



链滴

进程，系统性能和计划任务 3

作者: [Carey](#)

原文链接: <https://ld246.com/article/1609832052223>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

3 任务计划

通过任务计划，可以让系统自动的按时间或周期性任务执行任务

注意: 学习本节需要实现邮件通知,学习内容前必须安装并启动邮件服务

范例：环境准备

```
[root@centos8 ~]#yum -y install postfix  
[root@centos8 ~]#systemctl enable --now postfix
```

未来的某时间点执行一次任务

- at 指定时间点，执行一次任务
- batch 系统自行选择空闲时间去执行此处指定的任务

周期性运行某任务

- cron

3.1 一次性任务

at工具

- 由包at提供
- 依赖与atd服务,需要启动才能实现at任务
- at队列存放在/var/spool/at目录中,ubuntu存放在/var/spool/cron/atjobs目录下
- 执行任务时PATH变量的值和当前定义任务的用户身份一致

at 命令:

```
at [option] TIME
```

常用选项:

```
-V 显示版本信息  
-t time 时间格式[[CC]YY]MMDDhhmm[.ss]  
-l 列出指定队列中等待运行的作业; 相当于atq  
-d N 删除指定的N号作业; 相当于atrm  
-c N 查看具体作业N号任务  
-f file 指定的文件中读取任务  
-m 当任务被完成之后, 将给用户发送邮件, 即使没有标准输出
```

注意:

- 作业执行命令的结果中的标准输出和错误以执行任务的用户身份发邮件通知给 root
- 默认CentOS 8 最小化安装没有安装邮件服务,需要自行安装

TIME: 定义出什么时候进行 at 这项任务的时间

HH:MM [YYYY-mm-dd]
noon, midnight, teatime (4pm) ,tomorrow
now+#{minutes,hours,days, OR weeks}

范例：at 时间格式

HH:MM 在今日的 HH:MM 进行，若该时刻已过，则明天此时执行任务
02:00

HH:MM YYYY-MM-DD 规定在某年某月的某一天的特殊时刻进行该项任务
02:00 2016-09-20

HH:MM[am|pm] [Month] [Date]
06pm March 17
17:20 tomorrow

HH:MM[am|pm] + number [minutes|hours|days|weeks]，在某个时间点再加几个时间后才进行
项任务
now + 5 min
02pm + 3 days

at 任务执行方式：

- 交互式
- 输入重定向
- at -f file

/etc/at.{allow,deny} 控制用户是否能执行at任务

- 白名单：/etc/at.allow 默认不存在，只有该文件中的用户才能执行at命令
- 黑名单：/etc/at.deny 默认存在，拒绝该文件中用户执行at命令，而没有在at.deny 文件中的使用者则可执行
- 如果两个文件都不存在，只有 root 可以执行 at 命令

范例: ubuntu at任务存放路径

```
[root@ubuntu2004 ~]#ll /var/spool/cron/  
[root@ubuntu2004 ~]#ll /var/spool/cron/atjobs/
```

3.2 周期性任务计划cron

周期性任务计划cron相关的程序包：

- crone: 主程序包，提供crond守护进程及相关辅助工具
- crontabs: 包含CentOS提供系统维护任务
- crone-anacron: crone的补充程序，用于监控crone任务执行状况，如:crone中的任务在过去该行的时间点未能正常运行，则anacron会随后启动一次此任务

cron 依赖于crond服务，确保crond守护处于运行状态：

#CentOS 7 以后版本：

```
systemctl status crond
```

```
#CentOS 6:  
service crond status
```

cron任务分为

- 系统cron任务：系统维护作业， /etc/crontab 主配置文件， /etc/cron.d/ 子配置文件
- 用户cron任务：保存在 /var/spool/cron/USERNAME(ubuntu 系统存放在/var/spool/cron/crontabs/USERNAME)， 利用 crontab 命令管理

计划任务日志： /var/log/cron

3.2.1 系统cron计划任务

/etc/crontab 格式说明， 详情参见 man 5 crontab

注释行以 # 开头

```
[root@centos8 ~]#cat /etc/crontab  
SHELL=/bin/bash           #默认的SHELL类型  
PATH=/sbin:/bin:/usr/sbin:/usr/bin #默认的PATH变量值,可修改为其它路径  
MAILTO=root               #默认标准输出和错误发邮件给root,可以指向其它用户  
  
# 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
```

计划任务时间表示法：

(1) 特定值

给定时间点有效取值范围内的值

(2) *

给定时间点上有效取值范围内的所有值,表示“每...” ,放在星期的位置表示不确定

(3) 离散取值

#,#,#

(4) 连续取值

#-#

(5) 在指定时间范围上， 定义步长

/#: #即为步长

(6) 特定关键字

```
@yearly    0 0 1 1 *
@annually  0 0 1 1 *
@monthly   0 0 1 * *
@weekly    0 0 * * 0
@daily     0 0 * * *
@hourly    0 * * * *
@reboot    Run once after reboot
```

范例：每个月日期和星期几字段的关系

星期几如果和日期冲突它俩的关系为或者，其余的关系都为并且
[root@centos8 ~]#man 5 crontab

范例:

```
#晚上9点10分运行echo命令,输出信息仍会发送到root 邮箱
10 21 * * * wang /bin/echo "Howdy!"
```

```
#每3小时echo和wall命令
0 */3 * * * wang /bin/echo "howdy" ; wall "welcome to Magedu!"
```

crond任务相关文件:

```
/etc/crontab    配置文件
/etc/cron.d/    配置文件
/etc/cron.hourly/ 脚本
/etc/cron.daily/ 脚本
/etc/cron.weekly/ 脚本
/etc/cron.monthly/ 脚本
```

3.2.2 anacron

运行计算机关机时cron不运行的任务，CentOS6以后版本取消anacron服务，由crond服务管理，对笔记本电脑、台式机、工作站、偶尔要关机的服务器及其它不一直开机的系统很重要对很有用

由/etc/cron.hourly/0anacron执行，当执行任务时，更新/var/spool/anacron/cron.daily 文件的时戳

配置文件：/etc/anacrontab，负责执行/etc/ cron.daily /etc/cron.weekly /etc/cron.monthly中统任务

/etc/anacrontab格式说明

字段1：如果在这些日子里没有运行这些任务.....

字段2：在重新引导后等待这么多分钟后运行它

字段3：任务识别器，在日志文件中标识

字段4：要执行的任务

3.2.3 管理临时文件

CentOS 7 使用 systemd-tmpfiles-setup服务实现

CentOS 6 使用/etc/cron.daily/tmpwatch定时清除临时文件

配置文件:

```
/etc/tmpfiles.d/*.conf  
/run/tmpfiles.d/*.conf  
/usr/lib/tmpfiles/*.conf
```

/usr/lib/tmpfiles.d/tmp.conf

```
d /tmp 1777 root root 10d  
d /var/tmp 1777 root root 30d
```

命令:

```
systemd-tmpfiles --clean|remove|create configfile
```

3.2.4 用户计划任务

crontab命令:

- 每个用户都有专用的cron任务文件: /var/spool/cron/USERNAME
- 默认标准输出和错误会被发邮件给对应的用户,如: wang创建的任务就发送至wang的邮箱
- root能够修改其它用户的作业
- 用户的cron 中默认 PATH=/usr/bin:/bin,如果使用其它路径,在任务文件的第一行加PATH=/path者加入到计划任务执行的脚本中
- 第六个字段指定要运行的命令。该行的整个命令部分,直至换行符或“%”字符,指定的shell执行。除非使用反斜杠(\)进行转义,否则该命令中的“%”字符将变为换行符,并且第一个%之后的数据将作为标准输入发送到该命令。

crontab命令格式:

```
crontab [-u user] [-l | -r | -e] [-i]
```

常用选项:

```
-l  列出所有任务  
-e  编辑任务  
-r  移除所有任务  
-i  同-r一同使用,以交互式模式移除指定任务  
-u user 指定用户管理cron任务,仅root可运行
```

控制用户执行计划任务:

```
/etc/cron.{allow,deny}
```

范例: 修改默认的cron的文本编辑工具

```
root@ubuntu1804:~# crontab -e
no crontab for root - using an empty one
```

Select an editor. To change later, run 'select-editor'.

```
1. /bin/nano      <----- easiest
2. /usr/bin/vim.basic
3. /usr/bin/vim.tiny
4. /bin/ed
```

```
Choose 1-4 [1]:
root@ubuntu1804:~# cat /etc/profile.d/env.sh
export EDITOR=vim
```

范例：PATH变量

```
#方法1,在计划任务配置中指定PATH
[root@centos8 ~]# crontab -l
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/bin
```

```
* * * * * useradd hehe;echo $PATH
```

```
#方法2,在脚本中指定PATH变量
[root@centos8 ~]# crontab -l
```

```
* * * * * /data/test.sh
```

```
[root@centos8 ~]# cat /data/test.sh
#!/bin/bash
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/bin
useradd hehe
echo $PATH
```

面试题：11月每天的6-12点之间每隔2小时执行/app/bin/test.sh

```
[root@centos8 ~]# crontab -l
0 6-12/2 * 11 * /app/bin/test.sh
```

注意：运行结果的标准输出和错误以邮件通知给相关用户

```
(1) COMMAND > /dev/null
(2) COMMAND &> /dev/null
```

cron任务中不建议使用%，它有特殊用途，它表示换行的特殊意义，且第一个%后的所有字符串会被成当作命令的标准输入,如果在命令中要使用%，则需要用\转义

注意：将%放置于单引号中是不支持的

范例：在crontab中%的用法

```
30 2 * * * /bin/cp -a /etc/ /data/etc`date +%F_%T`
30 2 * * * /bin/cp -a /etc/ /data/etc`date + '%F_%T' ` 有问题
```

范例：在crontab中%的用法

```
[root@centos8 ~]#crontab -l
```

```
* * * * * mail -s "test" wang%wang,%%how are you?%
```

```
[root@centos8 ~]# cat /var/spool/mail/wang
From root@centos8.localdomain Sat Jul 4 23:58:01 2020
Return-Path:
X-Original-To: wang
Delivered-To: wang@centos8.localdomain
Received: by centos8.localdomain (Postfix, from userid 0)
id 0B03860272; Sat, 4 Jul 2020 23:58:01 +0800 (CST)
Date: Sat, 04 Jul 2020 23:58:01 +0800
To: wang@centos8.localdomain
Subject: test
User-Agent: Heirloom mailx 12.5 7/5/10
MIME-Version: 1.0
Content-Type: text/plain; charset=us-ascii
Content-Transfer-Encoding: 7bit
Message-Id: <20200704155801.0B03860272@centos8.localdomain>
From: root@centos8.localdomain (root)
```

wang,

```
how are you?
[root@centos8 ~]#
```

思考:

如何在秒级别运行任务?

```
for min in 0 1 2; do echo "hi"; sleep 20; done
```

如何实现每7分钟运行一次任务?

sleep命令:

```
sleep NUMBER[SUFFIX]...
```

SUFFIX:
s: 秒, 默认
m: 分
h: 小时
d: 天

范例:

```
[11:01:57 root@centos8 ~]#usleep
warning: usleep is deprecated, and will be removed in near future!
warning: use "sleep 1e-06" instead...
```

范例:

```
[10:31:58 root@centos8 ~]#time usleep 1000000
warning: usleep is deprecated, and will be removed in near future!
warning: use "sleep 1" instead...
```



```
real 0m1.014s
user 0m0.001s
sys 0m0.001s
[11:01:37 root@centos8 ~]#time ls
1.sh nohup.out scanhost.sh webmin-1.962-1.noarch.rpm
```

```
real 0m0.003s
user 0m0.001s
sys 0m0.001s
[11:01:47 root@centos8 ~]#time sleep 0.2
```

```
real 0m0.302s
user 0m0.002s
sys 0m0.001s
```

练习:

1. 每周的工作日1: 30, 将/etc备份至/backup目录中, 保存的文件名称格式为“etcbak-yyyy-mm-d-HH.tar.xz”, 其中日期是前一天的时间

```
[13:57:19 root@centos8 ~]#cat back-etc.sh
#!/bin/bash
#
#*****
#Author:zhangzhuo
#QQ: 1191400158
#Date: 2021-01-05
#FileName: back-etc.sh
#URL: https://www.zhangzhuo.ltd
#Description: The test script
#Copyright (C): 2021 All rights reserved
#*****
DIR=/etc
BACKDIR=/backup/etcbak-`date -d '-1 days' +%F-%H-%M-%S`
{
tar Jcvf $BACKDIR.tar.xz $DIR &>/dev/null
}&
```

```
[13:53:42 root@centos8 backup]#crontab -l
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin
30 1 * * 1-5 /root/back-etc.sh
```

2. 每两小时取出当前系统/proc/meminfo文件中以S或M开头的信息追加至/tmp/meminfo.txt文件中

```
[14:01:21 root@centos8 ~]#crontab -l
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin
0 */2 * * * grep -E '^S|^M' /proc/meminfo >> /tmp/meminfo.txt
```

3. 工作日时间, 每10分钟执行一次磁盘空间检查, 一旦发现任何分区利用率高于80%, 就执行wall警报

```
[14:22:07 root@centos8 ~]#cat diskwall.sh
#!/bin/bash
```

```
#
#*****
#Author:zhangzhuo
#QQ: 1191400158
#Date: 2020-01-05
#FileName: diskwall.sh
#URL: https://www.zhangzhuo.ltd
#Description: The test script
#Copyright (C): 2020 All rights reserved
#*****
df | sed -rn '/^\Vdev/s@([\^ ]*) .* ([0-9]+)%.*@\1 \2@p|' while read disk per;do
if [ $per -gt 80 ];then
echo Wall `hostname -l` host $disk disk the absence | mail -s "Wall" 1191400158@qq.com
fi
done

[14:22:26 root@centos8 ~]#crontab -l
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin
30 1 * * 1-5 /root/back-etc.sh
* */2 * * * grep -E '^S|^M' /proc/meminfo >> /tmp/meminfo.txt
*/10 * * * * /root/diskwall.sh
```