**cron调度进程**

**c r o n是系统主要的调度进程，可以在无需人工干预的情况下运行作业。有一个叫做 c r o n t a b的命令允许用户提交、编辑或删除相应的作业。每一个用户都可以有一个c r o n t a b文件来保存调度信息。可以使用它运行任意一个s h e l l脚本或某个命令，每小时运行一次，或一周三次，这完全取决于你。每一个用户都可以有自己的c r o nt a b文件，但在一个较大的系统中， 系统管理员一般会禁止这些文件，而只在整个系统保留一个这样的文件。系统管理员是通过 c r o n . d e n y和c r o n . a l l o w这两个文件来禁止或允许用户拥有自己的c r o n t a b文件。**

**启动定时服务**

**[root@]# service crond stop**

**[root@]# service crond start**

**1.1 crontab的域**

**为了能够在特定的时间运行作业，需要了解c r o n t a b文件每个条目中各个域的意义和格式。**

**下面就是这些域：(6个)**

**第1列分钟1～5 9**

**第2列小时1～2 3（0表示子夜）**

**第3列日1～3 1**

**第4列月1～1 2**

**第5列星期0～6（0表示星期天）**

**第6列要运行的命令**

**crontab的范例格式：**

**下面是c r o n t a b的格式：**

**分< >时< >日< >月< >星期< >要运行的命令**

**其中< >表示空格。**

**C r o n t a b文件的一个条目是从左边读起的，第一列是分，最后一列是要运行的命令，它位于星期的后面。在这些域中，可以用横杠-来表示一个时间范围，例如你希望星期一至星期五运行某个作业，那么可以在星期域使用1 - 5来表示。还可以在这些域中使用逗号“,”，例如你希望星期一 和星期四运行某个作业，只需要使用1 , 4来表示。可以用星号\*来表示连续的时间段。如果你对某个表示时间的域没有特别的限定，也应该在该域填入\*。**

**该文件的每一个条目必须含有5 个时间域，而且每个域之间要用空格分隔。该文件中所有的注释行要在行首用#来表示。**

**crontab条目举例**

**这里有c r o n t a b文件条目的一些例子：**

**10 \* \* \* \* /home/crontab/bin/logininfo.sh**

**上面是每小时10分的时候执行**

**30 21\* \* \* /apps/bin/cleanup.sh**

**上面的例子表示每晚的2 1 : 3 0运行/ a p p s / b i n目录下的c l e a n u p . s h。**

**45 4 1,10,22 \* \* /apps/bin/backup.sh**

**上面的例子表示每月1、1 0、2 2日的4 : 4 5运行/ a p p s**

**/ b i n目录下的b a c k u p . s h。**

**10 1 \* \* 6,0 /bin/find -name "core" -exec rm {} ;**

**上面的例子表示每周六、周日的1 : 1 0运行一个f i n d命令。**

**0,30 18-23 \* \* \* /apps/bin/dbcheck.sh**

**上面的例子表示在每天1 8 : 0 0至2 3 : 0 0之间每隔3 0分钟运行/ a p p s / b i n目录下的d b c h e c k . s h。**

**0 23 \* \* 6 /apps/bin/qtrend.sh**

**上面的例子表示每星期六的11 : 0 0 p m运行/ a p p s / bi n目录下的q t r e n d . s h。**

**你可能已经注意到上面的例子中，每个命令都给出了绝对路径。当使用c r o n t a b运行s h e l l 脚本时，要由用户来给出脚本的绝对路径，设置相应的环境变量。记住，既然是用户向c r o n 提交了这些作业，就要向c r o n提供所需的全部环境。不要假定c r o n知道所需要的特殊环境，它其实并不知道。所以你要保证在s h e l l脚本中提供所有必**

**要的路径和环境变量，除了一些自动设置的全局变量。**

**命令形式：**

**c r o n t a b命令的一般形式为：**

**Crontab [-u user] -e -l -r**

**其中：**

**-u 用户名。**

**-e 编辑c r o n t a b文件。**

**-l 列出c r o n t a b文件中的内容。**

**-r 删除c r o n t a b文件。**

**如果使用自己的名字登录，就不用使用- u选项，因为在执行c r o n t a b命令时，该命令能够知道当前的用户建一个新的crontab文件在考虑向c r o n进程提交一个c r o n t a b文件之前，首先要做的一件事情就是设置环境变量E D I TO R。c r o n进程根据它来确定使用哪个编辑器编辑c r o n t a b文件。9 9 %的U N I X和L I N U X用 户都使用v i，如果你也是这样，那么你就编辑$ H O M E目录下的. p r o f i l e文件，在其中加入这样 一行：**

**EDITOR=vi; export EDITOR**

**然后保存并退出。**

**不妨创建一个名为< u s e r > c r o n的文件，其中< u s e r >是用户名，例如， j a m e s c r o n。该文件中加入如下的内容。**

**＃。。。。这里是解释的话**

**0，15，30，45 18－06 \* \* \* /bin/echo**

**'time'>/dev/console**

**保存并退出。确信前面5个域用空格分隔。**

**在上面的例子中，系统将每隔1 5分钟向控制台输出一次当前时间。如果系统崩溃或挂起，**

**从最后所显示的时间就可以一眼看出系统是什么时间停止工作的。在有些系统中，用t t y 1来表 示控制台，可以根据实际情况对上面的例子进行相应的修改。**

**为了提交你刚刚创建的c r o n t a b文件，可以把这个新创**

**建的文件作为c r o n命令的参数：**

**$ crontab jamescron**

**现在该文件已经提交给c r o n进程，它将每隔1 5分钟运行**

**一次。**

**同时，新创建文件的一个副本已经被放在/ v a r / s p o o**

**l / c r o n目录中，文件名就是用户名（即，**

**james）。**

**列出crontab文件：**

**为了列出c r o n t a b文件，可以用：**

**你将会看到和上面类似的内容。可以使用这种方法在$ H O M**

**E目录中对c r o n t a b文件做一备**

**份：**

**$ crontab -l > $HOME/mycron**

**这样，一旦不小心误删了c r o n t a b文件，可以用上一节**

**所讲述的方法迅速恢复。**

**编辑crontab文件：**

**如果希望添加、删除或编辑c r o n t a b文件中的条目，而**

**E D I TO R环境变量又设置为v i，那么**

**就可以用v i来编辑c r o n t a b文件，相应的命令为：**

**$ crontab -e**

**可以像使用v i编辑其他任何文件那样修改c r o n t a b文**

**件并退出。如果修改了某些条目或添加了新的条目，那么在**

**保存该文件时， c r o n会对其进行必要的完整性检查。如**

**果其中的某个域出现了超出允许范围的值，它会提示你。**

**保存并退出。最好在c r o n t a b文件的每一个条目之上加**

**入一条注释，这样就可以知道**

**它的功能、运行时间，更为重要的是，知道这是哪位用户的**

**作业。**

**可以使用前面讲过的crontab -l命令列出它的全部信息**

**为了删除c r o n t a b文件，可以用：**

**$ crontab -r**