[**每天一个linux命令（49）：at命令**](http://www.cnblogs.com/peida/archive/2013/01/05/2846152.html)

在windows系统中，windows提供了计划任务这一功能，在控制面板 -> 性能与维护 -> 任务计划， 它的功能就是安排自动运行的任务。 通过'添加任务计划'的一步步引导，则可建立一个定时执行的任务。

在linux系统中你可能已经发现了为什么系统常常会自动的进行一些任务？这些任务到底是谁在支配他们工作的？在linux系统如果你想要让自己设计的备份程序可以自动在某个时间点开始在系统底下运行，而不需要手动来启动它，又该如何处置呢？ 这些例行的工作可能又分为一次性定时工作与循环定时工作，在系统内又是哪些服务在负责？ 还有，如果你想要每年在老婆的生日前一天就发出一封信件提醒自己不要忘记，linux系统下该怎么做呢？

今天我们主要学习一下一次性定时计划任务的at命令的用法！

**1．命令格式：**

at[参数][时间]

**2．命令功能：**

在一个指定的时间执行一个指定任务，只能执行一次，且需要开启atd进程（

ps -ef | grep atd查看， 开启用/etc/init.d/atd start or restart； 开机即启动则需要运行 chkconfig --level 2345 atd on）。

**3．命令参数：**

-m 当指定的任务被完成之后，将给用户发送邮件，即使没有标准输出

-I atq的别名

-d atrm的别名

-v 显示任务将被执行的时间

-c 打印任务的内容到标准输出

-V 显示版本信息

-q<列队> 使用指定的列队

-f<文件> 从指定文件读入任务而不是从标准输入读入

-t<时间参数> 以时间参数的形式提交要运行的任务

at允许使用一套相当复杂的指定时间的方法。他能够接受在当天的hh:mm（小时:分钟）式的时间指定。假如该时间已过去，那么就放在第二天执行。当然也能够使用midnight（深夜），noon（中午），teatime（饮茶时间，一般是下午4点）等比较模糊的 词语来指定时间。用户还能够采用12小时计时制，即在时间后面加上AM（上午）或PM（下午）来说明是上午还是下午。 也能够指定命令执行的具体日期，指定格式为month day（月 日）或mm/dd/yy（月/日/年）或dd.mm.yy（日.月.年）。指定的日期必须跟在指定时间的后面。 上面介绍的都是绝对计时法，其实还能够使用相对计时法，这对于安排不久就要执行的命令是很有好处的。指定格式为：now + count time-units ，now就是当前时间，time-units是时间单位，这里能够是minutes（分钟）、hours（小时）、days（天）、weeks（星期）。count是时间的数量，究竟是几天，还是几小时，等等。 更有一种计时方法就是直接使用today（今天）、tomorrow（明天）来指定完成命令的时间。

TIME：时间格式，这里可以定义出什么时候要进行 at 这项任务的时间，格式有：

HH:MM

ex> 04:00

在今日的 HH:MM 时刻进行，若该时刻已超过，则明天的 HH:MM 进行此任务。

HH:MM YYYY-MM-DD

ex> 04:00 2009-03-17

强制规定在某年某月的某一天的特殊时刻进行该项任务

HH:MM[am|pm] [Month] [Date]

ex> 04pm March 17

也是一样，强制在某年某月某日的某时刻进行该项任务

HH:MM[am|pm] + number [minutes|hours|days|weeks]

ex> now + 5 minutes

ex> 04pm + 3 days

就是说，在某个时间点再加几个时间后才进行该项任务。

**4．使用实例：**

**实例1：三天后的下午 5 点锺执行 /bin/ls**

**命令：**

at 5pm+3 days

**输出：**

[root@localhost ~]# at 5pm+3 days

at> /bin/ls

at> <EOT>

job 7 at 2013-01-08 17:00

[root@localhost ~]#

**说明：**

**实例2：明天17点钟，输出时间到指定文件内**

**命令：**

at 17:20 tomorrow

**输出：**

[root@localhost ~]# at 17:20 tomorrow

at> date >/root/2013.log

at> <EOT>

job 8 at 2013-01-06 17:20

[root@localhost ~]#

**说明：**

**实例3：计划任务设定后，在没有执行之前我们可以用atq命令来查看系统没有执行工作任务**

**命令：**

atq

**输出：**

[root@localhost ~]# atq

8       2013-01-06 17:20 a root

7       2013-01-08 17:00 a root

[root@localhost ~]#

**说明：**

**实例4：删除已经设置的任务**

**命令：**

atrm 7

**输出：**

[root@localhost ~]# atq

8       2013-01-06 17:20 a root

7       2013-01-08 17:00 a root

[root@localhost ~]# atrm 7

[root@localhost ~]# atq

8       2013-01-06 17:20 a root

[root@localhost ~]#

**说明：**

**实例5：显示已经设置的任务内容**

**命令：**

at -c 8

**输出：**

[root@localhost ~]# at -c 8

#!/bin/sh

# atrun uid=0 gid=0

# mail     root 0

umask 22此处省略n个字符

date >/root/2013.log

[root@localhost ~]#

**说明：**

**实例6：**

**命令：**

**输出：**

**说明：**

**5．atd 的启动与 at 运行的方式：**

**5.1 atd 的启动**

要使用一次性计划任务时，我们的 Linux 系统上面必须要有负责这个计划任务的服务，那就是 atd 服务。 不过并非所有的 Linux distributions 都默认会把他打开的，所以，某些时刻我们需要手动将atd 服务激活才行。 激活的方法很简单，就是这样：

**命令：**

/etc/init.d/atd start

/etc/init.d/atd restart

**输出：**

[root@localhost /]# /etc/init.d/atd start

[root@localhost /]# /etc/init.d/atd

用法：/etc/init.d/atd {start|stop|restart|condrestart|status}

[root@localhost /]# /etc/init.d/atd stop

停止 atd：[确定]

[root@localhost /]# ps -ef|grep atd

root     25062 24951  0 14:53 pts/0    00:00:00 grep atd

[root@localhost /]# /etc/init.d/atd start

[确定]td：[确定]

[root@localhost /]# ps -ef|grep atd

root     25068     1  0 14:53 ?        00:00:00 /usr/sbin/atd

root     25071 24951  0 14:53 pts/0    00:00:00 grep atd

[root@localhost /]# /etc/init.d/atd restart

停止 atd：[确定]

[确定]td：[确定]

[root@localhost /]#

说明：

/etc/init.d/atd start 没有启动的时候，直接启动atd服务

/etc/init.d/atd restart 服务已经启动后，重启 atd 服务

**备注：配置一下启动时就启动这个服务，免得每次重新启动都得再来一次**

命令：

chkconfig atd on

输出：

[root@localhost /]# chkconfig atd on

[root@localhost /]#

**5.2 at 的运行方式**

既然是计划任务，那么应该会有任务执行的方式，并且将这些任务排进行程表中。那么产生计划任务的方式是怎么进行的? 事实上，我们使用 at 这个命令来产生所要运行的计划任务，并将这个计划任务以文字档的方式写入 /var/spool/at/ 目录内，该工作便能等待 atd 这个服务的取用与运行了。就这么简单。

不过，并不是所有的人都可以进行 at 计划任务。为什么? 因为系统安全的原因。很多主机被所谓的攻击破解后，最常发现的就是他们的系统当中多了很多的黑客程序， 这些程序非常可能运用一些计划任务来运行或搜集你的系统运行信息,并定时的发送给黑客。 所以，除非是你认可的帐号，否则先不要让他们使用 at 命令。那怎么达到使用 at 的可控呢?

我们可以利用 /etc/at.allow 与 /etc/at.deny 这两个文件来进行 at 的使用限制。加上这两个文件后， at 的工作情况是这样的：

先找寻 /etc/at.allow 这个文件，写在这个文件中的使用者才能使用 at ，没有在这个文件中的使用者则不能使用 at (即使没有写在 at.deny 当中);

如果 /etc/at.allow 不存在，就寻找 /etc/at.deny 这个文件，若写在这个 at.deny 的使用者则不能使用 at ，而没有在这个 at.deny 文件中的使用者，就可以使用 at 命令了。

如果两个文件都不存在，那么只有 root 可以使用 at 这个命令。

透过这个说明，我们知道 /etc/at.allow 是管理较为严格的方式，而 /etc/at.deny 则较为松散 (因为帐号没有在该文件中，就能够运行 at 了)。在一般的 distributions 当中，由于假设系统上的所有用户都是可信任的， 因此系统通常会保留一个空的 /etc/at.deny 文件，意思是允许所有人使用 at 命令的意思 (您可以自行检查一下该文件)。 不过，万一你不希望有某些使用者使用 at 的话，将那个使用者的帐号写入 /etc/at.deny 即可！ 一个帐号写一行。