项目四　管理用户和用户组

4.1　项目引导

腾翼网络公司的管理员经过不断的学习与实践，已经能够胜任 Linux 系统常日的管理维护工作。但是 Linux 是一个多用户多任务操作系统，可以在系统上创建多个用户，并允许这些用户同时登录到系统去执行不同的任务，这将有可能影响到服务器是否可以正常运行。因此，用户和用户组的管理也是系统管理员所必须了解和掌握的重要工作之一。

4.2　项目分析

通过项目引导可以看出，用户和用户组的管理是 Linux 系统中至关重要的工作。作为系统管理员，应该熟悉保存用户与用户组信息的文件，掌握用户和用户组的创建与管理以及如何设置 ACL。在 Linux 系统中，可通过命令来创建和管理用户与用户组，也可在图形界面利用用户管理器来管理。

4.3　技术准备

4.3.1　Linux 用户与组

1.多用户多任务操作系统

Linux 是一个多用户多任务的操作系统，下面简单介绍一下单用户多任务和多用户多任务的概念。

(1)单用户多任务

单用户多任务是指系统在某个时间只允许一个用户登入，但是这个用户可以在同一时间执行多个不同的任务。比如我们以 liming 登录系统，进入系统后，打开 gedit 文本编辑器来写文档，但在写文档的过程中，感觉有点枯燥，所以又打开 xmms 播放音乐，当然为了随时与朋友保持联系，QQ 还得打开。如此一来，虽然只有一个用户登录系统，但却执行了多个任务。

(2)多用户多任务

Linux 是一个多用户多任务的操作系统。多用户是指多个用户可以在同一时间使用计算机系统；多任务是指 Linux 可以同时执行几个任务，它可以在还未执行完一个任务时又执

行另一项任务。比如在某台 Linux 服务器上，同时存在 FTP 用户、系统管理员、web 用户、普通用户等不同用户，在同一时刻，web 用户可能在管理网站；FTP 用户则可能在上传软件包到服务器；与此同时，可能还会有系统管理员在使用普通帐号或超级用户（root 帐号）维护系统。不同用户所具有的权限也不相同，要完成不同的任务得需要不同的用户，也可以说不同的用户，可能完成的工作也不一样。

值得注意的是：多用户多任务并不是大家同时挤到一台计算机的的键盘和显示器前来操作机器，多用户可能通过远程登录来进行，比如对服务器的远程控制，只要有用户权限任何人都是可以上去操作或访问。

（3）用户的角色区分

用户在系统中是分角色的，在 Linux 系统中，由于角色不同，权限和所完成的任务也不同，具体分为以下三种角色：

- 超级用户（root）：系统唯一，是真实的，可以登录系统，可以操作系统任何文件和命令，拥有最高权限。
- 系统用户：这类用户也被称为虚拟用户，与真实用户区分开来，这类用户不具有登录系统的能力，但却是系统运行不可缺少的用户，比如 bin、daemon、adm、ftp、mail、nobody 等，这类用户都是系统自身创建的，而非后来添加的。
- 普通用户：这类用户能登录系统，但只能操作自己家目录的内容，权限有限，这类用户都是系统管理员自行添加的。

（4）多用户操作系统的安全

从安全角度来说，多用户管理的系统更为安全，比如 liming 用户下的某个文件不想让其他用户看到，只需设置该文件的权限为只有 liming 用户可读可写就行了，这样一来就只有 liming 一个用户可以对其私有文件进行操作。Linux 系统在多用户下表现最佳，能很好的保护每个用户的安全，当然，还要要求管理员具有一定的安全意识，否则，这样的系统也不是安全的。

2.用户和用户组概念

（1）用户（user）的概念

通过前面对 Linux 多用户的理解，应该明白 Linux 是真正意义上的多用户操作系统，所以在 Linux 系统中可以创建若干个用户（user）。如果要使用系统资源，就必须向系统管理员申请一个账户，然后通过这个账户进入系统。这个账户和用户是一个概念，通过建立不同属性的用户，一方面，可以合理的利用和控制系统资源，另一方面也可以帮助用户组织文件，提供对用户文件的安全性保护。每个用户都有一个唯一的账号和口令，只有正确输入了账号密码，才能登录系统并进入自己的家目录。管理员账户（超级用户）对系统具有绝对的控制权，能够对系统进行一切操作。

（2）用户组（group）的概念

用户组（group）就是具有相同特征的用户（user）的集合体。有时我们要让多个用户具有相同的权限，比如查看、修改某一文件或执行某个命令，这时我们需要用户组，我们把用户都定义到同一用户组，我们通过修改文件或目录的权限，让用户组具有一定的操作权限，这

样用户组下的用户对该文件或目录都具有相同的权限，这是我们通过定义组和修改文件的权限来实现的。

在 Linux 系统中存在两种组：私有组和标准组。当创建用户的时候，没有为其制定属于哪个组，Linux 就会建立一个和用户同名的私有组，此私有组中只含有该用户；若使用标准组，在创建新用户时，为其制定属于哪个组。一个用户可以属于多个组，当一个用户属于多个组时，其登录后所属的组称为主组，其他组称为附加组或附属组。

(3)用户和用户组关系

用户和用户组的对应关系是：一对一、多对一、一对多或多对多，如图 4-1 所示。

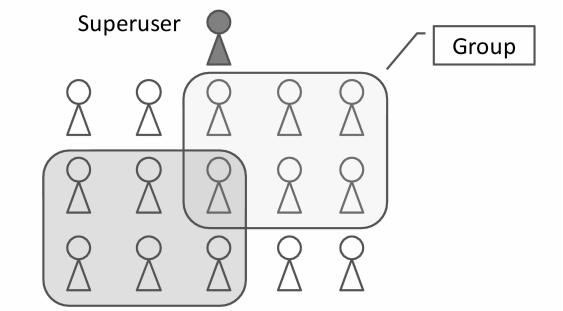

图 4-1 用户与组的关系

- 一对一：某个用户可以是某个组的唯一成员；
- 多对一：多个用户可以是某个唯一的组的成员，不归属其他用户组；
- 一对多：某个用户可以是多个用户组的成员；
- 多对多：多个用户对应多个用户组，并且几个用户可以是归属相同的组，其实多对多的关系是前面三条的扩展。

4.3.2 用户和用户组文件

在 Linux 中，用户账号、用户密码、用户组信息和用户组密码均是存放在不同的配置文件中，下面对这些配置文件作一简单介绍，以便对 Linux 系统有更深入的了解。

1.用户账号文件

在 Linux 系统中，所创建的用户账号及其相关信息(密码除外)均是存放在一个名叫 passwd 的配置文件中的，该文件位于/etc 目录。由于所有用户对 passwd 文件均有读取的权限，因此密码信息并未保存在该文件中，而是保存在了另外一个名叫 shadow 的配置文件中。

Linux 的配置文件均是文本文件，因此可使用文本文件内容查看命令来查看。在 passwd 文件中，一行定义一个用户账号，每行均由多个不同的字段构成，各字段值间用“:”分隔，每个字段均代表该账户某方面的信息。

在刚安装完成的 Linux 系统中，passwd 配置文件已有很多账户信息了，这些账户是由系统自动创建的，它们是 Linux 进程或部分服务程序程序正常工作所需要使用的账户，这些账户的最后一个字段的值一般为/sbin/nologin，表示该账户不能用来登录 Linux 系统。

由于 passwd 配置文件内容较多，下面使用 head 命令显示该文件前 10 行的内容，其显示结果如图 4-2 所示。

```
[root@localhost ~]# head /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
uucp:x:10:14:uucp:/var/spool/uucp:/sbin/nologin
```

图 4-2　passwd 用户文件部分内容

在 passwd 配置文件中，从左至右各字段的对应关系及其含义如表 4-1 所示。

表 4-1　passwd 配置文件各字段的对应关系

用户账号	用户密码	用户 ID	用户组 ID	用户名全称	用户主目录	用户所使用的 shell
root	x	0	0	root	/root	/bin/bash
bin	x	1	1	bin	/bin	/sbin/nologin
.	.	.	.	.	.	.

由于 passwd 不再保存密码信息，所以用 x 占位表示。用户 ID 是一个惟一代表该用户的数字，用户组 ID 代表该用户所属的私有组的组号，也是一个惟一代表该用户组的数字。

Shell 是用户登录后所使用的一个命令行界面。输入的命令由 Shell 进行解释，并发送给 Linux 内核，由内核进行具体操作，以实现命令所体现的功能。Linux 系统自带有多种 Shell，系统默认使用的是/bin/bash。若在配置文件中，该字段的值为空，则默认使用“/bin/sh”Shell。在/etc 目录下的 shells 文件中，列出了系统可以使用的 Shell 列表，可使用 more /etc/shells 或 chsh -l 命令来查看。

若要使某个用户账户不能登录 Linux，只需设置该用户所使用的 Shell 为/sbin/nologin 即可。比如，对于 FTP 账户，一般只允许登录和访问 FTP 服务器，不允许登录 Linux 操作系统。若要让某用户没有 telnet 权限，即不允许该用户利用 telnet 远程登录和访问 Linux 操作系统，则设置该用户所使用的 Shell 为/bin/true 即可。若要让用户没有 telnet 和 ftp 登录权限，则可设置该用户的 Shell 为/bin/false。

在/etc/shells 文件中，若没有/bin/false 或/bin/true，则可使用以下命令将其添加进去。

```
# echo “/bin/false” >> /etc/shells
# echo “/bin/true” >> /etc/shells
```

在安装 Linux 系统时，创建了一个普通用户 test，下面使用 tail 命令查看一下这个用户在 passwd 文件中的相关信息，如图 4-3 所示。新建账户的记录保存在 passwd 文件的末尾。

```
[root@localhost ~]# tail -1 /etc/passwd
test:x:500:500::/home/test:/bin/bash
```

图 4-3　passwd 文件中 test 用户信息

普通用户账户的用户 ID 是从 500 开始编号的，系统账号小于 500。

2.用户密码文件

为安全起见，用户真实的密码采用 MD5 加密算法加密后，保存在/etc/shadow 配置文件中，该文件只有 root 用户可以读取。

与 passwd 文件类似，shadow 文件也是每行定义和保存一个账户的相关信息，如图 4-4 所示。第 1 个字段为账户名，第 2 个字段为该账户的密码。

```
[root@localhost ~]# tail -1 /etc/shadow
test:$6$L41oenA9$Vtrp4Vnk8tggTxaLHF5rZxZKfAnRB1bahFJw8SGc2ihdJ9SCNoRCdxXzTHpfmLV
3jqambxi/pPpQfBV5KyJn//:16275:0:99999:7:::
```

图 4-4 shadow 文件中 test 用户信息

第 3 个字段的数字代表上次口令改变的时间距离 1970 年 1 月 1 日的天数；第 4 个字段的数字代表在这么长的天数内不能改变口令，0 代表可以随时修改；第 5 个字段代表在这么长的天数后必须改变口令；第 6 个字段代表口令到期之前的这么多天时会出现警告。

3.用户组账号文件

用户组账户信息保存在/etc/group 配置文件中，任何用户均可以读取。用户组的真实密码保存在/etc/gshadow 配置文件中。

在 group 文件中，第 1 个字段代表用户组的名称，第 2 列为 x，第 3 列为用户组 ID 号，第 4 列为该组中的用户成员列表，各用户名间用逗号分隔。

4.4 项目实施

4.4.1 任务 1：管理用户账号与密码

1.添加用户

(1)命令用法

在 Linux 中，创建或添加新用户使用 useradd 命令来实现，其命令用法为：

```
useradd [option] username
```

该命令的 option 可选项较多，常用的主要有：

[-c]：用于设置对该账户的注释说明文字。

[-d]：指定用来取代默认的/home/username 的主目录。

[-m]：若主目录不存在，则创建它。-r 与-m 相结合，可为系统账户创建主目录。

[-M]：不创建主目录。

[-e(expire_date)]：指定账户过期的日期。日期格式为 MM/DD/YY。

[-f(inactive_days)]：账号过期几日后永久停权。若指定为 0，则立即被停权；为-1，则关闭此功能。

[-g]：指定将该用户加入到哪一个用户组中。该用户组在指定时必须已存在。

[-G]：指定用户同时也是其中成员的其他用户组列表，各组间用逗号分隔。

[-n]：不为用户创建私有用户组。

[-s]：指定用户登录时所使用的 shell，默认为/bin/bash。

[-r]：创建一个用户 ID 小于 500 的系统账户，默认不创建对应的主目录。

[-u]：手工指定新用户的 id 值，该值必须惟一，且大于 499。

[-p]：为新建用户指定登录密码。此处的 password 是对登录密码经 MD5 加密后所得到的密码值，不是真实密码原文，因此在实际应用中，该参数选项使用较少，通常单独使用 passwd 命令来为用户设置登录密码。

(2)应用示例

例如，若要创建一个名为 liyan 的用户，并作为 student 用户组成员，则操作命令为：

```
# useradd -g student liyan
# tail -1 /etc/passwd                    # 显示 passwd 文件最后 1 行的内容
liyan:x:502:500::/home/liyan:/bin/bash
```

添加用户时，若未用-g 参数指定用户组，则系统默认会自动创建一个与用户账号同名的私有用户组。若不需要创建该私有用户组，则可选用-n 参数。

比如，添加一个名为 lijie 的账户，但不指定用户组，其操作结果为：

```
# useradd lijie
# tail -1 /etc/passwd
lijie:x:503:503::/home/lijie:/bin/bash
# tail -2 /etc/group                     # 显示 group 文件最后 2 行的内容 student:x:500:
lijie:x:503:              # 系统自动创建了名为 lijie 的用户组，ID 号为 503
```

创建用户账户时，系统会自动创建该账户对应的主目录，该目录默认放在/home 目录下，若要改变位置，可利用-d 参数指定；对于用户登录时所使用的 shell，默认为/bin/bash，若要更改，则使用-s 参数指定。

例如，若要创建一个名为 vod 的账户，主目录放在/var 目录下，并指定登录 shell 为/sbin/nologin，则操作命令为：

```
# useradd -d /var/vod -s /sbin/nologin vod
# tail -1 /etc/passwd
vod:x:504:504::/var/vod:/sbin/nologin
# tail -1 /etc/group
vod:x:504:
```

在 Linux 中，对于新创建的用户，在没有设置密码的情况下，账户密码是处于锁定状态的，此时用户账户将无法登录系统。在创建新用户时，对于没有指定的属性，其默认设置位于/etc/default/useradd 文件中。

2.设置账户属性

对于已经创建好的账户，可使用 usermod 命令来修改和设置账户的各项属性，包括登录

名、主目录、用户组、登录 shell 等，该命令的用法为：

```
usermod [option] username
```

命令参数选项 option 大部分与添加用户时所使用的参数相同，参数的功能也一样，下面按用途介绍该命令新增的几个参数。

(1)改变用户账户名

若要改变用户名，可使用-l 参数来实现，其命令用法为：

```
usermod -l 新用户名 原用户名
```

例如，若要将用户 lijie 更名为 lijunjie，则操作命令为：

```
# usermod -l lijunjie lijie
# tail -1 /etc/passwd
lijunjie:x:503:503::/home/lijie:/bin/bash
```

从输出结果可见，用户名已更改为了 lijunjie。主目录仍为原来的/home/lijie，若也要将其更改为/home/lijunjie，则可以通过执行以下命令来实现。

```
# usermod -d /home/lijunjie lijunjie                    #注意指定要修改属性的用户名
#tail -1 /etc/passwd
lijunjie:x:503:503::/home/lijunjie:/bin/bash           #主目录更改成功
# mv /home/lijie /home/lijunjie                        #注意将实际的目录也相应更改为 lijunjie
```

若要将 lijunjie 加入 student 用户组(用户组 ID 为 500)，则实现的命令为：

```
# usermod -g student lijunjie 或 usermod -g 500 lijunjie
#tail -1 /etc/passwd
lijunjie:x:503:500::/home/lijunjie:/bin/bash     #用户组已更改为了 500，操作成功
```

(2)锁定账户

若要临时禁止用户登录，可将该用户账户锁定。锁定账户可利用-L 参数来实现，其命令用法为：

```
usermod -L 要锁定的账户
```

例如，若要锁定 lijunjie 账户，则操作命令为 usermod -L lijunjie。

Linux 锁定账户，是通过在密码文件 shadow 的密码字段前加“!”来标识该用户被锁定。

(3)解锁账户

要解锁账户，可使用带-U 参数的 usermod 命令来实现，其用法为：

```
usermod -U 要解锁的账户
```

例如，若要解除对 lijunjie 账户的锁定，则操作命令为 usermod -U lijunjie。

3.删除账户

要删除账户，可使用 userdel 命令来实现，其用法为：

```
userdel [-r] 账户名
```

-r 参数为可选项，若带上该参数，则在删除该账户的同时，一并删除该账户对应的主目录。比如，若要删除 vod 账户，并同时删除其主目录，则操作命令为 userdel -r vod。

若要设置所有用户账户密码过期的时间，则可通过修改/etc/login.defs 配置文件中的配置项 PASS_MAX_DAYS 的值来实现，其默认值为 9999，代表用户账户密码永不过期。其中的 PASS_MIN_DAYS 配置项用于指定不能修改账户密码的天数，默认值为 0，代表用户账户的密码可以随时修改；PASS_MIN_LEN 配置项用于指定账户密码的最小长度，默认为 5 个字符；PASS_WARN_AGE 配置项用于指定账户密码到期前出现警告的天数，默认为密码到期前 7 天警告用户。

4.设置用户登录密码

Linux 的账户必须设置密码后，才能登录系统。设置账户登录密码，使用 passwd 命令，其用法为：

```
passwd ［账户名］
```

若指定了账户名称，则设置指定账户的登录密码，原密码自动被覆盖。只有 root 用户才有权限设置指定账户的密码，一般用户只能设置或修改自己账户的密码，使用不带账户名的 passwd 命令来实现设置当前用户的密码。

例如，若要设置 lijunjie 帐户的登录密码，则操作命令为：

```
# passwd lijunjie
Changing password for user lijunjiw.
New password:                                    #键入密码
Retype new password:                             #重复输入密码
passwd:all authentication tokens updated successfully.
```

账户登录密码设置后，该账户就可以登录系统了。此时，如果在图形登录界面下，可以选择注销用户，然后利用 lijunjie 账户重新登录系统；如果在字符界面下，按 Ctrl＋Alt＋F2 键，选择第 2 号虚拟控制台(tty2)，然后利用 lijunjie 账户登录系统，以检测能否登录系统。

5.锁定账户密码

在 Linux 中，除了用户账户可被锁定外，账户密码也可以被锁定，任何一方被锁定后，都将导致该账户无法登录系统。只有 root 用户才有权限执行该命令。锁定账户密码使用带-l 参数的 passwd 命令，其用法为：

```
passwd  -l  账户名
```

例如，若要锁定 lijunjie 账户的密码，则操作命令为 passwd -l lijunjie。

6.解锁账户密码

用户密码被锁定后，若要解锁，使用带-u 参数的 passwd 命令，该命令只有 root 用户才有权限执行，其用法为：

```
passwd  -u  要解锁的账户
```

7.查询密码状态

可以通过查询当前账户的密码状态来了解其是否被锁定，可使用带-S 参数的 passwd 命令来实现，其用法为：

```
passwd  -S  账户名
```

若账户密码被锁定，将显示输出“Password locked”，若未加密，则显示“Password set, MD5 crypt.”。

8.删除账户密码

若要删除账户的密码，使用带-d 参数的 passwd 命令来实现，该命令也只有 root 用户才有权限执行，其用法为：

```
passwd  -d  账户名
```

注意：账户密码被删除后，输入账户名后将会直接登录系统，系统不再要求输入密码。

4.4.2 任务 2：用户组管理

用户组是用户的集合，通常将用户进行分类归组，便于进行访问控制。正如前面提到的，用户与用户组属于多对多的关系，一个用户可以同时属于多个用户组，一个用户组也可以包含多个不同的用户。

1.创建用户组

使用 groupadd 命令来创建用户组，其命令用法为：

```
groupadd  [ -r ]  用户组名
```

若命令带有-r 参数，则创建的用户组为系统用户组，该类用户组的 GID 值小于 500；若没有-r 参数，则创建普通用户组，其 GID 值大于或等于 500。在前面创建的 student 用户组，由于是所创建的第 1 个普通用户组，故其 GID 值为 500。

如果要创建一个名为 sysgroup 的系统用户组，则操作命令为：

```
# groupadd -r sysgroup
# tail -1 /etc/group
sysgroup:x:101:                          # 该用户组的 GID 值为 101
```

2.修改用户组属性

用户组创建以后，根据需要可对用户组的相关属性进行修改。对用户组属性的修改，是修改用户组的名称和用户组的 GID 值。

(1)改变用户组名称

如果要对用户组重命名，可使用带-n 参数的 groupmod 命令来实现，其用法为：

```
groupmod  -n  新用户组名  原用户组名
```

对用户组更名，不会改变其 GID 的值。

比如，若要将 sysgroup 用户组更名为 teacher 用户组，则操作命令为：

```
# groupmod -n teacher sysgroup
# tail -1 /etc/group
teacher:x:101
```

(2)重设用户组的 GID

用户组的 GID 值可以重新进行设置修改，但不能与已有用户组的 GID 值重复。对 GID 进行修改，不会改变用户名的名称。

要修改用户组的 GID，使用带-g 参数的 groupmod 命令，其用法为：

```
groupmod  -g  new_GID  用户组名
```

比如，若要将 teacher 组的 GID 值更改为 501，则操作命令为：

```
# groupmod -g 501 teacher
# grep teacher /etc/group                    #在/etc/group 文件中查找并显示含有 teacher 的行
teacher:x:501
```

3.删除用户组

删除用户组使用 groupdel 命令来实现，其用法为：

```
groupdel  用户组名
```

比如，若要删除 teacher 用户组，则操作命令为 groupdel teacher。

在删除用户组时，被删除的用户组不能是某个账户的私有用户组，否则将无法删除，若要删除，则应先删除引用该私有用户组的账户，然后才能删除用户组。操作演示如下：

```
# groupadd teacher                          #新建 teacher 用户组
# useradd -g teacher cy                     #创建 cy 用户，并将 teacher 组作为其私有组
# groupdel teacher
groupdel:cannot remove user's primary group.  #提示不能删除用户的私有组
# userdel -r cy                             #删除 cy 用户
# groupdel teacher
# grep teacher /etc/group                   #没有输出，说明 teacher 组已经成功删除
```

4.添加用户到指定的组

可以将用户添加到指定的组，使其成为该组的成员。其实现命令为：

```
gpasswd  -a  用户账户  用户组名
```

比如，现创建一个名为 ftpusers 的用户组，然后将 liyang 用户添加到 ftpusers 用户组，其操作命令为：

```
# groupadd ftpusers                         #新建 ftpusers 用户组
# gpasswd -a liyan ftpusers                 #将 liyan 用户添加到 ftpusers 用户组
Adding user liyan to group ftpusers
# groups liyan                              #使用 groups 命令查看 liyan 用户所属的组
liyan:student ftpusers                      # liyan 同时属于 student 和 ftpusers 用户组
```

5.从指定的组中移除某用户

若要从用户组中移除某用户，其实现命令为：

```
gpasswd  -d  用户账户  用户组名
```

比如，若要从 ftpusers 用户组中，移除 liyan 用户，则操作命令为：

```
# gpasswd -d liyan ftpusers
Removing user liyan from group ftpusers
# groups liyan
liyan:student                                    # liyan 用户已不再属于 ftpusers 用户组
```

6.设置用户组管理员

添加用户到组和从组中移除某用户，除了 root 用户可以执行该操作外，用户组管理员也可以执行该操作。本项目中的命令，只要未特别申明，都只有 root 用户才有权限执行这些管理性操作。

若要将某用户指派为某个用户组的管理员，可使用以下命令来实现：

```
gpasswd  -A  用户账户  要管理的用户组名
```

命令功能：将指定的用户设置为指定用户组的用户管理员。用户管理员只能对授权的用户组进行用户管理（添加用户到组或从组中删除用户），无权对其他用户组进行管理。

比如，若要设置 liyan 为 ftpusers 用户组的用户管理员，则操作命令为：

```
# gpasswd -A liyan ftpusers
```

完成以上操作后，liyan 用户就可以对 ftpusers 用户组进行管理，但是无权对 student 用户组进行管理，操作演示如下（以 liyan 账户登录系统）：

```
$ gpasswd -a lijunjie ftpusers                  #将 lijunjie 添加到 ftpusers 用户组
Adding user lijunjie to group ftpusers          #操作成功，说明可以对 ftpusers 用户组进行管理
$ groups -d lijunjie student                    #试图将 lijunjie 用户从 student 用户组中移除
Permission denied                               #操作被拒绝，说明无权对其他用户组进行管理
```

另外，Linux 还提供了 id、whoami 和 groups 等命令，用来查看用户和组的状态。id 命令用于显示当前用户的 uid、gid 和所属的用户组的列表；whoami 用于查询当前用户的名称；“groups 用户账户”用于查看指定的用户所隶属的用户组。

4.4.5 任务 3：使用用户管理器管理用户和组

在 RHEL 6 的图形系统中，提供了用户管理器，利用该管理器，可以实现用户和用户组的创建与管理。其操作方式类似于 Windows 系统，下面做一简单介绍。

1.启动用户管理器

进入图形界面后，在菜单栏中依次单击“系统”→“管理”→“用户和组群”，即可启动用户管理器，如图 4-5 所示。

图 4-5　用户管理器

在用户管理器窗口显示有“用户”和“组群”两个选项卡，分别用于显示当前的用户账户和用户组。可单击选项卡的标题栏，进行选项卡的切换。

利用工具栏中的“添加用户”、“添加组群”、“属性”和“删除”按钮，可分别实现添加用户、添加用户组、修改用户账户属性、删除用户或用户组等功能。

2.添加用户

单击“添加用户”按钮，将显示图 4-6 所示的对话框，在对话框中分别输入要添加的用户的相关信息，然后单击“确定”按钮，即可实现用户的添加。

图 4-6　创建新用户

3.添加用户组

单击“添加组群”按钮，将显示图 4-7 所示的对话框，在输入框中输入用户组的名称，最后单击“确定”按钮即可。在此可以勾选“手动指定组群 ID”进行用户组 GID 值的设置。

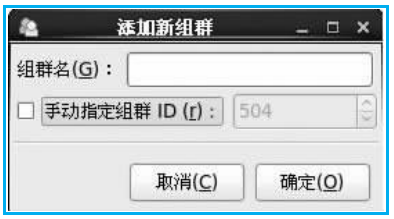

图 4-7　创建用户组

4.修改用户属性

在用户列表中，首先选中要修改属性的用户，然后单击“属性”按钮，此时将打开如图 4-8 所示的对话框。在该设置界面中，可对账户的各项属性进行设置和修改。

单击“账号信息”选项卡，切换到账户信息设置的对话框，在该对话框中可设置账户过期的日期以及账户是否加锁。

图 4-8　用户属性设置对话框

单击“密码信息”选项卡，将显示图 4-9 所示的对话框。

图 4-9　设置账户口令信息

在该对话框中，会显示用户最后一次更换密码的日期。勾选“启用密码过期”选项后，将可以设置与口令过期相关的信息，如：允许更换前的天数、需要更换的天数、更换前警告的天数、账号被取消激活前的天数。

单击“组群”选项卡，切换到如图 4-10 所示的对话框，在列表框中，可选择当前用户要加入的用户组。“主组群”组合框用于选择或输入当前用户的主要用户组（私有用户组）。

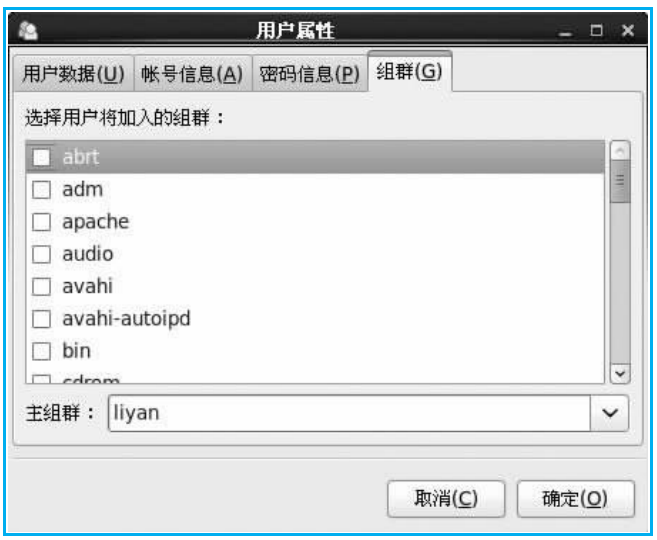

图 4-10　添加用户到用户组

5.删除用户或用户组

首先在用户列表框中选中要删除的用户，然后单击“删除”工具按钮，此时将弹出如图 4-11所示的对话框，询问是否要删除该用户的主目录，若单击“是”按钮，则同时删除当前用户的主目录。

图 4-11　删除用户

若要删除用户组，单击“组群”选项卡，切换到用户组列表，选中要删除的用户组，然后单击“删除”工具按钮即可删除。

4.5 技术拓展

4.5.1 任务 1:改变文件属主及属组

文件或目录的创建者,一般就是该文件或目录的所有者,称为文件拥有者或属主,对文件具有特别使用权。根据需要,可以更改文件或目录的所属关系,所有者或 root 用户可以将一个文件或目录的所有权转让给其他用户,使其他用户成为该文件或目录的属主。

在 Linux 中,使用 chown 命令可以改变文件或目录的所有者(属主)和所属的用户组(属组),利用参数-R,可以递归设置指定目录下的全部文件(包括子目录和子目录中的文件)的所属关系;chgrp 命令只能更改指定文件或目录所属的用户组。其命令用法为:

```
chown [ -R ] 新属主:新属组 要改变的文件或目录
或:chown [ -R ] 新属主.新属组 要改变的文件或目录
chgrp 新用户组 要改变所属用户组的目录或文件
```

为了便于测试该命令,下面创建两个普通用户账户和两个普通的用户组。

```
# groupadd pop
# groupadd pub
# useradd -g pop qc
# passwd qc
# useradd -g pop wy
# passwd wy
```

用户创建好后,系统就会自动在/home 目录下,为用户创建一个与账户同名的主目录。用户登录系统以后,其当前目录就是该主目录。

键入 logout 注销 root 用户(如果在图形界面下,可通过菜单栏中的选项注销 root),用 qc 用户重新登录,然后新建一个名为 myfile.txt 的文件。

```
$ touch myfile.txt
$ ll myfile.txt
-rw-r--r--  1  qc  pop  2 Oct  0  22:16 myfile.txt
```

从以上输出的信息可见,该文件的拥有者为 qc,所属的组为 pop。拥有者对该文件具有读和写的权限,其他用户仅有读的权限,由于文件不具有可执行性,因此不具有 x 属性。

若将该文件的拥有者修改为 wy,则操作命令为(使用 root 重新登录):

```
# chown wy /home/qc/myfile.txt
# ll /home/qc/myfile.txt
-rw-r--r--  1  wy  pop  2 Oct  0  22:19 myfile.txt
```

从以上输出的信息可见,尽管该文件仍位于 qc 用户的主目录中,但文件的所有者已变成为 wy 用户,此时的 wy 用户对该文件有读和写的权限,而原来的 qc 用户作为 pop 组的成

员，只有读的权限。

若要将 myfile.txt 文件所属的用户组修改为 pub 组，则实现命令为：

```
# chgrp pub /home/qc/myfile.txt
# ll /home/qc/myfile.txt
-rw-r--r--   1   wy   pub   2 Oct   0   22:22 myfile.txt
```

若要将 myfile.txt 文件的所有者和所属的用户组更改为 root 用户和 root 用户组，则实现命令为：

```
# chown root:root /home/qc/myfile.txt   或   chown root.root /home/qc/myfile.txt
# ll /home/qc/myfile.txt
-rw-r--r--   1   root   root   2 Oct   0   22:24 myfile.txt
```

chown 命令可同时更改所有者和所属的用户组，所有者和所属用户组之间可用冒号或小数点进行分隔表达。

4.5.2　任务 2：ACL 设置

1.ACL 简介

通过前面的内容，我们了解到传统的权限为三种身份（owner，group，others）搭配三种权限（r，w，x）以及三种特殊的权限（SUID，SGID，SBIT），随着应用的发展，这些权限组合已不能适应现在复杂的文件系统权限控制要求。

例如，目录 data 的权限为 drwxr-x--，所有者与所属组均为 root，在不改变所有者和所属组的前提下，要求用户 liulan 对该目录有完全访问权限（rwx），但又不能让其他用户拥有完全权限（rwx）。如果用传统的权限管理来设置，则这个要求不能实现。

为了解决这样的问题，Linux 开发出了一套新的文件系统权限管理方法，叫文件访问控制列表 ACL（Access Control Lists）。ACL 主要的目的是在提供传统的权限之外的局部权限设定。ACL 可以针对单个用户，单个文件或目录来进行 r，w，x 的权限设定，特别适用于需要特殊权限的使用情况。

要查系统是不是支持 ACL，可以通过下面的方法来查看。

```
# dumpe2fs /dev/sda1 |grep acl
dumpe2fs 1.41.12 (17-May-2010)
Default mount options: user_xattr acl
```

通过上面显示的内容可以看到，默认的挂载选项已经有 ACL 了，如果系统挂载的时候没有这个选项，可以通过重新挂载来解决。

```
# mount -o remount,acl /dev/sda1
```

2.ACL 的名词定义

ACL 是由一系列的 Access Entry 所组成的。每一条 Access Entry 定义了特定的类别

可以对文件拥有的操作权限。Access Entry 是由三个字段组成的：Entry tag type，qualifier (optional)，permission。其中最重要的字段是 Entry tag type，它有以下几个类型：

- ACL_USER_OBJ：相当于 Linux 里文件拥有者所具有的权限
- ACL_USER：定义了额外的用户可以对此文件拥有的权限
- ACL_GROUP_OBJ：相当于 Linux 里用户组所具有的权限
- ACL_GROUP：定义了额外的组可以对此文件拥有的权限
- ACL_MASK：定义了 ACL_USER，ACL_GROUP_OBJ 和 ACL_GROUP 的最大权限
- ACL_OTHER：相当于 Linux 里其他用户拥有的权限

下面使用 getfacl 命令来查看一个定义好的 ACL 文件(/home/test.txt)，通过该文件来理解各字段的含义与作用，操作命令如下：

```
[root@localhost ~]# getfacl /home/test.txt

# file: test.txt                    #定义了该文件的文件名
# owner: leonard                    #定义了该文件的拥有者
# group: admin                      #定义了该文件所属的用户组
user::rw-                           #定义了 ACL_USER_OBJ，说明文件拥有者拥有读和写的权限
user:john:rw-                       #定义了 ACL_USER，使用户 john 拥有对该文件的读写权限
group::rw-                          #定义了 ACL_GROUP_OBJ，说明文件所属的组拥有读写权限
group:dev:r—                        #定义了 ACL_GROUP，使得 dev 组拥有了对文件的读权限
mask::rw-                           #定义了 ACL_MASK 的权限为读写
other::r--                          #定义了 ACL_OTHER 的权限为读
```

以上显示的内容中，前面三个以 # 开头的行分别定义了文件名，拥有者和所属组，这些信息的作用并不大，可以用参数--omit-header 来省略掉，比如执行命令 getfacl --omit-header /home/test.txt，将不会再显示以 # 开头的这三行信息。

从以上内容就可以看出 ACL 提供了可以定义特定用户和用户组的功能。那么接下来就来介绍一下如何设置文件或目录的 ACL。

3.对文件或目录进行 ACL 设置

setfacl 用于设置文件或目录的 ACL 内容，其用法为：

```
# setfacl [参数选项] [acl 参数] 文件或目录名称
```

其中参数选项有：

[-m]：设置后续的 acl 参数

[-x]：删除后续的 acl 参数

[-b]：删除所有的 acl 设定参数

[-d]：设置预设 acl 参数(只对目录有效，在该目录新建的文件也会使用此 ACL 默认值)acl 参数格式为：

u：用户名：权限

g：用户组名：权限

m::权限

o::权限

(1)针对用户的 ACL 设置

首先将 root 家目录下的 install.log 文件复制到根目录下,然后以长格式显示该文件的信息,操作命令为:

```
[root@localhost /]# ls -l install.log
-rw-r--r--. 1 root root 41465 10 月 12 2014 install.log
```

使用 getfacl 命令查看 install.log 文件的原始权限,则操作命令为:

```
[root@localhost ~]# getfacl install.log
# file: install.log
# owner: root
# group: root
user::rw-
group::r--
other::r--
```

通过 ACL 使 liyan 用户对文件 install.log 拥有 rwx 权限,并使用 getfacl 命令查看设置了 ACL 的 install.log 文件,则操作命令为:

```
[root@localhost ~]# setfacl -m u:liyan:rwx /install.log
[root@localhost ~]# cd /
[root@localhost /]# getfacl install.log
# file: install.log
# owner: root
# group: root
user::rw-
user:liyan:rwx
group::r--
mask::rwx
other::r--
```

通过以上的显示内容,可以看出 liyan 用户已经拥有了 install.log 文件的 rwx 权限。再次使用 ls -l 命令查看 install.log 文件信息,操作命令如下:

```
[root@localhost /]# ls -l install.log
-rw-rwx r--+ 1 root root 41465 10 月 12 2014 install.log
```

通过 ls -l 查看的文件权限后面多了一个“+”号,这就表示了文件存在 ACL 权限。

(2)针对用户组的 ACL 设置

通过 ACL 使 test 用户组对文件 install.log 拥有 rw 权限,并使用 getfacl 命令查看设置了 ACL 的 install.log 文件,则操作命令为:

```
[root@localhost /]# setfacl -m g:test:rw install.log
[root@localhost /]# getfacl install.log
# file: install.log
# owner: root
# group: root
user::rw-
user:liyan:rwx
group::r--
group:test:rw-
mask::rwx
other::r--
```

通过以上的显示内容,可以看出 test 用户组已经拥有了 install.log 文件的 rwx 权限。

(3)删除 ACL 设置

如果要还原 install.log 文件的原始权限,需要删除用户 liyan 以及用户组 test 对 install.log 文件的相关权限,可以使用-x 参数来实现,则操作命令为:

```
[root@localhost /]# setfacl -x u:liyan install.log
[root@localhost /]# setfacl -x g:test install.log
[root@localhost /]# getfacl install.log
# file: install.log
# owner: root
# group: root
user::rw-
group::r--
mask::rw-
other::r--
```

这时候发现还有个 mask 的权限没有去掉,执行以下命令:

```
[root@localhost /]# setfacl -x m:: install.log
[root@localhost /]# getfacl install.log
# file: install.log
# owner: root
# group: root
user::rw-
group::r--
other::r--
```

经过了上面的操作才算把 install.log 文件的权限还原了,实在有点不方便,而且在使用参数-x 的时候,不能单独删除某个权限,否则会出现错误提示。比如命令 setfacl -x u:yufei:rwx install.log 在执行后,会提示无效的参数。

更为直接和便捷的方法是使用-b 参数删除所有的 ACL 权限,若要还原 install.log 文件的原始权限,则操作命令为:

```
[root@localhost /]# setfacl -b install.log
```

-b 参数，一次性把所有的 ACL 权限全部清空，还原成文件的原来权限。

(4)ACL 的 mask 设置与有效权限(effective)

在 Linux 权限里面，对于 rw-rw-r--来说，当中的那个 rw-是指用户组所拥有的权限。但是在 ACL 里面，这种情况只是在 ACL_MASK 不存在的情况下成立。如果文件有 ACL_MASK 值，那么当中那个 rw-代表的就是 mask 值而不再是用户组权限了。

下面这个例子将会详细解释 ACL 的 mask 设置。

在 tom 用户的家目录下有一个 test.sh 脚本文件(可以执行)，以常格式显示该文件，则操作命令为：

```
[tom@localhost ~]$ ls -l test.sh
-rwx r--r--. 1 tom admin 0 10 月 11 21:39 test.sh
```

通过以上显示内容，可以看到 test.sh 文件只有文件的拥有者 tom 拥有可读可写可执行的权限，而其所属的组 admin 只有可读的权限。现在想让用户 john 也对 test.sh 文件具有和 tom 一样的权限，则操作命令为：

```
[tom@localhost ~]$ setfacl -m u:john:rwx test.sh
[tom@localhost ~]$ getfacl --omit-header test.sh
user::rwx
user:john:rwx
group::r--
mask::rwx
other::r--
```

通过以上显示的内容，可以看到用户 john 已经拥有了 rwx 的权限。而此时的 mask 值也被设定为 rwx 的权限。这是因为 ACL_MASK 定义了 ACL_USER，ACL_GROUP_OBJ 和 ACL_GROUP 的最大权限，而此时只有 ACL_USER(user:john:rwx)，ACL_GROUP_OBJ(group::r--)两项，可以看出其最大值为 rwx，所以 mask 值显示为 rwx。现在再来看 test.sh 的 Linux 权限，使用 ls -l 显示该文件，命令如下：

```
[tom@localhost ~]$ ls -l test.sh
-rwxrwx r--+ 1 tom admin 0 10 月 11 21:39 test.sh
```

尽管以上文件的权限显示为 rwxrwx r--，但如果现在 admin 组的用户想要执行 test.sh 脚本文件，仍会被拒绝操作。原因在于实际上 admin 组的用户只有可写(r--)权限。这里当中显示的 rwx 是 ACL_MASK 的值而不是用户组的权限。所以从这里就可以知道，如果一个文件后面有+标记，都需要用 getfacl 来确认它的实际权限，以免发生混淆。

假如现在设置 test.sh 的 mask 值为只有读取的权限，则操作命令为：

```
[tom@localhost ~]$ setfacl -m m::r test.sh
[tom@localhost ~]$ getfacl --omit-header test.sh
user::rwx
user:john:rwx#effective:r--
group::r--
mask::r--
other::r--
```

此时可以看到 ACL_USER 旁边多了个#effective:r--，也就是说，现在 John 用户只有可读的权限。这是因为 ACL_MASK 定义了 ACL_USER，ACL_GROUP_OBJ 和 ACL_GROUP 的最大权限，而此时 mask 值显示为只有可读(r)权限，那么 ACL_USER 的最大权限也就是可读(r)权限。虽然这里给 ACL_USER 设置了其他权限，但是其真正有效果的只有可读权限。

(5)默认权限

上面所讲的都是 Access ACL，也就是对文件而言。下面简单介绍一下 Default ACL。Default ACL 是指对于一个目录进行默认权限设置，并且在此目录下建立的文件都将继承此目录的 ACL。

比如现在 tom 用户建立了一个 data 目录，操作命令如下：

```
[tom@localhost ~]$ mkdir data
```

tom 希望所有在此目录下建立的文件都可以被 john 用户所访问，那么就可以对 data 目录设置 Default ACL，操作命令为：

```
[tom@localhost ~]$ setfacl -m d:u:john:rw data
[tom@localhost ~]$ getfacl --omit-header data/
user::rwx
group::r-x
other::r-x
default:user::rwx
default:user:john:rw-
default:group::r-x
default:mask::rwx
default:other::r-x
```

若 tom 用户在 data 目录下建立一个 test.txt 文件，则操作命令为：

```
[tom@localhost ~]$ touch data/test.txt
```

再分别使用长格式显示该文件和使用 getfacl 命令显示该文件 ACL 设置，操作命令为：

```
[tom@localhost ~]$ ll data/test.txt
-rw-rw-r--+ 1 tom admin 0 10 月 12 00:21 data/test.txt
[tom@localhost ~]$ getfacl --omit-header data/test.txt
user::rw-
user:john:rw-
```

```
group::r-x #effective:r--
mask::rw-
other::r--
```

通过以上内容，可以看到 tom 用户在 data 目录下建立的文件，john 用户自动就拥有了读写权限。

4.6　项目小结

本项目首先介绍了用户账号文件、用户密码文件和用户组账号文件，接着对用户和用户组的相关操作进行了详细的讲解，最后在技术拓展部分介绍了如何改变文件的属主和属组以及如何对文件进行 ACL 的设置。

4.7　强化练习

本项目练习可对应工单 8 开展，工单列表见附录。可从随书配套光盘中获取电子版工单。