第5章 Linux 系统中的用户管理

前面内容对文件的操作都是有局限性的，不是本用户的文件，默认情况下不能被修改和执行；Shell 脚本程序虽然是用户自己书写的，但默认不能执行，必须增加执行权限后才可以运行该程序；有些命令也是只有 root 用户才可以使用。这些都是 Linux 系统的安全管理机制，它不仅管理文件的操作权限，也管理用户的操作权限。本章从用户、组和群的角度对用户进行全方位的管理。

5.1 安全机制

Linux 系统只允许授权的用户登录系统，登录进系统的用户也只能在自己的权限范围内访问文件和设备。每个系统都有一个 root 用户，Ubuntu 系统默认登录用户是安装系统时创建的普通用户，只有特殊情况下才可以使用 root 用户。

Linux 系统所采取的安全措施如下所述。

(1) 用户登录系统时必须提供用户名和密码。

(2) 以用户和群组来控制使用者访问文件和其他资源的权限。

(3) 每个文件都属于一个用户并与一个群组相关联。

(4) 每个进程都与一个用户和群组相关联。

5.2 用户管理

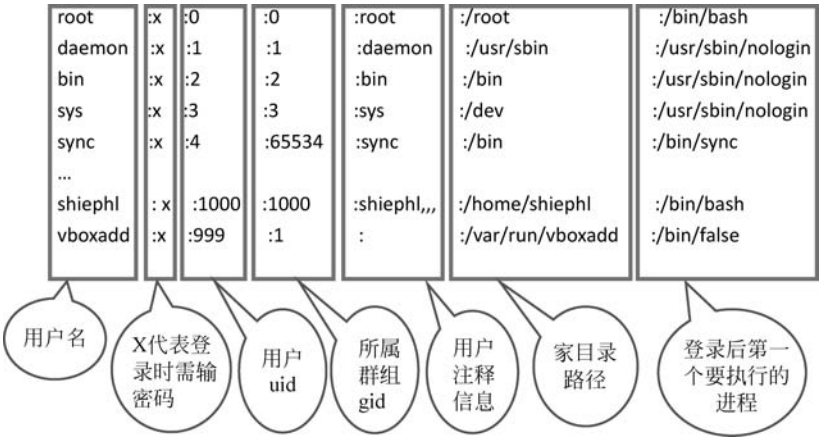
5.2.1 用户信息管理

在 Linux 系统中，用户具有如下特性。

每个用户都有一个唯一的用户标识符，用户名和 uid 都存放在/etc/passwd 文件中，其中还存放了每个用户的家目录以及该用户登录后第 1 个运行的程序。如果没有相应的权限，就不能读、写或执行其他用户的文件。

系统会将所有的用户信息自动记录在 passwd 文件中，这个文件保存在/etc/passwd 目录中，每个用户都占用一行记录，以冒号分隔成 7 个字段(列)，其中第 1 行记录是 root 用户的。

例 5.1 passwd 文件的信息解读。



5.2.2 root 用户管理

在每个 Linux 系统上都一定有一个 root 用户，root 用户也称为超级用户，有至高无上的权限，root 用户可以完全不受限制地访问任何用户的账户和所有的文件及目录。即使普通用户对自己的文件添加了访问权限也不能限制 root 用户对它的访问。

正是由于 root 用户的权限太过强大，如果使用不当或操作失误，就可能造成系统崩溃，因此一般系统都默认以普通用户身份登录，只在必须时才切换到 root 用户。这也是本书绝大多数例题都是以普通用户登录的原因，希望所有读者都养成良好的职业素养。另外，本书使用多个普通用户来运行程序，普通用户 liuhui 是英文版本系统下的用户，普通用户 shiephl 是在中文版本系统下的用户，两个 Ubuntu 版本系统下都各自有一个 root 用户，旨在让读者明白不同的用户管理理念、不同版本的操作技巧、不同系统下文件的共享使用。

但刚安装好的 Linux 系统没有设置 root 用户密码，Ubuntu 系统默认是没有激活 root 用户的，不能切换为 root 用户来执行命令，需要手工操作来激活 root 用户，在终端中输入 sudo passwd 或者 sudo passwd root 命令，系统会提示先输入当前用户的密码。当用户输入时显示器上没有任何提示，输入完毕按 Enter 键，下一条命令就是设置 root 用户的密码，输入时显示器上也没有任何提示，并且 root 用户权力强大，设置的密码可以非常简单，系统也会同意这个设置。

例 5.2 设置 root 用户的密码为 1234，激活 root 用户。

```
shiephl@ubuntuhl:~$ sudo passwd
[sudo] shiephl 的密码:
输入新的 UNIX 密码:
重新输入新的 UNIX 密码:
passwd: 已成功更新密码
shiephl@ubuntuhl:~$ su -
密码:
root@ubuntuhl:~#
```

只有激活了 root 用户，才可以正常使用 root 用户来执行一些命令。设置好 root 用户的密码后，root 用户就被激活了，用 su 命令切换到 root 用户后，可以增删用户、查看其他用户的文件、增删其他用户的文件，还可以修改其他用户的密码。

例 5.3 验证 root 用户的强大权限。

```
root@ubuntuhl:~# cp /home/shiephl/jerrymos1  /root/jryms
root@ubuntuhl:~# rm /home/shiephl/d1.txt
root@ubuntuhl:~# passwd shiephl
输入新的 UNIX 密码:
重新输入新的 UNIX 密码:
passwd: 已成功更新密码
root@ubuntuhl:~# _
```

使用终端命令切换到 root 用户后，在图形界面中第 1 次打开 root 目录，系统仍旧要进行认证，需要输入默认普通用户 shiephl 的登录密码，通过认证后才可以看到 root 目录下的内容。

5.2.3 增加和删除用户

在安装 Linux 操作系统时，实际上是以 root 用户的身份在操作，所以在安装过程中，输入的用户信息就是添加的第一个普通用户。安装完成后，第一次及以后的登录都是以第一个普通用户的身份信息在登录，如果需要增加或删除用户，需要切换到 root 用户。

1. 增加普通用户

useradd 或 adduser 命令用来建立用户账号和创建用户的起始目录，使用权限是 root 用户，其语法格式如下。

```
useradd [ - d home] [ - s Shell] [ - c comment] [ - m [ - k template]] [ - f inactive] [ - e expire ]
[ - p passwd] [ - r] name
```

主要参数说明如下。

(1) -c：加上备注文字，备注文字保存在 passwd 的备注栏中。

(2) -d：指定用户登录时的主目录，替换系统默认值/home/<用户名>。

(3) -D：变更预设值。

(4) -e：指定账号的失效日期，格式为 MM/DD/YY，如 06/30/12。默认值表示永久有效。

(5) -f：指定在密码过期后多少天即关闭该账号。如果为 0，账号立即被关闭；如果为 −1，则账号一直可用。默认值为 −1。

(6) -g：指定用户所属的群组，值可以使用组名，也可以是 GID。用户组必须是已经存在的，默认值为 100，即 users。

(7) -G：指定用户所属的附加群组。

(8) -m：自动建立用户的登录目录。

(9) -M：不要自动建立用户的登录目录。

(10) -n：取消建立以用户名为名的群组。

(11) -r：建立系统账号。

(12) -s：指定用户登入后所使用的 Shell。不同系统默认值不同，有的为/bin/sh，有的是/bin/bash。

(13) -u：指定用户 ID 号。该值在系统中必须是唯一的。0～499 默认是保留给系统用户账号使用的，所以该值必须大于 499。

注意

useradd 命令可用来创建用户账号，账号创建好之后，再用 passwd 命令设定账号的密码。使用 useradd 命令所创建的用户账号实际上保存在/etc/passwd 文件中。

例 5.4 增加新用户 Jenny，自动建立用户的登录目录，并建立系统账户，其他选项使用默认值。

```
root@ubuntuhl:~# useradd Jenny -m -r -u 888 -c"JennyHuang"
root@ubuntuhl:~# ls -l /home/*
/home/Jenny:
总用量 12
-rw-r--r-- 1 Jenny Jenny 8980 4月  16  2018 examples.desktop

/home/shiephl:
总用量 260
drwxr-xr-x 2 shiephl shiephl  4096 10月   4 09:55 backup
-rw-rw-r-- 1 shiephl shiephl 51200 10月   4 09:30 backup.tar
drwxr-xr-x 2 shiephl shiephl  4096 9月   19 10:38 deeplearning
```

用命令 useradd 增加用户后，查看/home 目录下有无 Jenny 用户的家目录，例 5.4 运行结果显示有 Jenny 用户和 shiephl 的家目录。

例 5.5 查看例 5.4 增加的新用户的信息。

```
root@ubuntuhl:~# cat /etc/passwd
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

gdm:x:121:125:Gnome Display Manager:/var/lib/gdm3:/bin/false
shiephl:x:1000:1000:shiephl,,,:/home/shiephl:/bin/bash
vboxadd:x:999:1::/var/run/vboxadd:/bin/false
Jenny:x:888:888:JennyHuang:/home/Jenny:/bin/sh
root@ubuntuhl:~#
```

这里只截取了重要的信息，最后一行是新增加的用户 Jenny 的信息。这里在增加用户时，用-u 设定用户 uid 为 888，没有设定用户登录后使用的 Shell，所以最后一项显示是/bin/sh，切换到该用户后，系统提示符变为只有一个 $ 的 sh 模式。笔者的两个 Ubuntu 系统的默认值不一样，以初始用户 liuhui 的英文版 Ubuntu 系统默认的新增用户的 Shell 是/bin/bash；而以初始用户 shiephl 的中文版 Ubuntu 系统默认的新增用户的 Shell 是/bin/sh。读者在使用时可以根据自己的系统进行选择，只要知道在/etc/passwd 文件中进行修改就可以了。

此时，新增的用户还没有自己的登录密码，第 1 次切换到新用户时也没有要求输入密码，是因为由 root 切换为任何一个用户都不需要输入密码，切换后系统提示符变为只有一个 $ 符，是因为在新增用户时没有设定 Shell，系统默认使用/bin/sh，可以让 root 用户修改这一项，使提示符变为任何需要的形式。由 root 用户切换为 Jenny 用户后，如果企图自己修改自己的密码，输入 passwd 命令，系统会提示操作错误，因为新增用户时并没有设置新用户的密码，所以提示输入当前 UNIX 密码时无法提供正确的密码，需要切换到 root 用户来修改用户 Jenny 的密码才可以成功。

例 5.6 新用户的第一次登录。

```
root@ubuntuhl:~# su Jenny

$ passwd
更改 Jenny 的密码。
（当前）UNIX 密码:
passwd: 认证令牌操作错误
passwd: 密码未更改

$ su -
密码:
root@ubuntuhl:~# passwd Jenny
输入新的 UNIX 密码:
重新输入新的 UNIX 密码:
passwd: 已成功更新密码
root@ubuntuhl:~#
```

用户密码被激活后，就可以正常使用该用户了。现在再次回到 Jenny 用户登录状态下来修改自己的密码就可以了。

例 5.7 修改密码。

```
$ passwd
更改 Jenny 的密码。
（当前）UNIX 密码:
输入新的 UNIX 密码:
重新输入新的 UNIX 密码:
必须选择更长的密码
输入新的 UNIX 密码:
重新输入新的 UNIX 密码:
passwd: 已成功更新密码
$ _
```

例 5.8 查看新用户的权限。

```
$ cd /home/Jenny
$ pwd
/home/Jenny
$ users
shiephl
$ whoami
Jenny
$ touch myfile1
$ mkdir mydir
$ ls -l
总用量 16
-rw-r--r-- 1 Jenny Jenny 8980 4月  16  2018 examples.desktop
drwxrwxr-x 2 Jenny Jenny 4096 10月 10 21:26 mydir
-rw-rw-r-- 1 Jenny Jenny    0 10月 10 21:08 myfile
-rw-rw-r-- 1 Jenny Jenny    0 10月 10 21:26 myfile1
$ ls -l /home/shiephl
总用量 260
drwxr-xr-x 2 shiephl shiephl  4096 10月  4 09:55 backup
-rw-rw-r-- 1 shiephl shiephl 51200 10月  4 09:30 backup.tar
```

例 5.8 运行结果表明，切换到新用户 Jenny 后，可以进行常规操作，如更换目录，在自己的家目录下新建文件和子目录，查看自己的目录下的内容，还可以查看同组用户 shiephl 的家目录。但是，用 users 命令查看登录用户时，仍然只显示 shiephl 用户，这个用户是在进入 Linux 系统时的登录用户；在终端中用命令 su 只是切换了用户，但是没有登录 Linux 系统，而 users、w 等命令的作用是查看登录 Linux 系统的用户，所以用 users、w 命令都不能显示 su 命令切换的用户。

2. 删除用户

命令 userdel 可以删除用户账号，和 deluser 命令功能一样，其语法格式如下。

```
userdel [ 选项]  用户账号
```

userdel 命令还可以删除与用户账号相关的文件，若不加参数，则仅删除用户账号，而不删除相关文件，使用权限是 root 用户。

常用选项说明如下。

(1) -r：删除用户登录目录以及目录中的所有文件。

(2) -f：强制删除用户(甚至当用户已经登录 Linux 系统时此选项仍旧可以生效)。

如果新用户创建后没有设置密码，也就是没有激活该用户，或者设置了新密码，但是没有切换到该用户，也即要删除的用户没有登录终端，那么用 userdel 命令可以直接将其删除。

例 5.9 删除没有登录终端的用户 Jenny。

```
root@ubuntuhl:~# useradd Jenny -m -r -u 888 -c "JennyHuang"
root@ubuntuhl:~# passwd Jenny
输入新的 UNIX 密码:
重新输入新的 UNIX 密码:
passwd: 已成功更新密码
root@ubuntuhl:~# useradd Jenny
useradd: 用户"Jenny"已存在
root@ubuntuhl:~# userdel Jenny
root@ubuntuhl:~# cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
gdm:x:121:125:Gnome Display Manager:/var/lib/gdm3:/bin/false
shiephl:x:1000:1000:shiephl,,,:/home/shiephl:/bin/bash
vboxadd:x:999:1::/var/run/vboxadd:/bin/false
root@ubuntuhl:~#
```

删除成功后系统没有任何提示，查看 passwd 文件，文件中没有看到 Jenny 用户的信息，说明用户 Jenny 被删除了。

但是，如果已经成功切换到新用户，也即新用户已经在终端运行了，那么会有对应的进程占用这个用户，执行 userdel 命令时将无法成功删除。

例 5.10 删除用户 Jenny。

```
root@ubuntuhl:~# userdel Jenny
userdel: user Jenny is currently used by process 2222
root@ubuntuhl:~#
root@ubuntuhl:~# w
 20:24:29 up  1:33,  1 user,  load average: 0.06, 0.05, 0.01
USER     TTY      来自             LOGIN@   IDLE   JCPU   PCPU WHAT
shiephl  :0       :0               18:51   ?xdm?  49.86s  0.02s /usr/lib/gdm3/g
root@ubuntuhl:~#
```

由例 5.10 运行结果可见，系统提示用户还处于登录状态，无法删除。此时需要先强制用户退出。

3. 强制已登录用户退出

要强制已登录用户退出，首先需要查看运行该用户进程的终端，可以用 w 命令或 ps 命令。

例 5.11 查看要删除的用户的进程和终端。

```
root@ubuntuhl:~# userdel Jenny
userdel: user Jenny is currently used by process 1839
root@ubuntuhl:~# kill 1839
root@ubuntuhl:~# w
 10:00:04 up 2 min,  1 user,  load average: 0.97, 0.56, 0.22
USER     TTY      来自             LOGIN@   IDLE   JCPU   PCPU WHAT
shiephl  :0       :0               09:58   ?xdm?  11.87s  0.03s /usr/lib/gdm3/g
root@ubuntuhl:~# ps
  PID TTY          TIME CMD
 1838 pts/0    00:00:00 su
 1864 pts/0    00:00:00 su
 1865 pts/0    00:00:00 bash
 1950 pts/0    00:00:00 ps
root@ubuntuhl:~# userdel Jenny
userdel: user Jenny is currently used by process 1851
root@ubuntuhl:~# kill -t pts/0
```

例 5.11 运行结果表明，只杀死进程号还是不行，因为用户会又被新的进程占用，所以需要把占用该用户的终端杀掉才可以。

例 5.12 关闭当前终端。

```
root@ubuntuhl:~# userdel Jenny
userdel: user Jenny is currently used by process 2296
root@ubuntuhl:~# w
 10:31:10 up 33 min,  1 user,  load average: 0.04, 0.03, 0.06
USER     TTY      来自             LOGIN@   IDLE   JCPU   PCPU WHAT
shiephl  :0       :0               09:58   ?xdm?  33.73s  0.03s /usr/lib/gdm3/g
root@ubuntuhl:~# ps
  PID TTY          TIME CMD
 2044 pts/1    00:00:00 su
 2057 pts/1    00:00:00 bash
 2295 pts/1    00:00:00 su
 2305 pts/1    00:00:00 su
 2306 pts/1    00:00:00 bash
 2321 pts/1    00:00:00 ps
root@ubuntuhl:~# kill 2296
root@ubuntuhl:~# userdel Jenny
userdel: user Jenny is currently used by process 2296
root@ubuntuhl:~# pkill -kill -t pts/1
```

命令 pkill -kill -t pts/1 是把占用用户 Jenny 的终端关闭，这个命令一执行，当前终端就关闭。然后重新打开终端，切换到 root 用户，再次执行删除用户的操作，便可以删除成功。

例 5.13 查看例 5.12 中删除用户的结果。

```
shiephl@ubuntuhl:~$ users
shiephl
shiephl@ubuntuhl:~$ userdel Jenny
userdel: Permission denied.
userdel: 无法锁定 /etc/passwd，请稍后再试。
shiephl@ubuntuhl:~$ su -
密码：
root@ubuntuhl:~# userdel Jenny
root@ubuntuhl:~# cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
gdm:x:121:125:Gnome Display Manager:/var/lib/gdm3:/bin/false
shiephl:x:1000:1000:shiephl,,,:/home/shiephl:/bin/bash
vboxadd:x:999:1::/var/run/vboxadd:/bin/false
root@ubuntuhl:~#
```

已经没有了Jenny用户的信息

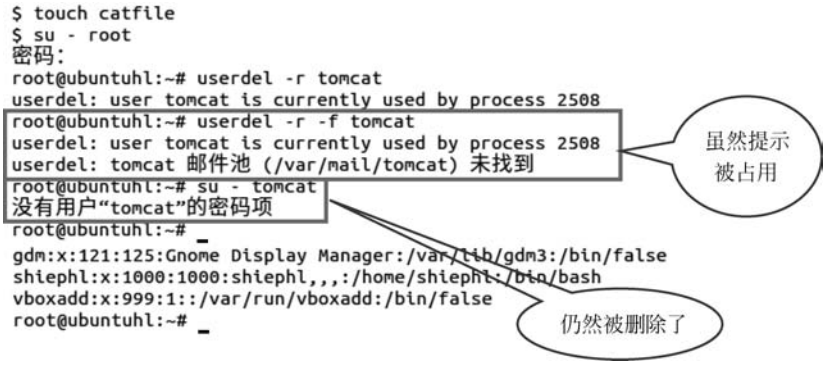
在使用 userdel 命令删除用户时，默认只删除 4 个配置文件(/etc/password、/etc/group、/etc/shadow、/etc/gshadow)中有关用户的信息，而不会删除用户家目录/home/用户名，可以使用 userdel -r 用户名来删除用户以及和用户相关的所有文件。如果用户已经被删除，只余下用户的家目录，可以用 rm -r 命令来删除目录。如果用户已经登录到终端，则无法删除该用户，使用 userdel -f 用户名命令才能强制删除。

例 5.14 新建一个用户 tomcat，切换登录后再用 userdel -r -f 命令强制删除已经登录到终端的用户 tomcat 和用户的相关文件。

```
root@ubuntuhl:~# ls -l /home
总用量 8
drwxr-xr-x  2     888     888 4096 10月 11 10:30 Jenny
drwxr-xr-x 22 shiephl shiephl 4096 10月 11 09:59 shiephl
root@ubuntuhl:~# useradd tomcat -m -r -u 889 -c "tomcatliu"
root@ubuntuhl:~# ls -l /home
总用量 12
drwxr-xr-x  2     888     888 4096 10月 11 10:30 Jenny
drwxr-xr-x 22 shiephl shiephl 4096 10月 11 09:59 shiephl
drwxr-xr-x  2 tomcat  tomcat  4096 10月 11 11:06 tomcat
root@ubuntuhl:~# passwd tomcat
输入新的 UNIX 密码：
重新输入新的 UNIX 密码：
passwd: 已成功更新密码
root@ubuntuhl:~# su - tomcat
```

被删除用户的家目录仍存在

新增用户的家目录



5.3 密码管理

5.3.1 用户密码管理

passwd 文件记录了用户的基本信息,包括用户是否设置密码的信息,但是用户的密码并不保存在该文件中,而是保存在/etc/shadow 文件中。shadow 文件实际上是存放用户密码的数据库,普通用户无权查看。

例 5.15 查看 shadow 文件的后 5 行。

```
shiephl@ubuntuhl:~$ tail -5 /etc/shadow
tail: 无法打开'/etc/shadow' 读取数据: 权限不够
shiephl@ubuntuhl:~$ su - root
密码:
root@ubuntuhl:~# tail -5 /etc/shadow
gdm:*:18113:0:99999:7:::
shiephl:$6$KcrI0P56$RP39dhpYOUCDwSAWTzJQRwsaPD9bkxb4I.sDFHhLgIRDDHn
vh/Mcz7ASmGbZfJBCM9ctCHZEaKoCza3lzINiU0:18179:0:99999:7:::
vboxadd:!:18178::::::
Jenny:!:18180:0:99999:7:::
jerrymos:$6$TaG3.EJU$c62xamgGecAAirv91gOjwx6J8utjBb5u2.87frXoffdLJT
EKQPkHXZLZ/i/bljBBxxcJchBCAMh2RLQm5WofD1:18180::::::
root@ubuntuhl:~#
```

例 5.15 运行结果分 8 列显示,第 1 列是用户名,第 2 列是用户的密码,当然是加密后的密文。可以看出,root 用户和安装系统时设置的第 1 个用户 shiephl 的密码密文都很长;而系统创建的其他系统用户密码列都用一个 * 表示;用 useradd 命令增加的用户 Jenny,由于没有激活,没有设置密码,所以第 2 列用 ! 列出,表示没有设置密码;用 useradd 命令增加的用户 jerrymos,设置了密码并登录过终端,所以密码列也很长。仔细观察密码列可以发现,密码都以 6 开头,表示密码使用 SHA512 加密算法。

5.3.2 密码修改

passwd 命令可以修改用户的密码,可以激活新添加的用户(root@ubuntuhl:~# passwd 用户名),可以激活 root 用户(shiephl@ubuntuhl:~$ sudo passwd 或者 sudo passwd root)。另外,对于有些用户,系统管理员不知道是否设置了密码,怎么查看?怎么操作?这些都可以用 passwd 命令来是实现。

例 5.16 增加新用户 tomcat，设定 Shell 为/bin/bash。

```
root@ubuntuhl:~# useradd tomcat -m -r -s "/bin/bash"
root@ubuntuhl:~# tail -5 /etc/passwd
shiephl:x:1000:1000:shiephl,,,:/home/shiephl:/bin/bash
vboxadd:x:999:1::/var/run/vboxadd:/bin/false
Jenny:x:1001:1001::/home/Jenny:/bin/sh
jerrymos:x:998:998:jerrymouseusers:/home/jerrymos:/bin/sh
tomcat:x:997:997::/home/tomcat:/bin/bash
root@ubuntuhl:~#
```

例 5.17 用 passwd -S 命令查看用户的密码状态。

```
root@ubuntuhl:~# passwd -S tomcat
tomcat L 10/11/2019 -1 -1 -1 -1
root@ubuntuhl:~# passwd -S jerrymos
jerrymos P 10/11/2019 -1 -1 -1 -1
root@ubuntuhl:~#
```

例 5.17 运行结果表明，已经激活正常使用的普通用户 jerrymos 的密码状态是 P，表明有密码状态；而新增用户 tomcat 的密码状态是 L，表明被锁定状态，因为是刚刚添加的用户，还没有激活该用户。

例 5.18 用 passwd -u 命令激活账户。

```
root@ubuntuhl:~# passwd -u tomcat
passwd: 解锁密码将产生一个没有密码的账户。
您应该使用 usermod -p 来为此账户设置密码并解锁。
root@ubuntuhl:~# passwd -uf tomcat
passwd: 不适用的选项 -- f
```

Linux 系统的某些版本可以强行设置某用户登录时不用输入密码，但在作者的两个 Ubuntu 系统上都无法实现。在此列出此方法，可以开阔读者的视野，给别有用心人的一点暗示。但是一般系统是不允许用户无密码登录的，所以需要激活用户并设置他的登录密码。

例 5.19 激活并设置新用户登录密码。

```
root@ubuntuhl:~# su - tomcat
tomcat@ubuntuhl:~$ passwd
更改 tomcat 的密码。
（当前）UNIX 密码:
passwd: 认证令牌操作错误
passwd: 密码未更改
tomcat@ubuntuhl:~$ su - root
密码:
root@ubuntuhl:~# passwd tomcat
输入新的 UNIX 密码:
重新输入新的 UNIX 密码:
passwd: 已成功更新密码
root@ubuntuhl:~# su - jerrymos
$ su - tomcat
密码:
tomcat@ubuntuhl:~$
```

例 5.19 运行结果表明，在没有激活新用户的情况下，可以从 root 用户直接切换到新用户 tomcat，但是新用户 tomcat 无法更改自己的密码，只有让 root 用户来激活新用户；通过 root 用户激活新用户后，借助另一个用户 jerrymos，从普通用户切换到普通用户，需要输入用户的密码，正确输入密码后就可进入该用户并进行相应的操作。

用户激活后再来查看他的密码状态，如例 5.20 所示。

例 5.20 查看用户自己和别人的密码状态。

```
tomcat@ubuntuhl:~$ passwd -S tomcat
tomcat P 10/12/2019 -1 -1 -1 -1
tomcat@ubuntuhl:~$ passwd -S jerrymos
passwd: 您不能查看或更改 jerrymos 的密码信息。
tomcat@ubuntuhl:~$
```

例 5.20 运行结果表明，普通用户可以查看自己的密码状态，不可以查看别人的密码状态，只有 root 用户才可以查看别人的信息。

5.4 群组管理

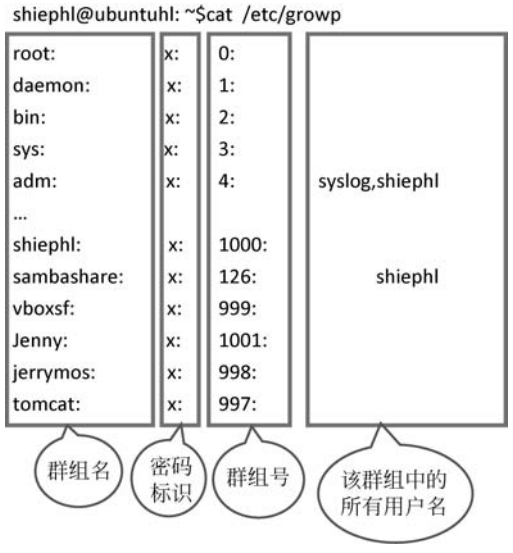
5.4.1 群组的文件

Linux 系统对用户的分组管理类似于学校把学生按照专业分班级管理一样，同一个班级成员有相同的专业课、相同的老师、相同的考试时间，学生可以使用的学校的公共资源也基本相同，并且同班同学之间的信息共享也方便。

Linux 系统中，每个用户一定隶属于至少一个群组，而每个群组都有一个 group 标识符，即 gid；群组和对应的 gid 都存放在/etc/group 文件中；系统创建用户时为每个用户创建一个同名的群组，并将该用户加入这个群组中，即每个用户至少会加入一个与它同名的群组中，也可以加入其他群组；如果有一个文件属于某个群组，那么该群组中的所有用户都可以访问这个文件。

/etc/group 文件中存放了所有群组的信息，它实际上是一个存放群组信息的数据库，每个群组占用一行记录，每个记录以冒号分隔成 4 个字段，每个字段中若有多个用户，每个用户名之间以逗号分隔。

例 5.21 查看群组信息文件 group。

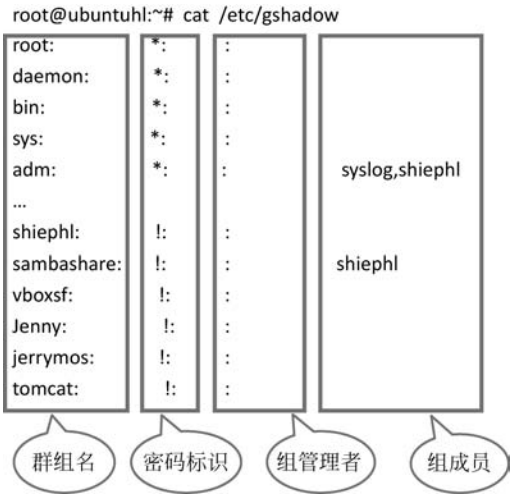


一个群组中可以有多个用户，有些系统下与群组同名的用户不显示。例如，笔者所用的两个系统的运行结果都没有把同名的用户名显示出来，有些系统会显示与群组同名的用户名。每次新增一个用户时都会自动创建一个同名的群组名，这个用户自然地包含在同名的群组中，如 root 组中一定有个 root 用户。一个用户可以属于多个群组，如安装系统时生成的那个特殊用户既属于 adm 组，也属于 sambashare 组，还属于其他一些组。如果某一个群组只有一个同名的用户，则用户信息中第 4 个字段会空缺。

Linux 系统对于群组的管理和对用户的管理一样，把群组基本信息放在 group 文件中，把密码放在另一个文件/etc/gshadow 中；使用权限也和对用户的管理一样，这个密码文件只有

root 用户才可以查看，其他用户无权查看。每个群组的密码信息占一行，每列用冒号分隔，共有 4 列，每个字段中若有多个用户，每个用户名之间以逗号分隔。

例 5.22 查看群组密码文件 gshadow。



具体说明如下。

第 1 字段：用户组。

第 2 字段：用户组密码。这个字段可以是空或!，如果是空或!，表示没有密码；是 * 号则表示群组设有密码。

第 3 字段：用户组管理者。这个字段也可为空，如果有多个用户组管理者，用逗号分隔。

第 4 字段：组成员。如果有多个成员，用逗号分隔。

5.4.2 改变所有者和群组

用户拥有的文件和所属的群组都可以更改，使用系统提供的命令 chown 和 chgrp 即可改变文件的所有者和群组。

在更改文件的所有者或所属群组时，可以使用用户名称和用户识别码设置。普通用户不能将自己的文件改变成其他所有者，其操作权限为 root 用户。

1. chown 命令

chown 命令的语法格式如下。

```
chown [ - R] [账号名称] [文件或目录]
chown [ - R] [账号名称]:[群组名称] [文件或目录]
```

chown 命令必要参数说明如下。

(1) -c：显示更改的部分的信息。

(2) -f：忽略错误信息。

(3) -h：修复符号链接。

(4) -R：处理指定目录及其子目录下的所有文件。

(5) -v：显示详细的处理信息。

(6) -deference：作用于符号链接的指向，而不是链接文件本身。

例 5.23 将文件 jerrymos1 的所有者由 shiephl 更改为 tomcat，不更改所在的群组，将文件 myfil1 的所有者和群组都更改为 root，将目录 files 及目录下的所有文件都变更为 root:root。

```
shiephl@ubuntuhl:~$ chown tomcat jerrymos1
chown: 正在更改'jerrymos1' 的所有者: 不允许的操作
shiephl@ubuntuhl:~$ su -
密码:
root@ubuntuhl:~# chown tomcat /home/shiephl/jerrymos1
root@ubuntuhl:~# chown root:root /home/shiephl/myfil1
root@ubuntuhl:~# chown root:root /home/shiephl/files
root@ubuntuhl:~# ls -l /home/shiephl
总用量 80
drwxr-xr-x 2 shiephl shiephl 4096 10月 13 13:03 backup
drwxr-xr-x 2 shiephl shiephl 4096 9月  19 10:38 deeplearning
-rw-r--r-- 1 shiephl shiephl 8980 9月  18 19:14 examples.desktop
drwx------ 2 root    root    4096 10月 13 13:22 files
drwxr-xr-x 3 shiephl shiephl 4096 9月  25 19:22 hlprivacy
-rw-r--r-- 1 tomcat  shiephl   82 10月 10 10:01 jerrymos1
drwxr-xr-x 2 shiephl shiephl 4096 9月  17 10:29 linuxdata
-rw-r--r-- 1 shiephl shiephl 1918 9月  19 19:34 linuxlearn
-rw-r--r-- 1 root    root       0 10月  3 15:29 myfil1
```

例 5.23 运行结果表明，更改所有者和群组的操作只有 root 用户有权限；使用 chowm tomcat /home/shiephl/jerrymos1 命令把本属于 shiephl 用户的文件的所有权给了 tomcat 用户，但文件还属于 shiephl 群组；使用 chown root:root /home/shiephl/myfil1 命令就把 myfil1 的所有者和群组都给了 root 用户和 root 群组。

2. chgrp 命令

chgrp 命令为变更群组的命令，用于变更文件的群组。变更文件的群组也可用 chown 命令，但建议用 chgrp 命令。chgrp 命令语法格式如下。

```
chgrp [选项]  用户组 文件
chgrp [选项]   --reference=参考文件 文件
```

chgrp 命令用来改变指定文件或目录所属的用户组，群组名可以是用户组的 ID，也可以是用户组的组名；文件名可以是由空格分开的要改变属组的文件列表，也可以是由通配符描述的文件集合。如果用户不是该文件的所有者或 root 用户，则不能改变该文件的群组。chgrp 命令的常用选项说明如下。

(1) -c:或--changes，效果类似于-v 参数，但仅显示更改的部分。

(2) -f:或-quiet 或--silent，不显示错误信息。

(3) -h:或-no-dereference，只对符号连接的文件做修改，而不更改其他任何相关文件。

(4) -R:或--recursive，递归处理，将指令目录下的所有文件及子目录一并处理。

(5) -v:或--verbose，显示指令执行过程。

(6) -reference=<参考文件或目录>：把指定文件或目录的所属群组全部设成和参考文件或目录的所属群组相同。

注意

> 命令中被更改的目标群组名必须是真实存在的。

新建一个用户时，如果没有特别说明，系统会自动创建一个同名的群组，那么在用户 shiephl 名下的所有文件都属于群组 shiephl。

例 5.24 把 shiephl 群组下的文件 shiephlf1 更改为 jerrymos 群组。

```
shiephl@ubuntuhl:~$ ls -l shiep*
-rw-r--r-- 1 shiephl shiephl 0 10月  4 20:19 shiephlf1
-rw-r--r-- 1 shiephl shiephl 0 10月  4 20:19 shiephlf1.lnx
-rw-r--r-- 1 shiephl shiephl 0 10月  3 15:29 shiephlf2
shiephl@ubuntuhl:~$ chgrp jerrymos shiephlf1
chgrp: 正在更改'shiephlf1' 的所属组: 不允许的操作
shiephl@ubuntuhl:~$ sudo  chgrp jerrymos shiephlf1
shiephl@ubuntuhl:~$ ls -l shiep*
-rw-r--r-- 1 shiephl jerrymos 0 10月  4 20:19 shiephlf1
-rw-r--r-- 1 shiephl shiephl  0 10月  4 20:19 shiephlf1.lnx
-rw-r--r-- 1 shiephl shiephl  0 10月  3 15:29 shiephlf2
shiephl@ubuntuhl:~$
```

例 5.24 运行结果表明，chgrp 命令的使用权限是 root 用户，shiephl 用户把自己的文件的群组更改时，系统会提示不允许，所以使用 sudo 命令暂时行使 root 用户的权限，修改后用 ls 命令查看一下文件所属的群组可知确实变为了 jerrymos。

如果想让两个文件有相同的群组，但不知道目标群组的名字，可以直接用 chgrp 命令从文件中检索组信息，这可以通过--reference 命令行选项来操作，只是会要求指定引用文件的名称。

例 5.25 把文件 shiephlf1.lnx 和 shiephlf2 所属的群组更改为和 shiephlf1 的群组一样。

```
shiephl@ubuntuhl:~$ sudo chgrp --reference=shiephlf1 shiephlf2 shiephlf1.lnx
[sudo] shiephl 的密码:
shiephl@ubuntuhl:~$ ls -l shiep*
-rw-r--r-- 1 shiephl jerrymos 0 10月  4 20:19 shiephlf1
-rw-r--r-- 1 shiephl jerrymos 0 10月  4 20:19 shiephlf1.lnx
-rw-r--r-- 1 shiephl jerrymos 0 10月  3 15:29 shiephlf2
shiephl@ubuntuhl:~$
```

如果想把整个目录的群组都更改一下，可以使用 -R 参数让命令递归执行，从而实现对目录中所有文件群组的更改。

例 5.26 把 files 目录下所有文件的群组更改为 jerrymos 用户。

```
shiephl@ubuntuhl:~$ sudo ls -l files
总用量 0
-rw-r--r-- 1 shiephl shiephl 0 10月  3 15:29 myfil1
-rw-r--r-- 1 shiephl shiephl 0 10月  3 15:29 myfil2
-rw-r--r-- 1 shiephl shiephl 0 10月  4 20:19 shiephlf1.lnx
shiephl@ubuntuhl:~$ sudo chgrp -R jerrymos files
shiephl@ubuntuhl:~$ sudo ls -l files
总用量 0
-rw-r--r-- 1 shiephl jerrymos 0 10月  3 15:29 myfil1
-rw-r--r-- 1 shiephl jerrymos 0 10月  3 15:29 myfil2
-rw-r--r-- 1 shiephl jerrymos 0 10月  4 20:19 shiephlf1.lnx
shiephl@ubuntuhl:~$
```

例 5.26 运行结果表明，原本属于 shiephl 用户的文件和目录所属群组通过 chgrp 命令更改为了其他用户，而原有的所有者没有变化。

5.5 文件权限管理

Linux 是一个支持多用户、多任务的系统，这也是它最优秀的特性，即可能同时有很多人都在系统上进行工作，所以千万不要强制关机。同时，为了保护每个人的隐私和工作环境，针对某一个文档（文件、目录），Linux 系统定义了 3 种身份，分别是所有者（owner）、群组（group）、其他人（others），每一种身份又对应 3 种权限，分别是可读（readable）、可写（writable）、可执行（excutable），这样的设计可以保证每个使用者所拥有数据的隐秘性，保证

数据的安全性，非法用户不可以随便更改别人的数据，同时又保证了数据的共享性，只有授权的用户才可以对别人的文件进行查看甚至修改。

5.5.1 文件权限的查询

每个文件(或目录)具有 3 种不同的使用者权限，即这个文件(或目录)的所有者的权限、与所有者用户在同一个群组的其他用户的权限、既不是所有者也不与所有者在同一个群组的其他用户的权限。

Linux 系统的文件操作权限包括如下几种。

(1) r：表示 read 权限，既可阅读文件，也可以使用 ls 命令列出目录内容的权限。

(2) w：表示 write 权限，即可编辑文件或在一个目录中创建和删除文件的权限。

(3) x：表示 execute 权限，即可运行程序或使用 cd 命令切换到该目录以及使用带有 -l 选项的 ls 命令列出该目录中详细内容的权限等。

(4) -：表示没有相应的权限，与所在位置的 r、w、或 x 相对应。

系统上的每个文件都一定属于一个用户(一般该用户就是文件的创建者)并与一个群组相关。通常一个用户可以操作属于自己的文件(或目录)，也可以访问其他同组用户共享的文件，但是一般是不能访问非同组的其他用户的文件。root 用户并不受这个限制，root 用户可以不受限制地访问 Linux 系统上的任何资源。

使用带有-l 选项的 ls 命令可以查看一个用户对某些文件的使用权限。

例 5.27 查看当前目录下文件和目录的操作权限。

```
shiephl@ubuntuhl:~$ ls -l
总用量 80
drwxr-xr-x 2 shiephl shiephl  4096 10月 13 13:03 backup
drwxr-xr-x 2 shiephl shiephl  4096 9月  19 10:38 deeplearning
-rw-r--r-- 1 shiephl shiephl  8980 9月  18 19:14 examples.desktop
drwx------ 2 root    jerrymos 4096 10月 13 13:22 files
drwxr-xr-x 3 shiephl shiephl  4096 9月  25 19:22 hlprivacy
-rw-r--r-- 1 tomcat  shiephl    82 10月 10 10:01 jerrymos1
drwxr-xr-x 2 shiephl shiephl  4096 9月  17 10:29 linuxdata
-rw-r--r-- 1 shiephl shiephl  1918 9月  19 19:34 linuxlearn
-rw-r--r-- 1 root    jerrymos    0 10月  3 15:29 myfil1
-rw-r--r-- 1 shiephl jerrymos    0 10月  4 20:19 shiephlf1
```

例 5.27 运行结果将每个文件的详细信息显示在一行上，每一行都有 7 方面的信息，以空格分隔，具体说明如下。

第 1 列：共 10 位，第 1 位表示文档类型，d 表示目录，-表示文件，l 表示链接文件，b 表示可随机存取的设备(如 U 盘等)，c 表示一次性读取设备(如鼠标、键盘等)。后 9 位依次对应 3 种身份所拥有的权限，身份顺序为 owner、group、others，权限顺序为 readable、writable、excutable。如-r-xr-x---的含义为当前文档是一个文件，所有者可读、可执行，同一个群组下的用户可读、可执行，其他人没有任何权限。

第 2 列：硬链接数 i-node，表示有多少个文件链接到 inode。硬连接数表示占用 i-node 数。i-node 是文件内容的真实表达，而 filename 是 inode 上层的表示方法。因此，每个文件名只能对应一个 i-node，一个 i-node 可以对应多个文件名。

第 3 列：表示所有者。

第 4 列：表示所属群组。

第 5 列：表示文档容量的大小，单位为字节。

第 6 列：表示文档的最后修改时间，不是文档的创建时间。

第 7 列：表示文档名称。以点(.)开头的是隐藏文档。

5.5.2 权限掩码的查看和更改

1. 权限掩码的查看

在 ls 命令显示的结果中，第 2～10 个字符定义了用户对文件的操作权限，每个字符的含义说明如下。

第 2～4 个字符为第 2 组，定义了文件所有者的权限，使用 u 代表所有者(owner)对文件的所有权限。

第 5～7 个字符为第 3 组，定义了文件所有者所在群组中其他成员所具有的权限，使用 g 代表这一组权限。

第 8～10 个字符为第 4 组，定义了其他用户对文件所具有的权限，使用 o 代表这一组(other)权限。

在第 2～4 组中，每组的第 1 个字符都是 r，表示具有读权限，若是 - ，表示没有读权限；在第 2～4 组中，每组的第 2 个字符都是 w，表示具有写权限，若是 - ，表示没有写权限；在第 2～4 组中，每组的第 3 个字符都是 x，表示具有执行权限，若是 - ，表示没有执行权限。

那么，怎么知道一个文件在新建以后的默认权限呢？除了使用 ls 命令可以查看外，还可以使用 umask 命令可以查看，其语法格式如下。

```
umask: umask [-p] [-S] [模式]
    显示或设定文件模式掩码。

    设定用户文件创建掩码为 MODE 模式。如果省略了 MODE，则
    打印当前掩码的值。

    如果 MODE 模式以数字开头，则被当作八进制数解析；否则是一个
    chmod(1) 可接收的符号模式串。

    选项:
      -p        如果省略 MODE 模式，以可重用为输入的格式输入
      -S        以符号形式输出，否则以八进制数格式输出

    退出状态:
    返回成功，除非使用了无效的 MODE 模式或者选项。
```

权限掩码 umask 就是要屏蔽掉权限值，也即在原有权限的基础上去除 umask 的值代表的权限。简单来说，新建一个目录时对目录的执行权限就是可以进入该目录，因此对目录的预权限为 rwxrwxrwx，亦即 777；新建一个文件时，预设没有可执行(x)权限，亦即只有 rw- rw- rw-，也就是最大为 666。默认情况下，umask 值是 022，可以用 umask 命令查看，此时建立的目录的默认权限是(rwx rwx rwx)-(--- -w -w-)＝rwx r-x r-x，用数字表示就是 755(777－022)；建立文件的默认权限是(rw- rw- rw-)-(--- -w -w-)＝rw- r-- r--，用数字表示就是 644(666－022)。

例 5.28 查看所有文件的默认权限掩码。

```
shiephl@ubuntuhl:~$ umask
0022
shiephl@ubuntuhl:~$ umask -S
u=rwx,g=rx,o=rx
shiephl@ubuntuhl:~$ umask -p
umask 0022
```

例 5.28 中，系统默认的权限掩码是 022，即去除的权限为--- -w- -w-；用-S 参数输出的权限掩码以字符的形式表示，比较直观；用-P 参数输出的结果可以重用其他命令的输入。

例 5.29 新建目录和文件的默认权限。

```
root@ubuntuhl:~# mkdir testdir
root@ubuntuhl:~# ls -l
总用量 12
-rw-r--r-- 1 root root   82 10月 10 19:22 jryms
-rw-r--r-- 1 root root  831 10月 12 20:52 t2.txt
drwxr-xr-x 2 root root 4096 10月 17 15:30 testdir
root@ubuntuhl:~# touch rufile1
root@ubuntuhl:~# ls -l
总用量 12
-rw-r--r-- 1 root root   82 10月 10 19:22 jryms
-rw-r--r-- 1 root root    0 10月 17 15:30 rufile1
-rw-r--r-- 1 root root  831 10月 12 20:52 t2.txt
drwxr-xr-x 2 root root 4096 10月 17 15:30 testdir
```

例 5.29 运行结果表明，新建目录和文件的默认权限分别为 755 和 644。

例 5.30 暂时更改系统权限掩码。

```
root@ubuntuhl:~# umask 077  #临时更改权限掩码
root@ubuntuhl:~# umask -S   #用字符显示当前的权限掩码
u=rwx,g=,o=
root@ubuntuhl:~# mkdir hldir
root@ubuntuhl:~# ls -l #查看新创建的目录的权限，777-077
总用量 20
drwx------ 2 root root 4096 10月 17 15:43 hldir
-rw-r--r-- 1 root root   82 10月 10 19:22 jryms
drwx------ 2 root root 4096 10月 17 15:38 mydir
-rw-r--r-- 1 root root    0 10月 17 15:30 rufile1
-rw-r--r-- 1 root root  831 10月 12 20:52 t2.txt
drwxr-xr-x 2 root root 4096 10月 17 15:31 testdir
root@ubuntuhl:~# touch hldata  #新建文件
root@ubuntuhl:~# ls -l    # 查看新建文件的权限 666-077
总用量 20
-rw------- 1 root root    0 10月 17 15:45 hldata
drwx------ 2 root root 4096 10月 17 15:43 hldir
-rw-r--r-- 1 root root   82 10月 10 19:22 jryms
drwx------ 2 root root 4096 10月 17 15:38 mydir
-rw-r--r-- 1 root root    0 10月 17 15:30 rufile1
-rw-r--r-- 1 root root  831 10月 12 20:52 t2.txt
drwxr-xr-x 2 root root 4096 10月 17 15:31 testdir
root@ubuntuhl:~#
```

例 5.30 运行结果表明，更改了权限掩码为 077，即对文件的所有者不限制默认的权限，对群组和其他用户所有的权限全部被剥夺，这样新建的文件的初始权限就是 rw- --- ---，新建目录的初始权限就是 rwx--- ---。用字符表示的权限来进行计算也是正确的，把字符表示的权限转为数字表示，就是文件的权限是 600，目录的权限是 700，但是，用数字的计算，对于目录，初始权限是 rwx，即拥有全部权限，所以在权限掩码 077 的剥夺下，第 1 组的权限没有变化，后两组的权限全部剥夺，所以初始权限减去掩码权限，就是 700(777－077)，是正确的。但是对于文件来说，初始权限只有读写 rw，在权限掩码的剥夺下，第 1 组用户所有者权限不变，仍为读写 rw，而群组用户和其他用户原本就只有读写 rw 权限，现在要剥夺读写执行 rwx，只能剥夺读写 rw 权限，所以不能简单地用 666－077 来计算权限掩码，而应该是 600(666－066)。如此，用数字表示和用字符就一样了。

2. 永久更改文件权限掩码

用 umask 命令只能临时更改系统默认保留权限，当关闭 Shell 再重新打开时，就会重置。若要永久更改系统默认的权限掩码，需要修改配置文件。

1）修改配置文件

如果想在当前系统中长时间使用自己设置的权限掩码，可以 root 用户的身份来修改系统配置和 Shell 配置文件。

第 1 步：修改系统配置，以 root 用户的身份打开系统配置文件/etc/profile，在文件的最后加入如下语句。

```
if [ $UID -gt 199 ] %% [ "`/usr/bin/id -gn`" = "`/usr/bin/id -un `" ]
   then
       umask 002   #普通用户，修改umask的值为002;
   else
       umask 077   #超级用户，修改umask的值为077;
fi
```

第 2 步：修改 Shell 配置，以超级 root 用户的身份打开 Shell 配置文件/etc/bash. bashrc，在文件的最后加入如下语句。

```
if [ $UID -gt 199 ] %% [ "`/usr/bin/id -gn`" = "`/usr/bin/id -un `" ]
   then
       umask 002   #普通用户，修改umask的值为002;
   else
       umask 077   #超级用户，修改umask的值为077;
fi
```

2）使配置生效

文件修改完成后保存退出，这两个文件是在系统启动时才能生效的，所以可以使用 source 命令来使配置立即生效。

source 命令语法格式如下。

```
source FileName
```

source 命令（从 C Shell 而来）是 bash Shell 的内置命令。该命令通常用命令“.”来替代，用于重新执行刚修改的初始化文档，如. bash_profile 和. profile 等。例如，在登录后对 profile 中的权限掩码做了修改，则能够用 source 命令重新执行 profile 中的命令使修改生效，而不用注销并重新登录。

例 5.31 永久修改系统权限掩码。

```
root@ubuntuhl:~# gedit /etc/profile
root@ubuntuhl:~# gedit /etc/bash.bashrc
root@ubuntuhl:~# umask
0022
root@ubuntuhl:~# source /etc/profile
root@ubuntuhl:~# source /etc/bash.bashrc
root@ubuntuhl:~# umask
0077
root@ubuntuhl:~#
root@ubuntuhl:~# umask -S
u=rwx,g=,o=
root@ubuntuhl:~#
```

3. 目录文件的权限更改

使用 chmod 命令可以更改文件的操作权限，其语法格式如下。

```
chmod [选项] mode 文件或目录名
```

1）常用选项

- -R：对目前目录下的所有档案与子目录进行相同的权限变更(即以递归的方式逐个变更)，不但要设置该目录权限，而且还要递归地设置该目录中所有文件和子目录的权限。
- -c：若该档案权限确实已经更改，才显示其更改动作。
- -f：若该档案权限无法被更改也不要显示错误信息。
- -v：显示权限变更的详细资料。
- --help：显示辅助说明。
- --version：显示版本。

2）mode 权限设定字符串

语法格式如下。

```
[ugoa…][[ +-= ][rwxX]…][,…]
```

共有 3 组数值，第 1 组表示设定谁的权限，具体如下。

- u：表示所有者(owner)的权限。
- g：表示群组(group)的权限。
- o：表示既不是所有者(owner)，也不和所有者在同一个群组(group)的其他用户的权限。
- a：表示所有用户(all)的权限。

第 2 组是运算符，具体含义如下。

- ＋：加入权限。
- －：删除权限。
- ＝：设定权限。

第 3 组表示具体的权限值，具体如下。

- r：read，读的权限。
- w：write，写的权限。
- x：execute，执行的权限。

3）以字符表示文件操作权限

例 5.32 用字符分别为每个用户更改文件的操作权限。

```
shiephl@ubuntuhl:~$ ls -l shi*
-rwxr-x-w- 1 shiephl jerrymos 0 10月  4 20:19 shiephlf1
-rw-r--r-- 1 shiephl jerrymos 0 10月  4 20:19 shiephlf1.lnx
-rw-r--r-- 1 shiephl jerrymos 0 10月  3 15:29 shiephlf2
shiephl@ubuntuhl:~$ chmod ug+x,o-r shiephlf2
shiephl@ubuntuhl:~$ chmod o+r shiephlf1
shiephl@ubuntuhl:~$ chmod a+x shiephlf1.lnx
shiephl@ubuntuhl:~$ ls -l shi*
-rwxr-xrw- 1 shiephl jerrymos 0 10月  4 20:19 shiephlf1
-rwxr-xr-x 1 shiephl jerrymos 0 10月  4 20:19 shiephlf1.lnx
-rwxr-x--- 1 shiephl jerrymos 0 10月  3 15:29 shiephlf2
shiephl@ubuntuhl:~$ _
```

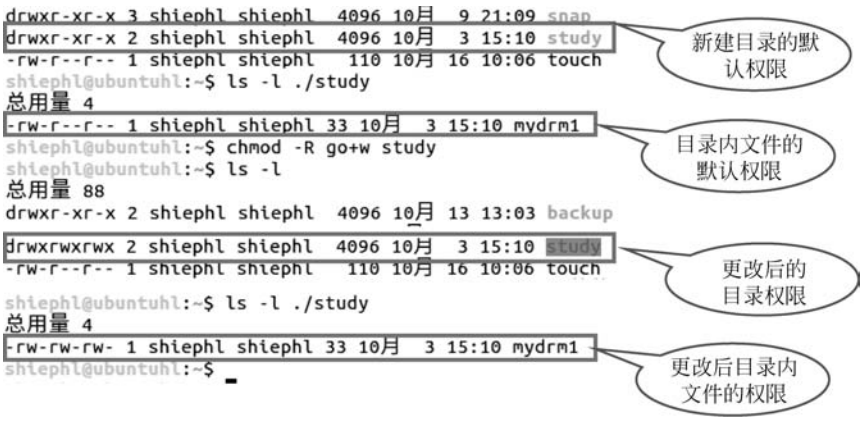
u g+x，o-r 后的权限

先查看当前用户的工作目录下以 shi 开头的文件的权限，然后对 u 和 g 增加执行 shiephlf2 文件的权限，对 o 删除 shiephlf2 文件的读取权限；对文件 shiephlf1，让 o 增加读取权限；对所有用户 a 增加对文件 shiephlf1.lnx 的执行权限。

注意

对多个用户增删不同的权限，用户名用逗号分隔；对多个用户增删相同的权限，用户名可以连写，后面紧跟增删的权限；命令中的 u(所有者)是指当前登录的用户。

例 5.33 更改目录的操作权限。



例 5.33 中，目录在创建时，系统默认为 3 组用户都分配了读取和执行权限，只有所有者有写的权限，目录的写权限实际是打开查看的权限；对于目录内的文件，默认所有者有读写权限，其他两组用户只有读取的权限。利用命令 chmod -R 递归更改后，目录和其下的文件的权限都发生了相同的变化。

4) 以数字表示文件的操作权限

每个用户都有 3 种权限，每个用户的权限又由 3 个数字表示，分别如下所述。

- 4：表示具有读(read)权限。
- 2：表示具有写(write)权限。
- 1：表示具有执行(execute)权限。
- 0：表示没有相应的权限。

把拥有的权限对应的数字相加，就得到介于 0 和 7 之间的一个数字，而这个数字就是用户对文件的操作权限；对于所有者、群组和其他用户这三组用户，可以分别用 3 个数字表示他们各自的权限。如 7(4＋2＋1)代表拥有读写执行权限，3(2＋1)代表拥有写和执行的权限，5(4＋1)代表拥有读和执行的权限。

注意

这里面的 421 可以看作是八进制的数字，读取权限对应于 4，写入权限对应于 2，执行权限对应于 1。也可以用二进制的思想来考虑，对于每种权限，如果拥有就用二进制 1 表示，没有这种权限就用二进制 0 表示，则 3 种权限都有 rwx 就是二进制的 111，也即八进制的 7。不同进制数字表示权限的具体形式及含义如表 5.1 所示。

表 5.1 数字表示用户权限

权　　限	八进制表示的数字	二进制的具体含义
rwx	7	111
rw-	6	110
r-x	5	101
r--	4	100
-wx	3	011
-w-	2	010
-x-	1	001
---	0	000

例 5.34 更改文件的操作权限。

```
shiephl@ubuntuhl:~$ ls -l shi*
-rwxr-xrw- 1 shiephl jerrymos 0 10月  4 20:19 shiephlf1
-rwxr-xr-x 1 shiephl jerrymos 0 10月  4 20:19 shiephlf1.lnx
-rwxr-x--- 1 shiephl jerrymos 0 10月  3 15:29 shiephlf2
shiephl@ubuntuhl:~$ chmod 400 shiephlf1
shiephl@ubuntuhl:~$ chmod 420 shiephlf1.lnx,shiephlf2
chmod: 无法访问'shiephlf1.lnx,shiephlf2': 没有那个文件或目录
shiephl@ubuntuhl:~$ chmod 420 shiephlf1.lnx  shiephlf2
shiephl@ubuntuhl:~$ ls -l shie*
-r-------- 1 shiephl jerrymos 0 10月  4 20:19 shiephlf1
-r---w---- 1 shiephl jerrymos 0 10月  4 20:19 shiephlf1.lnx
-r---w---- 1 shiephl jerrymos 0 10月  3 15:29 shiephlf2
shiephl@ubuntuhl:~$
```

先查看目前 3 组用户对 3 个文件的操作权限，然后使用数字分别对这 3 个文件更改权限，让所有者对文件 shiephlf1 只有读取权限，其他两组用户没有任何权限，就是 r-- --- ---，所以数字组合为 400；对于文件 shiephlf1.lnx 和 shiephlf1，让所有者只有读取权限，让群组只有写的权限，其他用户没有任何权限，就是 r-- -w- ---，所以使用数字 420。

注意

对多个文件同时更改，文件名之间用空格分隔。

例 5.34 运行结果表明，使用数字可以一次性设置 3 种用户的操作权限，比用字符更简捷。但数字含义不容易理解，使用字符更改权限，含义清晰简单，但操作复杂。两种方式各有所长。

例 5.35 更改目录的操作权限。

```
drwxr-xr-x 3 shiephl shiephl  4096 10月 16 12:04 linuxdata
shiephl@ubuntuhl:~$ chmod -R 721 linuxdata
shiephl@ubuntuhl:~$ ls -l
总用量 88
drwx-w---x 3 shiephl shiephl  4096 10月 16 12:04 linuxdata
shiephl@ubuntuhl:~$ ls -l ./linuxdata
总用量 12
drwx-w---x 2 shiephl shiephl 4096 9月  25 19:20 hldata
-rwx-w---x 1 shiephl shiephl    0 9月  17 10:40 jerrymos1
-rwx-w---x 1 shiephl shiephl  266 9月  22 13:23 play.txt
-rwx-w---x 1 shiephl shiephl  116 9月  22 13:40 tomcat
shiephl@ubuntuhl:~$
```

三组用户的权限变为721

先查看目录 Linuxdata 的权限信息，权限为 rwxr-xr-x，使用 chmod -R 命令递归地对目录及其下的文件更改权限，让所有者拥有 rwx 权限，群组拥有写(w)权限，其他用户拥有读取的权限，更改后再查看目录及内部文件的权限，权限掩码都更改成了 721。

注意

通过以上例子的运行可以看到，使用 chmod 命令可以针对某一个文件或目录来更改某一组用户的操作权限，一次只能针对特定的文件来更改权限。使用 umask 命令可以更改系统对后续新建的所有文件和目录的权限掩码，设置了 umask 的值以后再新建的文件和目录都被剥夺了 umask 定义的权限。umask 命令从全局上控制了所有用户对后续新建的所有文件和目录的操作权，在目前的 umask 权限掩码的限制下，新建的文件和目录的权限修改，只能用 root 用户或文件的所有者的身份使用 chmod 命令来修改。

5.6 用户的安全管理

Linux 系统是一个真正的多用户操作系统，那么多的用户在一个系统上工作，是否会出现越权的操作？会不会窃取或私自删除别人的文件？对于这些问题系统设计者早就考虑到了，并做了严格的控制，首先就是对用户进行严格的审查，并分别给不同的用户以不同的操作权限；其次就是用户身份的确认，坚决杜绝不合法用户的存在，对于可疑的用户要坚决删除，遵循“宁可错杀千人，不可放过一人”的原则；再次就是加强密码的管理，防止别有用心的人破解合法用户的密码。

常见的用户安全管理措施如下所述。

(1) 只有超级用户 root 才可以新增用户。

(2) 新增用户的权限很小，无法直接登录 Linux 系统，只能在授权用户登录后使用 su 命令切换到新增的用户。

(3) 每一个用户都属于某一个群组，群组的权限在创建群组时赋予。

(4) 用户信息和密码分别放置在不同的文件中，不同的用户有不同的权限。用户基本信息在/etc/passwd 文件中，一般用户可以查看，不可以修改；root 用户可以查看和修改/etc/passwd 文件。密码文件存放在/etc/shadow 中，一般用户无权查看，root 用户可以查看和修改。

(5) 普通用户只可以操作属于自己的文件，被其他用户授权的权限有严格限制。

(6) root 用户拥有最高的权限，使用时一定要小心，密码一定要保存好，千万不能泄露。

上机实验：Linux 中用户的安全管理

1. 实验目的

(1) 理解 Linux 系统中用户的安全管理策略。

(2) 理解所有者、群组、其他用户的权限设置。

(3) 掌握用户信息的存放文件和密码文件。

(4) 掌握群组的管理方法。

(5) 掌握文件权限的管理策略。

2. 实验任务

(1) 用户信息管理和用户的增删。
(2) 用户的密码管理文件和密码修改命令。
(3) 用户群组文件和所属群组的更改。
(4) 文件权限的管理。

3. 实验环境

装有 Windows 系统的计算机；虚拟机安装 VirtualBox+ Linux Ubuntu 操作系统。

4. 实验题目

任务 1：用户信息管理和用户的增删。

查看用户信息，并在信息文件中对用户信息进行修改；使用 root 用户身份增减用户并设置其相应的权限。

任务 2：用户的密码管理文件和密码修改命令。

查看密码管理文件，使用 passwd 命令修改密码。

任务 3：用户群组文件和所属群组的更改。

查看用户所属的群组，并使用 chgrp 命令更改群组；使用 chown 命令更改文件的所有者。

任务 4：文件权限的管理。

查看用户对文件的操作权限，并使用 umask 命令修改系统权限。

5. 实验心得

总结上机中遇到的问题及解决问题过程中的收获和心得体会等。