



10年口碑积累，成功培养50000多名研发工程师，铸就专业品牌形象

华清远见的企业理念是不仅要做好良心教育、做专业教育，更要做受人尊敬的职业教育。

《Linux Shell 编程从初学到精通》

作者：华清远见

专业始于专注 卓识源于远见

第 2 章 Linux 文件系统和文本编辑器

本章目标

本章首先介绍 Linux 用户和用户组的概念以及管理的常用命令；其次，重点介绍文件和目录操作，包含文件和目录复制、移动和删除等常用命令、文件和目录权限的概念以及管理的常用命令、查找文件命令——find；最后，介绍两种文本编辑器：命令行环境下的 vi 编辑器和 GNOME 桌面环境的 Gedit 编辑器。熟悉 Linux 系统基础命令的读者可以跳过此章，直接进入下一章学习。

2.1 用户和用户组管理

Linux 是多用户多任务的操作系统，用户（user）和用户组（group）的管理是 Linux 使用者应该了解和掌握的基础之一。本节介绍 Linux 用户和用户组的管理，它也是 2.2 节文件操作的基础。

2.1.1 用户管理常用命令

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。值得注意的是，用户的角色是通过 UID 来识别的，用户的 UID 是全局唯一的。Linux 用户可以分为三类。

root 用户（也称为超级用户）：系统唯一，是真实的。该用户既可以登录系统，可以操作系统任何文件和命令，拥有最高权限。

虚拟用户：这类用户也被称为伪用户或假用户，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 等；这类用户是系统自身拥有的，而非后来添加的，当然，我们也可以添加虚拟用户。

普通真实用户：这类用户能登录系统，但只能操作其根目录的内容，权限受到限制，这类用户都是系统管理员自行添加的。

Linux 用户管理的常用命令主要有：用户账号添加命令 `useradd` 或 `adduser`、修改用户命令 `usermod`、删除用户命令 `userdel` 及用户口令管理命令 `passwd` 等，下面将详细介绍这些命令。

1. 用户账号添加命令——`useradd` 或 `adduser`

`useradd` 和 `adduser` 是完全等价的两条命令，都是用于创建新的用户账号。我们以 `useradd` 为代表介绍它们的用法，命令格式如下：

```
useradd [option] [username]
```

其中，[option]为 `useradd` 命令选项，[username]是要创建的用户名。执行该命令后，将在系统中做以下一些事情：

在 `/etc/passwd` 文件中增添了一行记录。

在 `/home` 目录下创建新用户的主目录，并将 `/etc/skel` 目录中的文件复制到该目录中。

使用了该命令后，新建的用户暂时无法登录，因为还没有为该用户设置口令，需要再用 `passwd` 命令为其设置口令后，才能登录。用户的 UID 和 GID 是 `useradd` 自动选取的，它是将 `/etc/passwd` 文件中的 UID 加 1，将 `etc/group` 文件中的 GID 加 1。

表 2-1 列出了 `useradd` 命令中各选项的含义。

表 2-1 `useradd` 或 `adduser` 命令的选项及其意义

选 项	意 义
-g [initial_grp]	用于添加用户账号时指定该用户的私有组。如不指定-g 参数， <code>useradd</code> 命令将自动建立与用户账号同名的组作为该账号的私有组
-G [grp...]	用于添加附属组
-D	用于显示或设置 <code>useradd</code> 命令所使用的默认值
-d [directory]	指定用户主目录，如果此目录不存在，则同时使用-m 选项来创建主目录
-m	使用者目录若不存在，则自动建立
-u UID	指定用户的用户号，如果同时有-o 选项，则可以重复使用其他用户的标识号。注意，ID 值不能为负值，预设值为最小不得小于 99 而逐次增加。0~99 传统上保留给系统账号使用

使用 `useradd` 或 `adduser` 命令增加新用户时，系统将为用户创建一个与用户名相同的组，称为私有组，这一方法是为了能让新用户与其他用户隔离，确保安全性的措施。如果要改变私有组的名字，可以使用 `-g` 选项来完成。

例 2-1 通过增加一个用户 `wang`，并查看其相关信息，来帮助用户理解该命令所执行的操作。可以通过 `tail -l` 命令查看文件 `/etc/passwd` 来查看文件新创建的用户和新创建的用户主目录，而通过查看 `/etc/shadow` 创建用户的密码。

```
#例 2-1: 创建用户 wang 并查看其相关信息
[root@localhost ~]# useradd wang

#查看 passwd 文件中添加的用户账号信息
[root@localhost ~]# tail -l /etc/passwd
sshd:x:74:74:Privilege-separated SSH:/var/empty/ssh:/sbin/nologin
tcpdump:x:72:72:::/sbin/nologin
pulse:x:496:494:PulseAudio System Daemon:/var/run/pulse:/sbin/nologin
smolt:x:495:489:Smolt:/usr/share/smolt:/sbin/nologin
torrent:x:494:488:BitTorrent Seed/Tracker:/var/lib/bittorrent:/sbin/nologin
haldaemon:x:68:68:HAL daemon:~/sbin/nologin
sqUID:x:23:23::/var/spool/sqUID:/sbin/nologin
gdm:x:42:42::/var/lib/gdm:/sbin/nologin
wyq:x:500:500:wyq:/home/wyq:/bin/bash
wang:x:501:501::/home/wang:/bin/bash           #新创建的 wang 用户
#查看加密后的用户账号及密码信息
[root@localhost ~]# tail -l /etc/shadow
sshd:!!:14630::::::
tcpdump:!!:14630::::::
pulse:!!:14630::::::
smolt:!!:14630::::::
torrent:!!:14630::::::
haldaemon:!!:14630::::::
sqUID:!!:14630::::::
gdm:!!:14630::::::
wyq:
$6$NXWGCok7f.G2KQOk$LZjQAXpkC2SoXlF1gbgrUaP.b6rWkXMyU7bNQZnQZAKp1h6pwpwew65Y3Cwb77H0WfV.ftl2mh
v2LqlFlI1TP0:14630:0:99999:7:::
wang:!!:14671:0:99999:7:::
#查看所建立账号的主目录
[root@localhost ~]# ls /home
wang wyq
[root@localhost ~]#
```

例 2-1 中，首先通过 `useradd` 命令创建了一个新的用户 `wang`，然后通过“`tail -l /etc/passwd`”命令查看文件 `/etc/passwd`，可以看到，为 `wang` 用户的 UID 为 501，创建的新目录为 `/home/wang`，接着通过“`tail -l /etc/shadow`”命令查看文件 `/etc/shadow`，可以获得用户 `wang` 的密码，由于还没有为 `wang` 用户创建密码，可以看到 `wang` 后为“!!”，表示用户密码不可用，最后通过 `ls` 命令查看 `/home` 目录，可以看到为 `wang` 用户创建的主目录已经存在。

2. 修改用户账号——`usermod`

`usermod` 命令可用来修改用户账号的各种属性，包括用户主目录、私有组、登录 Shell 等内容，`usermod` 的命令格式如下：

```
usermod [option] [username]
```

其中, [option]为 useradd 命令选项, 而[username]是需修改的用户名。表 2-2 列出了 usermod 命令选项及其意义。

表 2-2 usermod 命令的选项及其含义

选 项	意 义
-d [directory]	修改用户登入时的目录
-e [days]	修改账号的有效期限, days 表示天数
-g [group]	修改用户所属的群组
-l [login_name]	变更用户登录时的名称为 login_name
-p [password]	修改用户密码
-s [shell]	指定用户登录的 Shell, 如果不设置, 则选用系统预设的 Shell

需要注意的是, 最好不要使用 usermod 命令修改用户密码, 因为如果用 usermod 命令, 则显示在文件“/etc/shadow”中的密码是明密码, 应该用 passwd 命令修改密码。下面的例 2-2 使用 usermod 来修改密码, 可以看到通过 tail 命令查看时, 该用户的密码为明码, 这样是很危险的, 如果其他用户打开文件“/etc/shadow”, 就可使用该用户名和密码登录到该账号上。

#例 2-2: 使用 usermod 修改密码

```
[root@localhost ~]# usermod -p 123456 wang
```

#通过 tail -l 查看用户 wang 的密码

```
[root@localhost ~]# tail -l /etc/shadow
```

```
sshd:!!:14630:::::::
```

```
tcpdump:!!:14630:::::::
```

```
pulse:!!:14630:::::::
```

```
smolt:!!:14630:::::::
```

```
torrent:!!:14630:::::::
```

```
haldaemon:!!:14630:::::::
```

```
sqUID:!!:14630:::::::
```

```
gdm:!!:14630:::::::
```

```
wyq:
```

```
$6$NXWGCok7f.G2KQOk$LzjQAXpkC2SoX1F1gbgrUaP.b6rWkXMyU7bNQZnQZAKp1h6pwpwew65Y3Cwb77H0WfV.fTl2mh  
v2LqlFlI1TP0:14630:0:99999:7:::
```

#可以看到 wang 用户的密码为 123456

```
wang:123456:14672:0:99999:7:::
```

```
[root@localhost ~]#
```

还需注意, usermod 不允许改变已登录用户的用户账号名称, 当用户修改 UID 时, 也必须确认这个用户没有在电脑上执行任何程序。下面的例 2-3 演示了修改正在登录用户的账名称时所产生的错误。

#例 2-3: 删除正在登录用户的用户账号所产生的错误

#wyq 是当前登录用户, 用 usermod 修改时发生错误

```
[wyq@localhost root]$ usermod wyq
```

```
bash: /usr/sbin/usermod: Permission denied
```

```
[wyq@localhost root]$
```

3. 删除用户账号命令——userdel

userdel 命令非常简单, 只有一个可选项-r, 如果在 userdel 后加上-r 选项, 则在删除用户的同时也一并删除存储在/home 目录下的该用户目录和文件。下面的例 2-4 说明了使用-r 选项和不使用-r 选项的区别。

#例 2-4: userdel 命令使用-r 选项和不使用-r 选项的区别

```
[root@localhost ~]# ls /home
```

```
wang wang1 wang2 wyq
[root@localhost ~]# userdel -r wang1
[root@localhost ~]# ls /home
wang wang2 wyq
[root@localhost ~]# userdel wang2
[root@localhost ~]# ls /home
wang wang2 wyq
[root@localhost ~]#
```

从例 2-4 中可以看出, 当使用 `-r` 选项删除用户账号 `wang1` 时, 目录 `/home` 下的子目录 `wang1` 被删除了; 删除 `wang2` 用户账号时若不使用 `-r` 选项, 则 `wang2` 子目录仍然存在。

4. 用户口令管理命令——`passwd`

用户管理一个重要的内容就是用户口令管理。用户账号刚建立时是没有口令的, 但是会被系统锁定, 必须为其指定口令才能使用, 这时需使用 `passwd` 命令, 下面是 `passwd` 命令的语法格式:

```
passwd [option] [username]
```

其中, `[option]` 为 `passwd` 命令选项, `[username]` 为用户名。表 2-3 为 `passwd` 命令选项及其意义。

表 2-3 `passwd` 命令的选项及其意义

选 项	意 义
<code>-l</code>	锁定用户口令, 即禁止使用该用户账号
<code>-u</code>	口令解锁
<code>-d</code>	关闭使用者的密码确认功能, 使用者在登录时可以不用输入密码, 只有具备 <code>root</code> 用户的使用者才可使用
<code>-f</code>	强迫用户下次登录时修改密码
<code>-l [login_name]</code>	变更用户登录时的名称为 <code>login_name</code>
<code>-s</code>	显示指定使用者的密码认证种类, 同样只有具备 <code>root</code> 权限的用户才可使用

当用户作为普通用户修改自己的密码时, 首先会提示用户的原密码, 然后会要求用户重新输入两次来验证用户新口令, 如果口令一致, 则新口令设置成功。而超级用户 `root` 为用户设定密码时, 则不需要原密码。

例 2-1 中创建的用户账号 `wang` 还无法使用, 还需要使用 `passwd` 命令来创建账号密码。下面的例 2-5 使用 `passwd` 命令为 `wang` 用户创建账号密码。

```
#例 2-5: 为 wang 用户创建账号密码
[root@localhost ~]# passwd wang
Changing password for user wang.
New password:
Retype new password:
BAD PASSWORD: it is based on a dictionary word
BAD PASSWORD: is too simple
passwd: all authentication tokens updated successfully.

#查看加密后的用户账号及密码信息, 可以看出该账号后无“!!”, 表示可以使用该账号了
[root@localhost ~]# tail -1 /etc/shadow
sshd:!!:14630::::::
tcpdump:!!:14630::::::
pulse:!!:14630::::::
smolt:!!:14630::::::
torrent:!!:14630::::::
```

```

haldaemon:!!:14630:::
sqUID:!!:14630:::
gdm:!!:14630:::
wyq:          $6$NXWGC0k7f.G2KQ0k$LZjQAXpkC2SoX1F1gbgrUaP.b6rWkXMyU7bnQZnQZAKplh6pwpXew65Y3Cw
b77H0WfV.fTl2mhv2Lq1F1I1TP0:14630:0:99999:7:::
wang:         $6$s2PKmNwn$wjpv2JzAA5EPOGkU4Iir4/hqvTlYkTARuB6B23OwavXGeqUKumG8BPHXBRkDggxAlx
po2gXEt4ltFmHwbVy100:14671:0:99999:7:::
[root@localhost ~]#
    
```

通过例 2-1 和例 2-5 对比可以看出，用户 wang 在文件“/etc/shadow”中已经发生了改变，凡是在文件“/etc/shadow”的用户名含“!!”时，该用户账号不可使用，而为加密信息时是可以使用的。

2.1.2 用户组管理常用命令

用户组就是具有相同特征的用户集合体，用户和用户组的关系是多对多的，一个用户可以属于多个用户组，同样，一个用户组可以包含多个用户。Linux 下每个文件都有一个用户组，当创建一个文件或目录时，系统会赋予其一个用户组关系。

用户组的管理命令包含用户组添加命令 `groupadd`、用户组修改命令 `groupmod` 和用户组删除命令 `groupdel`，下面将对这些命令进行详解。

1. 用户组添加命令——groupadd

`groupadd` 可指定用户组名称来建立新的用户组，需要时可从系统中取得新用户组值。其语法格式为：

```
groupadd [option] [groupname]
```

其中，[option]为 `groupadd` 命令选项，[groupname]是将要创建的用户组名，表 2-4 为 `groupadd` 命令的选项及其意义。

表 2-4 `groupadd` 命令的选项及其意义

选 项	意 义
-g GID	除非使用-o 参数，否则 GID 值必须是唯一且数值不可为负，预设值以/etc/login.defs 为准
-o GID	运行 GID 不唯一
-r	加入组 GID 号，且其 GID 号低于 499 系统账号
-f	新增一个已经存在的用户组账号，系统会出现错误信息，然后结束

`groupadd` 命令其实用起来非常简单，下面的例 2-6 添加了一个 GID 为 666 的用户组 wangyq，可以在文件/etc/group 的目录中查看到 GID 为 666 的用户组 wangyq。

```

#例 2-6: 使用 groupadd 命令添加用户组 wangyq
[root@localhost ~]# groupadd -g 666 wangyq
[root@localhost ~]# tail -1 /etc/group
wbpriv:x:88:sqUID
smolt:x:489:
torrent:x:488:
haldaemon:x:68:
sqUID:x:23:
gdm:x:42:
wyq:x:500:
wang:x:501:
wangyq:x:666:
wangl:x:345:
    
```

```
[root@localhost ~]#
```

如果调用 `groupadd` 命令时不设置 GID 号，如下面命令：

```
groupadd group1
```

则在系统中增加一个新组 `group1`，新组的组标识 GID 是在当前最大组标识的基础上加 1。

2. 用户组修改命令——`groupmod`

`groupmod` 可指定用户组名称来修改新的用户组号或用户组名称，其语法格式为：

```
groupmod [option] [groupname]
```

其中，[option]为 `groupmod` 命令选项， [groupname]为用户组名，表 2-5 为 `groupmod` 命令选项及其意义。

表 2-5 `groupmod` 命令的选项及其意义

选 项	意 义
-g GID	用户指定新的 GID
-o GID	重复使用 GID
-n	为群组改名

例 2-7 是将用户组 `wangyq` 的群组号修改为 555，通过命令 `tail` 可以看出该用户组的 GID 修改成功。

#例 2-7: 使用 `groupmod` 修改用户组号

```
[root@localhost ~]# groupmod -g 555 wangyq
```

```
[root@localhost ~]# tail -l /etc/group
```

```
wbpriv:x:88:sqUID
```

```
smolt:x:489:
```

```
torrent:x:488:
```

```
haldaemon:x:68:
```

```
sqUID:x:23:
```

```
gdm:x:42:
```

```
wyq:x:500:
```

```
wang:x:501:
```

```
wangyq:x:555:
```

```
wangl:x:345:
```

```
[root@localhost ~]#
```

3. 用户组删除命令——`groupdel`

`groupdel` 可指定用户组名称来删除已有的用户组，其语法格式为

```
groupdel [groupname]
```

该命令非常简单，但需要注意的是，如果该用户组中包含某些用户，则必须先删除这些用户，然后才能删除该用户组。下面的例 2-8 是一个删除用户组 `wangyq` 的例子，通过 `tail` 命令可以看出该用户组已找不到了。

#例 2-8: 用 `groupdel` 删除用户组

```
[root@localhost ~]#groupdel wangyq
```

#查看文件/etc/group 可以看到用户组 `wangyq` 已被删除

```
[root@localhost ~]# tail -l /etc/group
```

```
stapusr:x:490:
```

```
wbpriv:x:88:sqUID
```

```
smolt:x:489:
```

```
torrent:x:488:  
haldaemon:x:68:  
sqUID:x:23:  
gdm:x:42:  
wyq:x:500:  
wang:x:501:  
wangl:x:345:  
[root@localhost ~]#
```

2.2 文件和目录操作



在多数操作系统中都有文件的概念。文件是 Linux 用来存储信息的基本结构，它被命名（称为文件名）的存储在某种介质（如磁盘、光盘和磁带等）上的一组信息的集合。Linux 文件均为无结构的字符流形式。文件名是文件的标识，它由字母、数字、下画线和圆点组成的字符串构成。Linux 要求文件名的长度限制在 255 个字符以内。用户应该选择有意义的文件名。

为了便于识别和管理，用户可以把扩展名作为文件名的一部分，文件名与扩展名之间用圆点分开，扩展名对于文件分类是十分有用的。用户可能已经对某些大众已接纳的标准扩展名比较熟悉了，例如，C 语言编写的源代码文件总是具有 C 的扩展名。用户可以根据自己的需要，加入自己的文件扩展名。

在计算机系统存有大量的文件，如何有效地组织与管理它们，并为用户提供一个使用方便的接口是文件系统的一大任务。Linux 系统以文件目录的方式来组织和管理系统中的所有文件。所谓文件目录，就是将所有文件的说明信息采用树形结构组织起来，即我们常说的目录。也就是说，整个文件系统有一个“根”（root），然后在根上分“杈”（directory），任何一个分杈上都可以再分杈，杈上也可以长出“叶子”。“根”和“杈”在 Linux 中被称为“目录”或“文件夹”，而“叶子”则代表一个个的文件。实践证明，此种结构的文件系统效率比较高。

对文件进行访问时，需要用到“路径”（Path）的概念。何谓路径？顾名思义，路径是指从树形目录中的某个目录层次到某个文件的一条道路。路径的主要构成是目录名称，中间用“/”符号分开。任一文件在文件系统中的位置都是由相应的路径决定的。用户在对文件进行访问时，要给出文件所在的路径，这又分相对路径和绝对路径。相对路径是从用户工作目录开始的路径；绝对路径是指从“根”开始的路径，也称为完全路径。

应该注意到，在树形目录结构中到某一确定文件的绝对路径和相对路径均只有一条。绝对路径是确定不变的，而相对路径则随着用户工作目录的变化而不断变化。这一点对于我们以后使用某些命令如 cp 和 tar 等大有好处。

理解了文件、目录和路径的概念后，下面讲述文件和目录操作中经常用到的各种命令。

2.2.1 文件操作常用命令

文件操作常用命令包括文件清单命令 ls、文件复制命令 cp、文件移动命令 mv 和删除文件命令 rm，下面我们逐一讲述这些命令。

1. 文件清单命令——ls

ls 命令是英文单词 list 的简写，其功能是列出目录下的文件和子目录的相关信息。ls 命令是用户最常用的命令之一，因为用户需要不时地查看某个目录中的信息，该命令类似于 DOS 下的 dir 命令，其语法格式为：

```
ls [option] [file or directory]
```


其中, [option]为命令选项, [file or directory]对应于文件或目录。如果[file or directory]为目录, 该命令将列出该目录下的所有子目录与文件; 如果[file or directory]为文件, 则ls命令将输出该文件名及相关信息。默认情况下, 输出条目按字母顺序排序, 当未给出目录名或是文件名时, 就显示当前目录的信息。表 2-6 给出了ls命令选项及其意义。

表 2-6 ls 命令的选项及其意义

选 项	意 义
-a	显示指定目录下所有的子目录与文件, 包括隐藏文件
-A	显示指定目录下所有的子目录与文件, 包括隐藏文件, 且不列出“.”和“..”
-b	对文件名中的不可显示字符用八进制逃逸字符显示
-c	按文件的修改时间排序
-C	分成多列显示各项
-d	如果参数是目录, 只显示其名称, 而不显示其下的各文件。往往与l选项一起使用, 以得到目录的详细信息
-f	不排序, 该选项将使lts选项失效, 使用aU选项有效
-i	在输出的第一列显示文件的i节点号
-l	以长格式来显示文件的详细信息, 每行列出的信息依次是: 文件类型与权限、链接数、文件属主、文件属组、文件大小、最近修改的时间、名字
-L	若指定的名称为一个符号链接文件, 则显示链接所指向的文件
-m	输出按字符流格式, 文件跨页显示, 以逗号分开
-n	输出格式与l选项相同, 只不过在输出文件属主和属组是用相应的UID号和GID号来表示, 而不是实际名称
-o	与l选项相同, 只是不显示拥有者信息
-p	在目录后面加一个“/”
-q	将文件名中的不可显示字符用“?”代替
-r	按字母逆序或最早优先的顺序显示输出结果
-R	递归式地显示指定目录的各个子目录中的文件
-s	给出每个目录项所用的块数, 包括间接块
-t	显示时按修改时间(最近优先)而不是按名字排序; 若文件修改时间相同, 则按字典顺序, 修改时间取决于是否使用了c或u选项。默认的时间标记是最后一次修改时间
-u	显示时按文件上次存取的时间(最近优先)而不是按名字排序。即将-t的时间标记修改为最后一次访问的时间
-x	按行显示出各排序项的信息

下面 2-9 演示了ls命令-l选项的用法, -l是ls命令最常用的选项, 用于列出文件和目录的详细信息。

#例 2-9: 列出/home/wyq/shell/chapter15目录下的所有文件和目录的详细信息

```
[root@localhost ~]# ls -l /home/wyq/shell/chapter15
total 16
-rwxr-xr-x. 1 root root 543 2010-03-02 03:26 dev1.sh
-rwxr-xr-x. 1 root root 114 2010-03-02 04:01 dev2.sh
lrwxrwxrwx. 1 root root 9 2010-03-02 04:00 file1.sh -> /dev/null
-rwxr-xr-x. 1 root root 221 2010-03-02 01:48 proc1.sh
-rwxr-xr-x. 1 root root 169 2010-03-02 02:33 proc2.sh
[root@localhost ~]#
```

从例 2-9 的结果可以看出, ls -l命令列出的详细信息包含文件或目录的权限、所属用户和用户组、创建时间等。需要指出的是, Linux系统将ll命令作为ls -l命令的别名, 即ll命令也可以列出文件或目录的详细信息。

2. 文件复制命令——cp

cp 命令可以将给出的文件或目录复制到另一文件或目录中，就如同 DOS 下的 copy 命令一样，功能非常强大。cp 命令的一般格式如下所示：

```
cp [option] [source] [destination]
```

该命令把指定的源文件复制到目标文件或把多个源文件复制到目标目录中，其中，[option]为 cp 命令选项，[source]为源文件，而[destination]为目标目录或目标文件。表 2-7 给出了 cp 命令选项及其意义。

表 2-7 cp 命令的选项及其意义

选 项	意 义
-a	该选项通常在复制目录时使用，它保留链接、文件属性，并递归地复制目录
-d	复制时保留链接
-f	删除已经存在的目标文件而不提示
-i	在覆盖目标文件之前将给出提示要求用户确认。回答 y 时，目标文件将被覆盖，是交互式复制
-p	此时 cp 除复制源文件的内容外，还将把其修改时间和访问权限也复制到新文件中
-r	若给出的源文件是目录文件，此时 cp 将递归复制该目录下所有的子目录和文件。此时目标文件必须为一个目录名
-l	不进行复制操作，只是链接文件

需要说明的是，为防止用户在不经意的情况下用 cp 命令破坏另一个文件，如用户指定的目标文件名已存在，用 cp 命令复制文件后，这个文件就会被新源文件覆盖。因此，建议用户在使用 cp 命令复制文件时，最好使用 i 选项，在覆盖文件时进行最后确认。

下面的例 2-10 演示了将文件 chapter13/file.txt 复制到目录 testdir 中，可以看出，在不加任何选项的情况下，文件的权限改变了，同时文件修改的时间也改变了。

#例 2-10: 将文件 chapter13/file.txt 复制到目录 testdir 中

```
[root@localhost shell]# cp chapter13/file.txt testdir
[root@localhost shell]# ls -l chapter13/file.txt testdir/file.txt
-rw-rw-r--. 1 root root 51 2010-03-03 23:36 chapter13/file.txt
-rw-r--r--. 1 root root 51 2010-03-04 00:46 testdir/file.txt
[root@localhost shell]#
```

下面的例 2-11 在 cp 命令的基础上加了 -a 选项，则两个文件权限和修改时间未发生改变。

#例 2-11: 将文件 chapter13/file.txt 复制到目录 testdir 中，且不改变权限和修改日期

```
[root@localhost shell]# ls -l chapter13/file.txt testdir/file.txt
-rw-rw-r--. 1 root root 51 2010-03-03 23:36 chapter13/file.txt
-rw-rw-r--. 1 root root 51 2010-03-03 23:36 testdir/file.txt
[root@localhost shell]#
```

cp 命令的 -r 选项可实现目录的复制，如果直接复制目录，将产生如下错误：

```
[root@localhost shell]# cp rmdir testdir
cp: omitting directory `rmdir'          #直接复制目录产生的错误
[root@localhost shell]#
```

下面的例 2-12 演示了正确的目录复制过程，将目录 chapter13 中的全部内容复制到目录 testdir 中。

#例 2-12: 将目录中全部内容复制到另一个目录中

#查看 shell 目录下的文件和子目录

```
[root@localhost shell]#ls
chapter13 chapter15 chapter16 functions.new rmdir testdir tt.sh
```

#查看 testdir 目录下的文件和子目录

```
[root@localhost shell]#ls testdir
file.txt
```

```
#使用 cp -r 命令复制目录 chapter13 及其目录下的子目录和文件到 testdir
[root@localhost shell]# cp -r chapter13 testdir
#使用 cp 命令后再次查看目录 testdir
[root@localhost shell]# ls testdir
chapter13  file.txt
[root@localhost chapter13]#
```

可以看出,在复制前, `testdir` 目录仅仅有一个文件 `file.txt`, 而使用“`cp -r chapter13 testdir`”命令时, `testdir` 目录中存在目录 `chapter13`, 说明 `chapter13` 目录复制成功。

3. 文件移动命令——`mv`

`mv` 命令可用于将文件或目录从一个位置移动到另一个位置, 同时可移动多个文件。 `mv` 命令还可用于重命名, 其一般的格式为:

```
mv [option] [source] [destination]
```

其中, `[option]` 为 `mv` 命令选项, `[source]` 为源目录或文件, `[destination]` 为目标目录或目标文件, 如果 `[destination]` 类型是文件时, `mv` 命令将所给的源文件或目录重命名为给定的目标文件名, 此时, 源文件只能有一个 (也可以是源目录); 如果 `[destination]` 是已存在的目录名称, 源文件或目录参数可以有多个, `mv` 命令将各参数指定的源文件全部移至目标目录中。在跨文件系统移动时, `mv` 先复制, 再将原有文件删除, 从而导致该文件的链接丢失。表 2-8 给出了 `mv` 命令选项及其意义。

表 2-8 `mv` 命令选项及其意义

选 项	意 义
-i	交互方式操作。如果 <code>mv</code> 操作将导致对已存在的目标文件的覆盖, 此时系统询问是否重写, 要求用户回答 <code>y</code> 或 <code>n</code> , 这样可以避免误覆盖文件
-f	禁止交互操作。在 <code>mv</code> 操作要覆盖某已有的目标文件时不给任何指示, 指定此选项后, <code>i</code> 选项将不再起作用
-p	移动时保持权限

如果所给目标文件 (不是目录) 已存在, 此时该文件的内容将被新文件覆盖。为防止用户用 `mv` 命令破坏另一个文件, 使用 `mv` 命令移动文件时, 最好使用 `i` 选项, 例 2-13 是交互式 `mv` 命令的应用, 该例使用 `mv` 命令将文件 `file1.txt` 重命名为 `file2.txt`。

```
#例 2-13: 将文件 file1.txt 重命名为 file2.txt
[root@jselab chapter13]# ls -il file*
 427 -rw-r--r--. 1 root root 0 2010-05-19 06:18 file1.sh
28476 -rw-r--r--. 1 root root 0 2010-05-19 06:19 file1.txt
28477 -rw-r--r--. 1 root root 0 2010-05-19 06:20 file2.txt
28478 -rw-r--r--. 1 root root 0 2010-05-19 06:21 file5
28479 -rwxr-xr-x. 1 root root 51 2010-05-18 09:22 file.txt
#使用 mv 交互式命令将 file1.txt 重命名为 file2.txt
[root@jselab chapter13]# mv -i file1.txt file2.txt
mv: overwrite `file2.txt'? y
#再次查看 chapter13 中含 file 的文件, 可以看出 file1.txt 消失, file2.txt 的修改时间改变
[root@jselab chapter13]# ls -il file*
 427 -rw-r--r--. 1 root root 0 2010-05-19 06:18 file1.sh
28476 -rw-r--r--. 1 root root 0 2010-05-19 06:20 file2.txt
28478 -rw-r--r--. 1 root root 0 2010-05-19 06:19 file5
28479 -rwxr-xr-x. 1 root root 51 2010-03-04 01:05 file.txt
[root@jselab chapter13]#
```

对于目录的重命名，同样可以使用 `mv` 命令实现，例 2-14 将目录 `testdir` 命名为 `chapter2`，而且目录中的内容也未发生改变。

例 2-14: 将目录 `testdir` 命名为 `chapter2`

```
[root@localhost shell]# ls
chapter13 chapter15 chapter16 functions.new rmdir testdir tt.sh
[root@localhost shell]# mv testdir chapter2
[root@localhost shell]# ls
chapter13 chapter15 chapter16 chapter2 functions.new rmdir tt.sh
[root@localhost shell]#
```

4. 删除文件命令——`rm`

`rm` 命令提供删除文件功能，该命令可以删除目录中的一个或多个文件或子目录，它也可以将某个目录及其下的所有文件及子目录均删除。删除单个文件不用带任何参数；如果是删除整个目录及目录下的所有文件，需要带 `-rf` 参数。其一般格式为：

```
rm [option] [fileName or directoryName]
```

其中，`[option]` 为 `rm` 命令选项，`[fileName or directoryName]` 为文件名或子目录名。表 2-9 列出了 `rm` 命令选项及其意义。

表 2-9 `rm` 命令的选项及其意义

选项	意义
<code>-f</code>	忽略不存在的文件，从不给出提示
<code>-r</code>	指示 <code>rm</code> 将参数中列出的全部目录和子目录均递归地删除
<code>-i</code>	进行交互式删除

使用 `rm` 命令要小心，因为一旦文件被删除，它是不能被恢复的。为了防止这种情况的发生，可以使用 `i` 选项来逐个确认要删除的文件。如果用户输入 `y`，文件将被删除，而如果输入任何其他字段，文件则不会被删除。

如果删除的是文件，则其格式为：

```
rm fileName
```

如果删除的是目录，则其格式为：

```
rm -rf directoryName
```

例 2-15 演示了 `rm` 命令的用法，该例删除目录 `shell` 下的文件 `function.new` 和目录 `rmdir`。可以看出，在删除文件时，Linux 会提示用户是否要删除该文件。

#例 2-15: 使用 `rm` 删除一个目录及该目录下的文件和子目录

```
[root@jselab shell]# ls
chapter13 file1.sh file1.txt file2.txt file5 testvi
#使用交互式 rm 选项删除空目录 chapter13
[root@jselab shell]# rm -ir chapter13
rm: remove directory `chapter13'? y
[root@jselab shell]# ls
file1.sh file1.txt file2.txt file5 testvi
[root@jselab shell]#
```

2.2.2 目录操作常用命令

目录操作常用命令包括创建目录命令 `mkdir`、删除目录命令 `rmdir` 和目录切换命令 `cd`，下面我们逐一讲述这些命令。

1. 创建目录命令——mkdir

mkdir 命令用于创建目录，为文件分门别类地放在这些创建的目录中，其执行要指定需要创建的目录即可，与 DOS 下的 md 命令类似，其语法格式如下所示：

```
mkdir [option] [directoryName]
```

其中，[option]为 mkdir 命令选项，[directoryName]是需要创建的目录名称。需要说明的是，该命令创建由[directoryName]命名的目录，要求创建目录的用户在当前目录中(directoryName 的父目录中)具有写权限，并且[directoryName]不能是当前目录中已有的目录或文件名称，表 2-10 列出了 mkdir 命令选项及其意义。

表 2-10 mkdir 命令的选项及其意义

选 项	意 义
-m	对新建目录设置存取权限
-p	可以是一个路径名称。此时若路径中的某些目录尚不存在，加上此选项后，系统将自动建立好那些尚不存在的目录，即一次可以建立多个目录
-v	每次创建新目录都显示信息

在进行目录的创建时，可以使用-m 选项设置目录的权限，下面的例 2-16 创建了一个目录名为 tsk，让所有的用户都有 rwx（即读、写、执行的权限）。

```
#例 2-16: 通过 mkdir 创建一个名为 tsk 的目录，所有的用户都可读和执行。
[root@localhost shell]# ls
chapter13 chapter15 chapter16 chapter2
[root@localhost shell]# mkdir -m 777 tsk          #创建目录，并指定权限
[root@localhost shell]# ls -l
total 20
drwxr-xr-x. 2 root root 4096 2010-03-04 00:08 chapter13
drwxr-xr-x. 2 root root 4096 2010-03-02 04:01 chapter15
drwxr-xr-x. 2 root root 4096 2010-03-03 19:51 chapter16
drwxr-xr-x. 3 root root 4096 2010-03-04 01:05 chapter2
drwxrwxrwx. 2 root root 4096 2010-03-04 03:07 tsk
[root@localhost shell]#
```

上面的例 2-16 用-m 777 指定了目录的读、写、执行权限，2.2.3 节将介绍文件和目录的权限控制。

如果在创建目录的过程中父目录不存在，则可使用-p 选项一起创建父目录和需要创建的子目录，下面的例 2-17 演示了 mkdir 命令-p 选项的用法。

```
#例 2-17: 演示 mkdir 命令-p 选项的用法
#第 1 条命令: testdir 不存在，mkdir 不带-p 选项将出错
[root@localhost shell]# mkdir testdir/test
mkdir: cannot create directory `testdir/test': No such file or directory
#第 2 条命令: 加上-p 选项成功创建 testdir 目录及其子目录 test
[root@localhost shell]# mkdir -p testdir/test
[root@localhost shell]# ls
chapter13 chapter15 chapter16 chapter2 testdir tsk
[root@localhost shell]# cd testdir
[root@localhost testdir]# ls
test
[root@localhost testdir]#
```

在例 2-17 中，由于 testdir 目录是不存在的，因而，mkdir 不带-p 选项创建 testdir/test 时产生语法错误；当带上-p 选项后，mkdir 能够成功地创建 testdir 目录及其子目录 test。

2. 删除目录命令——rmdir

`rmdir` 命令可以删除一个或多个目录，在删除目录时，目录必须为空。`rm` 命令可以同时删除多个目录，在删除某一目录时，必须拥有该目录的父目录的写和执行的权限。其语法格式如下所示：

```
rmdir [option] [directoryName]
```

其中，[option]为 `rmdir` 命令选项，[directoryName]为目录名，表 2-11 列出了 `rmdir` 命令选项及其意义。

表 2-11 `rmdir` 命令的选项及其意义

选 项	意 义
-p	递归删除。当子目录删除后，其父目录为空时，也将会一同被删除。如果整个路径被删除或者由于某种原因保留部分路径，则系统会在标准输出上显示相应的信息
--ignore-fail-on-non-empty	忽略非空目录的错误信息

由于 `rmdir` 仅仅能删除父目录中只包含空子目录的情况，如果目录中存在文件，则使用 `rmdir` 和 `rmdir -p` 命令是无法删除该目录的，所以，需使用“`rm -rf [directoryName]`”命令代替 `rmdir`，删除某目录时，同样要求该用户拥有对父目录的写权限。

下面的例 2-18 演示了 `rmdir -p` 和 `rm -rf` 的用法。例子中首先使用 `rmdir -p` 命令删除只存在一个空子目录 `test1` 的目录 `testdir1`，可以看出该命令递归地删除了目录 `testdir1` 和其子目录 `test1`；然后再次使用 `rmdir -p` 命令删除含有文件 `test.txt` 和空的子目录 `test2` 的目录 `testdir2`，可以看出 `testdir2` 的空的子目录 `test2` 成功地被删除，但提示目录 `testdir2` 不为空，无法删除；最后使用 `rm -rf` 命令则成功地删除了不为空的目录 `testdir2`。

```
#例 2-18: 使用 rmdir 删除存在子目录的目录
#目录 testdir1 中仅仅存在一个空子目录 test1, 使用 rmdir -p 命令循环删除目录 testdir1 和 test1
[root@localhost wyq]# rmdir -p testdir1/test1
[root@localhost wyq]# ls
chapter13 Chapter7 Chapter8 myfile nusers sn.txt testdir2
#查看目录 testdir2 目录下的文件和子目录
[root@localhost wyq]# ls testdir2
test.txt test2
#由于 testdir2 中存在文件, 使用 rmdir -p 命令仅仅删除了空目录 testdir2
[root@localhost wyq]# rmdir -p testdir2/test2
rmdir: failed to remove directory "testdir2": 目录非空
#使用 rm -rf 命令则成功地删除了含有文件的目录 testdir2
[root@localhost wyq]# rm -rf testdir2
[root@localhost wyq]# ls
chapter13 Chapter7 Chapter8 myfile nusers sn.txt
[root@localhost wyq]#
```

3. 目录切换命令——`cd`

在 Linux 操作系统的文本模式下，在目录之间进行切换都需要通过命令来完成。显然，这没有图形化界面下使用鼠标操作那么方便。但是，Shell 中提供了一个目录切换字符——`cd`，在该字符的帮助下，系统管理员可以轻松地在不同的目录之间进行切换。其语法格式如下所示：

```
cd [directoryName]
```

该命令用于变换工作目录，其中，[directoryName]可以为绝对路径，也可为相对路径，绝对路径从“/”（指代根）开始，然后切换到所需的目录；相对路径从当前目录开始，当前目录可以是文件系统的任何地方。需要注意的是，想要访问目录或文件的相对路径之前，要确保知道当前工作目录的位置，如果标明的是到另一个目录或文件的绝对路径，则不必担心文件系统所在的位置，如果不能肯定，可输入 `pwd` 命令，就会显示当前的工作目录。下面的表 2-12 是 `cd` 命令的具体使用说明。

表 2-12 `cd` 命令的具体使用说明

命 令	使用说明
cd	返回登录目录
cd ~	同样是返回登录目录
cd /	返回系统根目录
cd /root	返回到根用户或超级用户（在安装时创建的账号）的主目录，但必须是根用户才能访问该目录
cd /home	返回到 home 目录，home 目录通常为用户登录目录的上级目录
cd ..	向上移动一级目录
cd -	返回上次访问的目录

在目录切换的过程中，还有两个比较重要的特殊字符，分别为“.”与“..”符号，其中“.”表示当前目录。这个符号很重要，在很多地方都需要用到。如在定义 PATH 环境变量的时候，在路径的最后需要加上这个“.”号，这表示当前目录。如果系统管理员想运行当前目录下的一个脚本文件，如 setup.sh，则可以不用绝对路径，而直接使用“./setup.sh”命令，其中这个“.”符号就代表当前目录。在 cd 命令中也可以使用“.”符号，如“cd ./setup”命令表示进入到当前目录的下一个子目录 setup 下面。

另外一个特殊的字符就是“..”（英文状态下的双点号），它在系统中表示的是上一级目录。如果管理员利用 cd 命令定义到一个目录后，又想回到上一级目录中，则可以使用命令“cd ..”命令来实现，注意：cd 命令与点号之间要有空格。下面举例说明 cd 命令的用法，例 2-19 通过使用 cd 命令返回到根目录中，并通过 ls 命令返回根目录下的文件和子目录，然后通过“cd -”命令回到上次访问的目录。

```
#例 2-19: 一个使用 cd 的例子
[wyq@jsselab ~]$ cd / #查看根目录
[wyq@jsselab /]$ ls
bin  etc  lost+found  opt  sbin  sys  var
boot home media      proc selinux tmp
dev  lib  mnt          root srv   usr
[wyq@jsselab /]$ cd - #返回到上次访问的目录
/home/wyq
[wyq@jsselab ~]$
```

“~”符号也是一个比较有用的符号，如系统管理员想从任何目录中回到用户的主目录下，除了按原路返回外，要回到用户的主目录下，有一个很便捷的方式，就是通过一个特殊的字符“~”来完成。通常情况下，当管理员创建某个用户后，在系统的/home 目录中会以这个用户的名字建立一个文件夹，这个文件夹所在的目录就是用户的主目录。当用户不知道自己所处的是哪个目录，而需要迅速回到自己的主目录时，可以使用命令“cd ~”来实现。下面的例 2-20 用来说明“~”及其用法，该例首先使用 su 命令切换到 root 用户下，然后通过“cd ~”命令迅速回到了 root 用户的主目录。

```
#例 2-20: 使用“~”快速到达用户主目录
[wyq@localhost home]$ su
密码: #输入密码后通过“cd ~”命令回到 root 用户的主目录
[root@localhost home]# cd ~
[root@localhost ~]#
```

2.2.3 文件和目录权限管理

Linux 系统中的每个文件和目录都有访问许可权限，用它来确定用户能以何种方式对文件和目录进行访问和操作。

文件或目录的访问权限分为只读、只写和可执行三种。以文件为例，只读权限表示只允许读其内容，而禁止对其做任何的更改操作；可执行权限表示允许将该文件作为一个程序执行。文件被创建时，文件所有者自动拥有对该文件的读、写和可执行权限，以便于对文件的阅读和修改。用户也可根据需要把访问权限设置为任何组合。

有三种不同类型的用户可对文件或目录进行访问：文件所有者、同组用户、其他用户。文件所有者一般是文件的创建者，他可以允许同组用户访问文件，还可以将文件的访问权限赋予系统中的其他用户，从而使系统中每一位用户都能访问该所有者拥有的文件或目录。

每一文件或目录的访问权限都有三组，每组用三位表示，分别为文件属主的读、写和执行权限，与属主同组的用户的读、写和执行权限，以及系统中其他用户的读、写和执行权限。下面的例 2-21 列出一个文件和一个目录的详细信息。

```
#例 2-21: 查看文件和目录的权限
[root@jselab shell-book]# ls -l CARGO.db
-rw-r--r--. 1 root root 225 03-15 13:34 CARGO.db
[root@jselab shell-book]# ls -l example/
总计 4
drwxr-xr-x. 2 root root 4096 2009-09-05 ch01
[root@jselab shell-book]#
```

例 2-21 中，CARGO.db 是一个文件，详细信息的第 1 个字符是横线 (-)；而 example 是一个目录，详细信息的第 1 个字符是 d。权限字段 r 代表只读，w 代表写，x 代表可执行。

文件和目录都有三组权限，以 CARGO.db 为例，第 1 组是 rw，表示该文件的属主具有读写权限；第 2 组是 r，表示与文件属主同组的用户只有读权限；第 3 组是 r，其他用户也只有读权限；ch01 目录也是同样的，图 2-1 显示了 CARGO.db 文件和 ch01 目录的文件权限对应关系。

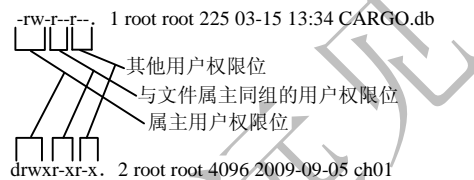


图 2-1 文件权限位示意图

文件和目录权限管理依赖于两个极其重要的命令：更改文件（目录）权限命令 chmod 和更改文件（目录）用户命令 chown，下面我们详细讲述这两个命令。

1. 更改文件（目录）权限命令——chmod

chmod 命令用于更改文件或目录的访问权限，它有两种用法：一种是包含字母和操作符表达式的文字设定法；另一种是包含数字的数字设定法。文字设定法的格式为：

```
chmod [userType] [signal] [type] [filename]
```

chmod 命令三种参数类型如表 2-13 所示。

表 2-13 chmod 三种参数类型

用户类型 (userType)	数学符号 (signal)	文件类型 (type)
u 表示用户 (user)，即文件或目录的所有者	+ 添加某个权限	r 可读
g 表示同组 (group)，即与文件属主同组的用户	- 取消某个权限	w 可写
o 表示其他 (others) 用户	= 赋予给定权限并取消其他所有权限	x 可执行
a 表示所有 (all) 用户。它是系统默认值		

下面的例 2-22 利用 chmod 命令改变 testvi 文件的权限，将 testvi 文件的权限改成：属主可读、可写、可执行，同组用户可读、可写。

```
#例 2-22: 修改文件 testvi 的属性
[root@localhost chapter13]# ls -l testvi
-rw-r--r--. 1 root root 0 2010-03-04 04:11 testvi
```



```
[root@localhost chapter13]# chmod u+x,g+w testvi
[root@localhost chapter13]# ls -l testvi
-rwxrw-r--. 1 root root 0 2010-03-04 04:11 testvi
[root@localhost chapter13]#
```

注意：`chmod` 命令的不同参数之间需要用逗号隔开。

那么什么是 `chmod` 的数字设定法呢？首先需要了解用数字表示属性的含义：`0` 表示没有权限，`1` 表示可执行权限，`2` 表示可写权限，`4` 表示可读权限，然后将其相加。所以，数字属性的格式应为 3 个从 `0` 到 `7` 的八进制数，其顺序是 `u`，`g`，`o`。比如例 2-22 中的

```
chmod u+x,g+w testvi
```

命令等价于下面的命令：

```
chmod 764 testvi
```

下面的例 2-23 新创建了一个文件 `testvi1`，然后使用了 `chmod` 数字设定法对 `testvi1` 进行修改权限，可以看出，执行后的文件 `testvi1` 和上面 `testvi` 文件的权限相同，其中 `touch` 命令可以用于文件的创建。

#例 2-23: 使用 `chmod` 数字设定法对文件 `testvi1` 进行权限控制

```
[root@localhost chapter13]# touch testvi1      #创建文件 testvi1
[root@localhost chapter13]# ls -l testvi1      #查询文件 testvi1 的权限
-rw-r--r--. 1 root root 0 2010-03-04 04:22 testvi1
[root@localhost chapter13]# chmod 764 testvi1  #修改 testvi1 的权限
[root@localhost chapter13]# ls -l testvi1      #查看修改后的 testvi1 的权限
-rwxrw-r--. 1 root root 0 2010-03-04 04:22 testvi1
[root@localhost chapter13]#
```

2. 更改文件（目录）属主命令——`chown`

利用 `chown` 命令可以改变文件或目录的属主。一般来说，这个指令只由系统管理者（`root`）使用，一般使用者没有权限改变别人的文件或目录属主，也没有权限将自己的文件属主更改为别人的，只有系统管理者（`root`）才有这样的权限。`chown` 命令的一般形式为：

```
chown [option] [owner] [filename]
```

其中，`[option]` 为 `chown` 命令选项，`[owner]` 为改变后的用户属主，而 `[filename]` 为需要改变属主的文件或目录。表 2-14 是 `chown` 命令的选项及其意义。

表 2-14 `chown` 命令的选项及其意义

命 令	意 义
-c	若该文件或目录属主确实已经更改，才显示其更改动作
-h	改变符号链接文件的属主时不影响该链接所指向的目标文件
-f	若该文件或目录属主无法被更改，也不要显示错误信息
-v	显示属主变更的详细资料
-R	对目前目录下的所有文件与子目录进行相同的拥有者变更（即以递归的方式逐个变更）

需要注意的是，当用户要改变一个文件的属主时，所使用的用户除了是 `root` 用户外，还可以是目标属组的成员。下面的例 2-24 是使用 `chown` 命令将属于 `root` 用户的文件 `file.txt` 改为 `wyq` 用户的。

#例 2-24: 使用 `chown` 改变用户属主

```
[root@localhost chapter2]# ls -l
total 8
drwxr-xr-x. 2 root root 4096 2010-03-04 01:34 chapter13
-rwxr-xr-x. 1 root root 51 2010-03-03 23:36 file.txt
#更改 file.txt 文件的属主为 wyq 用户
[root@localhost chapter2]# chown -h wyq file.txt
```

```
[root@localhost chapter2]# ls -l
total 8
drwxr-xr-x. 2 root root 4096 2010-03-04 01:34 chapter13
-rwxr-xr-x. 1 wyq root 51 2010-03-03 23:36 file.txt
[root@localhost chapter2]#
```

3. 特殊权限命令——SUID 与 SGID

除了上面提到的基本权限操作外，还有所谓的特殊权限存在。由于特殊权限会拥有一些“特权”，因而，用户若无特殊需要，不应该去打开这些权限，避免安全方面出现严重漏洞，甚至摧毁系统。但有时却需要没有被授权的用户完成某项任务，例如 `passwd` 程序，它允许用户改变口令，这就要求改变 `/etc/passwd` 文件的口令域。然而系统管理员决不允许普通用户拥有直接改变这个文件的权利，为了解决这个问题，SUID/SGID 便应运而生，下面介绍了这两个特殊权限的说明。

SUID: 当一个设置了 SUID 位的可执行文件被执行时，该文件以所有者的身份运行，也就是说，无论谁来执行这个文件，它都拥有文件所有者的特权，可以任意使用该文件所有者能使用的全部系统资源。如果所有者是 `root`，那么执行人就有超级用户的特权了。

SGID: 当一个设置了 SGID 位的可执行文件被执行时，该文件将具有所属组的特权，任意存取整个组所能使用的系统资源；若一个目录设置了 SGID，则所有被复制到这个目录下的文件，其所属的组都会被重设为和这个目录一样，除非在复制文件时加上 `-p` 选项，才能保留原来所属的群组设置。还可以使用符号方式来设置 SUID/GUID。

SUID 和 SGID 占据了 `ls -l` 清单中 `x` 位相同的空间，如果开始设置了可执行权限 `x` 位，则其相应的位置用小写的 `s` 表示；如果没有设置可执行权限 `x` 位，则其相应位置表示为大写的 `S`。设置和除去 SUID 和 SGID 很间接，设置 SUID 位和去除 SUID 位分别使用如下命令：

```
chmod u+s [filename] #设置[filename]的 SUID 位
chmod u-s [filename] #去除[filename]的 SUID 位
同样设置和去除 SGID 的命令分别为：
```

```
chmod g+s [filename] #设置[filename]的 SGID 位
chmod g-s [filename] #去除[filename]的 SGID 位
```

下面举一个例子讲解如何设置 SUID 位，例 2-25 中的 `file.txt` 开始的权限没有设置 SUID，使用 `chmod` 命令后就改变了 SUID。

```
#例 2-25: 使用 chmod 命令实现 SUID/GUID
[root@localhost chapter2]# ls -l
total 8
drwxr-xr-x. 2 root root 4096 2010-03-04 01:34 chapter13
-rwxr-xr-x. 1 wyq root 435 2010-03-04 15:48 file.txt
[root@localhost chapter2]# chmod u+s file.txt
[root@localhost chapter2]# ls -l
total 8
drwxr-xr-x. 2 root root 4096 2010-03-04 01:34 chapter13
-rwsr-xr-x. 1 wyq root 435 2010-03-04 15:48 file.txt
[root@localhost chapter2]#
```

SUID 的程序往往伴随着一定的安全问题。有时一个 SUID 程序与一个系统程序（或库函数）之间的交互作用会产生连程序的编制者也不知道的安全漏洞。一个典型的例子是 `/usr/lib/preserve` 程序，它被 `vi` 和 `ex` 编辑器使用，当用户在写出对文件的改变前被意外地与系统中断时，它可以自动制作一个正被编辑的文件的副本。这个保存的 `preserve` 程序将改变写在一个专门的目录内的临时文件中，然后利用 `/bin/mail` 程序发送给用户一个“文件已经被存”的通知。

2.2.4 查找文件命令——find

find 命令是 Linux 系统查找文件的命令，find 命令能帮助用户在使用、管理 Linux 的日常事务时方便地找出用户所需要的文件。find 命令的基本格式是：

```
find [路径] [选项] [操作]
```

在上述 find 命令中，路径是 find 命令所查找的目录路径，例如，用“.”来表示当前目录，用“/”来表示系统根目录。选项用于指定查找条件，如：可以指定按照文件属主、更改时间、文件类型等条件来查找，下面的表 2-15 列出了 find 命令的常用选项及其意义。

表 2-15 find 命令常用选项及其意义

选 项	意 义
name	根据文件名查找文件
perm	根据文件权限查找文件
prune	使用这一选项可以使 find 命令不在当前指定的目录中查找，如果同时使用 -depth 选项，那么 -prune 将被 find 命令忽略
user	根据文件属主查找文件
group	根据文件所属的用户组查找文件
mtime -n +n	根据文件的更改时间查找文件，-n 表示文件更改时间距今在 n 天之内，+n 表示文件更改时间距今在 n 天前
nogroup	查找无有效所属组的文件，即该文件所属的组在 /etc/groups 中不存在
nouser	查找无有效属主的文件，即该文件的属主在 /etc/passwd 中不存在
-newer file1 ! file2	查找更改时间比文件 file1 新但比文件 file2 旧的文件
type	查找某一类型的文件，type 后跟的子选项及其意义如下： b:块设备文件 d:目录 c:字符设备文件 p:管道文件 l:符号链接文件 f:普通文件
size n:[c]	查找文件长度为 n 块的文件，带有 c 时表示文件长度以字节计
depth	在查找文件时，首先查找当前目录中的文件，然后在其子目录中查找

find 命令的操作用于指定结果的输出方式，表 2-16 列出了 find 命令的操作名称及其意义。

表 2-16 find 命令的操作名称及其意义

操作名称	意 义
print	将匹配的文件输出到标准输出
exec	对匹配的文件执行该参数所给出的 Shell 命令。相应命令的形式为 'command' { } \;。注意，{ } 和 \; 之间的空格
ok	和 -exec 的作用相同，只不过以一种更安全的模式来执行该参数所给出的 Shell 命令，在执行每一个命令之前，都会给出提示，让用户来确定是否执行

下面我们举几个例子来说明 find 命令的用法，以及其路径、选项、操作的用法。首先，请看下面的例 2-26。

```
#例 2-26: 演示 find 命令的 print 操作
#第 1 条命令: 查找当前目录下文件名以 t 开头的，且文件属主具有读、写、执行权限的文件
[root@jselab shell-book]# find . -name 't*' -perm 744 -print
./DEBUG/trapdebug.sh
./test.sh
./traploop.sh
./testvar.sh
#第 2 条命令: 查找更改时间距今 90 天内的文件
```

```
[root@jsselab shell-book]# find . -mtime -90 -print
.
./hfile
./loggg
./newfile
[root@jsselab shell-book]#
```

例 2-26 中的两条命令变换了 `find` 的查找条件，但是，操作都是用 `print`，即打印出满足查找条件的所有结果，第 1 条命令查找当前目录下文件名以 `t` 开头的，且文件属主具有读、写、执行权限的文件，命令用“.”表示当前目录，`-name 't*'`表示以 `t` 开头的文件名，其中的“*”字符表示任意字符，这种用法称为通配，本书第 3 章将会介绍这一概念。`-perm 744`表示文件属主具有读、写、执行权限，744 是表示文件权限的数字设定法。第 1 条命令结果的 `./`表示当前目录下的文件，`./DEBUG`则表示子目录 `DEBUG` 下的文件。第 2 条命令查找更改时间距今 90 天内的文件，使用了 `-mtime -n` 用法。

`find` 命令 `exec` 操作将对匹配的文件执行该参数所给出的 Shell 命令，下面的例 2-27 演示了 `exec` 操作的用法。

```
#例 2-27: 演示 find 命令的 exec 操作
[root@jsselab shell-book]# find /etc -type f -name "rc*" -exec ls -l {} \;
-rwxr-xr-x. 1 root root 220 2009-05-02 /etc/rc.d/rc.local
-rwxr-xr-x. 1 root root 2596 2009-05-02 /etc/rc.d/rc
-rwxr-xr-x. 1 root root 23756 2009-05-02 /etc/rc.d/rc.sysinit
-rw-r--r--. 1 root root 401 2009-05-02 /etc/event.d/rc4
-rw-r--r--. 1 root root 401 2009-05-02 /etc/event.d/rc3
-rw-r--r--. 1 root root 817 2009-05-02 /etc/event.d/rcS
-rw-r--r--. 1 root root 644 2009-05-02 /etc/event.d/rc1
-rw-r--r--. 1 root root 420 2009-05-02 /etc/event.d/rc6
-rw-r--r--. 1 root root 401 2009-05-02 /etc/event.d/rc2
-rw-r--r--. 1 root root 550 2009-05-02 /etc/event.d/rc0
-rw-r--r--. 1 root root 401 2009-05-02 /etc/event.d/rc5
-rw-r--r--. 1 root root 519 2009-05-02 /etc/event.d/rcS-sulogin
[root@jsselab shell-book]#
```

上面例 2-27 中的命令表示查找 `/etc` 目录下文件名以 `rc` 开头的普通文件，并对查找结果执行 `ls -l` 命令，即列出查找结果的详细文件信息。例 2-27 的结果表明该命令列出了 `/etc` 目录及其子目录下所有以 `rc` 开头的普通文件的详细信息，列出详细信息是 `exec` 操作进一步处理的结果。

`ok` 操作和 `exec` 的作用相同，只不过以一种更安全的模式来执行该参数所给出的 Shell 命令，在执行每一个命令之前，都会给出提示，让用户来确定是否执行。在执行一些危险操作时，建议使用 `ok` 操作，比如：下面的例 2-28 试图删除 `/var/log` 目录下更改时间距今 3 天内的所有文件，由于删除文件的操作是不可恢复的，因此，`find` 命令使用 `ok` 操作在删除文件之前给出提示，等待用户确定后再执行删除操作。

```
#例 2-28: 演示 find 命令的 ok 操作
#删除/var/log目录下更改时间距今3天内的所有文件
[root@jsselab shell-book]# find /var/log -mtime -3 -ok rm {} \;
< rm ... /var/log > ? n
< rm ... /var/log/audit/audit.log > ? y
< rm ... /var/log/spooler > ? y
< rm ... /var/log/rpmpkgs > ? y
< rm ... /var/log/wtmp > ? y
< rm ... /var/log/secure-20100705 > ? y
.....
[root@jsselab shell-book]#
```

由例 2-28 的结果可以看出，一旦 `find` 命令使用了 `ok` 操作，在对查找结果执行进一步操作前，Shell 出现提示信息，用户可输入 `y` 或 `n` 选项确认是否执行该项操作。

总之，find 命令的选项在指定查找条件时显得灵活多变，可以根据文件的各种属性来查找文件；对于查找到的结果，find 命令可以有三种处理方式，print 仅将结果输出到屏幕，而 exec 和 ok 可以对结果进一步处理，区别在于：exec 直接进行处理，而 ok 在处理之前出现提示信息，供用户最终确定是否执行该操作。

2.3 文本编辑器



编写 Shell 脚本需要使用文本编辑器，Linux 系统中的文本编辑器种类极多，如 Emacs、Kvim、vi、Gedit、Kate 等。本节介绍两种文本编辑器：第一种是 vi 编辑器，它适用于命令行界面，当我们使用 SSH Secure Shell 或 PuTTY 登录 Linux 系统时，vi 编辑器是唯一选择；第二种是 Gedit 编辑器，它适用于图形化界面，当我们登录 GNOME 桌面环境时，一般使用 Gedit 编辑文件。

2.3.1 vi 编辑器

vi 是 UNIX 世界上最通用的全屏编辑器，Linux 中用的是 vi 的加强版 vim，vim 同 vi 完全兼容。在 Linux 系统中，vi 和 vim 是完全等价的两条命令，都可以启动 vi 编辑器。

vi 编辑器可以执行输出、删除、查找、替换、块操作等众多文本操作，而且用户可以根据自己的需要对其进行定制，这是其他编辑程序所没有的。vi 编辑器以命令行的方式处理文本，尽管不如图形化处理方式直观，但它具有操作速度快、功能全面等优点。另外，vi 不是一个排版程序，它不像 Word 或 WPS 那样可以对字体、格式、段落等其他属性进行编排，它只是一个文本编辑程序。vi 或 vim 的基本格式如下：

```
vi [option] [filename...]  
或 vim [option] [filename...]
```

其中，[option]是选项，[filename]是需要编辑的一个或多个文件名。如果在启动 vi 时没有指定文件名，则 vi 命令会自动产生一个无名的空文件。如果指定的[filename]文件不存在，则 vi 将创建一个名为[filename]的新文件。启动 vi 后，消息行会显示文件的名称、文件中的行数和字数。消息行显示的信息随着所运行命令的不同而不同，如果文件中的任何一行上有一个波浪线 (~)，就说明没有足够的行来填满屏幕。注意，vi 并不锁住所编辑的文件，因此，多个用户可能同时编辑一个文件，最后保存的文件版本将被保留。表 2-17 列出了 vi 命令选项及其意义。

表 2-17 vi 命令的选项及其意义

选 项	说 明
-c command	在对文件进行编辑前，先执行 command 命令
-r filename	恢复文件 filename
-R	以只读方式编辑文件
+n file	编辑 file 文件，并将光标置于第 n 行
+ file	编辑 file 文件，并将光标置于最后一行
+/string file	编辑 file 文件，并将光标置于第一个保护 string 所表示的字符串的行

输入 vi 命令打开 vi 编辑器后，vi 编辑器的运行状态共有以下两种模式。

(1) 一般模式 (Normal mode)

输入 vi 命令进入 vi 文本编辑器的时候，就是一般模式了。该模式将用户的输入看做命令，这个模式允许用户移动游标，且允许搜索文本功能；图 2-2 是用 vi 编辑器打开 stack.sh 文件时的一般模式，此时，vi 编辑器的最后一行是文件名、文件包含的字符数和字节数。退出 vi 编辑器、保存当前的修改也是在一般模式中进行的，在一般模式下按冒号按钮，将出现如图 2-3 所示的界面，vi 编辑器的最后一行显示冒号，冒号后面输入保存、退出等命令，这些命令共有四种，如表 2-18 所示。

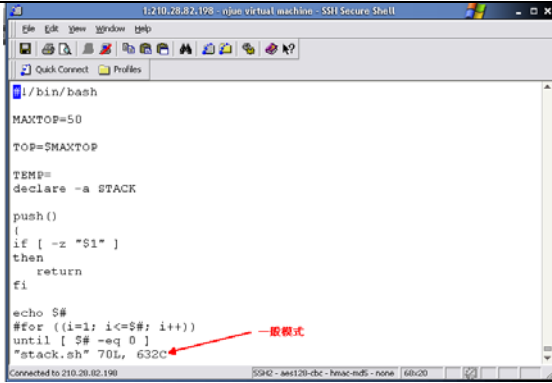


图 2-2 vi 编辑器的一般模式：打开文件时

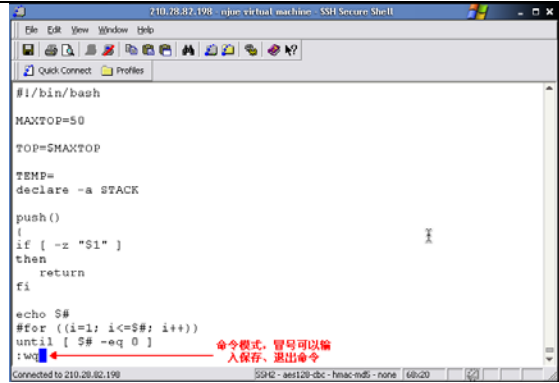


图 2-3 vi 编辑器的一般模式：保存和退出

表 2-18 vi 编辑器文本保存和退出命令

保存和退出命令	描述
w	将编辑的文本存储
q	离开 vi 文本编辑器
q!	曾修改过文本，但是不想存储，使用该命令强制离开 vi 文本编辑器
wq	存储文本并离开 vi 文本编辑器

(2) 插入模式 (Insert mode)

在一般模式下按下 I、o、a 等字母都可以进入编辑模式，在此模式下，vi 将用户的输入插入到当前光标位置，修改暂时保存到缓冲区，按“Esc”键则从编辑模式退回到一般模式。图 2-4 显示了 vi 编辑器的编辑模式，vi 编辑器的最后一行显示 INSERT，这表示能插入新字符。

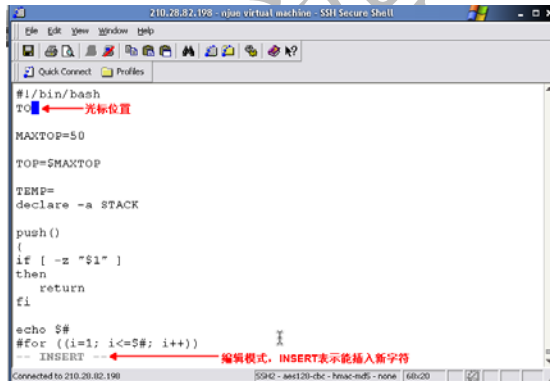


图 2-4 vi 编辑器的编辑模式

vi 编辑器在插入模式下编辑文件时存在两个关键技巧：移动光标到适当的位置和编辑文本。vi 编辑器提供了丰富的移动光标命令，如表 2-19 所示，对于较短的文件，使用 k、j、h、l 进行上下左右地移动就可以满足需求。但是，对于较长的文件，经常需要根据段落、句子、行数来移动光标。

表 2-19 用于移动光标的 vi 命令

命令	动作
h	将光标向左移动
j、加号 (+)、Enter	将光标向下移动
k、减号 (-)	将光标向上移动
l	将光标向右移动
}	将光标移动到当前段落的末尾
{	将光标移动到当前段落的开头
)	将光标移动到当前句子的末尾

(将光标移动到当前句子的开头
^	移动到当前行的第一个非空字符
\$	移动到当前行末尾
:n	移动到行 n

编辑文件似乎不需要展开多解释，一般情况下，我们可以像使用 Windows 系统的记事本一样编辑文件，表 2-20 列出了 vi 编辑器经常用到的编辑命令，熟练地使用这些命令能够提高编辑文件的效率，比如：dd 能够删除光标所在的整行文本，d\$能够删除当前光标位置到该行结束的所有文本等。

表 2-20 常用的 vi 编辑命令

命 令	动 作
x	删除光标当前位置的字符
dd	删除光标所在的整行文本
d\$	删除当前光标位置到该行结束的所有文本
dw	从当前光标位置向前删除单词
J	将下一行文本内容合并到本行行尾
a	在当前光标位置后附加内容
A	在当前光标所在行的后面附加内容

我们在编辑文件时常常需要搜索单词或替换单词，vi 编辑器也提供了搜索和替换功能。首先按下斜杠按钮 (/)，光标会自动移到 vi 编辑器下方的命令行，用户输入待搜索的字符串，按下“Enter”键开始搜索，vi 编辑器可能用以下三种方式响应用户的搜索。

当满足搜索条件的字符串出现在当前光标位置后面时，vi 编辑器将光标跳转到第一个满足搜索条件的位置。

当满足搜索条件的字符串出现在当前光标位置前面时，vi 编辑器将跳转到从头开始的第一个满足搜索条件的位置。

文件中不存在满足条件的字符串时，vi 编辑器提示错误信息。

图 2-5 显示了 vi 编辑器的搜索功能，它是在/etc/services 文件中搜索 ssh 字符串，vi 编辑器下方出现斜杠 (/)，在“/”后面输入待搜索的字符串。

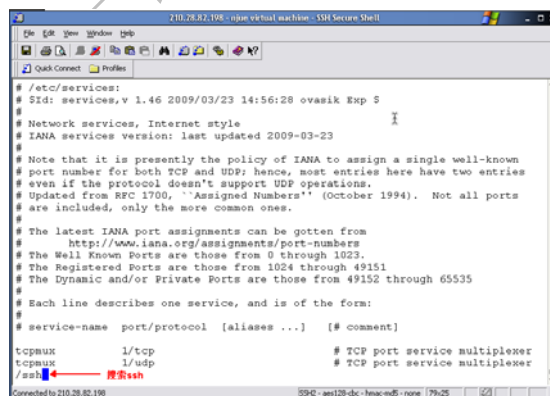


图 2-5 vi 编辑器的搜索功能

vi 编辑器的替换命令的基本格式为：

```
:s/old_string/new_string
```

上述替换命令表示将第一次出现的 old_string 替换成 new_string，当然，可以在上述替换命令后加上 g 选项，表示将所有的 old_string 替换成 new_string，命令为：

```
:s/old_string/new_string/g
```

图 2-6 显示了 vi 编辑器的替换功能，该命令将所有的 TOP 替换为 STACKTOP，表 2-21 归纳了 vi 编辑器的搜索和替换命令。我们可以看到，vi 编辑器在替换文本时还可以指定行号的范围。

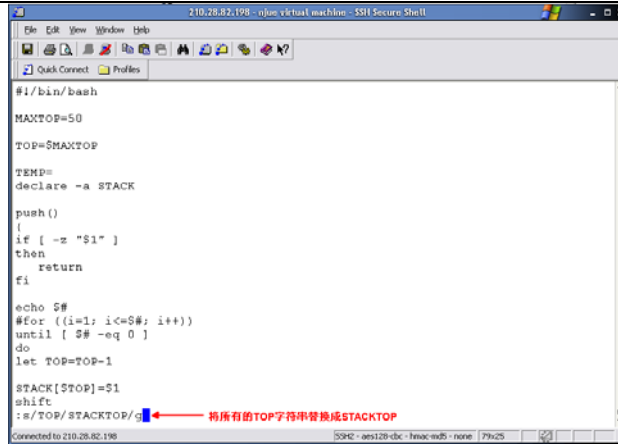


图 2-6 vi 编辑器的替换功能

表 2-21 vi 编辑器的搜索和替换命令

命令名称	意义
/word	自当前光标位置向下搜索名字为 word 的字符串
?word	自当前光标位置向上搜索名字为 word 的字符串
:n1,n2s/word1/word2/g	在 n1 行与 n2 行之间搜索名字为 word1 的字符串，并将其替换为 word2
:1,\$s/word1/word2/g	在第 1 行与最后一行之间搜索名字为 word1 的字符串，并将其替换为 word2

2.3.2 Gedit 编辑器

Gedit 是 GNOME 桌面环境下一个兼容 UTF-8 的文本编辑器。它简单易用，有良好的语法高亮显示功能，对中文支持很好，还支持包括 GB2313、GBK 在内的多种字符编码。Gedit 利用 GNOME VFS 库，它还可以编辑远程文件。它支持完整的恢复和重做系统以及查找和替换，还支持包括多语言拼写检查和一个灵活的插件系统，可以动态地添加新特性。Gedit 还增加了一些小特性，包括行号显示、括号匹配、文本自动换行、当前行高亮以及自动文件备份。Gedit 需要在 GNOME 图形化环境下使用，Gedit 的启动命令为：

```
gedit file1 file2 ... fileN
```

gedit 是 Gedit 的启动命令，可以同时打开多个文件，每个文件将在 Gedit 编辑器上生成一个标签，单击相应的标签可以看到相应的文件。Gedit 能将 Shell 脚本、C 语言等文件的关键字以不同的颜色显示，即所谓的语法高亮显示功能，极大地方便了程序员编辑文件。图 2-7 显示了 Gedit 编辑器的界面，图中打开了 stack.sh、CARGO.db 和 colon.sh 三个文件，可以看出，Gedit 的界面与 Windows 的文本编辑器风格相似，读者一定不会感到陌生。

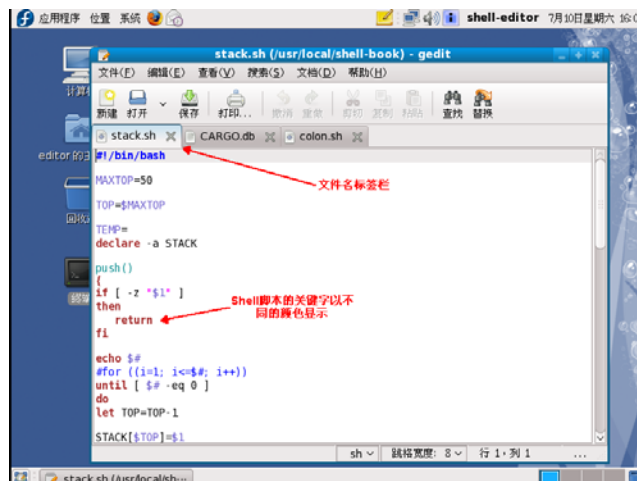


图 2-7 Gedit 编辑器的界面

Gedit 编辑器包含菜单栏、工具栏、显示区域、输出窗口和状态栏等部分，其菜单栏包括如下项目：

File (文件): 用于处理新文件、保存现有文件和打印文件。

Edit (编辑): 用于操作活动缓冲区中的文本，以及设置编辑器首选项。

View (查看): 用于设置在窗口中显示的编辑器特性，已经设置文本高亮模式。

Search (搜索): 用于查找和替换活动编辑器缓冲区中的文本。

Tools (工具): 用于访问 Gedit 中安装的插件工具。

Documents (文档): 用于管理在缓冲区中打开的文件。

Help (帮助): 用于访问完整的 Gedit 手册。

File 菜单提供了 **Open Location** 选项，允许使用 WWW 世界中流行的标准统一资源标识符 (URL) 格式从网络上打开文件，该格式表示可以用于访问文件的协议 (如 HTTP 或 FTP)、文件所在的服务器和该文件在服务器的完整访问路径。

Gedit 有若干种可用的插件，但并非所有的插件都是默认安装的。表 2-22 列出了当前可用的插件。

表 2-22 Gedit 插件列表

插 件	描 述
Change Case	更改所选文本的大小写
Document Tools	报告单词、行、字符和非空字符的数量
External Tools	在编辑器中提供一个 Shell 环境，用于执行命令和脚本
File Brower Pane	提供一个简单的文件浏览器，便于选择要编辑的文件
Indent Lines	提供高级缩进和不缩进功能
Insert Date/Time	在当前光标位置插入各种格式的当前日期和时间
Modelines	在编辑器窗口底部提供 Emacs 样式的消息行
Python Console	在编辑器窗口底部提供一个交互式控制台，用于使用 Python 编程语言输入命令
Snippets	允许保存经常使用的文本段，以便在文本的任何地方检索
Sort	对整个文件或所选文本快速排序
Spell Checker	为文本文件提供字典拼写检查
Tag List	提供经常使用的字符串列表，使用户可以方便地在文本中输入这些字符串
User name	在当前光标位置插入当前用户的登录名

2.4 本章小结



本章简明扼要地介绍了与 Shell 脚本编程相关的 Linux 基础知识，包含用户和用户组管理的常用命令、文件和目录操作的常用命令、文件和目录的权限管理、查找文件命令——find、两种文本编辑器——vi 和 Gedit。

由于篇幅所限，本章不可能全面、系统地介绍 Linux 基础知识，对此还需进一步学习的读者可以参考 Linux 基础类的参考书。

2.5 上机提议



1. 假定目录下有如下文件：

```
[root@localhost chapter16]# ls
bad_style_indent.sh      login1.sh
color_script1.sh        login2.sh
color_script2.sh        no_interaction_script1.sh
```

```

file1                no_interaction_script.sh
file11               no_shift.sh
file2                package_sed.sh
file_can_execute_or_not1.sh  print_ascii.sh
file_can_execute_or_not2.sh  root_or_not.sh
file_exist_or_not1.sh      shift_exam1.sh
file_exist_or_not2.sh      shift_exam1.sh2.sh
getopts_exam1.sh         shift_exam2.sh
getopts_exam2.sh         show_num1.sh
getopts_exam3.sh         test.txt
ip_login.sh             use_shift.sh
[root@localhost chapter16]#
    
```

则下面的命令将输出的内容各是什么？

- (1) echo *
- (2) echo f*
- (3) echo get*
- (4) echo *.*
- (5) echo *.sh

2. 使用文件操作命令创建一个文件，然后使用文件复制、移动和删除等命令对该文件进行操作。

3. 创建一个目录，然后在该目录下再创建一个文件，并对该文件进行删除操作。

4. 新建一个用户组 group1，并新建一个系统组 group2。

(1) 更改组 group2 的 GID 为 103，更改组名为 groupptest。

(2) 删除组 groupptest。

5. 新建用户 user1，指定 UID 为 777，目录为/home/user1，初始组为 group1，有效组为 root，指定 shell 为/bin/bash。

(1) 新建一个系统用户 user2。

(2) 查看用户 user1 的组群，切换到 user1，在主目录下新建文件 test1，切换有效组为 root，再新建文件 test2。

(3) 修改用户 user1 的个人说明为 This is a test（提示加-c 选项）。

(4) 修改用户密码过期时间为 2010-09-01。

(5) 更改用户 user1 的密码为 111111，加锁用户 user1 并查看/etc/shadow，用户 user1 通过 ssh 登录 127.0.0.1。

(6) 更改用户主目录/home/user1 为/home/user11。

(7) 列出用户 user1 的 UID、GID 等。

(8) 增加用户 user3、user4，增加组 testgroup，给组 testgroup 设定密码，将组 testgroup 管理权授予 user1，并同时加入 root、user1、user3 加入到 testgroup，检查结果，切换到 user1，将 user4 加入到 testgroup 组。

(9) 使用 passwd 将 user1 用户密码冻结，用 passwd 查看 user1 相关信息，最后用 passwd 将用户 user1 解冻。

(10) 切换 user1 用户，用 su 加命令行直接查看 shadow 的尾。

联系方式

集团官网: www.hqyj.com

嵌入式学院: www.embedu.org

移动互联网学院: www.3g-edu.org

企业学院: www.farsight.com.cn

物联网学院: www.topsight.cn

研发中心: dev.hqyj.com

集团总部地址：北京市海淀区西三旗悦秀路北京明园大学校内 华清远见教育集团

北京地址：北京市海淀区西三旗悦秀路北京明园大学校区，电话：010-82600386/5

上海地址：上海市徐汇区漕溪路 250 号银海大厦 11 层 B 区，电话：021-54485127

深圳地址：深圳市龙华新区人民北路美丽 AAA 大厦 15 层，电话：0755-25590506

成都地址：成都市武侯区科华北路 99 号科华大厦 6 层，电话：028-85405115

南京地址：南京市白下区汉中路 185 号鸿运大厦 10 层，电话：025-86551900

武汉地址：武汉市工程大学卓刀泉校区科技孵化器大楼 8 层，电话：027-87804688

西安地址：西安市高新区高新一路 12 号创业大厦 D3 楼 5 层，电话：029-68785218

广州地址：广州市天河区中山大道 268 号天河广场 3 层，电话：020-28916067

华清远见