crontab 定时任务

参考链接：<http://www.cnblogs.com/peida/archive/2013/01/08/2850483.html>

目录

[简介 1](#_Toc499113555)

[crond服务（安装，操作） 2](#_Toc499113556)

[crontab命令详解 3](#_Toc499113557)

[crontab文件的含义 3](#_Toc499113558)

[常用方法 4](#_Toc499113559)

[创建crontab文件 4](#_Toc499113560)

[恢复crontab文件 5](#_Toc499113561)

[使用实例 5](#_Toc499113562)

[注意问题 5](#_Toc499113563)

[环境变量 5](#_Toc499113564)

[百分号特殊意义 6](#_Toc499113565)

[防止误删crontab文件 6](#_Toc499113566)

[清理系统用户的邮件日志 6](#_Toc499113567)

# 简介

linux系统由 cron (crond) 这个系统服务来控制的循环运行的例行性计划任务， Linux 系统上面原本就有非常多的计划性工作，因此这个系统服务是默认启动的。另外, 由于使用者自己也可以设置计划任务，所以， Linux 系统也提供了使用者控制计划任务的命令 :crontab 命令，通过crontab 命令，我们可以在固定的间隔时间执行指定的系统指令或 shell script脚本。时间间隔的单位可以是分钟、小时、日、月、周及以上的任意组合。这个命令非常设合周期性的日志分析或数据备份等工作。

Linux下的任务调度分为两类，系统任务调度和用户任务调度。

①系统任务调度：系统周期性所要执行的工作，比如写缓存数据到硬盘、日志清理等。在/etc目录下有一个crontab文件，这个就是系统任务调度的配置文件。

②用户任务调度：用户定期要执行的工作，比如用户数据备份、定时邮件提醒等。用户可以使用 crontab 工具来定制自己的计划任务。所有用户定义的crontab 文件都被保存在 /var/spool/cron目录中。其文件名与用户名一致。

需要注意的是，如果要定义一个定时重启系统的任务，就必须将任务放到/etc/crontab文件，即使在root用户下创建一个定时重启系统的任务也是无效的。

# crond服务（安装，操作）

|  |  |
| --- | --- |
| linux | |
| 操作 | /sbin/service crond start //启动服务  /sbin/service crond stop //关闭服务  /sbin/service crond restart //重启服务  /sbin/service crond reload   //重新载入配置 /sbin/service crond status   //查看服务状态 |
| 开机启动 | 查看crontab服务是否已设置为开机启动：ntsysv  加入开机自动启动：chkconfig –level 35 crond on |

|  |  |
| --- | --- |
| ubuntu | |
| 操作 | $sudo /etc/init.d/cron(crond) start //启动服务  $sudo /etc/init.d/cron(crond) stop //关闭服务  $sudo /etc/init.d/cron(crond) restart //重启服务  $sudo /etc/init.d/cron(crond) reload //重新载入配置  ps aux | grep cron //查看cron是否已启动 |

|  |  |
| --- | --- |
| mac | |
| 操作 | $sudo /usr/sbin/cron start //启动服务  $sudo /usr/sbin/cron stop //关闭服务  $sudo /usr/sbin/cron restart //重启服务 |

mac 下的crontab使用见：<https://www.clarencep.com/2015/12/16/how-to-enable-crontab-on-osx/>

mac 定时任务：

<http://www.jianshu.com/p/4addd9b455f2>

<https://alvinalexander.com/mac-os-x/mac-osx-startup-crontab-launchd-jobs>

因为mac有自己的定时服务，并且可以定时到秒，同时支持Xcode界面操作。

# crontab命令详解

1. 命令格式：

crontab [ -u user ] file

crontab [ -u user ] [  -e  |  -l  |  -r  ]

② 命令参数：

|  |  |
| --- | --- |
| -u user | 用来设定某个用户的crontab服务，例如，“-u sophia”表示设定sophia用户的crontab服务，此参数一般有root用户来运行。 |
| file | file是命令文件的名字,表示将file做为crontab的任务列表文件并载入crontab。如果在命令行中没有指定这个文件，crontab命令将接受标准输入（键盘）上键入的命令，并将它们载入crontab。（是覆盖原有的，不是追加） |
| -e | 编辑某个用户的crontab文件内容。如果不指定用户，则表示编辑当前用户的crontab文件。 |
| -l | 显示某个用户的crontab文件内容，如果不指定用户，则表示显示当前用户的crontab文件内容。 |
| -r | 从/var/spool/cron目录中删除某个用户的crontab文件，如果不指定用户，则默认删除当前用户的crontab文件。 |
| -i | 在删除用户的crontab文件时给确认提示。 |

# crontab文件的含义

用户所建立的crontab文件中，每一行都代表一项任务，每行的每个字段代表一项设置，它的格式共分为六个字段，前五段是时间设定段，第六段是要执行的命令段，格式如下：

minute   hour   day   month   week   command



在以上各个字段中，还可以使用以下特殊字符：

|  |  |
| --- | --- |
| 星号（\*） | 代表所有可能的值，例如month字段如果是星号，则表示在满足其它字段的制约条件后每月都执行该命令操作。 |
| 逗号（,） | 可以用逗号隔开的值指定一个列表范围，例如，“1,2,5,7,8,9” |
| 中杠（-） | 可以用整数之间的中杠表示一个整数范围，例如“2-6”表示“2,3,4,5,6” |
| 正斜线（/） | 可以用正斜线指定时间的间隔频率，例如“0-23/2”表示每两小时执行一次。同时正斜线可以和星号一起使用，例如\*/10，如果用在minute字段，表示每十分钟执行一次。 |

# 常用方法

创建crontab文件

①在向cron进程提交一个crontab文件之前，首先要设置环境变量EDITOR。cron进程根据它来确定使用哪个编辑器编辑crontab文件。9 9 %的UNIX和LINUX用户都使用vi，编辑$ HOME目录下的. profile文件，在其中加入这样一行：

EDITOR=vi; export EDITOR

②然后保存并退出。不妨创建一个名为<user> cron的文件，其中<user>是用户名，例如， davecron。在该文件中加入如下的内容。

 #echo the date to the console every

      # 15minutes between 6pm and 6am

0,15,30,45 18-06 \* \* \* /bin/echo 'date' > /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。

③为了提交刚刚创建的crontab文件，可以把这个新创建的文件作为cron命令的参数：

     $ crontab davecron

现在该文件已经提交给cron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/var/spool/cron目录中，文件名就是用户名(即dave)。

恢复crontab文件

如果不小心误删了crontab文件，假设你在自己的$ H O M E目录下还有一个备份，那么可以将其拷贝到/var/spool/cron/<username>，其中<username>是用户名。如果由于权限问题无法完成拷贝，可以用：

     $ crontab <filename>

其中，<filename>是你在$ H O M E目录中副本的文件名。我建议你在自己的$ H O M E目录中保存一个该文件的副本。

# 使用实例

|  |  |
| --- | --- |
| \* \* \* \* \* myCommand | 每1分钟执行一次myCommand |
| 3,15 \* \* \* \* myCommand | 每小时的第3和第15分钟执行 |
| 3,15 8-11 \* \* \* myCommand | 每天的上午8点到11点的第3和第15分钟执行 |
| 3,15 8-11 \*/2 \* \* myCommand | 每隔两天的上午8点到11点的第3和第15分钟执行 |

# 注意问题

## 环境变量

有时我们创建了一个crontab，但是这个任务却无法自动执行，而手动执行这个任务却没有问题，这种情况一般是由于在crontab文件中没有配置环境变量引起的。

因为我们手动执行某个任务时，是在当前shell环境下进行的，程序当然能找到环境变量，而系统自动执行任务调度时，是不会加载任何环境变量的。因此，就需要在crontab文件中指定任务运行所需的所有环境变量。所以注意如下3点：

1. 脚本中涉及文件路径时写全局路径；
2. 脚本执行要用到java或其他环境变量时，通过source命令引入环境变量，如:

cat start\_cbp.sh #查看某个文件

!/bin/sh #定义执行脚本

source /etc/profile #

export RUN\_CONF=/home/d139/conf/platform/cbp/cbp\_jboss.conf

/usr/local/jboss-4.0.5/bin/run.sh -c mev &

1. 当手动执行脚本OK，但是crontab死活不执行时,可尝试在crontab中直接引入环境变量解决问题。如:

0 \* \* \* \* . /etc/profile;

/bin/sh /var/www/java/audit\_no\_count/bin/restart\_audit.sh

## 百分号特殊意义

在crontab中%表示换行的意思。如果要用的话必须进行转义\%，如经常用的date ‘+%Y%m%d’在crontab里是不会执行的，应该换成date ‘+\%Y\%m\%d’

防止误删crontab文件

我就有过类似的经历，有数次误删了crontab文件（因为r键紧挨在e键的右边）。这就是为什么有些系统文档建议不要直接编辑crontab文件，而是编辑该文件的一个副本，然后重新提交新的文件。

在使用crontab命令时要格外小心。如果遗漏了任何选项，crontab可能会打开一个空文件，或者看起来像是个空文件。这时敲delete键退出，不要按<Ctrl-D>，否则你将丢失crontab文件。

## 清理系统用户的邮件日志

每条任务调度执行完毕，系统都会将任务输出信息通过电子邮件的形式发送给当前系统用户，这样日积月累，日志信息会非常大，可能会影响系统的正常运行。可以在crontab文件中设置如下形式，将输出（标准错误和标准输出）重定向到一个不存在的文件/dev/null，忽略日志输出，：

0 \*/3 \* \* \* /usr/local/apache2/apachectl restart >/dev/null 2>&1

“/dev/null 2>&1”表示先将标准输出重定向到/dev/null，然后将标准错误重定向到标准输出，由于标准输出已经重定向到了/dev/null，因此标准错误也会重定向到/dev/null，这样日志输出问题就解决了。