

23.计划任务

什么叫计划任务

计划任务，字面意思就是在约定好的时间执行已经计划好的工作，如每天晚上12点备份一个日志文件，这就是个计划任务。

计划任务的类型

1. 系统执行的工作：系统周期性所要执行的工作，如备份系统数据、清理缓存
2. 个人执行的工作：某用户定期要做的工作，例如每隔10分钟检查邮件服务器是否有新信，这些工作可由每个用户自行设置。

计划任务的实现

常用语执行计划任务的命令——at, batch, 与周期性任务相关的命令crontab, **CentOS7的新特性timer**（讲到systemd的时候细讲）

1. at命令和batch命令的区别

at命令旨在特定的时间执行任务；batch命令设置的任务由系统选择空闲时间去执行，空闲时间默认为cpu利用率小于0.8的时候

2. at命令

at的任务是否能运行依赖于atd服务

我们可以先查看下atd服务是否运行

centos6:

service atd status #查看atd服务状态

chkconfig --list | grep atd #此命令在centos7上不能执行

centos7:

systemctl status atd #查看atd服务状态

systemctl start atd #开启atd服务

systemctl is-enabled atd #查看是否开始开机启动服务，如果弹出enabled，说明开机启动此服务

语法

at (选项) (参数)

选项

-f: 指定包含具体指令的任务文件；shell脚本

-q: 指定新任务的队列名称；

-l: 显示待执行任务的列表；atq, at -l

-d: 删除指定的待执行任务；atrm at -d

-m: 任务执行完成后向用户发送E-mail。

具体的使用方法:

at命令后想要输入执行程序的确切时间 (at 20:00 2019-3-1 在某天 ，也可以

24:32当天某个时间)，然后回车

接着在>后输入你想要执行的命令，最后用Ctrl+d组合键退出at

或者at now +10 minutes 十分钟后执行

minutes: 表示《分钟》

hours: 表示《小时》。

days: 表示《天》。

weeks: 表示《星期》。

months: 表示《月》。

years: 表示《年》。

实例：

```
[root@CentOS7 ~]# date
2019年 02月 28日 星期四 12:06:11 CST
[root@CentOS7 ~]# at 12:07
at> ls > /root/aaab.txt
at> <EOT>
job 5 at Thu Feb 28 12:07:00 2019
[root@CentOS7 ~]# ll
总用量 16
-rw-r--r-- 1 root root 38 2月 28 12:04 aaaa.txt
-rw-r--r-- 1 root root 47 2月 28 12:07 aaab.txt
-rw----- 1 root root 1486 1月 21 00:11 anaconda-ks.cfg
-rw-r--r-- 1 root root 1232 1月 27 22:05 grepdemo.txt
[root@CentOS7 ~]#
```

atq和atrm命令：列出和删除正在等待执行的at任务

我们每次使用at命令指定延时执行的命令，at都会给其分配一个job编号

具体操作：

```
[root@CentOS7 ~]# at now +10 minutes
at> ls > /root/abc
at> <EOT>
job 6 at Thu Feb 28 12:18:00 2019
[root@CentOS7 ~]# atq
6 Thu Feb 28 12:18:00 2019 a root
[root@CentOS7 ~]# atrm 6
[root@CentOS7 ~]# atq
[root@CentOS7 ~]#
```

把任务删除后就不会再去执行了。

我们使用at -l 同样可以查看要延时执行的任务

at还有黑名单和白名单功能

（黑名单）：当仅/etc/at.deny 存在时（默认）写入deny的用户不能配置at

（白名单）：当/etc/at.allow存在时/etc/at.deny失效，只有写在allow中的用户才能配置at

当/etc/at.allow及/etc/at.deny都不存在时，仅有root才能配置at

batch命令的用法和at相似，这里就不再介绍了。

3. crontab命令

使用at命令只能执行某个命令一次，而crontab却可以重复执行命令，例如：每分钟，每小时，每天，每周等。

crontab其实是一个命令，用来读取和修改名为crontab的文件。这个crontab文件包含了你要定时执行的程序列表，也包含了执行的时刻。

实际上，有两个命令，一个叫crontab，一个叫cron。crontab用于修改crontab文件，cron用于实际执行定时的程序。

注意：crontab命令只有在启动crond服务的前提下才能实用！

```
# systemctl start crond
```

```
# systemctl enable crond
```

crontab的常用参数：

```
crontab -l    #列出当前用户下的cron服务的详细内容
```

```
crontab -r    #删除cron服务
```

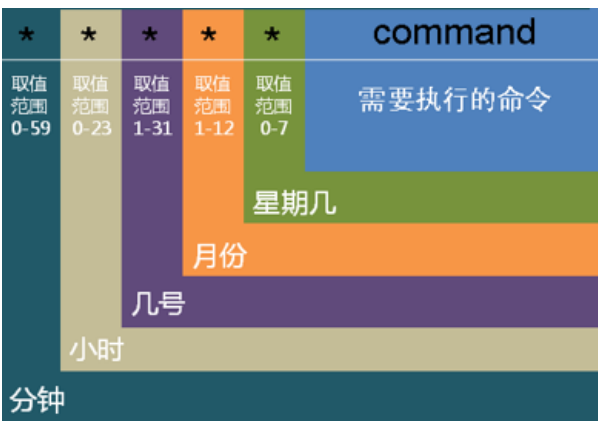
```
crontab -e    #编辑cron服务
```

例如：

```
crontab -l    # 查看自己的cron计划任务
```

```
crontab -r    # 想删除自己的cron计划任务
```

crontab -e 编辑时的语法（注意命令必须些绝对路径）



0或者7都是表示星期天，另外在这个时间的表达式中，还有些特殊符号，意思如下：

*	代表取值范围内的数字	(任意/每)
/	指定时间的间隔频率	*/10 0-23/2
-	代表从某个数字到某个数字	8-17
,	分开几个离散的数字	6,10-13,20

每天凌晨2点1分：1 2 * * *

每月1、10、22日的4 : 45: 45 4 1, 10, 22 * *

每月1到10日的4 : 45: 45 4 1-10 * *

每隔两天的上午8点到11点的第3和第15分钟：3, 15 8-11 */2 * *

晚上11点到早上7点之间，每隔一小时：0 23-7/1 * * *

周一到周五每天晚上 21:15: 15 21 * * 1-5

注：所有用户的计划任务，都会在/var/spool/cron/下产生对应的文件

系统级别的计划任务

系统级计划任务的主配置文件/etc/crontab

仅能通过 root定义，不过可以以普通用户的身份去执行计划任务中的命令。

```
SHELL=/bin/bash    #指定操作系统使用哪个shell
PATH=/sbin:/bin:/usr/sbin:/usr/bin    #系统执行命令的搜索路径
MAILTO=root    #将执行任务的信息通过邮件发送给xx用户
```

```
# For details see man 4 crontabs
```

```
# Example of job definition:
```

```
# .----- minute (0 - 59)
```

```
# | .----- hour (0 - 23)
```

```
# | | .----- day of month (1 - 31)
```

```
# | | | .----- month (1 - 12) OR jan, feb, mar, apr ...
```

```
# | | | | .---- day of week (0 - 6) (Sunday=0 or 7) OR
```

```
sun, mon, tue, wed, thu, fri, sat
```

```
# | | | | |
```

```
# * * * * * user-name command to be executed
```

要配置计划任务就在文件的最后一行添加内容，比如：

```
*/5 * * * * /usr/bin/ls # 表示每5分钟执行一次ls
```