着当进程启动后等待它结束。 respawn则意味着如果该进程不存在，则启动相应的进程，如果它存在，那么只要它一掉下来就立即重新启动它。

process域包含了实际要运行的命令。下面是 inittab文件的一部分。

```
$ pg /etc/inittab
id:3:initdefault:

# System initialization.
si::sysinit:/etc/rc.d/rc.sysinit
# run level 0
l0:0:wait:/etc/rc.d/rc 0
# run level 1
l1:1:wait:/etc/rc.d/rc 1
# run level 2
l2:2:wait:/etc/rc.d/rc 2
# run level 3
l3:3:wait:/etc/rc.d/rc 3
# run level 4
l4:4:wait:/etc/rc.d/rc 4
# run level 5
l5:5:wait:/etc/rc.d/rc 5
# runlevel 6
l6:6:wait:/etc/rc.d/rc 6
#Run gettys in standard runlevels
1:12345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty ttyS1 vt100
```

该文件的第一行是系统缺省的运行级别，这里是级别 3，一般都是这样。

以数字10到16开始的行启动或停止该运行级别所对应的全部运行级别脚本。例如，该文件中有这样一行：

```
l5:5:wait:/etc/rc.d/rc 5
```

它的意思是，在运行级别 5应该以参数5执行脚本/etc/rc.d/rc，即/etc/rc.d/rc执行/etc/rc.d/rc5.d目录中的所有脚本。

在上述文件的最后一行，在运行级别 2、3、4和5，该进程将会始终存在，即使暂时掉下来，大概也不会超过 1s。这一始终存在的进程是串口 ttyS1上的mingetty。该命令含有一个参数，即终端类型为vt100。

28.4 运行级别

init进程在系统完全就绪之前所做的最后几项工作之一就是执行缺省运行级别所包含的所有脚本。该进程是通过 /etc/rc.d/rc或/etc/rc.init来启动这些脚本的。它的作用是首先杀死该运行级别所包含的进程再启动这些进程。

但是它怎么知道该启动或停止哪些服务呢？ rc或rc.init脚本将会使用for循环来依次查看相应运行级别目录中的文件，给每一个链接名以 K开头的相应脚本赋予参数stop；给每一个链接名以S开头的相应脚本赋予参数 start。在运行级别切换时，上述脚本也会完成同样的工作，只不过根据相应的运行级别来启动或停止对应的脚本。

rcN.d目录中的脚本只是一些链接 —— 真正的脚本保存在其他的目录中。它们通常都放置

在/usr/sbin/init.d或/etc/init.d目录中。

> 如果是LINUX系统，那么……
> /etc/rc.d/init.d

在这个目录中含有一些能够启动或停止某一服务的脚本。这些脚本的名字最好能够表示出它所实现的功能，形如 rc.<功能>，其中rc表示运行命令（run command）或运行控制（run control），或者就像某些系统管理员所称的那样“真正关键的”（‘real crucial’）。

下面是这类文件的部分列表。

```
$ ls
rc.localfs        rc.halt        rc.reboot   rc.syslogd   rc.daemon
...
```

一般来说，rc.scripts都应当能够接受这样的参数：

rc.name stop：停止该服务。

rc.name start：启动该服务。

可选的参数包括restart和status。其他任何参数都应当给出相应的用法说明。注意，可以手工运行这些脚本。

现在我们已经知道运行级别脚本应当具有什么样的功能，下一步就是把它们放置在相应的rcN.d目录中。不过在此之前我们先来了解一下系统运行级别。

28.4.1 各种运行级别

系统含有七种运行级别(见表28-1)。不同的系统在某些运行级别上稍有差别。

在将一个脚本放置在不同的运行级别目录中之前，首先应当弄清楚打算在哪一个运行级别启动或停止相应的服务？一旦弄清楚这一点，就可以接着进行下面的步骤了。

表28-1 各个运行级别的用途

运行级别0	启动和停止整个系统
运行级别1	单用户或管理模式
运行级别2	多用户模式；部分网络服务被启动。有些系统将其作为正常运行模式，而不是级别 3
运行级别3	正常操作运行模式，启动所有的网络服务
运行级别4	用户定义的模式，可以使用该级别来定制所需要运行的服务
运行级别5	有些UNIX操作系统变体将其作为缺省 X-windows模式，还有些系统把它作为系统维护模式
运行级别6	重启动

28.4.2 运行级别脚本的格式

rcN.d目录中的脚本都是一些链接，这样是为了省去不必要的副本。这些链接的格式为：

```
Snnn.script_name
```

或

```
Knnn.script_name
```

其中，

S：代表启动相应的进程

K：代表杀死相应的进程

nn：是00至99的两位数字，不过在有些系统中是 000至999三位数字。在不同目录中的链

接应采用同一数字。例如，如果某个服务在 rc3.d中启动时名为S45.myscript，那么如果希望它在rc2.d中启动，应当使用链接名 S45.myscript。

script_name：相应脚本的文件名，根据所在操作系统的不同，它们可能位于下列目录中：

```
/usr/sbin/init.d
/etc/rc.d
/etc/init.d
```

当init进程调用相应的运行级别脚本时，杀进程按照从高到低的 K序号进行，即K23,myscript K12.named；而启动进程按照从低到高的序号进行。如果使用的是 LINUX系统，K序号将按照从高到低的顺序执行。

28.4.3 安装运行级别脚本

如果想要安装自己的运行级别脚本，必须：

- 编写该脚本，确保它符合调用标准。
- 确信它能够启动或终止相应的服务。
- 将该脚本放置于(取决于操作系统)/etc/init.d或/usr/sbin/init.d或/etc/rc.d中。
- 在相应的rcN.d目录中按照合理的命名方式创建链接。

下面的脚本能够启动或停止一个名为 rc.audit的审核应用程序。该服务运行于级别3、5、4，停止于级别 6、2、1。通过查看 rcN.d中的条目，我们发现序号 35空闲，于是就使用该序号。实际上，系统并不对使用已占用的序号作任何检查。

下面就是这个脚本。可以看到，该脚本使用了一个简单的 case语句来接收start和stop参数。

```
$ pg rc.audit
#!/bin/sh
# rc.audit start| stop
# script to start or stop zeega's audit application
#
case "$1" in
start)
  echo -n "Starting the audit system...."
  /apps/audit/audlcp -a -p 12
  echo
  touch /var/lock/subsys/rc.audit
  ;;
stop)
  echo -n "Stopping the audit system...."
  /apps/audit/auddown -k0
  echo
  rm -f /var/lock/subsys/rc.audit
  ;;
restart)
  $0 stop
  $0 start
  ;;
*)
  echo "To call properly..Usage: $0 {start|stop|restart}"
  exit 1
  ;;
esac
```

```
exit 0
```

> 如果是LINUX系统，那么……
> 有些LINUX变体在启动服务时要求创建一个锁文件。如果没有锁文件，在杀死该脚本时就可能会出现问题。

start选项将使该审核进程启动相应的审核系统，而 stop选项将使它终止该系统的运行。当然，在将自己的运行级别脚本放置在 init.d目录中之前，应该首先对该脚本进行测试。

```
$ rc.audit
To call properly..Usage:./rc.audit {start|stop|restart}
$ rc.audit start
Starting the audit system....
```

让我们假定该脚本已经通过了测试。它能够正确地启动和停止审核服务。现在我们把该脚本放置在相应的运行级别目录中。

在本系统中，rcN.d目录位于/etc/rc.d目录下，而我的运行级别脚本保存在 /etc/rc.d/init.d目录下。如果系统目录结构与上面的不同，那么需要对下面的命令作相应的调整。

我们首先启动该脚本 —— 记住启动脚本所使用的链接名是以 S打头的。

```
$ pwd
/etc/rc.d/rc3.d
$ ln -s../init.d/rc.audit S35rc.audit

$ ls -l
...
lrwxrwxrwx   1 root   root 27 May  8 14:37 S35rc.audit -> ../init.d/
rc.audit
...
```

我们已经创建了相应的链接。 ls -l命令的结果显示该链接指向 /etc/init.d/rc.audit文件。我本应该在链接命令中给出全路径，不过没有这个必要。现在我只要进入其他的相关目录 (rc4.d和rc5.d)使用同样的命令就可以启动其他相应的服务。

如果希望停止某个脚本的运行，可以使用如下命令：

```
$ pwd
/etc/rc.d/rc6.d
$ ln -s../init.d/rc.audit K35rc.audit
$ ls -l
...
lrwxrwxrwx   1 root   root  27 May  8 14:43 K35rc.audit -> ../init.d/
rc.audit
...
```

在其他相关目录中，也可以如法炮制，停止相应的审核服务。现在当系统重启动时 (运行级别6)，它将被停止；在运行级别切换到 2或1时也是如此。该服务在运行级别 4或5中同样也会被启动。

28.5 使用inittab来启动应用程序

我们还可以用其他的方法来启动应用程序。可以通过在 inittab文件中加入相应的条目来做到这一点。在我所管理的有些系统上，我就使用了这种方法，这倒不是因为这些系统中没有运行级别目录，而是由于我有几个用于系统检查的脚本需要在系统刚刚就绪之后运行。使用inittab是实现上述功能的理想途径。

这里我们给出一个例子，我打算在系统运行在级别 3时运行我的一个磁盘镜像检查脚本。首先我确定该脚本能够正确运行，然后对 inittab文件做备份。

```
$ cp /etc/initab /etc/inittab.bak
```

接下来编辑inittab文件，在该文件末尾加入这样一个条目：

```
# disk checker script, let's see if any of the mirrors are broken.
rc.diskchecker:3:once:/usr/local/etc/rc.diskchecker > /dev/console 2>&1
```

保存并退出。

上面的一条意思是：

行首的rc.diskchecker是该进程在运行级别3中的唯一标识。该进程只运行一次。所要运行的脚本是/usr/local/etc/rc.diskchecker，所有的输出都被送到控制台。

28.6 启动和停止服务的其他方法

如果不想把/etc/inittab文件弄得过于杂乱，还有其他的方法可以实现启动和停止服务的功能。大多数系统都含有一个名为 rc.local的文件，一般来说也是位于 /etc目录下。该脚本文件将在inittab和运行级别脚本之后运行。可以在该文件中加入任何命令，或从中调用最习惯用的启动脚本。

有些系统还在/bin目录下(更多的是在/usr/sbin目录下)含有一个名为shutdown的脚本文件。可以使用它来关闭某些服务。

28.7 小结

运行级别的确是一个系统管理问题。本章的目的在于：使你了解在系统启动时，如何按照需求灵活地控制各种服务和脚本的启动。

这还意味着在系统重启动时，能够手工启动和停止某些服务。