



# 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



## 第6章 Linux系统用户管理

---

### 6.1 Linux用户

### 6.2 Linux用户组

### 6.3 用户和用户组管理命令

### 6.4 su和sudo



## 6.1 Linux用户

---

### 6.1.1 用户和用户组

### 6.1.2 用户分类

### 6.1.3 用户相关文件



## 6.1.1 用户和用户组

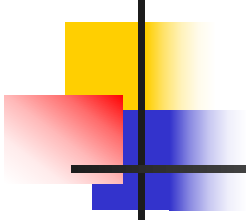
---

- Linux 系统是一个 多用户多任务 的 分时 操作系统，任何一个要使用系统资源的用户都必须先向系统管理员申请一个账号，然后以这个账号的身份进入系统。
- 每个用户账号都拥有一个唯一的用户名和各自的口令。用户在登录时键入正确的用户名和口令后，就能够进入系统和自己的主目录。
- 在Linux系统中，任何文件都属于某一特定用户，而任何用户都隶属于至少一个用户组，默认情况下，用户的组名和用户名相同。

## 6.1.1 用户和用户组

- 用户名 (username)：任何一个要使用系统资源的用户都有一个唯一的用户名和各自的口令，登录系统时，通过用户名来标记用户自身，进入自己的主目录。
- 例：系统在建立lex用户的同时，在home文件夹下产生一个以该用户名命名的文件夹home/lex/，与该用户相关的文件都存储在此文件夹下，登录时也是进入该目录。
- 如图所示，在/home/目录下有三个用户，在每个以用户名命名的文件夹下存放着各个用户自己的文件，登录时直接进入该目录。

```
lex@lex-virtual-machine:/home$ pwd
/home
lex@lex-virtual-machine:/home$ ls -l
total 12
drwxr-xr-x 19 lex  lex  4096  4月 13 08:31 lex
drwxr-xr-x  2 lex1 lex1  4096  4月 10 15:36 lex1
drwxr-xr-x  2 lex2 lex2  4096  4月 13 13:49 lex2
```

- 
- 用户ID：每个用户不仅有唯一的用户名，还有唯一的用户id，用户id缩写为uid。对于系统内核来说，它使用uid来记录拥有进程或文件的用户。
  - Linux系统分配的uid是一个32位的整数，即最多可以有 $2^{32}$ 个不同的用户。
  - 系统有一个数据库，存放着用户名与uid的对应关系，这个数据库存在配置文件/etc/passwd中，系统上的大多数用户都有权限读取这个文件，但是不能进行修改。

## 6.1.2 用户分类

- Ubuntu系统的安全性和多功能，依赖于如何给用户分配权限以及对其的使用方法。
- 用户分为3类：普通用户、超级用户、系统用户，其uid有不同的取值范围。

用户id范围	用户类型
0	根用户(超级用户)
1~999	系统用户
1000+	普通用户(Ubuntu 是1000+)

使用 `id root/lex1` 查看

```
root:x:0:0:root:/root:/bin/bash
```

```
lex1:x:1001:1001:,,,:/home/lex1:/bin/bash  
lex2:x:1002:1002:,,,:/home/lex2:/bin/bash
```



## ➤ 普通用户

普通用户是使用系统最多的人群，用户主目录为/home/用户名。普通用户的权限不是很高，一般情况下只在自己的主目录和系统范围内的临时目录中创建文件。

## ➤ 超级用户

超级用户又称为root用户或系统管理员，使用/root作为主目录。在系统上拥有最高权限：可以修改和删除任何文件、可以运行任何命令、可以取消任何进程、增加和保留其他用户、配置添加系统软硬件。超级用户的uid、gid都为0。





## ➤ 系统用户(伪用户)

- 大多数Linux系统会将一些低uid保留给系统用户。系统用户不代表人，而代表系统的组成部分，例如处理电子邮件的进程经常以用户名mail来运行；运行Apache网络服务器的进程经常作为用户apache来运行。
- 因为不是真正的用户，所以系统用户没有登录shell，其主目录也很少在/home中，而在属于相关应用的系统目录中，例如mail的目录在/var/mail。

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
```

## 6.1.3 用户相关文件

### 1、/etc/passwd文件

Linux系统的用户信息保存在配置文件/etc/passwd中，该文件是可读格式的文本，管理员可以利用文本编辑器来修改。而系统的大多数用户没有权限修改它，只能读取这个文件。

在passwd中，系统的每一个合法用户账号对应于该文件中的一行记录，这行记录定义了每个用户账号的属性。这些记录是按照uid排序的，首先是root用户，然后是系统用户，最后是普通用户。用户数据按字段以冒号分隔，格式如下：

格式：username:password:uid:gid:userinfo（普通用户通常省略）:home:shell

字段名	编号	说明
username	1	给一个用户可读的用户名称
password	2	加密的用户密码
uid	3	用户ID，Linux内核用这个整数来识别用户
gid	4	用户组ID，Linux内核用这个整数识别用户组
userinfo	5	用户名描述，
home	6	当用户登录时
shell	7	登录shell是用

```
sssd:x:126:131:sssd system user,,,:/var/lib/sss:/usr/sbin/nologin
lex:x:1000:1000:lex,,,:/home/lex:/bin/bash
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
samba1:x:1001:1001:,,,:/home/samba1:/bin/bash
_rpc:x:127:65534::/run/rpcbind:/usr/sbin/nologin
statd:x:128:65534::/var/lib/nfs:/usr/sbin/nologin
mysql:x:129:135:MySQL Server,,,:/nonexistent:/bin/false
samba2:x:1002:1002:,,,:/home/samba2:/bin/bash
```

## 6.1.3 用户相关文件

### 2、/etc/shadow文件

- 用户的加密密码被保存在/etc/passwd文件的第二个字段中。
- 在Linux和Unix系统中，采用更新的“影子密码”技术来保存密码，用户的密码被保存在专门的/etc/shadow文件中，只有超级管理员的root权限可以查看，普通用户无权查看其内容。
- /etc/shadow文件格式:username:password:lastchg:min:max:warn:inactive:expire:flag

字段名	编号	说明
username	1	用户的登录名
password	2	加密的用户密码
lastchg	3	自1970.1.1起到上次修改口令所经过的天数
min	4	两次修改口令之间至少经过的天数
max	5	口令还会有效的最大天数
warn	6	口令失效前多少天内向用户发出警告
inactive	7	禁止登录前用户还有效的天数
expire	8	用户被禁止登录的时间，如果为空，账户永久可用。
flag	9	保留

## 解释下图中，第一个用户lex的shadow信息

```
lex:$6$g/jME.ry1QL.pyPh$3aLC8USBhT/5kEWtR72ZRdBo.FQbLFj54G45pPxt.jWYqIb6TJBZ0dM  
dfPFgYKW9bfMvOYrhZ80hcS36w6nnT/:18706:0:99999:7:::
```

字段名	编号	说明
username	1	lex
password	2	加密口令: \$6\$g.....
lastchg	3	自1970.1.1起到上次修改口令所经过的天数: 18706天
min	4	需几天可以修改口令: 0天 (随时)
max	5	口令还会有效的最大天数: 99999天, 即永不过期
warn	6	口令失效前7天内向用户发出警告
inactive	7	禁止登录前用户还有效的天数未定义
expire	8	用户被禁止登录的时间未定义, 永久可用
flag	9	保留未使用



## 6.2 Linux用户组

### 6.2.1 用户组

### 6.2.2 用户组相关文件

- 每个用户都属于一个用户组，用户组就是具有相同特征的用户集合体。一个用户组可以包含多个用户，拥有一个自己专属的用户组id，缩写为gid。gid是一个32位的整数，Linux系统内核用其来标识用户组。
- 同属于一个用户组内的用户具有相同的地位，并可以共享一定的资源。一个用户只能有一个gid（主要组、基本组），但是可以归属于其他的附加群组（次要组）。默认情况下，用户的组名（主要组、基本组）和用户名相同，

## 6.2.1 用户组

■ 由于每个文件必须有一个组所有者，因此有一个与每个用户相关的默认组。这个默认组成为新建文件的组所有者，**被称作用户的主要组，又称为基本组**。也就是说如果没有指定用户组，**创建用户的时候系统会默认同时创建一个和这个用户名同名的组，这个组就是基本组。**不可以把用户从基本组中删除。

■ 除了主要组以外，用户也可以根据需要再隶属于其他组，这些组被称作**次要组或附加组**。用户是可以从附加组中被删除的。用户不论属于基本组中还是附加组中，就会拥有该组的权限？（作业，验证）

■ 一个用户可以属于多个附加组。但是一个用户只能有一个基本组。

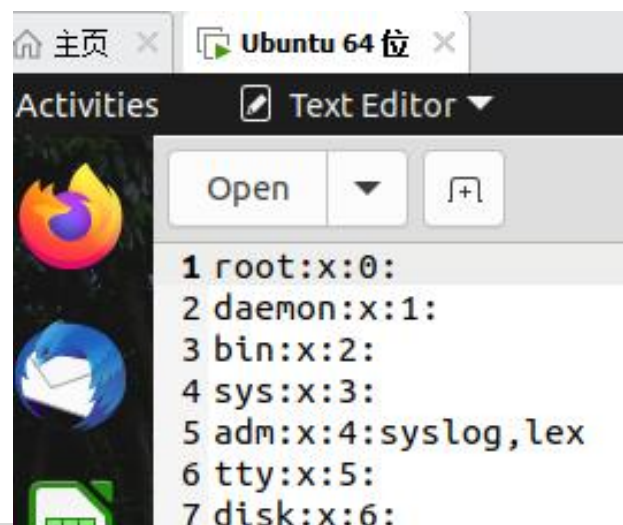


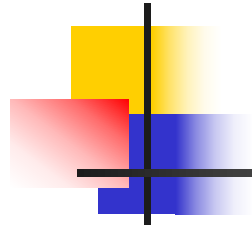
## 6.2.2 组相关文件

### 1、/etc/group

- ▶ Linux系统中，用户组的信息保存在配置文件/etc/group中，该文件是可读格式的文本，管理员可以利用文本编辑器来修改。而系统的大多数用户没有权限修改它，只能读取这个文件。
- /etc/group文件对组的作用相当于/etc/passwd文件对用户的作用，把组名与组ID联系在一起，并且定义了哪些用户属于哪些组。该文件是一个以行为单位的配置文件，每行字段用冒号隔开，格式如下：  
**group\_name:group\_password:group\_id:group\_members**
- 其中，每个字段的含义：

字段名	编号	说明
group_name	1	用户组名
group_password	2	加密后的用户组密码
group_id	3	用户组ID
group_members	4	逗号分隔开的组成员





## 2、/etc/gshadow

- 和用户账户文件/etc/passwd文件一样，为了保护用户组的加密密码，防止暴力破解，用户组文件也采用将组口令与组的其他信息分离的安全机制，即使使用/etc/gshadow文件存储各个用户组的加密密码。查看这个文件需要root权限。
- 例：在root下查看gshadow文件：

```
geoclue:!::  
pulse:!::  
pulse-access:!::  
gdm:!::  
lxd:!::lex  
lex:!::  
smbashare:!::lex  
systemd-coredump:!:::  
lex1:!::  
lex2:!::lex1
```

！表示组没有设置密码，可以用命令给组设置密码,格式为:sudo gpasswd lex1



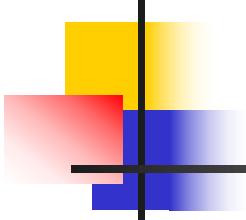


## 6.3 用户和用户组管理命令

---

### 6.3.1 用户管理命令

### 6.3.2 用户组管理命令



## 6.3.1 用户管理命令

### 1、useradd:创建一个新用户

选项	作用
-d	指定用户主目录。如果此目录不存在，则同时使用-m选项，可以创建主目录。
-g	指定gid
-u	指定uid
-G	指定用户所属的附加组。
-l	不要把用户添加到lastlog和faillog中，这个用户的登录记录不需要记载
-M	不要建立用户主目录
-m	自动创建用户主目录
-p	指定新用户的密码
-r	建立一个系统帐号
-s	指定shell

## 例1：创建新用户

# useradd lex3 创建新用户lex3

说明：

- 1) 只有超级用户root和具有超级用户权限的用户才能建立新用户。
- 2) useradd命令如果不加任何参数，建立的是“三无”用户：一无主目录，二无密码，三无系统Shell，虽然从上图能看到用户的目录是/home/lex3，但/home下显示没有这个目录。

```
lex3:x:1003:1003::/home/lex3:/bin/sh
```

```
lex@lex-virtual-machine:/home$ ls -l
```

```
total 12
```

```
drwxr-xr-x 19 lex lex 4096 4月 17 10:42 lex
drwxr-xr-x  2 lex1 lex1 4096 4月 10 15:36 lex1
drwxr-xr-x  2 lex2 lex2 4096 4月 13 13:49 lex2
```

```
lex@lex-virtual-machine:/home$ sudo useradd -m lex3
```

```
lex@lex-virtual-machine:/home$ ls
```

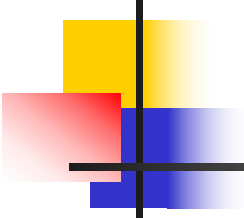
```
lex lex1 lex2 lex3
```

```
lex@lex-virtual-machine:/home$ ls -l
```

```
total 16
```

```
drwxr-xr-x 19 lex lex 4096 4月 17 10:42 lex
drwxr-xr-x  2 lex1 lex1 4096 4月 10 15:36 lex1
drwxr-xr-x  2 lex2 lex2 4096 4月 13 13:49 lex2
drwxr-xr-x  2 lex3 lex3 4096 4月 17 12:30 lex3
```

- 3) 加参数-m创建用户。



例：建立用户，指定组，并设置密码

**# useradd -m lex4 -g lex1 -p 123456**

新建用户lex4，创建主目录，指定组为lex1（基本组），设置密码为123456。

```
lex@lex-virtual-machine:/home$ sudo useradd -m lex4 -g lex1 -p 123456
lex@lex-virtual-machine:/home$ id lex4
uid=1004(lex4) gid=1001(lex1) groups=1001(lex1)
```

## 2、adduser

■ 功能描述：创建新用户。

使用adduser创建用户时显示了建立用户的详细进程，同时包含部分人机交互的对话过程，系统会提示用户输入各种信息，然后根据这种信息创建新用户。使用简单，不用加参数，建议初学者使用。

格式：**adduser 用户名**



### 3、passwd

- 功能描述：为用户设定口令，修改用户的口令，管理员还可以使用passwd命令锁定某个用户账户，该命令需要root权限。
- Ubuntu中登录用户时需要输入口令，也就是说只有指定了密码后才可以使用该用户，即使指定的是空口令也可以。

选项	作用
-l	管理员通过锁定口令来锁定已经命名的账户，即禁用该用户
-u	管理员解开账户锁定状态
-x	管理员设置最大密码使用时间（天）
-n	管理员设置最小密码使用时间（天）
-d	管理员删除用户的密码
-f (-e)	强迫用户下次登录时修改口令 (密码失效)

- **root用户具有超级权限，无需密码直接进入任何用户；**但普通用户之间的转换需要密码，如果没有口令，便无法切换到该用户

## 例6：锁定用户及解锁用户

#passwd -l lex5

锁定lex5账户

```
lex@lex-virtual-machine:/home$ passwd -l lex5
passwd: Permission denied.
lex@lex-virtual-machine:/home$ su
Password:
root@lex-virtual-machine:/home# passwd -l lex5
passwd: password expiry information changed.
root@lex-virtual-machine:/home# su lex5
lex5@lex-virtual-machine:/home$ su lex1
Password:
lex1@lex-virtual-machine:/home$ su lex5
Password:
su: Authentication failure
```

锁定用户后，密码前显示！  
如：锁定lex5用户。

```
lex5:!*$MuhVrJ62qnE05gpm$iiEiaZ
fGTxVknvZBNfVLK0u82/40Si6QBr1rC
"/etc/shadow":531:2141C
```

锁定账户后，这个账户不能使用了，  
解锁后可以使用。

```
root@lex-virtual-machine:/home# passwd -u lex5
passwd: password expiry information changed.
root@lex-virtual-machine:/home# exit
exit
lex1@lex-virtual-machine:/home$ su lex5
Password:
lex5@lex-virtual-machine:/home$
```

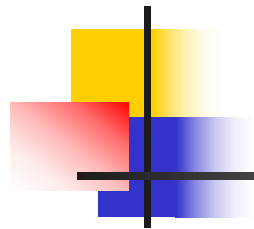


4、usermod:修改用户账户信息，如用户ID号、账号名称、主目录、用户组、登录shell等。

例 改变用户的组

选项	作用
-d	修改用户主目录
-e	修改帐号的有效期限
-f	修改在密码过期后多少天即关闭该帐号
-g	修改用户所属的组
-G	修改用户所属的附加组
-l	修改用户帐号名称
-L	锁定用户密码，使密码无效
-s	修改用户登入后所使用的shell
-u	修改用户ID
-U	解除密码锁定

```
root@lex-virtual-machine:/home# ls -l
total 24
drwxr-xr-x 19 lex  lex  4096  4月 17 10:42 lex
drwxr-xr-x  2 lex1 lex1  4096  4月 10 15:36 lex1
drwxr-xr-x  2 lex2 lex2  4096  4月 13 13:49 lex2
drwxr-xr-x  2 lex3 lex3  4096  4月 17 12:30 lex3
drwxr-xr-x  2 lex4 lex1  4096  4月 17 12:33 lex4
drwxr-xr-x  2 lex5 lex5  4096  4月 17 14:58 lex5
root@lex-virtual-machine:/home# usermod -g lex lex4
root@lex-virtual-machine:/home# ls -l
total 24
drwxr-xr-x 19 lex  lex  4096  4月 17 10:42 lex
drwxr-xr-x  2 lex1 lex1  4096  4月 10 15:36 lex1
drwxr-xr-x  2 lex2 lex2  4096  4月 13 13:49 lex2
drwxr-xr-x  2 lex3 lex3  4096  4月 17 12:30 lex3
drwxr-xr-x  2 lex4 lex  4096  4月 17 12:33 lex4
drwxr-xr-x  2 lex5 lex5  4096  4月 17 14:58 lex5
root@lex-virtual-machine:/home# usermod -g 1000 lex4
root@lex-virtual-machine:/home# ls -l
total 24
drwxr-xr-x 19 lex  lex  4096  4月 17 10:42 lex
drwxr-xr-x  2 lex1 lex1  4096  4月 10 15:36 lex1
drwxr-xr-x  2 lex2 lex2  4096  4月 13 13:49 lex2
drwxr-xr-x  2 lex3 lex3  4096  4月 17 12:30 lex3
drwxr-xr-x  2 lex4 lex  4096  4月 17 12:33 lex4
drwxr-xr-x  2 lex5 lex5  4096  4月 17 14:58 lex5
```



## 5、userdel

功能描述：删除用户。userdel命令可以删除已存在的用户账号，将/etc/passwd等文件系统中的该用户记录删除，必要时还删除用户的主目录。

选项：**-r**：将用户的主目录一起删除。

例11：建立test4和test5两个用户，然后删除。

**#userdel test4** 删除用户test4，保留主目录

**#userdel test5 -r** 删除用户test5及其主目录

```
root@lex-virtual-machine:/home# useradd -m test4
root@lex-virtual-machine:/home# useradd -m test5
root@lex-virtual-machine:/home# tail -3 /etc/passwd
lex5:x:1005:1004:,,,:/home/lex5:/bin/bash
test4:x:1006:1006::/home/test4:/bin/sh
test5:x:1007:1007::/home/test5:/bin/sh
root@lex-virtual-machine:/home# ls
lex lex1 lex2 lex3 lex4 lex5 test4 test5
```

```
root@lex-virtual-machine:/home# userdel test4
root@lex-virtual-machine:/home# userdel -r test5
userdel: test5 mail spool (/var/mail/test5) not found
root@lex-virtual-machine:/home# userdel test5
userdel: user 'test5' does not exist
root@lex-virtual-machine:/home# ls
lex lex1 lex2 lex3 lex4 lex5 test4
root@lex-virtual-machine:/home# userdel test4 -r 24
userdel: user 'test4' does not exist
```





## 6.3.2 用户组管理命令

1、groupadd:用指定的组名称来建立新的组账号

选项	作用
-g	指定组ID号，除非使用-o选项，否则该值必须唯一
-o	允许设置相同组id的群组，不必唯一
-r	建立系统组账号，即组ID低于999
-f	强制执行，创建相同id的组

2、groupmod:更改群组的信息

选项	作用
-g	指定组ID号
-o	与groupadd相同，重复使用群组识别码
-n	修改用户组名

3、groupdel: 从系统上删除组。

如果该组中仍包含某些用户，则必须先删除这些用户后，才能删除组。

4、gpasswd用来管理组。该命令可以把用户加入组（附加组），为组设定密码。



## 6.4 su和sudo

### 6.4.1 su命令:切换用户。

超级用户root向普通或虚拟用户切换不需要密码，而普通用户切换到其它任何用户都需要密码验证。

### 6.4.2 sudo命令：

■sudo命令为super user do的缩写，允许系统管理员分配给普通用户一些合理的权利，让他们执行一些只有超级用户或者其他特许用户才能完成的任务。

■sudo的流程为：当前用户切换到root（或其它指定切换到的用户），然后以root（或其它指定的切换到的用户）身份执行命令，执行完成后，直接退回到当前用户。而这些的前提是要通过sudo的配置文件/etc/sudoers来进行授权。

sudo工具由文件/etc/sudoers进行配置，该文件包含所有可以访问sudo工具的用户列表并定义了他们的特权。

1、把用户加到 /etc/sudoers

2、把用户加到 sudo 组

(1) `sudo gpasswd -a 用户 sudo`

(2) `sudo usermod -G sudo 用户`



## 第七章 硬盘和内存

---

7.1 硬盘

7.2 磁盘配额

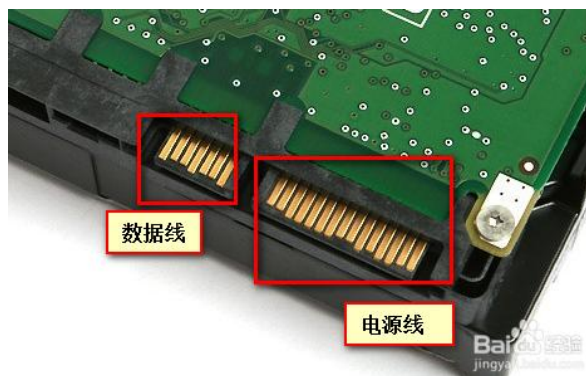
7.3 内存管理

7.4 进程管理

7.5 任务计划

## 7.1 硬盘

■ 硬盘接口分为IDE、SATA、SCSI和光纤通道四种，IDE接口硬盘多用于家用产品中，也部分应用于服务器，SCSI接口的硬盘则主要应用于服务器市场，而光纤通道只在高端服务器上，价格相对较贵，SATA是目前比较流行的的硬盘接口类型。





## 7.1.1 命名方式

---

### 1、硬盘的命名

Linux系统中，每一个设备都映射到一个系统文件，包括硬盘、光驱IDE，SCSI设备。在Linux下对IDE的设备是以hd命名的，一般主板上有两个IDE接口，一共可以安装四个IDE设备。主IDE上的主从两个设备分别为hda和hdb，第二个IDE口上的两个设备分别为hdc和hdd。SATA、SCSI接口设备是用sd命名的，第一个设备是sda，第二个是sdb。依此类推。

### 2、分区的命名

分区是用设备名称加数字命名的，如IDE接口的命名为hda1、hda2；SATA、SCSI接口的命名为sda1、sda2等。



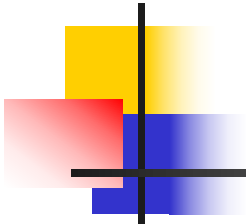
### 3、主分区、扩展分区、逻辑分区

■ 主分区：主分区是硬盘的启动分区，也称为引导分区，通常对应于计算机的第一个分区。主分区必须是可用的，也就是说它必须包含有效的文件系统，并且该文件系统必须是活动的。在Linux中，主分区可以有多种文件系统类型，如ext4、ntfs、vfat等。主分区通常用于存储操作系统、应用程序和数据，并且是计算机启动时所加载的第一个分区。

■ 逻辑分区：逻辑分区是在扩展分区内部创建的多个分区。扩展分区不能直接使用，必须划分成多个逻辑分区。逻辑分区可以有多种文件系统类型，并用于存储用户数据、应用程序或系统临时文件等。在Linux中，逻辑分区的创建和管理通常使用fdisk、parted等工具进行。

■ 扩展分区：扩展分区是一种特殊类型的分区，用于容纳逻辑分区。扩展分区本身不能直接使用，必须进一步划分成多个逻辑分区。扩展分区的容量等于其包含的所有逻辑分区的容量之和。在Linux中，扩展分区的创建和管理可以使用fdisk、parted等工具进行。

主分区是硬盘的启动分区，必须包含有效的文件系统，并且该文件系统必须是活动的；逻辑分区是在扩展分区内部创建的多个分区，用于存储用户数据、应用程序或系统临时文件等；而扩展分区是一种特殊类型的分区，用于容纳逻辑分区，其容量等于其包含的所有逻辑分区的容量之和。



### 3、主分区、扩展分区、逻辑分区

- **一个硬盘最多可以分4个主分区**；因此硬盘可以被分为1—3个主分区加一个扩展分区，或者仅有1—4个主分区。
- 对于扩展分区，可以继续对他进行划分，分成若干个逻辑分区，也就是说扩展分区只不过是逻辑分区的“容器”。主分区的名称分别是hda1、hda2、hda3和hda4，其中扩展分区也占用一个主分区的名称。**逻辑分区的名称一定是从hda5开始**，每增加一个分区，分区名称的数字就加1，如hda6代表第二个逻辑分区等等，SATA、SCSI接口的硬盘与此类似。



图7.1 硬盘的命名



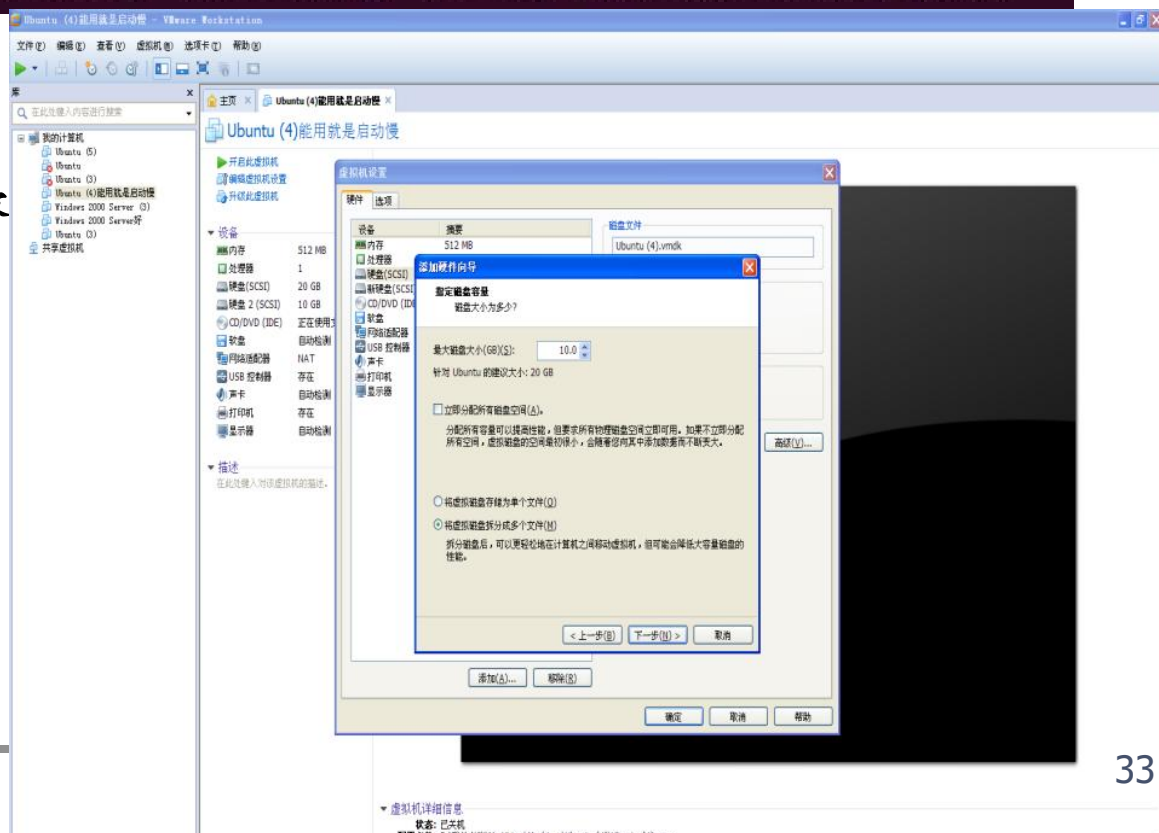
## 7.1.2 硬盘的分区

- 可以直接对硬盘分区，为了实验方便，我们在虚拟机下添加硬盘，并分区
- 先看下没添加硬盘之前的硬盘分区

Device	Start	End	Sectors	Size	Type
/dev/sda1	2048	4095	2048	1M	BIOS boot
/dev/sda2	4096	1054719	1050624	513M	EFI System
/dev/sda3	1054720	41940991	40886272	19.5G	Linux filesystem

### 1.添加硬盘

点击“**VM→Settings**”，选择“**add**”，添加一个硬盘，类型为**SCSI**，硬盘的容量为**10G**，添加完成后开启并进入系统。



2、查看硬盘信息 :添加10G硬盘后硬盘的信息。

```
Disk /dev/sdb: 10 GiB, 10737418240 bytes, 20971520 sectors
Disk model: VMware Virtual S
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk identifier: 0x4aeca60c
```

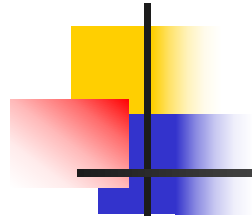
3、创建分区:

- (a)sudo fdisk /dev/sdb
- (b)输入m,显示帮助的命令

设备	启动	起点	终点	块数	Id	系统
----	----	----	----	----	----	----

```
命令(输入 m 获取帮助) : m
命令操作
a toggle a bootable flag
b edit bsd disklabel
c toggle the dos compatibility flag
d delete a partition
l list known partition types
m print this menu
n add a new partition
o create a new empty DOS partition table
p print the partition table
q quit without saving changes
s create a new empty Sun disklabel
t change a partition's system id
u change display/entry units
v verify the partition table
w write table to disk and exit
x extra functionality (experts only)

命令(输入 m 获取帮助) : n
```



(1)建立主分区：输入**n创建分区**，再输入**p创建主分区**，将sdb硬盘分成两个区：4G(主分区) 和6G（扩展分区---逻辑分区），注意直接输入大小时前面加上+，如+4G。

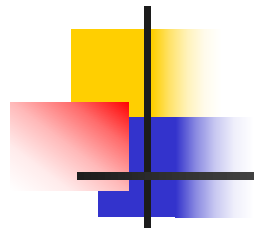
```
q quit without saving changes

Create a new label
g create a new empty GPT partition table
G create a new empty SGI (IRIX) partition table
o create a new empty DOS partition table
s create a new empty Sun partition table

Command (m for help): n
Partition type
  p primary (0 primary, 0 extended, 4 free)
  e extended (container for logical partitions)
Select (default p):
```

```
First sector (2048-20971519, default 2048):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-20971519,
default 20971519): +4G

Created a new partition 1 of type 'Linux' and of size 4 GiB.
```



(2)创建扩展分区：输入**n**（大写也可以）创建分区，再输入**e**创建扩展分区。

```
Command (m for help): n
Partition type
   p   primary (1 primary, 0 extended, 3 free)
   e   extended (container for logical partitions)
Select (default p): e
Partition number (2-4, default 2):
First sector (8390656-20971519, default 8390656):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (8390656-20971519, default 20971519):

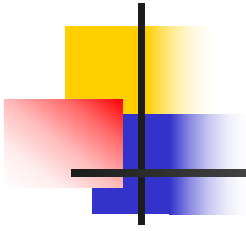
Created a new partition 2 of type 'Extended' and of size 6 GiB
.
```



(3)创建逻辑分区：再输入n创建分区，输入l创建逻辑分区,输入逻辑分区号5,指定逻辑分区大小，剩余的6G，采用的默认值。

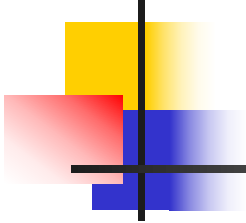
```
Command (m for help): n
All space for primary partitions is in use.
Adding logical partition 5
First sector (8392704-20971519, default 8392704):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (8392704-20971519, default 20971519):

Created a new partition 5 of type 'Linux' and of size 6 GiB.
```

- 
- P 显示分配完的分区，w存盘退出。

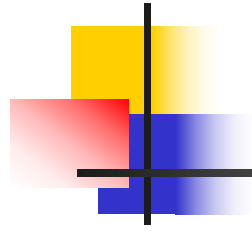
```
Command (m for help): p
Disk /dev/sdb: 10 GiB, 10737418240 bytes, 20971520 sectors
Disk model: VMware Virtual S
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x6c3a8f03
```

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sdb1		2048	8390655	8388608	4G	83	Linux
/dev/sdb2		8390656	20971519	12580864	6G	5	Extended
/dev/sdb5		8392704	20971519	12578816	6G	83	Linux



#### (4)格式化

- 分区完成后，需要对分区格式化、创建文件系统才能正常使用。格式化分区的主要命令是mkfs，格式为：
- mkfs -t [文件系统格式] 设备名
- 选项-t的参数用来指定文件系统格式，如ext3,ext4,nfs等
- 设备名称如/dev/sdb1， /dev/sdb2等



对/dev/sdb1进行格式化，如图所示。

```
lex@lex-virtual-machine:~/lesson$ sudo mkfs -t ext3 /dev/sdb1
[sudo] password for lex:
mke2fs 1.45.6 (20-Mar-2020)
Creating filesystem with 1048576 4k blocks and 262144 inodes
Filesystem UUID: 6174ad23-18f8-4891-a5f4-62c995bc2455
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: 0/done
```

接下来对/dev/sdb5进行格式化即可，**不能对扩展分区格式化。**





挂载分区：在使用分区前，需要挂载该分区，在挂载分区前，需要新建挂载点，在/mnt目录下新建/mnt/sdb1、 /mnt/sdb5 目录，作为分区的挂载点，**注意扩展分区也不能挂载，如图所示。**

```
lex@lex-virtual-machine:/$ sudo mount -t ext3 /dev/sdb1 /mnt/sdb1
lex@lex-virtual-machine:/$ sudo mount -t ext4 /dev/sdb5 /mnt/sdb5
lex@lex-virtual-machine:/$ sudo mount -t ext3 /dev/sdb2 /mnt/sdb2
mount: /mnt/sdb2: wrong fs type, bad option, bad superblock on /dev/sdb2, missing codepage or helper program, or other error.
lex@lex-virtual-machine:/$
```

## (5)卸载

- 卸载磁盘的命令为umount，格式为：
- umount 设备名或挂载点
- 可以直接卸载设备，#sudo umount /dev/sda1
- 也可以通过卸载挂载点卸载设备，#sudo umount /mnt/sda1



## 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：

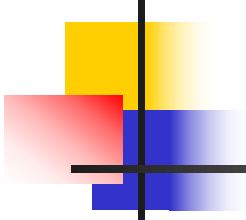
■ 建立目录，把要激活的分区挂载到此目录下。

```
# 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

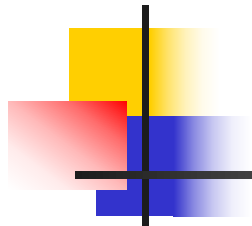
例：在磁盘分区/dev/sdb1启用磁盘配额，该分区为ext3文件系统（df-T查看），挂载到/myquota，使用如下命令修改/etc/fstab文件：

```
#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)

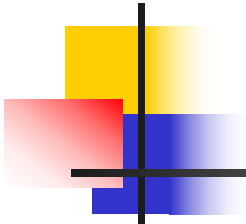
有时需要上面两条命令都执行。





## 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 -avu
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



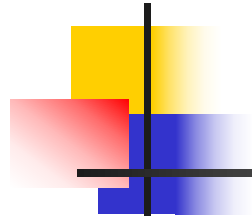
---

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

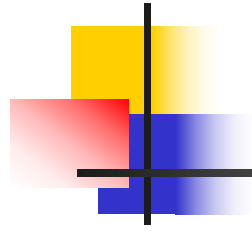
## 2、显示用户的配额

■ 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

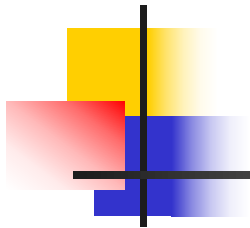
-axl 显示所有包含其他使用者的行程(包括优先级)

```
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。

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

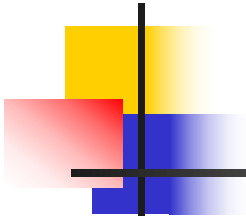
例如：终止 3710进程     **kill -s 9 3710**





## 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。

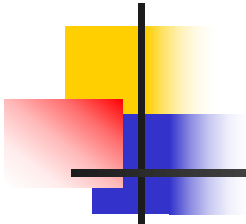


kill -l 显示信号的名称列表

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

安装at（默认是没安装）

sudo apt-get update

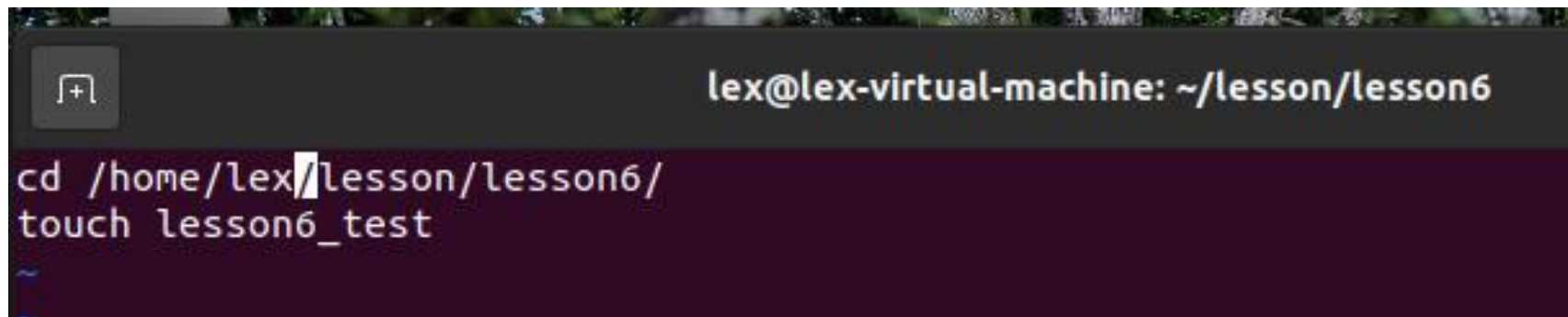
sudo apt-get install at

```
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]
```





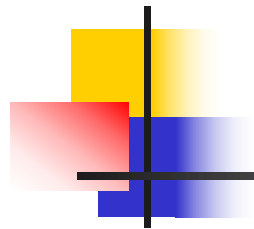
例：在11:18执行/home/lesson/lesson6/lt1



```
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$
```



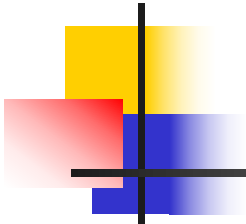


## (2)Batch命令：任意时间执行

- 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]

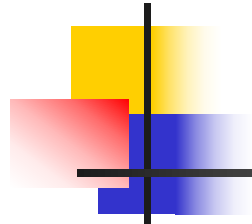


●格式：crontab [选项] [用户名]

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$
```