

# Linux\_crontd定时任务

## Crontd

Crontd 是 Linux 系统中用来定期执行命令和或者指定程序任务的一种服务或者软件

一般安装C5/C6系统时，crond默认存在

Crontd 服务默认每分钟，会检查系统中是否有需要执行的定时任务，如果有系统会根据定义好的规则来执行定时任务

秒级任务：

crond无能为力

自己写守护进程shell循环

Quartz可以实现秒级任务

## Linux定时任务分类

系统自身定期执行的任务

用户执行的定时任务

## Linux的定时任务分类

at适合执行一次就结束的调度任务

```
[root@node1 ~]# chkconfig --list atd
atd          0:off  1:off  2:off  3:on   4:on   5:on   6:off
```

anacron 适合于非7\*24小时开机的服务准备的，开机执行的

检测停机期间没有执行的任务，在开机后一次执行一遍

## crond

aasdfasd

服务默认每分钟，会检查系统中是否有需要执行的定时任务，如果有系统会根据定义好的规则来执行定时任务

```
[root@node1 ~]# chkconfig --list crond
crond       0:off  1:off  2:on   3:on   4:on   5:on   6:off
```

crond是一个定时任务的守护进程，crontab是用户用来设定定时任务规则的命令

## crontab

参数 <sup>↗</sup>	含义 <sup>↗</sup>	指定示例 <sup>↗</sup>
<b>-l(字母)</b> <sup>↗</sup>	查看 crontab 文件内容。提示：l 可理解为 list 的缩写。 <sup>↗</sup>	crontab -l❌❌❌ <sup>↗</sup>
<b>-e</b> <sup>↗</sup>	编辑 crontab 文件内容，提示：e 可理解为 edit 的缩写。 <sup>↗</sup>	crontab -e❌❌❌ <sup>↗</sup>
<b>-i</b> <sup>↗</sup>	删除 crontab 文件内容，删除前会提示确认。用的很少。 <sup>↗</sup>	crontab -ri <sup>↗</sup>
<b>-r</b> <sup>↗</sup>	删除 crontab 文件内容。用的很少。 <sup>↗</sup>	crontab -r <sup>↗</sup>
<b>-u user</b> <sup>↗</sup>	指定使用的用户执行任务。 <sup>↗</sup>	crontab -u boy -l <sup>↗</sup>
特别强调：-i,-r 参数在生产中很少用，没什么需求必须要用-e 进去编辑即可。 <sup>↗</sup>		

crontab -e 编辑的文件为 /var/spool/cron/root

```
head -1 /var/spool/cron/root
```

|文件|说明|||--||/etc/cron.deny|该文件中所列用户不允许使用 crontab 命令||/etc/cron.allow|该文件中所列用户允许是 crontab 命令，优先于/etc/cron.deny||/var/spool/cron/|所用用户 crontab配置文件默认都存放在此目录，文件以用户名命名|

## 定时任务指令使用格式

用户的定时任务6列

系统的定时任务8列

```
[root@node1 cron]# cat /etc/crontab
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
HOME=/

# 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
```

## 基本格式

```
[root@node1 ~]# crontab -e
0 9 * * 6,7 /bin/sh /server/scripts/printDate.sh >/dev/null 2>&1
[root@node1 ~]# cat /var/log/cron
Jul 8 04:29:01 node1 CROND[2731]: (root) CMD (/root/shan.sh)
Jul 8 04:30:01 node1 CROND[2739]: (root) CMD (/root/shan.sh)
Jul 8 04:30:01 node1 CROND[2740]: (root) CMD (/usr/lib/sa/sa1 1 1)
Jul 8 04:31:01 node1 CROND[2748]: (root) CMD (/root/shan.sh)
Jul 8 04:32:01 node1 CROND[2753]: (root) CMD (/root/shan.sh)
Jul 8 04:33:01 node1 CROND[2762]: (root) CMD (/root/shan.sh)
Jul 8 04:34:01 node1 CROND[2768]: (root) CMD (/root/shan.sh)
Jul 8 04:35:01 node1 CROND[2773]: (root) CMD (/root/shan.sh)
Jul 8 04:36:01 node1 CROND[2779]: (root) CMD (/root/shan.sh)
Jul 8 04:37:01 node1 CROND[2784]: (root) CMD (/root/shan.sh)
```

01 \* \* \* \* cmd

cmd 为要执行的命令或脚本，例：/bin/sh /server/scripts/shan.sh

每列之间必须要有一个空格

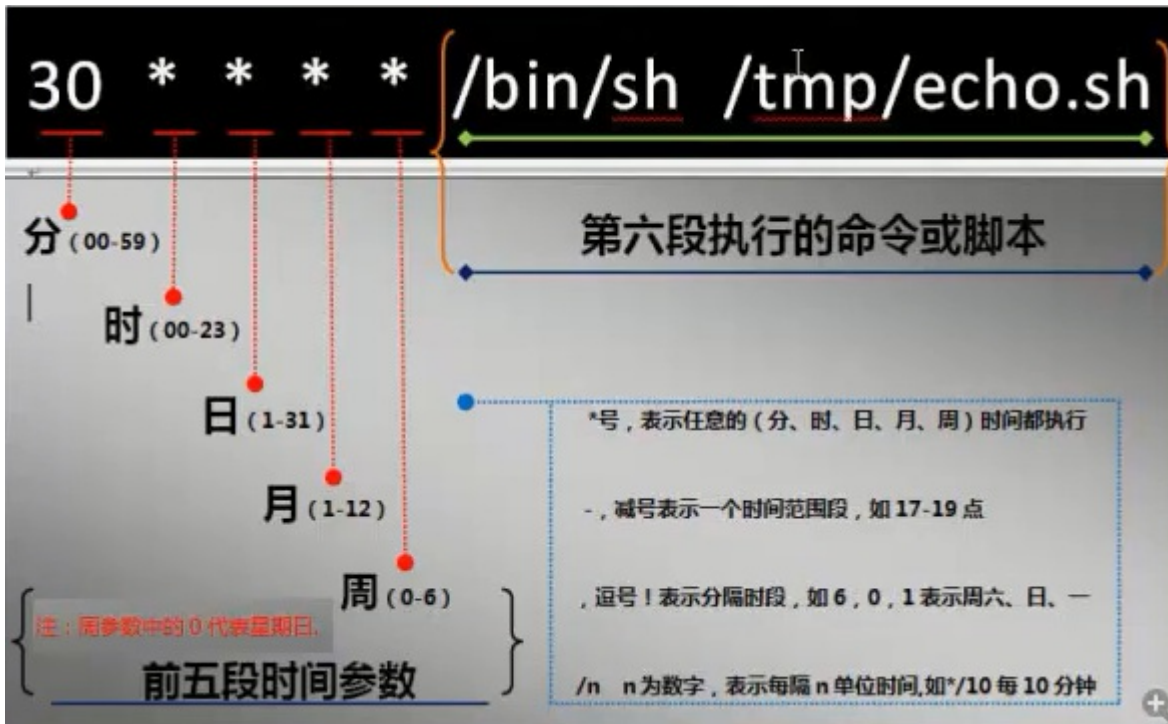
\* 表示任意时间，实际就是“每”时间的意思

分位上的\*就等价于每分钟

\*\*\_\*\*表示分隔符，表示一个时间范围，如17-19点，00 17-19 \* \* \* cmd，每天的17,18,19点的00分执行任务

\*\*,\*\*表示分隔时段的意思 30 17,18,19 \* \* \* cmd 表示每天的17,18,19点的30分执行任务

/n代表数字，\*/10 \* \* \* \* cmd 表示每10分钟执行任务



## 注意事项

在定时任务的最后加上 `/dev/null 2>&1` (条件：定时任务调用的必须是脚本文件，如果定时任务直接配的是命令不用加)

如果定时任务配的是脚本不用 `/dev/null 2>&1` 会在6.X上/var/spool/mail/root 文件中记录详细的错误日志

## 小结

要领 1：为定时任务规则加必要的注释

要领 2：执行 shell 脚本任务前加 `/bin/sh`

要领 3：定时任务命令或脚本的结尾加 `>/dev/null 2>&1`

要领 4：定时任务命令或程序最好写到脚本里执行

要领 5：在指定用户下执行相关定时任务

要领 6：生产任务程序不要随意打印输出信息 `tar zcf echo 123 >a.log`

要领 7：定时任务执行的脚本要规范路径 `/server/scripts`

要领 8：配置定时任务规范操作过程，防止出错

### 工作中调试定时任务的方法↓

- 1、增加执行任务频率调试任务（某些任务不能用于生产环境）↓
- 2、调整系统时间调试任务（不能用于生产环境）↓
- 3、通过脚本日志输出调试定时任务↓
- 4、注意一些任务命令带来的问题（\*/1 \* \* \* \* echo "==" >> /tmp/oldboy.log >/dev/null 2>&
- 5、注意环境变量导致的定时任务故障↓
- 6、通过crond定时任务服务日志调试定时任务↓
- 7、其他稀奇古怪的问题调试的办法↓

### crontab定时任务生产应用问题8箴言↓

1. 系统环境变量问题↓
2. 定时任务要用绝对路径↓
3. 脚本权限问题加/bin/sh↓
4. 时间变量问题用反斜线\%↓
5. >/dev/null 2>&1问题（1>/dev/null 2>/dev/null）↓
6. 定时任务规则之前加注释↓
7. 使用脚本程序替代命令行定时任务↓
8. 避免不必要的程序及命令输出↓
9. 切到目标目录的上一级打包目标↓
10. 定时任务脚本中的程序命令用全路径← I

```

JAVA_HOME="/opt/jdk1.6.0_18"
CLASSPATH=$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar
PATH=$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar:/opt/nginx-0.7.61/sbin:/opt/jdk1.6.0_18/bin:/opt/resin-3.0.25/bin:$PATH
export JAVA_HOME PATH USER LOGNAME MAIL HOSTNAME HISTSIZE INPUTRC CLASSPATH
/usr/local/bin/xxresin_stop.sh
/usr/local/bin/xxresin_start.sh

```

linux定时任务crond那些事

linux定时任务生产java服务无法执行问题群友案例