1.Linux计划任务

大家平常都会有一些比如说：你每天固定几点起床？每天按时上班打卡、每月15号准时开工资、每年2月14你俩口子某某纪念日等这些诸如此类，这些都是定时发生的。或者说是通俗点说：例行公事；还比如说我们还会遇到一些突发事件，临时几点过来加个班？刚好晚上几点聚个餐？
像上面这些情况，如果事少的话你大脑可以记住，如果事很多，像老板经理董事长每天的工作安排，通常都是记在一些本上，或者闹铃提醒等。
那么，咱们的LINUX系统和上面的情况也很类似，我们也可以通过一些设置。来让电脑定时提醒我们该做什么事了。或者我们提前设置好，告诉电脑你几点做什么几点做什么，这种我们就叫它定时任务。而遇到一些需要执行的事情或任务。我们也可以通过命令来告诉电脑一会临时把这个工作给做一下

总结：在我们LINUX中，我们可以通过crontab和at这两个东西来实现这些功能的

计划任务的作用：是做一些周期性的任务，在生产中的主要用来定期备份数据

CROND：这个守护进程是为了周期性执行任务或处理等待事件而存在

任务调度分两种：系统任务调度，用户任务调度

```
系统任务调度：系统周期性所要执行的工作，比如写缓存数据到硬盘、日志清理等。
在/etc目录下有一个crontab文件，这个就是系统任务调度的配置文件(/etc/crontab文件就是系统任务调度的配置文件)

用户任务调度：用户定期要执行的工作，比如用户数据备份、定时邮件提醒等。
用户可以使用 crontab 工具来定制自己的计划任务。所有用户定义的crontab文件都被保存
在/var/spool/cron目录中。其文件名与用户名一致。
```

计划任务的安排方式分两种:
一种是定时性的，也就是例行。就是每隔一定的周期就要重复来做这个事情
一种是突发性的，就是这次做完了这个事，就没有下一次了，临时决定，只执行一次的任务

at和crontab这两个命令：
at：它是一个可以处理仅执行一次就结束的指令
crontab：它是会把你指定的工作或任务，比如：脚本等，按照你设定的周期一直循环执行下去

2.at计划任务的使用

语法格式：at 时间

服务：atd

```
[root@exercise1 ~]# yum install -y at   #安装at服务
[root@exercise1 ~]# systemctl start atd   #开启atd服务
[root@exercise1 ~]# systemctl status atd   #查看atd服务状态
```

```
[root@exercise1 ~]# systemctl status atd
● atd.service - Job spooling tools
   Loaded: loaded (/usr/lib/systemd/system/atd.service; enabled; vendor preset: enabled)
   Active: active (running) since 三 2022-01-26 22:22:17 CST; 8s ago
 Main PID: 1143 (atd)
   CGroup: /system.slice/atd.service
           └─1143 /usr/sbin/atd -f

1月 26 22:22:17 exercise1 systemd[1]: Started Job spooling tools.
1月 26 22:22:17 exercise1 systemd[1]: Starting Job spooling tools...
```

实战-使用at创建计划任务

```
[root@exercise1 ~]# ls /opt/   #刚开始查的时候/opt/目录下是什么都没有的

[root@exercise1 ~]# date   #查看系统时间
2022年 01月 26日 星期三 22:26:04 CST

[root@exercise1 ~]# at 22:27 2022-01-26  #注意：如果是上午时间，后面加上am，比如9:20am
at> mkdir /opt/test  &&   #输入你要执行的命令
```

```
at> touch /opt/test/a.txt
at> <EOF>   #结束：ctrl+d
job 1 at Wed Jan 26 22:27:00 2022

[root@exercise1 ~]# ls /opt/test/   #检查at计划任务运行结果
a.txt

[root@home opt]# at 10:02 2022-07-20                #不能创建已过去的时间任务
at: refusing to create job destined in the past
```

3.查看和删除at将要执行的计划任务

这个查看，只能看到还没有执行的。如果这个任务已经开始执行或者执行完成了，是看不到的

```
[root@exercise1 ~]# at 11:21 2022-01-27
at> ls /opt/
at> <EOT>
job 5 at Thu Jan 27 11:21:00 2022
#其中atq等同于at -l
[root@exercise1 ~]# atq
5   Thu Jan 27 11:21:00 2022 a root
[root@exercise1 ~]# at -l
5   Thu Jan 27 11:21:00 2022 a root

[root@exercise1 ~]# at -c 7   #-c  打印任务的内容到标准输出,查看7号计划任务具体内容
```

```
[root@exercise1 ~]# at 11:21 2022-01-27
at> ls /opt/
at> <EOT>
job 5 at Thu Jan 27 11:21:00 2022
[root@exercise1 ~]# atq
5       Thu Jan 27 11:21:00 2022 a root
[root@exercise1 ~]# at -l
5       Thu Jan 27 11:21:00 2022 a root
```

3.1查看定时任务内容

```
[root@exercise1 ~]# at 11:28 2022-01-27   #创建一个at任务
at> ls /opt/
at> <EOT>
job 9 at Thu Jan 27 11:28:00 2022

[root@exercise1 ~]# ls /var/spool/at/   #at计时任务存放目录
a0000901a1e6b0  spool   #at任务创建成功且等待执行的时候才会产生像a0000901a1e6b0这样的文件

[root@exercise1 ~]# tail -5 /var/spool/at/a0000901a1e6b0   #就是查看at任务里面执行什么命令，在文件内容后面
}
${SHELL:-/bin/sh} << 'marcinDELIMITER64a6bde0'
ls /opt/

marcinDELIMITER64a6bde0
```

3.2删除at计划任务

语法：atrm 任务编号

```
[root@exercise1 ~]# date
2022年 01月 27日 星期四 13:56:02 CST
[root@exercise1 ~]# at 13:57 2022-01-27
at> ls /opt/
```

```
at> <EOT>
job 11 at Thu Jan 27 13:57:00 2022

[root@exercise1 ~]# at 13:57 2022-01-27
at> ll /opt/
at> <EOT>
job 12 at Thu Jan 27 13:57:00 2022

[root@exercise1 ~]#
[root@exercise1 ~]# at -l
11  Thu Jan 27 13:57:00 2022 a root
12  Thu Jan 27 13:57:00 2022 a root

[root@exercise1 ~]# atrm 11

或者

[root@exercise1 ~]# at -d 12
[root@exercise1 ~]# at -l   #查询是否还有待执行的任务
[root@exercise1 ~]# date   #通过系统时间显示还没有到之前设置的时间，证明已经删除掉了
2022年 01月 27日 星期四 13:56:50 CST
[root@exercise1 ~]#
#删除at任务两种方式：atrm at任务号 或者 at -d at任务号  ==> atrm 11等同于at -d 11
```

at计划任务的特殊写法

```
[root@exercise1 ~]#at 20:00 2021-8-8         #在某天
[root@exercise1 ~]#at now+10min                #在10分钟后执行
[root@exercise1 ~]#at 17:00 tomorrow          #明天下午5点执行
[root@exercise1 ~]#at 6:00pm +3days      #在3天以后的下午6点执行
[root@exercise1 ~]# at 14:18</opt/a.txt   #执行/opt/a.txt文件里面的内容
job 16 at Fri Jan 28 14:18:00 2022
```

4.crontab定时任务的使用

crond命令定期检查是否有要执行的工作，如果有要执行的工作便会自动执行该工作

cron是一个linux下的定时执行工具，可以在无需人工干预的情况下运行作业。

linux任务调度的工作主要分为以下两类：(其实就是前面提到的系统任务调度，用户任务调度)
系统执行的工作：系统周期性所要执行的工作，如更新locate数据库updatedb数据库，日志定期切割，收集系统状态信息，/tmp定期清理

个人执行的工作：某个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

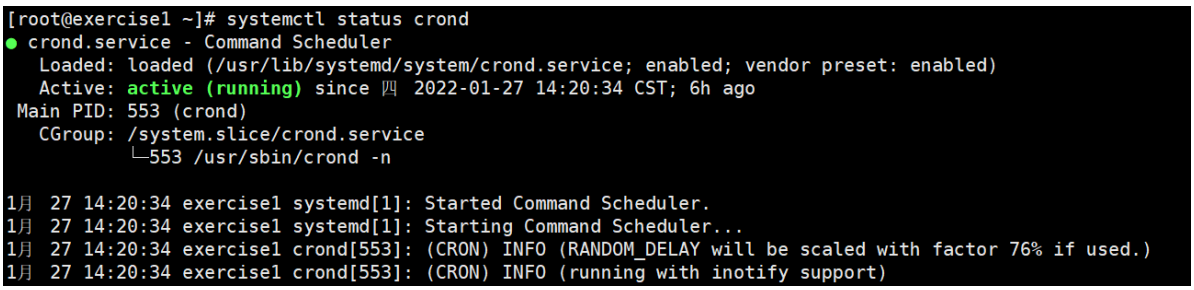
4.1启动crond服务

```
[root@exercise1 ~]# systemctl status crond   #查询crond服务状态
● crond.service - Command Scheduler
   Loaded: loaded (/usr/lib/systemd/system/crond.service; enabled; vendor preset: enabled)
   Active: active (running) since 四 2022-01-27 14:20:34 CST; 6h ago
 Main PID: 553 (crond)
   CGroup: /system.slice/crond.service
           └─553 /usr/sbin/crond -n

1月 27 14:20:34 exercise1 systemd[1]: Started Command Scheduler.
1月 27 14:20:34 exercise1 systemd[1]: Starting Command Scheduler...
1月 27 14:20:34 exercise1 crond[553]: (CRON) INFO (RANDOM_DELAY will be scaled with factor 76% if used.)
1月 27 14:20:34 exercise1 crond[553]: (CRON) INFO (running with inotify support)

```

```
[root@exercise1 ~]# systemctl start crond    #若是关闭状态输入此命令启动
[root@exercise1 ~]# systemctl enable crond    #设置crond服务为开机自启动
[root@exercise1 ~]# systemctl is-enabled crond    #查看crond服务的自启动状态
enabled
[root@exercise1 ~]# systemctl disable crond    #关闭crond.serivce的自启动状态
```



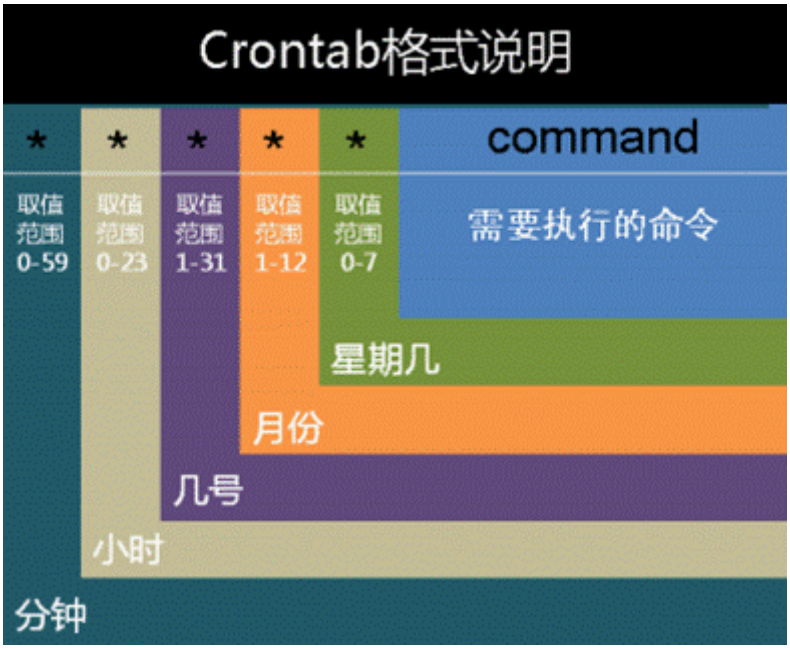
4.2.cron命令参数介绍

crontab的参数:
crontab -u hr　　#指定hr用户的cron服务
crontab -l　　#列出当前用户下的cron服务的详细内容
crontab -u abc -l　#只能root才能进行这个任务，列出指定用户abc下的cron服务的详细内容
crontab -r　　#删除cron服务（会全部删除计划任务）
crontab -e　　#编辑cron服务

例子

```
[root@exercise1 ~]# crontab -u abc -e    #编辑属于abc用户的计划任务
[root@exercise1 ~]# crontab -u root -l    #root查看自己的cron计划任务
no crontab for root
[root@exercise1 ~]# crontab -u lin05 -r    #root想删除lin05的cron计划任务
no crontab for lin05
[root@exercise1 ~]#
```

cron-e 编辑时的语法



注意

星期日用0或7表示

一行对应一个任务，特殊符号的含义:

符号	含义	举例
*	代表取值范围内的数字	(任意/每)
/	指定时间的间隔频率	*/6
-	代表从某个数字到某个数字	8-17
,	以分为例，第6和第10分钟	6,10

```
2.了解crontab的时间编写规范

00 02 * * * ls                    #每天的凌晨2点整执行

00 02 1 * * ls                    #每月的1日的凌晨2点整执行

00 02 14 2 * ls                   #每年的2月14日凌晨2点执行

00 02 * * 7 ls                    #每周日的凌晨2点整执行

```

```
00 02 * 6 5 ls                    #每年的6月周五凌晨2点执行

00 02 14 * 7 ls                   #每月14日和每周日的凌晨2点都执行

00 02 14 2 7 ls                   #每年的2月14日和每年2月的周天的凌晨2点执行

*/10 02 * * * ls                  #每天凌晨2点，每隔10分钟执行一次

* * * * * ls                      #每分钟都执行

00 00 14 2 * ls                   #每年2月14日的凌晨执行命令

*/5 * * * * ls                    #每隔5分钟执行一次

00 02 * 1，5，8 * ls               #每年的1月5月8月凌晨2点执行

00 02 1-8 * * ls                  #每月1号到8号凌晨2点执行
```

4.3.创建计划任务

例1：每天凌晨2点1分开始备份数据（编写计划任务的步骤）

```
第1步：书写对的命令
[root@exercise1 ~]# tar cvf/opt/grub2.tar.gz /boot/grub2

第2步：添加计划任务
```

```
[root@exercise1 ~]# crontab -e
#插入以下内容
1 2 * * *  tar cvf/opt/grub2.tar.gz  /boot/grub2
#保存退出提示的信息
no crontab for root - using an empty one
crontab: installing new crontab
[root@exercise1 ~]# crontab -l   #查看
1 2 * * * tar zcvf /opt/grub2.tar.gz /boot/grub2

第3步：测试与检查
[root@exercise1 ~]# date -s "02:00:50 2022-01-29"   #设置时间看效果
[root@exercise1 ~]# ll /opt/   #已成功备份
总用量 3044
-rw-r--r--. 1 root root 3114165 1月  29 02:01 grub2.tar.gz

看日志检查定时任务的日志
[root@exercise1 ~]# tail -f /var/log/cron

```

例2：黑客：以非root用户添加计划任务。最好使用已经存在系统用户添加。这里使用lin05用户来添加

```
[root@exercise1 ~]# crontab -u lin05 -e
#插入以下内容
*/1 * * * * echo "aaaaa" >> /opt/a.txt
#保存退出提示的信息
no crontab for lin05 - using an empty one
crontab: installing new crontab
[root@exercise1 ~]# crontab -u lin05 -l
*/1 * * * * echo "aaaaa" >> /opt/a.txt

[root@exercise1 ~]# crontab -l      #只能查当前用户的计划任务
```

互动：如何排查所有用户的计划任务?

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

[root@exercise1 ~]# ll /var/spool/cron/

总用量 8

-rw-------. 1 root root 39 1月 29 02:06 lin05
-rw-------. 1 root root 49 1月 27 21:26 root

所以后期可以使用这一招排查，黑客是否在你的机器中安装了定时任务

例3：此 run-parts 用法尽量写在/etc/crontab文件里，不要使用 crontab -e

```
* * * * *       root    /bin/echo "111111111" > /opt/a.txt       （直接写命令）
* * * * *       root    /opt/test.sh                （运行单个脚本）
* * * * *       root    run-parts    /opt            （运行此目录下的所有可执行程序）
```

注:写在这里的时间任务必须指明 运行该命令的用户身份
run-parts:可以执行某个目录下的所有脚本文件

```
01 * * * *      root            run-parts /etc/cron.hourly
02 4 * * *      root            run-parts /etc/cron.daily
22 4 * * 0      root            run-parts /etc/cron.weekly
42 4 1 * *      root            run-parts /etc/cron.monthly
```

/etc/crontab文件包括下面几行:

```
[root@exercise1 ~]# cat /etc/crontab

SHELL=/bin/bash                                 #执行命令的shell是哪个

PATH=/sbin:/bin:/usr/sbin:/usr/bin              #环境变量

MAILTO=root                                     #邮件发给谁

# For details see man 4 crontabs
```

```
10
11 # Example of job definition:
12 # .---------------- minute (0 - 59)
13 # |  .------------- hour (0 - 23)
14 # |  |  .---------- day of month (1 - 31)
15 # |  |  |  .------- month (1 - 12) OR jan,feb,mar,apr
   ...
16 # |  |  |  |  .---- day of week (0 - 6) (Sunday=0 or 7)
   OR sun,mon,tue,wed,thu,fri,sat
17 # |  |  |  |  |
18 # *  *  *  *  * user-name  command to be executed
```

前四行是用来配置crond任务运行的环境变量，第一行SHELL变量指定了系统要使用哪个shell，这里是bash，第二行PATH变量指定了系统执行命令的路径，第三行MAILTO变量指定了crond的任务执行信息将通过电子邮件发送给root用户，如果MAILTO变量的值为空，则表示不发送任务执行信息给用户，第四行的HOME变量指定了在执行命令或者脚本时使用的主目录。

---

crontab -e 命令与直接编辑/etc/crontab文件区别：

1. 使用crontab -e这个命令编辑的定时任务列表是属于用户级别的，初次编辑后在 /var/spool/cron 目录下生成一个与用户名相同的文件，文件内容就是我们的定时任务列表。如没有定时任务，这个文件就是空文件

2. 编辑 /etc/crontab 文件只有 root 用户才行，全局文件

3. crontab -e 会进行语法检查、直接编辑 /etc/crontab 文件则不会

4.在/etc/crontab文件里添加计划任务，用crontab -l是查不到的，也不会在 /var/spool/cron生成任务内容文件

注意环境变量问题

有时我们创建了一个crontab，但是这个任务却无法自动执行，而手动执行这个任务却没有问题，这种情况一般是由于在crontab文件中没有配置环境变量引起的。
在 crontab文件中定义多个调度任务时，需要特别注意的一个问题就是环境

变量的设置，因为我们手动执行某个任务时，是在当前shell环境下进行的，程序当然能找到环境变量，而系统自动执行任务调度时，是不会加载任何环境变量的，因此，就需要在crontab文件中指定任务运行所需的所有环境变量，这 样，系统执行任务调度时就没有问题了。
不要假定cron知道所需要的特殊环境，它其实并不知道。所以你要保证在shelll脚本中提供所有必要的路径和环境变量，除了一些自动设置的全局变量。所以注意如下3点：

1）脚本中涉及文件路径时写绝对路径；

2）脚本执行要用到java或其他环境变量时，通过source命令引入环境变量，如：
source /etc/profile
/usr/local/bin/run.sh

3）当手动执行脚本OK，但是crontab死活不执行时。这时必须大胆怀疑是环境变量惹的祸，并可以尝试在crontab中直接引入环境变量解决问题。
如：
*/1 * * * * source /etc/profile ; /opt/restart_audit.sh

crond注意的事项

1)给定时任务注释，表示说明定时任务主要作用

2)将需要定期执行的任务写入Shell脚本中，避免直接使用命令无法执行的情况tar date

3)定时任务的结尾一定要有 & > /dev/null或者将结果追加重定向 > /tmp/date.log文件

原因：会产生大量的邮件报警

4)注意有些命令是无法成功执行的 echo " 123 " > > tmp/test.log & > /dev/null

5)如果一定要是用命令，命令必须使用绝对路径

原因：定时任务执行命令的时候，只能识别/usr/bin 和/bin 下面的命令(针对crontab -e)

解决方法：1.命令使用绝对路径

2.重新配置下环境变量 source /etc/profile

6)如何拒绝某个用户使用

```
#1.使用root将需要拒绝的用户加入/etc/cron.deny

[root@exercise1 ~]# echo "abc">>/etc/cron.deny

#2.登陆该普通用户，测试是否能编写定时任务
[abc@exercise1 ~]$ crontab -e
You (abc) are not allowed to use this program (crontab)
See crontab(1) for more information

```

例4：

```
[root@exercise1 ~]# ls /etc/cron
cron.d/       cron.daily/   cron.deny     cron.hourly/
cron.monthly/ crontab
```

注：cron.d/　　#是系统自动定期需要做的任务，但是又不是按小时，按天，按星期，按月来执行的，那么就放在这个目录下面。
cron.deny　　#控制用户是否能做计划任务的文件,未记录到这个文件当中的用户，就可以使用crontab
cron.monthly/　　#每月执行的脚本;
cron.weekly/　　#每周执行的脚本;
cron.daily/　　#每天执行的脚本;
cron.hourly/　　#每小时执行的脚本;
crontab　　#主配置文件也可添加任务;

例5：crontab 最小执行时间是分钟，如果是需要半分钟执行 ，如果实现呢？，看如下

```
每30秒 把时间写入 /tmp/cron.txt 文件
*/1 * * * * date >> /tmp/cron.txt
* * * * * sleep 30s; touch data >> /tmp/cron.txt
```

计划任务如何调试

1.crond调试

- 增加频率：

  调整任务每分钟执行的频率，以便做后续的调试。

  调整系统时间

- *执行脚本输出到文件*（排查错误）

o如果使用cron运行脚本，请将脚本执行的结果写入指定日志文件，观察日志内容是否正常。

- 命令使用绝对路径，防止无法找到命令导致定时任务执行产生故障。
- 看日志：

o通过查看/var/log/cron日志，以便检查我们执行的结果，方便进行调试。

2.crond.编写思路

- 1.手动执行命令，然后保留执行成功的结果。
- 2.编写脚本

  o脚本需要绝对路径/scripts

  o脚本内容复制执行成功的命令（减少每个环节出错几率）

  o脚本内容尽可能的优化，使用一些变量或使用简单的判断语句

  o脚本执行的输出信息可以重定向至其他位置保留或写入/dev/null

- 3.执行脚本

  o使用bash命令执行，防止脚本没有增加执行权限(/usr/bin/bash)

  o执行脚本成功后，复制该执行的命令，以便写入cron

- 4.编写计划任务

  o加上必要的注释信息，人、时间、任务

  o设定计划任务执行的周期

  o粘贴执行脚本的命令（不要手敲）。

- 5.调试计划任务增加任务频率测试

  o检查环境变量问题

  o检查crond服务日志

8.实战-常见的计划任务写法和案例

**常见写法：

```
每天晚上21:00重启netowrk
* 21 * * * systemctl restart network

每月1、10、22日的4:45重启netowrk。
45 4 1,10,22 * * systemctl restart network

每月1到10日的4:45重启netowrk。
45 4 1-10 * * systemctl restart network

每隔两天的上午8点到11点的第3和第15分钟执行netowrk
3,15 8-11 */2 * * systemctl restart network

晚上11点到早上7点之间，每隔2小时重启netowrk
* 23-24,0-7/2 * * *  systemctl restart network

**周一到周五每天晚上21:15寄一封信给root@home
15 21 * * 1-5  mail -s root@home
```

案例要求：

每天22：00备份/etc/目录到/tmp/backup下面
将备份命令写入一个脚本中
在执行计划任务时，不要输出任务信息
每天备份文件名要求格式：2020-12-07_etc.tar.gz
存放备份内容的目录要求只保留三天的数据

```
第一种
tar -zcvf /tmp/backup/$(date +%F)_etc.tar.gz  /etc >/dev/null 2>&1
rm -rf /tmp/backup/$(date -d "-3days " +%F )_etc.tar.gz
```

```
第二种
date -d "-3days " +%F  |xargs -i rm -rf {}
```

```
第三种
mkdir  /tmp/backup
tar -zcvf /tmp/backup/$(date +%F)_etc.tar.gz   /etc
find  /tmp/backup -name "*.tar.gz" -mtime +3 -exec rm -f {} \;
```

[root@exercise1 ~]#crontab -l
* 22 * * * /opt/backup.sh >/dev/null
注：工作中备份的文件不要放到/tmp,因为过一段时间，系统会清空/tmp目录