

[来自网络,感谢分享者,不能保证以下文字没 bug]

## Linux crontab 命令格式与详细例子

基本格式 :

```
* * * * * command  
分 时 日 月 周 命令
```

第 1 列表示分钟 1~59 每分钟用\*或者 \*/1 表示

第 2 列表示小时 1~23 (0 表示 0 点)

第 3 列表示日期 1~31

第 4 列表示月份 1~12

第 5 列标识号星期 0~6 (0 表示星期天)

第 6 列要运行的命令

crontab 文件的一些例子:

```
30 21 * * * /usr/local/etc/rc.d/lighttpd restart
```

上面的例子表示每晚的 21:30 重启 apache。

```
45 4 1,10,22 * * /usr/local/etc/rc.d/lighttpd restart
```

上面的例子表示每月 1、10、22 日的 4 : 45 重启 apache。

```
10 1 * * 6,0 /usr/local/etc/rc.d/lighttpd restart
```

上面的例子表示每周六、周日的 1 : 10 重启 apache。

```
0,30 18-23 * * * /usr/local/etc/rc.d/lighttpd restart
```

上面的例子表示在每天 18 : 00 至 23 : 00 之间每隔 30 分钟重启 apache。

```
0 23 * * 6 /usr/local/etc/rc.d/lighttpd restart
```

上面的例子表示每星期六的 11 : 00 pm 重启 apache。

```
* */1 * * * /usr/local/etc/rc.d/lighttpd restart
```

每一小时重启 apache

```
* 23-7/1 * * * /usr/local/etc/rc.d/lighttpd restart
```

晚上 11 点到早上 7 点之间, 每隔一小时重启 apache

```
0 11 4 * mon-wed /usr/local/etc/rc.d/lighttpd restart
```

每月的 4 号与每周一到周三的 11 点重启 apache

```
0 4 1 jan * /usr/local/etc/rc.d/lighttpd restart
```

一月一号的 4 点重启 apache

名称 : crontab

使用权限：所有使用者

使用方式：

`crontab file [-u user]`-用指定的文件替代目前的 crontab。

`crontab [-u user]`-用标准输入替代目前的 crontab。

`crontab -l [user]`-列出用户目前的 crontab。

`crontab -e [user]`-编辑用户目前的 crontab。

`crontab -d [user]`-删除用户目前的 crontab。

`crontab -c dir`-指定 crontab 的目录。

crontab 文件的格式：M H D m d cmd.

M: 分钟 (0-59)。

H: 小时 (0-23)。

D: 天 (1-31)。

m: 月 (1-12)。

d: 一星期内的天 (0~6, 0 为星期天)。

cmd 要运行的程序，程序被送入 sh 执行，这个 shell 只有 USER, HOME, SHELL 这三个环境变量

说明：

crontab 是用来让使用者在固定时间或固定间隔执行程序之用，换句话说，也就是类似使用者的时程表。`-u user` 是指设定指定

user 的时程表，这个前提是你必须要有其权限(比如说是 root)才能够指定他人的时程表。如果不使用 `-u user` 的话，就是表示设

定自己的时程表。

参数：

`crontab -e`：执行文字编辑器来设定时程表，内定的文字编辑器是 VI，如果你想用别的文字编辑器，则请先设定 VISUAL 环境变数

来指定使用那个文字编辑器(比如说 `setenv VISUAL joe`)

`crontab -r` : 删除目前的时程表

`crontab -l` : 列出目前的时程表

`crontab file [-u user]`-用指定的文件替代目前的 `crontab`。

时程表的格式如下 :

```
f1 f2 f3 f4 f5 program
```

其中 `f1` 是表示分钟, `f2` 表示小时, `f3` 表示一个月份中的第几日, `f4` 表示月份, `f5` 表示一个星期中的第几天。`program` 表示要执

行的程序。

当 `f1` 为 `*` 时表示每分钟都要执行 `program`, `f2` 为 `*` 时表示每小时都要执行程序, 其余类推

当 `f1` 为 `a-b` 时表示从第 `a` 分钟到第 `b` 分钟这段时间内要执行, `f2` 为 `a-b` 时表示从第 `a` 到第 `b` 小时都要执行, 其余类推

当 `f1` 为 `*/n` 时表示每 `n` 分钟个时间间隔执行一次, `f2` 为 `*/n` 表示每 `n` 小时个时间间隔执行一次, 其余类推

当 `f1` 为 `a, b, c, ...` 时表示第 `a, b, c, ...` 分钟要执行, `f2` 为 `a, b, c, ...` 时表示第 `a, b, c, ...` 个小时要执行, 其余类推

使用者也可以将所有的设定先存放在档案 `file` 中, 用 `crontab file` 的方式来设定时程表。

例子 :

```
#每天早上 7 点执行一次 /bin/ls :
```

```
0 7 * * * /bin/ls
```

```
在 12 月内, 每天的早上 6 点到 12 点中, 每隔 3 个小时执行一次 /usr/bin/backup :
```

```
0 6-12/3 * 12 * /usr/bin/backup
```

```
周一到周五每天下午 5:00 寄一封信给 alex@domain.name :
```

```
0 17 * * 1-5 mail -s "hi" alex@domain.name < /tmp/maildata
```

每月每天的午夜 0 点 20 分, 2 点 20 分, 4 点 20 分.... 执行 echo "haha"

```
0 0-23/2 * * * echo "haha"
```

注意 :

当程序在你所指定的时间执行后, 系统会寄一封信给你, 显示该程序执行的内容, 若是 你不希望收到这样的信, 请在每一行空一格之

后加上 > /dev/null 2>&1 即可

例子 2 :

```
#每天早上 6 点 10 分
```

```
10 6 * * * date
```

```
#每两个小时
```

```
0 */2 * * * date
```

```
#晚上 11 点到早上 8 点之间每两个小时, 早上 8 点
```

```
0 23-7/2, 8 * * * date
```

```
#每个月的 4 号和每个礼拜的礼拜一到礼拜三的早上 11 点
```

```
0 11 4 * mon-wed date
```

```
#1 月份日早上 4 点
```

```
0 4 1 jan * date
```

范例

```
$crontab -l 列出用户目前的 crontab.
```