

第24学时 使用任务计划实现系统管理自动化

在本学时教程中，将完成系统管理技巧的学习。到目前为止，已经学习了很多系统管理员所使用的命令，而且对大部分需要由来完成为自己和其他用户所进行的系统维护任务都应该比较熟悉了。本学时教程将介绍怎样把这些知识融合到一起，目的是能够使用 cron程序包和其他 OpenLinux 操作系统的任务计划程序来自动地完成这些任务。

使用本学时教程介绍的程序和技能，将能够让许多种不同的系统管理任务自动地完成并维护一个健康的、运行良好的系统。我们第一个要讨论的话题就是 cron程序包，接着将学习到在系统上怎样才能为不同的用户管理好 at命令功能。

24.1 使用cron程序包

cron程序包(也就是crond)是在开机启动Linux操作系统之后由系统上的 /etc/rc.d/init.d子目录中的cron.init命令脚本程序启动的一个程序。这一切全都是自动进行的，因此不必操心每次开机启动Linux操作系统之后还要再去启动一个 cron程序包。crond程序包运行在后台，并且不断地检查着几个文件。其中的第一个文件就是子目录 /etc中的crontab文件。下面是这个文件的一部分：

```
...
03 3 1 * * root [ -x /usr/sbin/cronloop ] && /usr/sbin/
➔cronloop Monthly
04 4 * * 6 root [ -x /usr/sbin/cronloop ] && /usr/sbin/
➔cronloop Weekly
05 5 * * * root [ -x /usr/sbin/cronloop ] && /usr/sbin/
➔cronloop Daily
42 * * * * root [ -x /usr/sbin/cronloop ] && /usr/sbin/
➔cronloop Hourly
...
```

正如所看到的，这是一个包括了在每小时、每天、每星期和每月定时执行某些任务的命令清单。cronloop命令是一个命令脚本程序，它会检查 /etc/cron.d子目录，然后根据在 /etc/cron.d子目录中的Monthly、Weekly、Daily以及Hourly子目录中查到的文件运行相应的任务(以shell命令脚本程序的形式)。而真正的 shell命令脚本程序可以在 /etc/cron.d子目录下的lib子目录中找到。

举例来说，如果去查看子目录 Daily中的内容，就会发现一个名字叫做 50update-locatedb的文件，它被链接到/etc/cron.d/lib子目录中的一个叫做update-locatedb的shell命令脚本程序上。而根据前面/etc/crontab文件中的Daily数据项，这个命令脚本程序将会在每天早晨五点五十分为locate命令建立一个新的数据库。如下所示：

```
#!/bin/bash
# Make/update the file name database

. /etc/cron.d/lib/functions

TMPDIR=$(cronloop_mk_TMPDIR)
declare -rx TMPDIR
```

```
# remove TMPDIR on exit
trap cronloop_rm_TMPDIR EXIT

OMIT="/tmp /proc /mnt /var/tmp /var/spool /dev /net /auto /amd /NetWare"

/usr/bin/updatedb --prunepaths="$OMIT" --netuser=nobody 2> /dev/null
exit 0
```

可以看出这个命令脚本程序执行了 updatedb 命令来建立系统的 locate 数据库(请阅读第 4 学时教程“阅读与浏览命令”)。

cron 命令还会在子目录 /var/spool/cron 中按照用户名搜索用户私人的 crontab 文件。这些文件是使用保存在子目录 /usr/bin 中的 crontab 命令创建的, 你的用户们使用它们来定期安排执行他们自己的任务。但是怎样才能告诉 cron 在什么时候去运行这些命令脚本程序呢? 而又是怎样知道 locatedb 命令脚本程序会在每天早上的 5 点 50 分运行的呢? 请继续向下学习 cron 命令的格式。

24.1.1 管理用户的 cron 任务计划

/etc/crontab 文件只是用来计划安排那些系统全局性的任务的。但是, OpenLinux 操作系统也允许用户建立他们自己的 cron 计划安排。作为根操作员, 如果希望用户们能够使用 crontab 命令在子目录 /var/spool/cron 中建立他们自己的 cron 文件, 你首先需要创建两个文件: /etc/cron.allow 和 /etc/cron.deny。在文件 /etc/cron.allow 中, 输入根操作员姓名、根、还有你允许运行 cron 程序包的用户的名字。因为在缺省的 OpenLinux 操作系统的安装中这两个文件都不存在, 所以每个用户都可以存取他个人的 cron 文件。你可以使用 crontab 命令为用户们建立文件, 或者也可以让用户们自己去创建它。



在使用 crontab 命令的时候一定要使用它的 -u 命令行参数。如果以根操作员身份(执行了 su 命令之后)操作时运行了 crontab 这个命令却没有使用 -u 参数, 就会收到一个使用错误信息。

下一小节将介绍 crontab 文件的格式以及在 OpenLinux 系统和普通用户各自使用的 cron 数据项格式之间的区别。

24.1.2 使用 crontab 命令设置任务计划

crontab 数据项的格式在 crontab 命令的使用手册页的第五小节有详细的说明。如果想阅读这个使用手册页, 可以使用下面的命令:

```
# man 5 crontab
```

这个使用手册页给出了 crontab 数据项各个组成部分的详细说明。但是, 为了能够简单地把事情说得更明白一些, 我在这里举几个例子; 并且在本学时教程结束的时候, 还会再教你几个能够用得上的例子。一般说来, 这个数据项的各个组成部分是:

minutes hour day of month month day of week command

数据项之间通常使用空格作为分隔符。但是在 /etc/crontab 文件中, 在每个数据项的 day of week 和 command 两项之间必须插入一个用户名。而用户个人的 crontab 数据项中则不需要用户名。

先举个简单的例子, 如果在一台膝上电脑运行的 OpenLinux, 并且发现每次从节电休眠状态唤醒膝上电脑的时候, OpenLinux 操作系统的系统时间都会变慢; 那么就需要以根操作员的

身份编辑/etc/crontab文件，在其中插入一条命令，每隔一分钟就从膝上电脑的硬件时钟上刷新OpenLinux操作系统的时间。请按照下面的方法在 crontab数据项中使用 hwclock命令，如下所示：

```
* * * * * root /sbin/hwclock --hctosys
```

可以使用的另外一个简单的数据项为 whatis命令建立它的数据库。举例来说，如果知道在每天的早晨不会有几个用户工作在你的系统之上，那么就可以安排 makewhatis命令在凌晨二点执行，使用的设置方法如下面所示：

```
30 2 * * * root /usr/sbin/makewhatis
```

另一方面，作为一个普通的用户，可以让 OpenLinux每隔15分钟向你报一次时，方法是：先使用命令行参数 -e调用 crontab命令，然后在加上下面的内容：

```
0 , 15 , 30 , 45 * * * * /usr/local/bin/saytime %
```



在缺省的情况下，当你使用 crontab命令 -e参数的时候，OpenLinux操作系统会调入 vim编辑器程序运行。请阅读第 18学时教程“个人事务处理工具程序”中的关于如何使用 pico编辑器程序替换缺省的编辑器程序的介绍。

上面的命令告诉 cron程序包每隔十五分钟执行一次 saytime命令进行报时。你可以在下面站点找到 saytime命令：

<http://metalab.unc.edu/pub/Linux/apps/sound/speech/saytime.tgz>

如果想每分钟都能够听得见报时(这样也许会很烦人!)，可以使用：

```
* * * * * /usr/local/bin/saytime %
```

需要注意的是，如果在数据项中某一行的结尾没有使用回车换行，就必须使用一个百分号(%)符号。

下面是更多这类数据项的例子：

```
* 22 * * 1 - 4 /usr/bin/wall 'Time for bed ! Finished your homework yet ? ' %
```

```
0 1 * * * /usr/bin/find/ -xdev -name core -exec /bin/rm {} \; %
```

```
0 , 30 * * * /user/bin/tput bel > /dev/console %
```

```
* 12 25 12 * /bin/echo ' Happy Holidays ' | /bin/mail -s Greetings root %
```

第一个例子是在每个星期一到星期四的晚上十点钟使用 wall命令向所有的用户们广播一条轻柔的备忘录提示信息：数字 22代表按照24小时计算的钟点、数字 1 - 4代表星期一到星期四。第二个例子(可能更适合放在 /etc/crontab文件里)在每天的凌晨一点运行 find命令在系统中搜索 core文件并在找到之后删除它。第三个例子在整点或者整半点的时候 0和30分钟)敲响终端上的报时钟，使用 tput命令进行输出。最后一个例子在十二月二十五日中午向根操作员发出一个邮件消息。



crontab数据项的格式和细节等方面资料我们在第 18学时教程中做了说明。在 crontab使用手册页的第五部分也有关于这些设置的介绍。请阅读它的使用手册页中的其他例子。

使用不同的任务和时间进行练习。使用 OpenLinux，可以让搜索文件、报告在线用户情况

和报告最新登录记录等操作自动地完成。也可以使用 OpenLinux 通过电子邮件把这些信息报告给你。可以使用 cron 安排备份操作在不在场的时候执行，还可以让你的系统在某个预先设定的时间自动关机。

24.2 使用 atrun 命令管理用户的任务计划

cron 程序包非常适用于计划安排那些周期性运行的程序，或者执行周期性的任务，正象在第 18 学时教程中看到的那样。而 at 命令则适用于一次性的或者提示备忘录这类的任务。虽然 Linux 系统在安装后已经自动设置好处理用户的 at 计划安排请求，但是有必要了解如何管理 at 命令的各种功能。

在缺省的情况下，当某个用户使用 at 命令的时候，这个命令是以缺省的 CPU 优先权运行的。如果有过多的用户在后台启动运行具有比较高的优先权的任务或者大量占用 CPU 的任务，你的系统性能就会受到影响。下面将介绍怎样才能对系统中的 at 命令的功能进行管理，从而最大限度地提供最佳的性能和控制。

系统上的 at 命令的功能是由 cron 程序包激活的，而 cron 程序包在开机启动 Linux 的时候调入运行之后，就会去查找 /etc/crontab 文件中有没有下面这样的数据项：

```
...
# Run any at jobs every minute
***** root [ -x /usr/sbin/atrun ] && /usr/sbin/atrun
...
```

正如所看到的，这就是为什么 at 命令依赖于 cron 命令的原因：cron 命令在系统启动后每分钟执行一次 atrun 命令(你可以在子目录 /usr/sbin 找到这个命令)。atrun 命令则在子目录 /var/spool/atjobs 中依次寻找有没有任务需要执行。举例来说，如果根操作员在当天(下午 2:30)建立了一个在 14:46 分的时候运行的任务，就会看到下面的内容：

```
# at 14:46
at> echo Hello
at> <EOT>
warning: commands will be executed using /bin/sh
job 11 at 1998-12-14 14:46

# ls -l /var/spool/atjobs
total 1
-rwx----- 1 root    root      842 Dec 14 14:43 a0000b00e85b62
```

如果你想建立一个 at 命令，需要输入单词 at，后面再跟上一个时间。当按下回车键之后，at > 提示符就会出现。接着再输入一个命令并按下回车键。如果你想退出输入命令状态并保存任务，请按下 Ctrl + D 组合键。at 命令就会告诉这个任务的编号和将要执行的时间。正如在 atjobs 子目录中所看到的，输入这个命令之后，确实有一个任务在等着根操作员。



可以使用 atq 命令获得更多关于等待中的 atjobs 的特定信息。如果你是根操作员，将会看到用户们已经安排的所有任务计划，而不仅仅是你自己安排的任务计划。

控制 batch 和 at 命令

现在已经知道 at 命令的功能是如何实现的了。那么在系统上怎样才能控制 at 命令的运行，

又为了哪些人去进行控制呢？在 OpenLinux 操作系统中控制 at 命令如何运行的方法之一是使用 atrun 命令的 -l (负载平均值) 参数。这个参数将控制由用户使用 batch 命令 (在第 18 学时中与 at 命令一起介绍过) 提交的所有任务。通过设置一个小于 1.5 (缺省值) 的数值你就可以限制在什么时候运行 batch 任务, 这个数值告诉 atrun 只有当系统的负载平均值比较低的时候才运行 batch 任务。这个负载平均值是系统运行当中由 /proc/loadavg 文件中的一个数值确定的。

可以使用下面的命令看到当前负载平均值：

```
# cat /proc/loadavg
0.20 0.11 0.03 2/50 1228
```

上面给出了最近 5 分钟、10 分钟和 15 分钟中的负载平均值。也可以使用 uptime 命令获得这个负载平均值，如下所示：

```
# uptime
7:40 pm up 2:44, 3 users, load average: 0.13, 0.08, 0.02
```

如果你想在所有的时间里都允许用户使用 batch 命令，就需要把系统的 /etc/crontab 文件中的 atrun 命令数据项中的 -l 参数的值改为大于缺省值 1.5 的数值。

但是你又怎样才能控制在系统上是否允许用户使用 at 命令呢？在缺省的情况下，当你安装了 OpenLinux 操作系统之后，在系统上的任何人都能够使用 at 命令。有四种方法可以控制允许什么人可以使用 at 命令。请查看 /etc 子目录，找一个叫做 at.deny 的文件。

基于安全方面的考虑，这个 at.deny 文件中包含着一个不允许使用 at 命令的 OpenLinux 操作系统用户的清单，其中还包括了某些程序包或者其他的进程。而没有出现在清单中的其他人就可以使用 at 命令来安排计划任务。如果想限制某个用户使用 at 命令，把这个用户的用户名放入 /etc/at.deny 文件就可以了。如果不想让系统上的任何人使用 at 命令 (当然这对你不适用，因为你是根操作员)，可以在子目录 /etc 中建立一个空的 at.allow 文件，也可以从 /etc 子目录中删除 at.deny 文件。如果只想允许某些特定的用户使用 at 命令，可以在 /etc 子目录中建立一个叫做 at.allow 文件，并把那些用户的用户名放入这个文件。如果 at.allow 和 at.deny 文件在子目录 /etc 中都被删除了，那么就只有根操作员可以使用 at 命令了。

做为根操作员，控制 at 任务的最后一招是：使用 atq 命令列出系统上所有已经安排的 at 任务，如果看到已经安排了太多的 at 任务，就可以使用 atrm 命令把它们删除掉，如下所示：

```
# atq
Date                Owner    Queue    Job#
20:00:00 11/26/97   bball    c         12
20:10:00 11/26/97   bball    c         13
20:15:00 11/26/97   bball    c         14
20:30:00 11/26/97   bball    c         15
...
# atrm 12 13 14 15 ...
```

这里，假设你看到了一个很长很长的任务号清单，并删除了它们。对热情的用户来说，这么做是一种比较粗暴无礼的管理办法。一个比较好的办法是应该先向用户们发个电子邮件，询问他们正在运行的任务中有没有可以改在非工作时间或者非用机高峰时间自动运行的。

做为系统管理员，在为 OpenLinux 系统上的用户们安排各种命令的执行方面，一切都在你的掌握之中。请考虑一下那些需要安排在每小时、每天、每周、以及每月执行的任务吧。只需要一点点想象力，就能够使任何一个定制的工具程序、命令行以及各种用来帮助你管理系统的报告程序等等都可以自动地运行。你得到的结局将会是更多的磁盘空间、更好运行着的

系统还有更开心的用户。

24.3 课时小结

本学时教程介绍了一个要求所有的系统管理员掌握的有价值的技能——在任何日期和时间对任务计划进行安排并自动运行复杂的任务的能力。这是成长为一个成功的系统管理员的秘密之一——从最初开始就花费时间为进行系统维护精雕细琢一个能够反反复复地运用的通用方法或者工具。

24.4 专家答疑

问：我希望我的计算机能够在每天早晨自动地连接到因特网，然后再下载当天的新闻并把这些新闻通过电子邮件发送给我。这能不能做到？

答：能。在系统的 crontab 文件中加上一个命令脚本程序的名称，使用这个命令脚本程序初始化网络连接、暂时休眠、唤醒然后再下载并电邮这些新闻。举例来说，lynx 浏览器程序的 dump 参数可以象下面这样与 mail 命令使用在一起：

```
lynx http://some.newsource.com/news.html -dump | mail -s "news" root
```

24.5 练习题

1. 建立个人的 crontab 设置，在某个重要的日子，比如周年纪念或者生日来临几天之前提醒你。你认为让 OpenLinux 通知你的最好的办法是什么？

2. 列出一个某些需要自动进行处理的重要任务清单。你怎样才能做到在一点？你会在什么时间执行这些任务？

CD - ROM光盘上的内容

书后所附的CD-ROM光盘上包含有Caldera公司的OpenLinux 2.2，它可以从Caldera公司的FTP站点上下载。CD-ROM光盘上还包括下面所列的软件：

- KDE 1.1
- WordPerfect 8
- Netscape Communicator 4.51
- LIZARD，图形化的安装向导

下面是也包括在本书所附的CD-ROM光盘上的商业版的升级软件：

- PartitionMagic，Caldera Edition
- BootMagic，Caldera Edition

下面的软件包括在Caldera公司商业版本的OpenLinux 2.2软件包中：

- StarOffice 5.0
- BRU