

# 例行性工作排程

教师: 李军辉

发现两种工作排程的方式:

- 1 一种是例行性的，就是每隔一定的周期要来办的事项；
- 2 一种是突发性的，就是这次做完以后就没有的那一种。

发现两种工作排程的方式:

- 1 一种是例行性的，就是每隔一定的周期要来办的事项；
- 2 一种是突发性的，就是这次做完以后就没有的那一种。

Linux 如何实现工作排程:

- at: at 是个可以处理仅运行一次就结束排程的命令，不过要运行 at 时，必须要有 atd 这个服务的支持才行。
- crontab: crontab 这个命令所配置的工作将会循环的一直进行下去！可循环的时间为分钟、小时、每周、每月或每年等。

# Linux 上常见的例行性工作

- 1 进行登录档的轮替 (log rotate)
- 2 登录档分析 logwatch 的任务
- 3 创建 locate 的数据库
- 4 whatis 数据库的创建
- 5 RPM 软件登录档的创建
- 6 移除缓存档
- 7 与网络服务有关的分析行为

# 仅运行一次的工作排程

要使用单一工作排程时，我们的 Linux 系统上面必须要有负责这个排程的服务，那就是 atd.

```
/etc/init.d/atd restart
```

注意: at命令默认是未安装, 故不存在/etc/init.d/atd, 运行sudo apt-get install at安装.

哪些用户可以运行at命令，进行一次工作排程？

- /etc/at.allow
- /etc/at.deny
- 如果没有以上两个文件,表明只有root可以

at [-mldv] TIME

选项与参数：

- m：当 at 的工作完成后，即使没有输出信息，亦以 email 通知使用者该工作已完成。
- l：at -l 相当於 atq，列出目前系统上面的所有该使用者的 at 排程；
- d：at -d 相当於 atrm，可以取消一个在 at 排程中的工作；
- v：可以使用较明显的时间格式列出 at 排程中的工作列表；
- c：可以列出后面接的该项工作的实际命令内容。

at [-mldv] TIME

选项与参数：

TIME：时间格式，这里可以定义出『什么时候要进行 at 这项工作』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 进行此工作。

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工作！

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！

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。



范例一：再过五分钟后，将 /root/.bashrc 寄给 root 自己

```
at now + 5 minutes
```

```
at> /bin/mail root -s "testing at job" < /root/.bashrc
```

```
at> <EOT> <==这里输入 [ctrl] + d 就会出现 <EOF> 的字样！代表  
结束！
```

范例二：将上述的第 4 (?) 项工作内容列出来查阅

```
at -c 4
```

范例三：由於机房预计於 2017/12/05 停电，我想要在 2017/12/04 23:00 关机？

```
at 23:00 2009-03-17
```

```
at> /bin/sync      (刷新文件系统缓冲区)
```

```
at> /bin/sync
```

```
at> /sbin/shutdown -h now
```

```
at> <EOT>
```

查看有多少个at工作排程

- atq

取消某个工作排程

- atrm 1

范例三：由於机房预计於 2017/12/05 停电，我想要在 2017/12/04 23:00 关机？

```
batch 23:00 2009-03-17
```

```
at> /bin/sync      (刷新文件系统缓冲区)
```

```
at> /bin/sync
```

```
at> /sbin/shutdown -h now
```

```
at> <EOT>
```

batch: 系统有空时才进行后台任务.

它会在CPU工作负载小于0.8的时候, 才执行任务.

# 例行性工作排程

循环运行的例行性工作排程则是由 cron (crond) 这个系统服务来控制的。

```
/etc/init.d/crond restart
```

哪些用户可以运行crontab命令，进行一次工作排程？

- /etc/cron.allow
- /etc/cron.deny
- 如果没有以上两个文件, 表明只有root可以

```
crontab [-u username] [-l|-e|-r]
```

选项与参数：

-u：只有 root 才能进行这个任务，亦即帮其他使用者创建/移除 crontab 工作排程；

-e：编辑 crontab 的工作内容

-l：查阅 crontab 的工作内容

-r：移除所有的 crontab 的工作内容，若仅要移除一项，请用 -e 去编辑。



范例一：用 dmtsai 的身份在每天的 12:00 发信给自己

```
crontab -e
```

# 此时会进入 vi 的编辑画面让您编辑工作！注意到，每项工作都是一行。

```
0 12 * * * mail dmtsai -s "at 12:00" < /home/dmtsai/.bashrc
```

```
#分 时 日 月 周 |<=====命令串=====>|
```

代表意义	分钟	小时	日期	月份	周	命令
数字范围	0-59	0-23	1-31	1-12	0-7	呀就命令啊

特殊字符	代表意义
*(星号)	代表任何时刻都接受的意思！举例来说，范例一内那个日、月、周都是*，就代表著『不论何月、何的礼拜几的 12:00 都运行后续命令』的意思！
,(逗号)	<p>代表分隔时段的意思。举例来说，如果要下达的工作是 3:00 与 6:00 时，就会是：</p> <p><code>0 3,6 * * * command</code></p> <p>时间参数还是有五栏，不过第二栏是 3,6，代表 3 与 6 都适用！</p>
-(减号)	<p>代表一段时间范围内，举例来说，8 点到 12 点之间的每小时的 20 分都进行一项工作：</p> <p><code>20 8-12 * * * command</code></p> <p>仔细看到第二栏变成 8-12 喔！代表 8,9,10,11,12 都适用的意思！</p>
/n(斜线)	<p>那个 n 代表数字，亦即是『每隔 n 单位间隔』的意思，例如每五分钟进行一次，则：</p> <p><code>* /5 * * * * command</code></p> <p>很简单吧！用 * 与 /5 来搭配，也可以写成 0-59/5，相同意思！</p>

例题：

假如每五分钟需要运行 `/home/dmtsai/test.sh` 一次，又该如何？

答：

同样使用 `crontab -e` 进入编辑：

```
*/5 * * * * /home/dmtsai/test.sh
```

假如你每星期六都与朋友有约，那么想要每个星期五下午 4:30 告诉你朋友星期六的约会不要忘记。

```
30 16 * * 5 mail friend@his.server.name < /home/dmtsai/friend.txt
```

查询使用者目前的 crontab 内容.

```
crontab -l
```

删除某一项工作.

```
crontab -e
```

```
#通过编辑的方式删除
```

```
crontab -r
```

```
#删除所有的工作
```

# 可唤醒停机期间的工作任务

为什么要唤醒停机期间的工作任务？

什么是anacron？

anacron 并不是用来取代 crontab 的，anacron 存在的目的就在于我们上头提到的，在处理非 24 小时一直启动的 Linux 系统的 crontab 的运行！所以 anacron 并不能指定何时运行某项任务，而是以天为单位或者是在启动后立刻进行 anacron 的动作，他会去侦测停机期间应该进行但是并没有进行的 crontab 任务，并将该任务运行一遍后，anacron 就会自动停止了。

`anacron -u [job]..`

选项与参数：

`-s`：开始一连续的运行各项工作 (job)，会依据时间记录档的数据判断是否进行；

`-f`：强制进行，而不去判断时间记录档的时间戳记；

`-n`：立刻进行未进行的任务，而不延迟 (delay) 等待时间；

`-u`：仅升级时间记录档的时间戳记，不进行任何工作。

`job`：由 `/etc/anacrontab` 定义的各项工作的名称。

- 1 今天假设我有一个命令程序，名称为：ping.sh 这个档名！我想要让系统每三分钟运行这个文件一次，但是偏偏这个文件会有很多的信息显示出来，所以我的 root 帐号每天都会收到差不多四百多封的信件，光是收信就差不多快要疯掉了！那么请问应该怎么配置比较好呢？
- 2 您预计要在 2020 年的 2 月 14 日寄出一封给 kiki，只有该年才寄出！该如何下达命令？
- 3 下达 crontab -e 之后，如果输入这一行，代表什么意思？
  - \* 15 \* \* 1-5 /usr/local/bin/tea\_time.sh
- 4 我用 vi 编辑 /etc/crontab 这个文件，我编辑的那一行是这样的：

```
25 00 * * 0 /usr/local/bin/backup.sh
```

这一行代表的意义是什么？



- 1 今天假设我有一个命令程序，名称为：ping.sh 这个档名！我想要让系统每三分钟运行这个文件一次，但是偏偏这个文件会有很多的信息显示出来，所以我的 root 帐号每天都会收到差不多四百多封的信件，光是收信就差不多快要疯掉了！那么请问应该怎么配置比较好呢？

这个涉及数据流重导向的问题，我们可以将他导入文件或者直接丢弃！如果该信息不重要的话，那么就予以丢弃，如果信息很重要的话，才将他保留下来！假设今天这个命令不重要，所以将他丢弃掉！因此，可以这样写：

```
*/3 * * * * root /usr/local/ping.sh > /dev/null 2>&1
```

- 1 您预计要在 2020 年的 2 月 14 日寄出一封给 kiki，只有该年才寄出！该如何下达命令？

```
at 1am 2020-02-14
```

1 下达 `crontab -e` 之后，如果输入这一行，代表什么意思？

■ `* 15 * * 1-5 /usr/local/bin/tea_time.sh`

在每星期的 1~5，下午 3 点的每分钟，共运行 60 次  
`/usr/local/bin/tea_time.sh` 这个文件。要特别注意的是，每个星期  
1~5 的 3 点都会进行 60 次！应该是要写成：

```
30 15 * * 1-5 /usr/local/bin/tea_time.sh
```

- 1 我用 vi 编辑 /etc/crontab 这个文件，我编辑的那一行是这样的：

```
25 00 * * 0 /usr/local/bin/backup.sh
```

这一行代表的意义是什么？

这一行代表.....没有任何意义！因为语法错误！您必须要了解，在 /etc/crontab 当中每一行都必须要有使用者才行！所以，应该要将原本那行改成：

```
25 00 * * 0 root /usr/local/bin/backup.sh
```