



Linux操作系统

授课教师：刘二小 副教授

单位：杭州电子科技大学

通信工程学院



7.2 磁盘配额

■磁盘配额就是管理员可以为用户所能使用的磁盘空间进行配额限制，每一用户只能使用最大配额范围内的磁盘空间，避免了某些用户因为存储垃圾文件浪费磁盘空间导致其他用户无法正常工作，在Linux系统发行版本中使用quota来对用户进行磁盘配额管理。

■设置用户和组配额的分配量对磁盘配额的限制一般是从一个用户 **占用磁盘大小和所有文件的数量**两个方面来进行的。

■设置磁盘配额时，“某用户在系统中共计只能使用50MB磁盘空间”，这样的限制要求是无法实现的，只能设置“某用户在/dev/sda7（挂载到/home）分区能使用30MB，在/dev/sda5（挂载到/）分区能使用20MB”。磁盘配额的设置单位是分区，针对分区启用配额限制功能后才可以对用户设置，而不理会用户文件放在该文件系统中的哪个目录中，其他系统，如Unix、Windows，原理与Linux相同。



设置磁盘配额步骤：

- 查看内核是否支持配额；
- 安装磁盘配额工具
- 激活分区的配额功能
- 建立配额数据库
- 启动分区磁盘配额功能
- 设置用户和组磁盘配额；
- 设置宽限期

7.2.1 查看内核是否支持配额

■在配置磁盘配额前，需要检查系统内核是否支持quota，查看Ubuntu内核是否支持配额的命令如下：

■`#grep CONFIG_QUOTA /boot/config-3.2.0-29-generic`

说明：

(1) CONFIG_QUOTA一定要大写

(2) 版本不同文件名config-3.2.0-29-generic略有不同，**到/boot目录下查看。**

■在查看结果中CONFIG_QUOTA和CONFIG_QUOTACTL两项都等于y，说明当前的内核支持quota，如图所示。

```
lex@lex-virtual-machine:/boot$ grep CONFIG_QUOTA /boot/config-5.8.0-50-generic
CONFIG_QUOTA=y
CONFIG_QUOTA_NETLINK_INTERFACE=y
# CONFIG_QUOTA_DEBUG is not set
CONFIG_QUOTA_TREE=m
CONFIG_QUOTACTL=y
CONFIG_QUOTACTL_COMPAT=y
```

7.2.2 安装磁盘配额工具

- 配额软件默认是没有安装的，因此，需要安装quota和quotatool软件包来管理硬盘配额，步骤如下：
- 更新软件包 `sudo apt-get update`
- 安装 `sudo apt-get install quota quotatool`

```
lex@lex-virtual-machine:/boot$ sudo apt-get install quota quotatool
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
下列软件包是自动安装的并且现在不需要了：
  libboost-program-options1.71.0 linux-headers-5.8.0-45
  linux-headers-5.8.0-45-generic linux-image-5.8.0-45-generic
  linux-modules-5.8.0-45-generic
  linux-modules-extra-5.8.0-45-generic
使用'sudo apt autoremove'来卸载它(它们)。
建议安装：
  libnet-ldap-perl rpcbind default-mta | mail-transport-agent
下列【新】软件包将被安装：
  quota quotatool
升级了 0 个软件包，新安装了 2 个软件包，要卸载 0 个软件包，有 180 个
软件包未被升级。
需要下载 272 kB 的归档。
解压缩后会消耗 1,583 kB 的额外空间。
获取:1 http://mirrors.tuna.tsinghua.edu.cn/ubuntu groovy/main amd64 q
uota amd64 4.05-1 [254 kB]
获取:2 http://mirrors.tuna.tsinghua.edu.cn/ubuntu groovy/universe amd
```

7.2.3 激活分区的配额功能

1、激活分区的配额功能步骤如下：

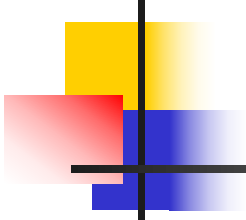
■ 建立目录，把要激活的分区挂载到此目录下。

```
# mkdir /myquota
```

2、因为用root用户建立的目录，而要对lex用户在这个目录中挂载磁盘配额，则这目录的owner、group都要改为lex，如下图所示

```
# chown lex:lex /myquota
```

```
root@lex-virtual-machine:/# ls -l
total 945460
lrwxrwxrwx 1 root root 7 3月 20 15:05 bin -> usr/bin
drwxr-xr-x 4 root root 4096 4月 17 10:44 boot
drwxrwxr-x 2 root root 4096 3月 20 15:08 cdrom
drwxr-xr-x 19 root root 4280 4月 18 09:24 dev
drwxr-xr-x 136 root root 12288 4月 17 22:52 etc
drwxr-xr-x 9 root root 4096 4月 17 15:15 home
lrwxrwxrwx 1 root root 7 3月 20 15:05 lib -> usr/lib
lrwxrwxrwx 1 root root 9 3月 20 15:05 lib32 -> usr/lib32
lrwxrwxrwx 1 root root 9 3月 20 15:05 lib64 -> usr/lib64
lrwxrwxrwx 1 root root 10 3月 20 15:05 libx32 -> usr/libx32
drwx----- 2 root root 16384 3月 20 15:05 lost+found
drwxr-xr-x 3 root root 4096 4月 9 21:08 media
drwxr-xr-x 6 root root 4096 4月 17 22:32 mnt
drwxr-xr-x 2 lex lex 4096 4月 18 09:26 myquota
drwxr-xr-x 2 root root 4096 10月 22 21:58 opt
dr-xr-xr-x 382 root root 0 4月 18 09:24 proc
drwx----- 7 root root 4096 4月 17 22:48 root
drwxr-xr-x 37 root root 960 4月 18 09:24 run
lrwxrwxrwx 1 root root 8 3月 20 15:05 sbin -> usr/sbin
drwxr-xr-x 8 root root 4096 10月 22 22:19 snap
drwxr-xr-x 2 root root 4096 10月 22 21:58 srv
-rw----- 1 root root 968064000 3月 20 15:05 swapfile
dr-xr-xr-x 13 root root 0 4月 18 09:24 sys
drwxrwxrwt 20 root root 4096 4月 18 09:25 tmp
drwxr-xr-x 14 root root 4096 10月 22 21:58 usr
drwxr-xr-x 15 root root 4096 3月 20 21:42 var
```

3、对分区使用磁盘配额，选择进行磁盘配额的分区后，要让分区的文件系统支持配额，就要修改/etc/fstab文件。

/etc/fstab文件的作用：系统开机时会主动读取/etc/fstab这个文件中的内容，根据文件里面的配置挂载磁盘。这样我们只需要将磁盘的挂载信息写入这个文件中我们就不需要每次开机启动之后手动进行挂载了。

先用cp命令做个备份

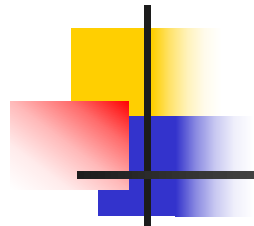
cp /etc/fstab etc/fstab1

例：在磁盘分区/dev/sdb1启用磁盘配额，该分区为ext3文件系统（df-T查看），挂载到/myquota，使用如下命令修改/etc/fstab文件：

```
#vi /etc/fstab （gedit /etc/fstab ）
```

在/etc/fstab文件末尾添加如下行：

```
/dev/sdb1    /myquota    ext3    defaults,usrquota 0 0
```



重启系统让/etc/fstab文件生效，或执行命令：

(1) `#sudo mount -a` (挂载 /etc/fstab 里面的内容)

也可以使用如下命令：

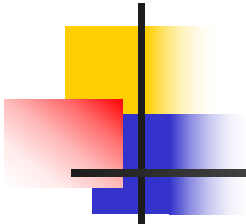
(2) `mount -o remount,usrquota /myquota`

重新挂载/myquota,启用用户配额usrquota, grpquota则是是启用组配额)

有时需要上面两条命令都执行。

7.2.4 建立配额数据库

- 实现磁盘配额，系统**必须生成**并维护相应的数据库文件aquota.user，用户的配额设置信息及磁盘使用的块、索引结点等相关信息被保存在**aquota.user 数据库中**，实现组磁盘配额，组的配额设置信息及磁盘使用的块、索引结点等相关信息被保存在**aquota.grp数据库中**。
- 扫描相应文件系统，**用quotacheck命令生成基本配额文件**，运行quotacheck命令，quotacheck命令检查启用了配额的文件系统，并为每个文件系统建立一个当前磁盘用的表（数据库）。该表会被用来更新操作系统的磁盘用量文件，此外，文件系统的磁盘配额文件也被更新。
- 格式： `quotacheck -avug` 建立配额数据库
- 所用选项如下：
 - a—指定每个启用了配额的文件系统都应该创建配额文件
 - v—在检查配额过程中显示详细的状态信息
 - u—检查用户磁盘配额信息
 - g—检查组群磁盘配额信息



```
root@lex-virtual-machine:/myquota# ls
lost+found
root@lex-virtual-machine:/myquota# sudo quotacheck -av
quotacheck: Your kernel probably supports journaled quota but you are
it. Consider switching to journaled quota to avoid running quotacheck
unclean shutdown.
quotacheck: Scanning /dev/sdb1 [/myquota] done
quotacheck: Cannot stat old user quota file /myquota/aquota.user: No
or directory. Usage will not be subtracted.
quotacheck: Old group file name could not be determined. Usage will
b Software & Updates
quotacheck: checked 3 directories and 0 files
quotacheck: Old file not found.
root@lex-virtual-machine:/myquota# ls -l
total 24
-rw----- 1 root root 6144 4月 18 09:49 aquota.user
drwx----- 2 root root 16384 4月 17 22:28 lost+found
root@lex-virtual-machine:/myquota#
```



7.2.5 启动磁盘配额

■ 使用quotaon命令启动磁盘配额，格式为：

quotaon [选项] [设备名或挂载点]

#quotaon -av

■ 其中常用选项及含义如下：

-a：不用指明具体的分区，在启用配额功能的所有文件系统上创建数据库

-v：显示启动过程

```
root@lex-virtual-machine:/myquota# sudo quotaon -av
/dev/sdb1 [/myquota]: user quotas turned on
root@lex-virtual-machine:/myquota#
```



7.2.6 编辑用户磁盘配额

1、编辑用户磁盘配额

■ 要为用户配置配额，以超级用户身份在 shell 提示下执行以下命令：

`edquota username`

■ 其中常用选项及含义如下：

-u : 配置用户配额

-g : 配置组配额

-t : 编辑宽限时间

-p : 复制quota资料到另一用户上

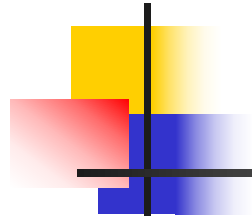


在具体操作之前，先了解一下磁盘配额的两个基本概念：软限制和硬限制。

◆软限制：一个用户在一定时间范围内（默认为一周，可以使用命令“edquota -t”重新设置，时间单位可以为天、小时、分钟、秒）超过其限制的额度，在不超出硬限制的范围内可以继续使用空间，系统会发出警告（警告信息设置文件为“etc/warnquota.conf”），但如果用户达到时间期限仍未释放空间到限制的额度下，系统将不再允许该用户使用更多的空间。

例如：我们对文件数量设置的配额，最小文件数是2，最大是5，在规定的时间内（宽限期），可以超过2个文件，不能超过5个文件。

◆硬限制：一个用户可拥有的磁盘空间或文件的绝对数量，绝对不允许超过这个限制。



配置lex1用户的磁盘配额，输入：

#edquota -u lex1
soft :10240
hard :40960
soft:2
hard :10

```
root@lex-virtual-machine: /myquota
GNU nano 5.2 /tmp//EdP.acZ5mFp
Disk quotas for user lex1 (uid 1001):
Filesystem      blocks      soft      hard      inodes      soft      hard
/dev/sdb1         0      10240    409600         0         2        10
```

Rhythmbox

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^_ Go To Line M-E Redo

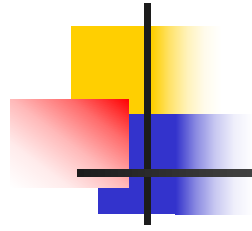
2、显示用户的配额

■ 编辑磁盘配额完成后，可以显示用户的配额，命令如下：

quota -u 用户名

quota 命令显示磁盘使用情况和限额。缺省情况下，或者带 -u 标志，只显示用户限额。

```
disk quotas for user rootnet (uid 1000):  
文件系统块数量 配额 规限宽限期文件节点 配额 规限宽限期  
/dev/sdb1      4 10240 40960      1 2 10  
root@maline: virtual-machine: /myquota#
```

7.2.7配额宽限期设置

- 使用容量超过soft limit, 宽限时间自动启动, 使用者将容量降低到soft limit以下, 宽限时间自动关闭, 假如使用者没有在宽限时间内将容量降低到soft limit, 那么他将无法再写入数据, 即使使用容量没有到达hard limit。

编辑宽限时间的命令为:

```
#edquota -t
```

7.2.8关闭磁盘配额

- 使用quotaoff命令终止磁盘配额的限制, 例如, 关闭/mnt/sdb1磁盘空间配额的命令:

```
# sudo quotaoff /dev/sdb1
```

提示: 机器启动后如果磁盘配额有问题, 先关闭磁盘配额再重新启动。



7.3 内存管理

- 直接从物理内存读写数据要比从硬盘读写数据要快的多，因此，我们希望所有数据的读取和写入都在内存完成，而内存是有限的，这样就有了物理内存与虚拟内存。
- 物理内存就是系统硬件提供的内存大小，是真正的内存，相对于物理内存，在linux下还有一个虚拟内存，虚拟内存就是为了满足物理内存的不足而提出的，它是利用**磁盘空间**虚拟出的一块逻辑内存，用作虚拟内存的磁盘空间被称为交换空间（Swap Space）。
- 作为物理内存的扩展，linux会在物理内存不足时，**使用交换分区的虚拟内存**，内核会将**暂时不用的物理内存块信息**写到交换空间，这样，物理内存得到了**释放**，这块内存就可以用于其它目的，当需要用到原始的内容时，这些信息会被重新从交换空间读入物理内存。
- linux的内存管理采取的是**分页存取机制**，为了保证物理内存能得到充分的利用，内核会在适当的时候将物理内存中不经常使用的数据块自动交换到虚拟内存中，而将经常使用的信息保留到物理内存。



交换分区swap

- SWAP就是Linux下的虚拟内存分区,它的作用是在物理内存使用完之后,将磁盘空间(也就是SWAP分区)虚拟成内存来使用。
- 虽然这个SWAP分区能够作为"虚拟"的内存,但它的速度比物理内存慢,因此如果需要更快速度的话,SWAP分区不能满足,最好的解决办法是加大物理内存.SWAP分区只是临时的解决办法。
- 交换分区 (swap) 的合理值一般在物理内存的2 倍左右,可以适当加大,具体还是以实际应用为准。
- Linux下可以创建两种类型的交换空间, 一种是swap分区, 一种是swap文件。前者适合有空闲的分区可以使用, 后者适合于没有空闲的硬盘分区, 硬盘的空间都已经分配完毕



7.4 进程管理

- 进程是指处于运行状态的程序，一个源程序经过编译、链接后，成为一个可以运行的程序。当该可执行的程序被系统加载到内存空间运行时，就称为进程。
- 程序是静态的保存在磁盘上的代码和数据的组合，而进程是动态概念。

7.4.1 常用进程管理命令

1、ps命令查看进程

使用权限：所有使用者

使用方式：ps [options] [--help]

说明：显示瞬间进程 (process) 的动态

参数：

ps 的参数非常多, 在此仅列出几个常用的参数并介绍含义：

-A 列出所有的行程（包括：

PID,TTY,TIME,COMMAND）

-u 用户名：显示指定用户的所有进程

-aux详细显示包含所有使用者的进程

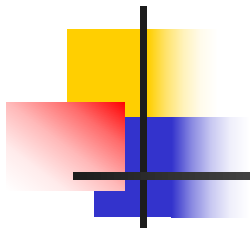
-axl 显示所有包含其他使用者的行程(包括优先级)

```
root@lex-virtual-machine:/home/lex# ps -a
  PID TTY          TIME CMD
 1132 tty2        00:00:15 Xorg
 1249 tty2        00:00:00 gnome-session-b
 4908 pts/1        00:00:00 su
 4909 pts/1        00:00:00 bash
 4916 pts/1        00:00:00 ps
root@lex-virtual-machine:/home/lex#
```

显示指定用户的进程

```
root@lex-virtual-machine:/home/lex# ps -u lex
```

PID	TTY	TIME	CMD
1110	?	00:00:00	systemd
1112	?	00:00:00	(sd-pam)
1118	?	00:00:06	pulseaudio
1121	?	00:00:00	tracker-miner-f
1123	?	00:00:00	gnome-keyring-d
1127	tty2	00:00:00	gdm-x-session
1129	?	00:00:00	dbus-daemon
1132	tty2	00:00:16	Xorg
1140	?	00:00:00	gvfsd
1155	?	00:00:00	gvfsd-fuse
1158	?	00:00:00	gvfs-udisks2-vo
Rhythmbox		00:00:00	gvfs-goa-volume
1175	?	00:00:00	goa-daemon
1184	?	00:00:00	goa-identity-se
1189	?	00:00:00	gvfs-afc-volume
1195	?	00:00:00	gvfs-mtp-volume
1199	?	00:00:00	gvfs-gphoto2-vo
1249	tty2	00:00:00	gnome-session-b
1314	?	00:00:00	ssh-agent
1332	?	00:00:00	at-spi-bus-laun



详细显示所有用户的进程

```
root@lex-virtual-machine:/home/lex# ps -axu
USER          PID  %CPU  %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root           1   0.0   0.5 169044 11708 ?        Ss   09:24   0:03 /sbin/i
root           2   0.0   0.0     0     0 ?        S    09:24   0:00 [kthrea
root           3   0.0   0.0     0     0 ?        I<   09:24   0:00 [rcu_gp
root           4   0.0   0.0     0     0 ?        I<   09:24   0:00 [rcu_pa
root           6   0.0   0.0     0     0 ?        I<   09:24   0:00 [kworke
root           9   0.0   0.0     0     0 ?        I<   09:24   0:00 [mm_per
root          10   0.0   0.0     0     0 ?        S    09:24   0:00 [ksofti
root          11   0.0   0.0     0     0 ?        I    09:24   0:01 [rcu_sc
root          12   0.0   0.0     0     0 ?        S    09:24   0:00 [migrat
root          13   0.0   0.0     0     0 ?        S    09:24   0:00 [idle_i
root          14   0.0   0.0     0     0 ?        S    09:24   0:00 [cpuhp/
root          15   0.0   0.0     0     0 ?        S    09:24   0:00 [cpuhp/
root          16   0.0   0.0     0     0 ?        S    09:24   0:00 [idle_i
root          17   0.0   0.0     0     0 ?        S    09:24   0:00 [migrat
root          18   0.0   0.0     0     0 ?        S    09:24   0:00 [ksofti
root          20   0.0   0.0     0     0 ?        I<   09:24   0:00 [kworke
root          21   0.0   0.0     0     0 ?        S    09:24   0:00 [kdevtm
```




2. 用top命令监控进程

top命令用来监控Linux进程的整体状态，从终端中运行，用当前运行进程一览表取代命令行，每隔几秒更新一次，对任何按下的单键做出反应。

q	退出
h or ?	帮助
s	设定两次更新之间的时间（以秒为单位）
space	更新显示
M（大写）	根据内存大小对进程排序
P（大写）	根据CPU（处理器）占用对进程排序
u	显示特定用户的进程
k	杀死进程（给进程发送信号）
r	更改进程优先级



3、kill命令结束进程

当需要中断一个前台进程的时候，通常是使用< Ctrl+c >组合键；但是对于一个后台进程用组合键不能中断了，这时就必须使用于kill命令。该命令可以终止后台进程。终止后台进程的原因很多，或许是该进程占用的CPU时间过多；或许是该进程已经挂死。总之这种情况是经常发生的。

(1) kill命令是通过向进程发送指定的信号来结束进程的。

例如： **kill 2330** 或者 **kill -15 2330**

当程序接收到SIGTERM的信号后，将会发生以下的事情：

程序立刻停止；当程序释放相应资源后再停止；程序可能仍然继续运行。

(2) 指定信号终止指定进程。

格式： **kill [-s 信号] 进程号 ...**

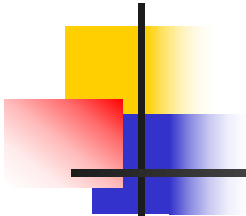
-s 指定需要送出的信号。

例如：终止 3710进程 **kill -s 9 3710**



注意：

- (1) 不指定信号将发送SIGTERM (15) 终止指定进程。如果仍无法终止该进程，发送信号SIGKILL(9)，将强制结束进程。
- (2) 注意，信号使进程强行终止，这常会带来一些副作用，如数据丢失或者终端无法恢复到正常状态。

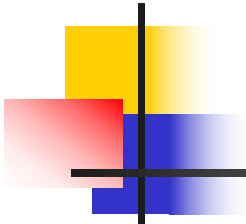


kill -l 显示信号的名称列表

例：显示信号的名称列表

```
root@lex-virtual-machine:/home/lex# kill -l
 1) SIGHUP      2) SIGINT      3) SIGQUIT     4) SIGILL      5) SIGTRAP
 6) SIGABRT     7) SIGBUS     8) SIGFPE      9) SIGKILL     10) SIGUSR1
11) SIGSEGV    12) SIGUSR2   13) SIGPIPE    14) SIGALRM    15) SIGTERM
16) SIGSTKFLT  17) SIGCHLD   18) SIGCONT    19) SIGSTOP    20) SIGTSTP
21) SIGTTIN    22) SIGTTOU   23) SIGURG     24) SIGXCPU    25) SIGXFSZ
26) SIGVTALRM  27) SIGPROF   28) SIGWINCH   29) SIGIO       30) SIGPWR
31) SIGSYS     34) SIGRTMIN  35) SIGRTMIN+1 36) SIGRTMIN+2 37) SIGRTMIN+3
38) SIGRTMIN+4 39) SIGRTMIN+5 40) SIGRTMIN+6 41) SIGRTMIN+7 42) SIGRTMIN+8
43) SIGRTMIN+9 44) SIGRTMIN+10 45) SIGRTMIN+11 46) SIGRTMIN+12 47) SIGRTMIN+13
4 LibreOffice Writer 49) SIGRTMIN+15 50) SIGRTMAX-14 51) SIGRTMAX-13 52) SIGRTMAX-12
53) SIGRTMAX-11 54) SIGRTMAX-10 55) SIGRTMAX-9  56) SIGRTMAX-8  57) SIGRTMAX-7
58) SIGRTMAX-6  59) SIGRTMAX-5 60) SIGRTMAX-4  61) SIGRTMAX-3  62) SIGRTMAX-2
63) SIGRTMAX-1  64) SIGRTMAX
```

<https://www.cnblogs.com/wangcp-2014/p/5146343.html> 详细内容



只有第9种信号(SIGKILL)才可以无条件终止进程，其他信号进程都有权利忽略。下面是常用的信号：同上

SIGINT 2 中断（同 Ctrl + C）

SIGQUIT 3 退出（同 Ctrl + \）

SIGTERM 15 终止

SIGKILL 9 强制终止

SIGSTOP 19 暂停（同 Ctrl + Z）



4、nice启动优先值命令

格式：nice [-n] 优先级的范围

说明：

- 1、优先级的范围为-20 ~ 19等40个等级，其中数值越小优先级越高，数值越大优先级越低，既-20的优先级最高，19的优先级最低。若调整后的程序运行优先级高于-20，则就以优先级-20来运行命令行；若调整后的进程优先级低于19，则就以优先级19来运行命令行。
- 2、若 nice命令未指定优先级的调整值，则以缺省值10来调整程序运行优先级，既在当前进程运行优先级基础之上增加10，优先级变低。
- 3、若不帶任何参数运行命令nice，则显示出当前的程序运行优先级。



5、renice改变正在运行的进程

- 通过修改pid来重新指定（不是修正）一个或多个进程 (Process)的优先级（值）。
- -p pid 重新指定进程pid 的优先级。
- -g pgrp 重新指定进程群组(process group)的 Pid 为 pgrp 的进程 (一个或多个) 的优先级。
- -u user 重新指定进程所有者为 user 的进程的优先级。

6、jobs显示后台执行的任务

■ 显示后台正在运行的进程，如图所示。 & 表示后台运行

```
ubuntu@linux:~/桌面$ sudo shutdown -h 21:00 &
[1] 3108
ubuntu@linux:~/桌面$
来自ubuntu@linux的广播信息
(/dev/pts/0) 于 18:11 ...

The system is going down for halt in 169 minutes!
^C
ubuntu@linux:~/桌面$ jobs
[1]+  运行中                  sudo shutdown -h 21:00 &
ubuntu@linux:~/桌面$
```

7.5任务计划

■对于密集访问磁盘的进程，希望它能够在每天非负荷的高峰时间段运行，可以通过指定任务计划使某些进程在后台运行。

(1)at命令： at命令用来向atd守护进程（自动运行任务）提交需要在特定时间运行的作业，在一个指定的时间执行任务，只能执行一次。

at [选项] [时间日期]

选项	作用
-f filename	运行由filename指定的脚本
-m	完成时，用电子邮件通知用户，即便没有输出
-l	列出所提交的作业
-r	删除一个作业

安装at（默认是没安装）

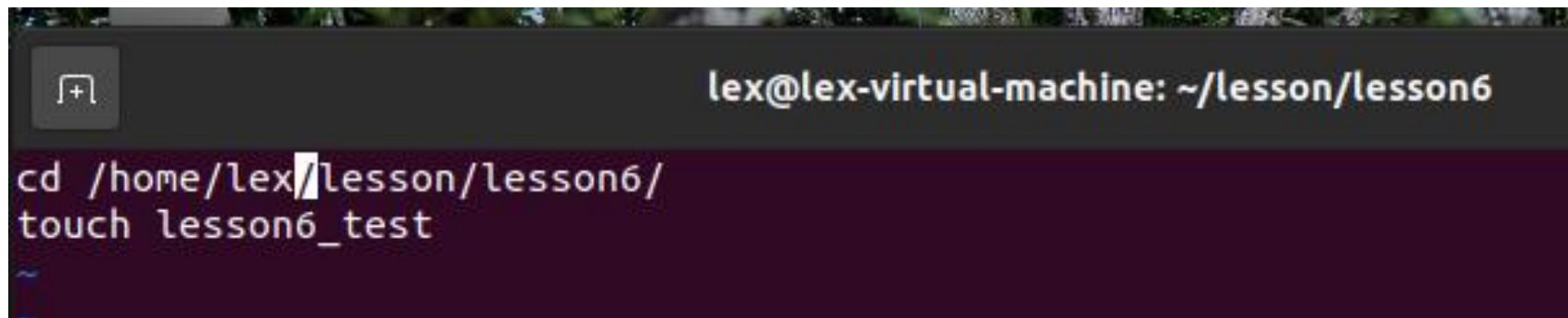
sudo apt-get update

sudo apt-get install at

```
lex@lex-virtual-machine:~$ sudo apt-get install at
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
下列软件包是自动安装的并且现在不需要了:
  libboost-program-options1.71.0
使用'sudo apt autoremove'来卸载它(它们)。
将会同时安装下列软件:
  libfl2
建议安装:
  default-mta | mail-transport-agent
下列【新】软件包将被安装:
  at libfl2
升级了 0 个软件包，新安装了 2 个软件包，要卸载 0 个软件包，有 180 个软件包未被升级。
需要下载 50.0 kB 的归档。
解压后会消耗 241 kB 的额外空间。
您希望继续执行吗？ [Y/n] y
获取:1 http://mirrors.tuna.tsinghua.edu.cn/ubuntu groovy/main amd64 libfl2 amd64 2.6.4-4 [38.5 kB]
获取:2 http://mirrors.tuna.tsinghua.edu.cn/ubuntu groovy/main amd64 at amd64 3.1.23-1 [38.5 kB]
```

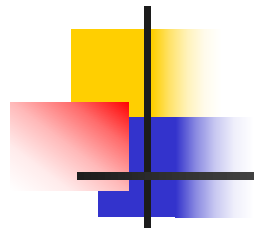


例：在11:18执行/home/lesson/lesson6/lt1



```
lex@lex-virtual-machine: ~/lesson/lesson6
cd /home/lex/lesson/lesson6/
touch lesson6_test
~
```

```
lex@lex-virtual-machine:~/lesson/lesson6$ at -f /home/lex/lesson/lesson6/lt1 11:18
warning: commands will be executed using /bin/sh
job 3 at Sun Apr 18 11:18:00 2021
lex@lex-virtual-machine:~/lesson/lesson6$ ls
a.txt  lt1
lex@lex-virtual-machine:~/lesson/lesson6$ ls
a.txt  lesson6_test  lt1
lex@lex-virtual-machine:~/lesson/lesson6$
```

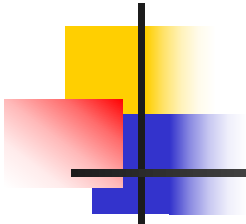


(2)Batch命令：任意时间执行

- batch命令不在特定时间运行，而是等到系统不忙于别的任务时运行，batch守护进程会监控系统的平均负载。
- **batch命令（不用安装）**，键入batch命令后，“at>”提示就会出现。键入要执行的命令，可以指定多条命令，方法是键入每一条命令后按Enter键。键入所有命令后，按Enter键转入一个空行，然后再键入“**Ctrl+D**”退出。
- 也可以输入shell脚本。

(3)crontab命令:定时任务配置

- cron是系统主要的调度进程，可以在无需人工干预的情况下运行任务计划，由crontab命令来设定cron服务。
- crontab命令允许用户提交、编辑或删除相应的作业。每一个用户都可以有一个crontab文件来保存调度信息。可以使用它周期性的运行任意一个shell脚本或某个命令。系统管理员是通过cron.deny和cron.allow这两个文件来禁止或允许用户拥有自己的crontab文件。
- 格式：crontab [选项] [用户名]



●格式：crontab [选项] [用户名]

选项	用法
-l	显示用户的crontab文件的内容（显示自己的crontab文件的内容）
-i	删除用户的Crontab文件前给提示
-r	从Crontab目录中删除用户的Crontab文件
-e	编辑用户的Crontab文件

用户建立的crontab文件名与用户名一致，存于/var/spool/cron/crontabs/中。crontab文件格式共分为六个字段(先后顺序)，**第六个字段为所要执行的命令**：

字段	含义	取值范围
1	分钟	0~59
2	小时	0~23
3	日期	1~31
4	月份	1~12
5	星期	0~6

例：crontab -e编辑文件，在57分把/home/lex/lesson/lesson6/1.txt强制复制为该目录下的1_crontest.txt文件。

```
# (/tmp/crontab.HF15Tr/crontab installed on Sun Apr 18 11:52:53 2021)
# (Cron version -- $Id: crontab.c,v 2.13 1994/01/17 03:20:37 vixie Exp $)
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
57 * * * * cp -f /home/lex/lesson/lesson6/1.txt /home/lex/lesson/lesson6/1_crontest.txt
"root" 27L, 1178C
```

```
lex@lex-virtual-machine:~/lesson/lesson6$ ls -l
total 16
-rw-r--r-- 1 root root 40 4月 18 11:57 1_crontest.txt
-rw-r--r-- 1 root root 40 4月 18 11:53 1.txt
-rw-rw-r-- 1 lex lex 2882 4月 13 08:31 a.txt
-rw-rw-r-- 1 lex lex 0 4月 18 11:18 lesson6_test
-rw-rw-r-- 1 lex lex 48 4月 18 11:17 lt1
-rw-rw-r-- 1 lex lex 0 4月 18 11:28 t1_1
lex@lex-virtual-machine:~/lesson/lesson6$
```