

# 计划任务

- 计划任务主要是做一些周期性的任务，目前主要用途是定期备份数据

## 一次调度执行at

- 语法格式：

```
1 at <TIMESPEC>
2   now +5min
3   teatime tomorrow (teatime is 16:00)
4   noon +4 days
5   5pm august 3 2021
```

- 例1

```
1 [root@localhost ~]# systemctl start atd
2 [root@localhost ~]# systemctl enable atd
3 [root@server1 ~]# at now +2min
4 at> useradd zhangsan
5 at> <EOT>
6 job 1 at Tue Feb  2 11:29:00 2021
7 [root@server1 ~]# atq
8 1 Tue Feb  2 11:29:00 2021 a root
9 [root@server1 ~]# id zhangsan
10 id: zhangsan: no such user
11 [root@server1 ~]# id zhangsan
12 uid=1000(zhangsan) gid=1000(zhangsan) 组=1000(zhangsan)
13
```

- 例2

```
1 [root@localhost ~]# vi at.jobs
2 touch /root/`date +%F`.txt
3 useradd eagleslab
4 [root@localhost ~]# at now +1min < at.jobs
5 [root@localhost ~]# ls
6 2020-08-27.txt anaconda-ks.cfg at.jobs
7 [root@localhost ~]# id eagleslab
8 uid=1001(eagleslab) gid=1001(eagleslab) 组=1001(eagleslab)
```

## 循环调度执行cron 用户级

- `crond` 在使用之前必须要启动守护进程

```
1 [root@localhost ~]# systemctl start crond
2 [root@localhost ~]# systemctl enable crond
3 [root@localhost ~]# ps aux|grep crond
4 root      6242  0.3  0.0 126380 1656 ?        Ss   16:27   0:00
   /usr/sbin/crond -n
```

- `cron` 进程每分钟会处理一次计划任务
- 存储位置

```
1 [root@localhost ~]# ls /var/spool/cron
```

- 管理命令

```
1 [root@localhost ~]# crontab -l      # 列出当前用户所有计划任务
2 [root@localhost ~]# crontab -r      # 删除当前用户计划任务
3 [root@localhost ~]# crontab -e      # 编辑当前用户计划任务
4 管理员可以使用 -u username, 去管理其他用户的计划任务
5 [root@localhost ~]# vi /etc/cron.deny # 这个文件中加入的用户名无法使用cron
```

- `cron` 语法格式

```
1 分 时 日 月 星期 命令
2 * 表示任何数字都符合
3 0 2 * * * /run.sh      # 每天的2点
4 0 2 14 * * /run.sh     # 每月14号2点
5 0 2 14 2 * /run.sh     # 每年2月14号2点
6 0 2 * * 5 /run.sh      # 每个星期5的2点
7 0 2 * 6 5 /run.sh      # 每年6月份的星期5的2点
8 0 2 2 * 5 /run.sh      # 每月2号或者星期5的2点    星期和日同时存在，那么就是或的关系
9 0 2 2 6 5 /run.sh      # 每年6月2号或者星期5的2点
10
11 */5 * * * * /run.sh   # 每隔5分钟执行一次
12 0 2 1,4,6 * * /run.sh  # 每月1号，4号，6号的2点
13 0 2 5-9 * * /run.sh   # 每月5-9号的2点
14
15 * * * * * /run.sh     # 每分钟
16 0 * * * * /run.sh     # 每整点
17 * * 2 * * /run.sh     # 每月2号的每分钟
```

## 循环调度执行cron系统级

- 临时文件的清理 `/tmp` `/var/tmp`
- 系统信息的采集 `sar`
- 日志的轮转(切割) `logrotate`
- 通常不是由用户定义
- 文件的位置

```
1 [root@localhost ~]# vim /etc/crontab      # 默认没有定义任何计划任务
2 [root@localhost ~]# ls /etc/cron.d      # 定义的计划任务每小时会执行
3 0hourly sysstat
4 [root@localhost ~]# cat /etc/cron.d/0hourly
5 # Run the hourly jobs
6 SHELL=/bin/bash
7 PATH=/sbin:/bin:/usr/sbin:/usr/bin
8 MAILTO=root
9 01 * * * * root run-parts /etc/cron.hourly      # 每小时01分以root身份执
   行/etc/cron.hourly/目录下的所有脚本
```

- `cron` 仅仅会执行每小时定义脚本 `/etc/cron.hourly`

```
1 [root@localhost ~]# ls /etc/cron.hourly/
2 [root@localhost ~]# cat /etc/cron.hourly/0anacron
3 /usr/sbin/anacron -s          # anacron是用来检查是否有错过的计划任务需要被执行
4 [root@localhost ~]# vi /etc/anacrontab
5 1 5 cron.daily nice run-parts /etc/cron.daily
6 #每天开机 5 分钟后就检查 /etc/cron.daily 目录内的文件是否被执行，如果今天没有被执行，那就执行
7 7 25 cron.weekly nice run-parts /etc/cron.weekly
8 #每隔 7 天开机后 25 分钟检查 /etc/cron.weekly 目录内的文件是否被执行，如果一周内没有被执行，就会执行
9 @monthly 45 cron.monthly nice run-parts /etc/cron.monthly
10 #每隔一个月开机后 45 分钟检查 /etc/cron.monthly 目录内的文件是否被执行，如果一个月内没有被执行，那就执行
```