

Linux 常用基本命令

侯在钱

目 录

1. 命令格式	2
2. 文件和目录操作命令	3
2.1. pwd 命令	3
2.2. ls 命令	4
2.3. ll 命令	4
2.4. cd 命令	5
2.5. locate 命令	5
2.6. find 命令	6
2.7. clear 命令	6
2.8. cat 命令	6
2.9. more 命令	7
2.10. less 命令	8
2.11. head 命令	8
2.12. tail 命令	8
2.13. grep 命令	8
2.14. chmod 命令	9
2.15. chown 命令	10
2.16. cp 命令	11
2.17. scp 命令	11
2.18. mv 命令	12
2.19. mkdir 命令	13
2.20. rm 命令	13
3. 文件归档和压缩命令	13
3.1. zip/unzip	13
3.2. gzip/gunzip	14
3.3. tar	14
4. 系统管理命令	14
4.1. free	15
4.2. shutdown	15
4.3. reboot	16
4.4. date	16
4.5. cal	16
4.6. top	17
5. 用户管理命令	18
5.1. groupadd	18

5.2.	groupdel.....	18
5.3.	groups	18
5.4.	useradd	18
5.5.	userdel	19
5.6.	passwd	19
5.7.	usermod.....	19
5.8.	su.....	20
5.9.	who	20
5.10.	whoami	20
6.	网络管理命令	20
6.1.	finger.....	20
6.2.	ftp.....	21
6.3.	hostname	21
6.4.	netstat	21
6.5.	ping	22
6.6.	ssh.....	22
6.7.	telnet.....	22
6.8.	wget	23
6.9.	ifconfig	23
7.	进程管理命令	23
7.1.	ps.....	23
7.2.	kill.....	25
8.	服务管理命令	26

要执行 Linux 命令,首先要打开 Linux 的命令窗口(通常也称之为 Shell 窗口,或终端),在 Shell 窗口中执行命令。例如我们先执行一个最简单的命令 ls,只要在键入“ls”,然后回车,此命令的功能是列出当前目录下的所有子目录和文件。

Linux 中命令很多,例如:ls, pwd, cd, rm, ping 等等。本节就介绍这些命令的作用及使用方法。

1. 命令格式

命令的一般格式: **command** [options] [arguments]

说明:

command: 命令名。

options: 命令的选项,一般是一个单词或字母。有的命令有选项,有的命令没有选项。选项前面一般有“-”符号。选项是对命令参数的补充,当存在参数

时才可能有选项。

arguments: 命令的参数，有时候选项也带参数。有的命令有参数，有的命令没有参数。

[]: 方括号表示可有可无的意思。**[options]**表示有的命令有选项，有的命令也可能没有选项。**[arguments]**表示有的命令有参数，有的命令可能没有参数。

举例:

(1) 没有参数的命令

像 **ls**, **pwd** 都没有选项和参数。直接输入命令，回车即可执行命令。

(2) 有参数没有选项的命令

例如删除文件 **myfile.txt** 的命令，**myfile.txt** 就是参数: **rm myfile.txt**

(3) 有参数也有选项的命令

通过命令 “**rm myfile.txt**” 删除文件，系统会有确认提示，询问是否确定要删除。可以通过一个选项，在执行命令时不再需要确认提示，命令格式如下：

```
rm -f myfile.txt
```

此处的 **-f** 就是选项，作用是进行强制删除，也就是没有确认提示。

2. 文件和目录操作命令

需掌握的常用命令：**cd, ls, locate, less, grep, chmod, cp, mv, mkdir, rm**。Linux 不像 Windows 有 C 盘、D 盘这么一说，它只有一个根目录 “/”。

2.1. pwd 命令

pwd 命令代表 “print working directory”（打印工作目录）。当键入 **pwd** 时，系统显示当前工作目录。例如：

```
[root@localhost]# pwd
```

```
/root
```

表明当前的工作目录是：**/root**。

2.2.ls 命令

ls 命令用于显示当前工作目录下的内容，包括子目录和文件。使用方法：

```
[root@localhost]# ls
```

2.3.ll 命令

ll 命令和 ls 命令功能一样，只是显示方式不一样，ll 命令会列出当前目录下子目录和文件列表的详细信息。使用方法如下：

```
[root@localhost]# ll
```

显示的文件列表格式如下：

```
drwxr-xr-x  3  user  group  102  Mar11  22:56  Filename
```

文件列表显示了 7 个段，每一段分别的含义说明如下：

(1) 第 1 段，drwxr-xr-x，表示文件属性，第 1 个字母 d 表示这个是个目录，如果是“-”表示文件。

文件类型	说明
d	目录
-（短线）	常规文件（而不是目录或链接）
l	到系统上其它位置的另一个程序或文件的符号链接。类似于 Windows 的快捷方式。

d 后面有 9 个字符，分成三组，每 3 个字符为一组。我们使用括号把他括起来：**d(rwx)(r-x)(r-x)**，对应含意：文件类型（所有者权限）（组权限）（其他人权限）。这三组分别说明的是：文件的所有者权限，文件所属的组权限，和“其他人”的权限。每一组中每一位字符的含意如下：

第 1 位表示是否有读权限，有读权限，显示“r”，没有读权限，显示“-”。

第 2 位表示是否有写权限，有写权限，显示“w”，没有写权限，显示“-”。

第 3 位表示是否有执行权限，有执行权限，显示“x”，没有，显示“-”。

d(rwx)(r-x)(r-x)含意说明：第 1 位 d：表示目录。第 1 组(rwx)：表示文件所有者对这个文件的权限是，有读，写，执行权限。第 2 组(r-x)：表示文件所属用户组对这个文件的权限是：有读，没有写权限，有执行权限。第 3 组(r-x)：表示其他人员对这个文件的权限是：有读，没有写权限，有执行权限。

(2) 第 2 段, 目录下的文件和子目录个数: 3 表示目录下有 3 个文件或目录 (注意: 每个目录都有一个指向它本身的子目录"." 和指向它上级目录的子目录"..")

(3) 第 3 段, 显示文件的所有者: user。

(4) 第 4 段, 显示文件所属用户组: group。

(5) 第 5 段, 文件大小: 102 byte。

(6) 第 6 段, 修改时间: Mar11 22:56。

(7) 第 7 段, 文件名: Filename。

2.4.cd 命令

cd 命令来改变工作目录。

例如把当前目录更改到/usr 目录下:

```
[root@localhost]# cd /usr
```

例如把当前目录更改到整个系统的根目录下:

```
[root@localhost]# cd /
```

几个常用 cd 用法:

命令	功能
cd tmp	进入当前目录下的 tmp 目录。
cd ~	回到当前用户的 HOME 目录下。
cd /	回到整个系统的根目录。
cd /etc	进入到/etc 目录
cd ..	回到上一级目录。

2.5.locate 命令

使用 locate 命令来搜寻文件或目录。locate 命令直接在文件索引数据库里查找, 所以搜索文件的速度很快。如果需要更新文件索引数据库, 则使用 updatedb 命令。

例如, 如果想搜寻所有名称中带有 finger 这个词的文件或目录, 命令如下:

```
[root@localhost]# locate finger
```

locate 有一个十分有用的选项 `-r`，它可以让你在搜索文件时使用正则表达式。locate 是 Linux 系统中的一个查找文件命令，若在查找文件时提示：`locate: can not open '/var/lib/mlocate/mlocate.db': No such file or directory`，原因是 locate 是通过生成一个文件和文件夹的索引数据库，当用户在执行 locate 命令查找文件时，它会直接在索引数据库里查找，若该数据库太久没更新或不存在，则会提示以上错误。

解决方法是执行如下的命令，更新文件索引数据库。

```
[root@localhost]# updatedb
```

2.6.find 命令

这是另一个 Linux 系统中重要的文件查找命令。其一般命令格式为：

`find 位置 -name 文件名称`。

例如，在 “/” 这个根目录中查找 “linux.html” 文件，命令如下：

```
[root@localhost]# find / -name linux.html。
```

你除了可以按文件名称来使用 find 查找文件外，也可以根据文件大小（通过 `-size n` 选项指定）、时间（如 `-atime n` 表示查找 n 天前访问过的文件）来搜索文件。

此外，find 命令同样支持在搜索文件时使用正则表达式，你只需指定 `-regex` 选项即可。

find 和 locate 命令的区别：locate 是在建立好的索引的基础上进行查找，查找要快很多。find 功能强大，但它是直接在硬盘上搜寻，查找速度比较慢。

2.7.clear 命令

clear 命令用于清除终端窗口。

2.8.cat 命令

cat 是 concatenate（连锁）的简写，意思是合并文件。该命令可以显示文件的内容（经常和 more 搭配使用），或者是将多个文件合并成一个文件。

cat 主要有三大功能：

(1) 一次显示整个文件。\$ cat filename

(2) 从键盘创建一个文件。\$ cat > filename << 结束符

只能用于创建新文件，不能编辑已有文件。

例如设置结束符为 EOF: cat > filename.txt << EOF

编辑文件完后，只需在最后一行键入“EOF”，然后回车即可。

(3) 将几个文件合并为一个文件: \$cat file1 file2 > file

参数:

-n 或 --number 由 1 开始对所有输出的行数编号

-b 或 --number-nonblank 和 -n 相似，只不过对于空白行不编号

-s 或 --squeeze-blank 当遇到有连续两行以上的空白行，就代换为一行的空白行

-v 或 --show-nonprinting

例:

(1) 把 textfile1 的档案内容加上行号后输入 textfile2 这个档案里

cat -n textfile1 > textfile2

(2) 把 textfile1 和 textfile2 的档案内容加上行号（空白行不加）之后将内容附加到 textfile3 里。

cat -b textfile1 textfile2 >> textfile3

和 more 一起使用，使用管道连起来。例如: cat /etc/passwd | more。

不过一般不这么用，使用麻烦。直接使用 more 命令即可，例如: more /etc/passwd。

2.9.more 命令

功能: 浏览超过一页的文件。

Ø 在显示满一页时暂停，此时可按空格键继续显示下一个画面。

Ø 按 Q 键退出显示。

Ø Enter 键滚动显示下一行。

例如:

[root@localhost]# more <filename>

2.10. less 命令

less 命令的用法与 **more** 命令类似，也可以用来浏览超过一页的文件。所不同的是 **less** 命令除了可以按空格键向下显示文件外，还可以利用上下键来滚动文件。当需要结束浏览时，只要在 **less** 命令的提示符 “:” 下按 **Q** 键即可。

例如：

```
[root@localhost]# less <filename>
```

2.11. head 命令

你可以使用 **head** 命令来查看文件的开头部分。此项命令是：

```
head <filename>
```

head 是一个有用的命令，但是由于它只限于文件的最初几行，你看不到文件实际上有多长。按照默认设置，你只能阅读文件的前十行。你可以通过指定一个数字选项来改变显示的行数，如下面的命令所示：

```
[root@localhost]# head -20 <filename>
```

2.12. tail 命令

与 **head** 命令恰恰相反的是 **tail** 命令。使用 **tail** 命令，你可以查看文件结尾的十行。这有助于查看日志文件的最后十行来阅读重要的系统消息。你还可以使用 **tail** 来观察日志文件被更新的过程。使用 **-f** 选项，**tail** 会自动实时地把打开文件中的新消息显示到屏幕上。例如，要即时观察 **/var/log/messages** 的变化，以根用户身份在 **shell** 提示下键入以下命令：

```
[root@localhost]# tail -f /var/log/messages。
```

2.13. grep 命令

grep 命令用于在文件中查找指定的字符串。例如，在 **temp.txt** 文件中查找每一个 “hello”，使用的命令如下：

```
[root@localhost]# grep hello temp.txt
```

如果要显示行号，则使用如下命令：


```
[root@localhost]# grep -n 要查找的内容 文件名
```

2.14. chmod 命令

chmod 命令用于改变文件或目录的访问权限。

更改权限：

给文件设置新的权限。

例如，给 test.sh 文件设置权限

```
[root@localhost]#chmod 644 test.sh
```

644：由 3 为数字组成，第 1 位数字代表文件所有者对这个文件的权限，第 2 位数字代表文件所属用户组对这个文件的权限，第 3 位数字代表其他人对这个文件的权限。权限属性对应的数字如下：

权限属性	对应数字	说明
r	4	文件可以被读取
w	2	文件可以被写入
x	1	文件可以被执行（如果它是程序的话）
-	0	相应的权限还没有被授予。

例如，仅有读权限，就是 4。仅有写权限就是 2。如果既有读，也有写权限，则把权限值加起来，4+2 就是 6。

644 权限说明：文件所有者对这个文件的权限是：读、写；文件所属用户组对这个文件的权限：仅有读；其他人对这个文件的权限：仅有读。

可通过 ll 命令查看权限更改后的情况。

加减权限：

在原有权限的基础之上，增加或减少权限，使用命令如下：

chmod o +w test.sh #o+w 命令告诉系统给其他人对文件 test.sh 增加写权限。

chmod go -rw test.sh #组和其他人减去读取和写入权限。

下面是一个速记符号含义的列表：

身份

u ：拥有文件的用户（User）

g : 所有者所在的组群 (Group)

o : 其他人 (不是所有者或所属用户组的其他用户, Other)

a : 全部 (All, 包括 u、g、和 o)

权限

r : 读取权

w : 写入权

x : 执行权

行动

+ — 添加权限

- — 删除权限

= — 使它成为唯一权限

常用的权限设置:

这里是一个某些常用设置、数值、以及它们的含义的列表:

-rw-----(600)—只有所有者才有读取和写入的权限。

-rw-r---(644)—只有所有者才有读取和写入的权限: 组群和其他人只有读取的权限。

-rwx----- (700)—只有所有者才有读取、写入和执行的权限。

-rwxr-xr-x(755)—所有者有读取、写入和执行的权限; 组群和其他人只有读取和执行的权限。

-rwx--x--x(711)—所有者有读取、写入和执行权限; 组群和其他人只有执行权限。

-rw-rw-rw-(666)—每个人都能够读取和写入文件。(请谨慎使用这些权限。)

-rwxrwxrwx(777)—每个人都能够读取、写入和执行。(再重申一次, 这种权限设置可能会很危险。)

下面列举了一些对目录的常见设置:

drwx----(700)—只有所有者能在目录中读取、写入。

drwx-xr-x(755)—每个人都能够读取目录, 但是其中之一的内容却只能被告所有者改变。

2.15. chown 命令

chown 用于改变文件的所有者。

命令格式:

chown [-R] user[:group] file...

user: 新的档案拥有者的使用者 ID

group: 新的档案拥有者的使用者群体(group)

-R: 对目前目录下的所有档案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)

2.16. cp 命令

cp (**copy**) 命令可以将文件或目录复制到其他目录中。在使用 **cp** 命令时, 只需要指定源文件名与目标文件名或目标目录即可。

命令格式:

cp <源> <目标目录>

例如, 把当前目录下的 **test.txt** 文件拷贝到 **/tmp** 目录下:

cp test.txt /tmp

拷贝目录 (命令中需要使用 **-r** 选项):

cp -r <源目录名> <目标目录>

例如, 把当前目录下的 **test** 目录拷贝到 **/tmp** 目录下:

cp -r test /tmp

2.17. scp 命令

scp 是 **linux** 中功能最强大的文件传输命令, 可以实现从本地到远程以及远程到本地的文件传输操作。

Ø 本地到远程的文件复制:

文件拷贝命令格式:

scp local_file remote_username@remote_ip:remote_folder

或者

scp local_file remote_username@remote_ip:remote_file

local_file: 本地文件。

remote_username: 远程计算机的用户名。

remote_ip: 远程计算机的 IP 地址。

remote_folder: 远程计算机的目录。

使用例子:

```
scp 1.mp3 root@192.168.1.2:/tmp/
```

目录拷贝命令格式:

```
scp -r local_folder remote_username@remote_ip:remote_folder
```

例如将本地/home/music/目录复制到远程/tmp/目录下:

```
scp -r /home/music/ root@192.168.1.2:/tmp/
```

复制后远程有/tmp/music/目录。

Ø 远程到本地的复制

从远程服务器复制文件到本地, 只要将 2 个参数调换顺序即可:

拷贝文件, 例如把 192.168.1.2 上的/home/music/1.mp3 拷贝到本地/tmp/目录下:

```
scp root@192.168.1.2:/home/music/1.mp3 /tmp/
```

拷贝目录, 例如把 192.168.1.2 上的/home/music/目录拷贝到本地/tmp/目录下:

```
scp -r root@192.168.1.2:/home/music/ /tmp/
```

2.18. mv 命令

mv 命令用于移动文件。**mv** 的常见选项包括:

-i—互动。如果你选择的文件会覆盖目标中的现存文件, 它会提示你。这是一个实用的选项, 因为它像 **cp** 中的 **-i** 选项一样, 会给你一个确认替换已存文件的机会。

-f—强制。它会超越互动模式, 不提示地移动文件。除非你知道自己在干什么, 这个选项很危险。在你对系统信心十足之前, 请谨慎使用这个选项。

-v—详细。显示文件的移动进度。

命令格式:

```
mv <源文件> <目标目录>
```

如果你想把文件从你的主目录中移到另一个现存的目录中, 使用以下命令:

```
[root@localhost]# mv sneakers.txt /home/newuser/
```

移动后同时给文件改名:

```
[root@localhost]# mv sneakers.txt /home/newuser/new_sneakers.txt
```

2.19. mkdir 命令

mkdir(make directory)命令用来建立目录。例如在系统中当前目录下建立 data 子目录：

```
[root@localhost]# mkdir data
```

2.20. rm 命令

rm 命令用于将文件或目录删除。如果是链接文件，只是删除了该链接，原有文件保持不变。删除文件和目录的选项包括：

选项	说明
-i	互动。提示你确认删除。这个选项可以帮助你避免误删文件。
-f	Force, 强制。代替互动模式，不提示地删除文件。除非你知道自己在干什么，使用这个选项通常不是明智之举。
-v	Verbose, 显示操作过程的详细信息。显示文件的删除进度。
-r	Recursive, 递归。将会删除某个目录及其中所有的文件和子目录。

例如要使用 rm 命令来删除文件 tmp.txt，命令如下：

```
rm tmp.txt
```

如果要删除目录，命令如下：

```
rm -r 目录
```

3. 文件归档和压缩命令

需掌握的常用命令：tar, zip, unzip, gzip, gunzip。

3.1. zip/unzip

zip 命令用于打包和压缩文件。

压缩例子：zip myfile.zip 目录/*

unzip 命令用于解压缩扩展名为.zip 的文件。可以在 Linux 下解压 Windows 中用 WinZip 压缩的文件。

解压例子：**unzip myfile.zip**

3.2. gzip/gunzip

gzip 命令用于压缩".gz"文件，**gunzip** 用于解开被 **gzip** 压缩过的文件，这些压缩文件预设最后的扩展名为".gz"。

压缩：**gzip usr.tar**

解压：**gunzip usr.tar.gz**

3.3. tar

tar 命令用于文件打包，不是压缩。

打包例子：**tar -cvf myfile.tar** 目录

解包例子：**tar -xvf myfile.tar**

如果一个压缩包是.tar.gz 格式，则可以先通过 **gunzip** 解压，然后再通过 **tar** 解包。也可以通过 **tar** 命令直接解压解包，只需在解包加个参数 **z**：

tar -xzvf myfile.tar.gz

tar 命令常用选项：

选项	说明
-c	建立新的归档文件
-x	从归档文件中解出文件
-v	处理过程中输出相关信息
-f	指定的文件名
-z	与-x 联用时调用 gzip 完成对 gz 文件的解压

4. 系统管理命令

需掌握的常用命令：**free, top, shutdown**。

4.1.free

free 命令用于查看当前系统内存的使用情况，它可以显示系统中剩余和已用的物理内存、交换内存和内核缓冲区。

例如：

```
[root@localhost]# free
```

交换区：我们如果没有足够的内存，也许就不能运行某些大型的软件，解决的办法是在硬盘上划出一个区域来当作临时的内存，好像内存变大了。Windows 操作系统把这个区域叫做虚拟内存，Linux 把它叫做交换分区 **swap**。

4.2.shutdown

该命令用于关机。语法规则如下：

shutdown [-cfFhknr(参数名称)] [-t 秒数] 时间 [警告信息]

具体各参数功能：

-c 取消前一个 **shutdown** 命令。值得注意的是，当执行一个如 “**shutdown -h 11: 10**” 的命令时，只要按 “**Ctrl+C**” 键就可以中断关机的命令。若是执行如 “**shutdown -h 11: 10 & ”** 的命令将 **shutdown** 转到后台时，则需要使用 **shutdown -c** 将前一个 **shutdown** 命令取消。

-f 重新启动时不执行 **fsck**(注：**fsck** 是 Linux 下的一个检查和修复文件系统的程序)。

-F 重新启动时执行 **fsck**。

-h 将系统关机，在某种程度上功能与 **halt** 命令相当。

-k 只是送出信息给所有用户，但并不会真正关机。

-n 不调用 **init** 程序关机，而是由 **shutdown** 自己进行(一般关机程序是由 **shutdown** 调用 **init** 来实现关机动作)，使用此参数将加快关机速度，但是不建议用户使用此种关机方式。

-r **shutdown** 之后重新启动系统。

-t<秒数> 送出警告信息和关机信号之间要延迟多少秒。警告信息将提醒用户保存当前进行的工作。

[时间] 设置多久时间后执行 **shutdown** 命令。时间参数有 **hh: mm** 或 **+m** 两种模式。**hh: mm** 格式表示在几点几分执行 **shutdown** 命令。例如 “**shutdown 10:**

45”表示将在 10:45 执行 shutdown.+m 表示 m 分钟后执行 shutdown.比较特别的用法是以 now 表示立即执行 shutdown. 值得注意的是这部分参数不能省略。

[警告信息] 要传送给所有登入用户的信息。

应用举例：

立即关机：

```
# shutdown -h now
```

定时关机(11 点 10 分关机，如果要取消按 Ctrl+C 键)：

```
# shutdown -h 11:10, 或者 # shutdown 11:10
```

指定 5 分钟后关机，同时送出警告信息给登入用户：

```
# shutdown +5 "System will shutdown after 5 minutes"
```

重启计算机：

```
# shutdown -r
```

4.3.reboot

重启计算机。用法：

```
# reboot
```

4.4.date

该命令可以显示当前系统的日期和时间，也可以设置日期和时间。clock 命令也可以用于显示系统的日期与时间（默认情况下，普通用户无法执行 clock 命令，必须用 root 账号登录执行）。

例如，显示日期和时间：

```
# date
```

设置日期和时间：

```
# date -s "2016-07-22 11:28:30"
```

4.5.cal

该命令可显示计算机中的日历。

显示当前月日历：

```
# cal
```


显示指定年份的日历命令：

```
# cal 2016
```

显示指定月份的日历命令：

```
# cal 9 2016
```

4.6. top

top 命令是 Linux 下常用的性能分析工具，能够实时显示系统中各个进程的资源占用状况，类似于 Windows 的任务管理器。

例如：

```
[root@localhost]# top
```

5. 用户管理命令

需掌握的常用命令：useradd, passwd, su。

5.1.groupadd

groupadd 命令用于创建新的用户组。

语法格式：groupdel 组名

例如：

```
[root@localhost]# groupadd mygroup
```

5.2.groupdel

groupdel 命令用于删除用户组；

语法格式：groupdel 组名

例如：

```
[root@localhost]# groupdel mygroup
```

5.3.groups

groups 命令用于显示当前用户所在的组。

例如：

```
[root@localhost]# groups
```

5.4.useradd

useradd 命令用于建立用户帐号。

主要参数

- c: 加上备注文字，备注文字保存在 passwd 的备注栏中。
- d: 指定用户登入时的启始目录。
- D: 变更预设值。
- e: 指定账号的有效期限，缺省表示永久有效。
- f: 指定在密码过期后多少天即关闭该账号。
- g: 指定用户所属的群组。

- G: 指定用户所属的附加群组。
- m: 自动建立用户的登入目录。
- M: 不要自动建立用户的登入目录。
- n: 取消建立以用户名称为名的群组。
- r: 建立系统账号。
- s: 指定用户登入后所使用的 shell。
- u: 指定用户 ID 号。

添加用户: `useradd newuser`

添加用户, 指定相关参数, 建立 test 用户, 并把此用户加入 root 组:

```
useradd -m -d /home/share -g root test
```

5.5. userdel

`userdel` 命令用来删除用户帐号及其相关文件。

例如:

```
[root@localhost]# userdel newuser
```

5.6. passwd

`passwd(password)` 命令用于修改用户的密码。一般来说, 设置账户密码失败有几种情况: 密码太简单、密码太短、密码中的字符多数相同。

语法格式: `passwd 用户名`

修改密码, 命令如下:

```
[root@localhost]# passwd newuser
```

5.7. usermod

`usermod` 可用来修改用户帐号的各项设定。可以用来设定用户归属哪个组。

例如使得 test 用户加入 mail 组:

```
[root@localhost]# usermod -G mail test
```

5.8. su

su (switch user) 命令可以使得当前用户切换到另一个用户。如果要退出切换后的用户登录，可以执行 exit 命令。

切换到 user1 用户：

```
[root@localhost]# su user1
```

切换到 root 用户：

```
[root@localhost]# su
```

5.9. who

该命令可以查看当前计算机有哪些用户登录。

5.10. whoami

该命令是用来查看当前用户的登录名。

6. 网络管理命令

需掌握的常用命令：ping, netstat, ftp, ssh, telnet, wget。

6.1. finger

finger 用来查询一台主机上的登录账号的信息，通常会显示用户名、主目录、停滞时间、登录时间、登录 Shell 等信息，使用权限为所有用户。

命令格式：

```
finger [选项] [使用者] [用户@主机]
```

主要参数说明：

-s: 显示用户注册名、实际姓名、终端名称、写状态、停滞时间、登录时间等信息。

-l: 除了用-s 选项显示的信息外，还显示用户主目录、登录 Shell、邮件状态等信息，以及用户主目录下的.plan、.project 和.forward 文件的内容。

-p: 除了不显示.plan 文件和.project 文件以外，与-l 选项相同。

应用实例：

直接键入如下命令：

finger

查询远程机上的用户信息应用说明：

如果要查询远程机上的用户信息，需要在用户名后面接“@主机名”，采用[用户名@主机名]的格式，不过要查询的网络主机需要运行 **finger** 守护进程的支持。

6.2.ftp

该命令是标准的文件传输协议的用户接口，是在 **TCP/IP** 网络上传输文件最简单有效的方法。

6.3.hostname

该命令用于显示或设置系统的主机名。

6.4.netstat

netstat 命令用于显示网络连接、路由表和网络接口信息，用户可以知道目前有哪些网络连接正在运行。

常用选项有：

选项	说明
-a	显示所有 socket ，包括正在监听的。
-c	每隔 1 秒钟就重新显示一遍网络信息，直到用户中断它。
-l	仅列出有在 Listen (监听) 的服务状态，格式同“ ifconfig -e ”命令。
-n	以 IP 地址代替名称，显示网络连接信息。
-r	显示核心路由表，格式同“ route -e ”命令。
-t	显示 TCP 协议的连接信息。
-u	显示 UDP 协议的连接信息。
-v	显示正在进行的网络协议。

应用示例：

列出所有端口: `netstat -a`

通过管道方式查看每一页内容: `netstat -a|more`

列出所有 `tcp` 端口: `netstat -at`

列出所有 `udp` 端口: `netstat -au`

只显示监听端口: `netstat -l`

只列出所有监听 `tcp` 端口: `netstat -lt`

只列出所有监听 `udp` 端口: `netstat -lu`

查看 80 是否被占用: `netstat -l|grep 80`

6.5. ping

`ping` 命令可用来测试计算机和网络上的其他计算机是否连通。

例如测试本机和 192.168.1.2 那台机器是否连通:

```
[root@localhost]# ping 192.168.1.2
```

6.6. ssh

`ssh` 命令(安全的 Shell)用于通过网络登录远程计算机,如同操作本地计算机一样。

命令格式:

`ssh 用户@服务器 IP 地址`

例如使用 `root` 用户登录到 192.168.1.2 上:

```
ssh root@192.168.1.2
```

然后根据系统提示,输入登录的用户名和密码。

还有一个 `rsh` 命令和 `ssh` 命令功能类似,常用的是 `ssh`。`ssh` 和 `rsh` 的区别:安全级别不同,`ssh` 的密码等都是加密传输,而且还有密钥认证的机制,而 `rsh` 是明文传输,而且没有密钥的机制。

6.7. telnet

`telnet` 命令用于通过网络登录远程计算机,如同操作本地计算机一样。和 `ssh` 命令功能类似。`ssh` 和 `telnet` 区别: `ssh` 是加密的。`telnet` 是明码传输的,发送的数

据被监听后不需要解密就能看到内容。现在不建议使用 `telnet` 。

6.8. wget

`wget` 命令用于 Linux 环境下从 WWW 上下载文件，支持 HTTP 和 FTP 协议，支持代理服务器和断点续传功能，能够自动递归远程主机的目录，查找合乎要求的文件并下载到本地硬盘上，`wget` 命令可在后台运行，截获并忽略 `HANGUP` 信号，因此在用户退出登录之后，仍可继续运行。

例如：`wget http://url/filename`

6.9. ifconfig

可以查看网卡的 IP 地址。例如：

```
[root@localhost]# ifconfig
```

也可以临时修改 IP，重启系统后，恢复原始 IP。

命令格式：

`ifconfig IP 地址 netmask 子网掩码`

7. 进程管理命令

常用命令：`ps`, `kill`。

7.1. ps

`ps` 命令用于显示程序的状态。

例如查找所有进程：

`ps -aux`

命令常用选项：

选项	说明
-a	显示所有用户的所有进程（包括其它用户）。
-x	显示没有控制终端的进程，同时显示各个命令的具体路径。
-u	按用户名和启动时间的顺序来显示进程。

`-aux` 显示所有包含其他使用者的进程。

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
进 程 拥 有 者	进 程 ID	占用的 CPU	占用的 内存	占 用 的 虚 拟 记 忆 体 大 小	占 用 的 记 忆 体 大 小	终 端 的 次 要 装 置 号 码	该 行 程 的 状态	进程开 始时间	执 行 的 时 间	所执行的指 令

STAT，状态，分别表示的含意：

D: 不可中断的静止

R: 正在执行中

S: 静止状态

T: 暂停执行

Z: 不存在但暂时无法消除

W: 没有足够的记忆体分页可分配

<: 高优先序的行程

N: 低优先序的行程

L: 有记忆体分页分配并锁在记忆体内（即时系统或握 A I/O）

应用举例：

使用管道和 more 连用，进行分页查看：

```
[root@localhost ~]# ps -aux | more
```

使用管道和 grep 连用，在进程中查找指定进程。例如查找 Tomcat 进程：

```
ps -aux|grep tomcat
```

注意事项：如果有两条查询结果，则说明 Tomcat 已经启动，如果没有，则 Tomcat 没有启动。如果只查询出一条结果，并且以“grep tomcat”结尾，则此进程不是

Tomcat 进程，而是查找进程。

把查询到的进程导出到文件：

```
[root@localhost ~]# ps -aux > tmp.txt
```

7.2. kill

该命令用于终止一个程序。

一般用法格式：

kill PID

PID 是指进程 ID。例如杀掉进程号为 2629 的进程：

kill 2629

向进程发送信号的方式，杀进程命令格式：

kill -s signal PID

signal 是指关闭 PID 所指定的进程时发送的信号，PID 是指进程 ID。

例如通过上面的进程查到 Tomcat 的进程 ID 是 2629，要杀死此进程的命令：

```
[root@localhost ~]# kill -s SIGINT 2629
```

或者写成：

```
[root@localhost ~]# kill -INT 2629
```

或者写成：

```
[root@localhost ~]# kill -2 2629
```

使用信号数字杀掉进程：

signal 也可以使用对应的数字代替，例如上面的命令可以写成：

kill -2 2629

常用的几个信号（**signal**，可通过 **kill -l** 命令信号有哪些，及信号与数字对应关系）：

SIG 信号	对应数字	说明
SIGINT	2	这个就是在 shell 下用 Ctrl+C 来结束一个程序时，发送的信号，进程收到这个程序会结束。 你可以用 kill -INT pid 来发这个信号。
SIGQUIT	3	这个是在 shell 下用 Ctrl+\ 来结束程序时，发送的信号，

		进程收到这个信号后也是结束。退出时会产生 CORE 文件，类似于一个程序生产一个错误日志。
SIGKILL	9	这个信号之所以被称为“强杀”，就是因为无法改变进程收到这个信号后所执行的动作，进程只能退出。（前面说的两个信号，虽然默认是退出，但是应用程序自己可以通过 signal 系统调用来修改成其他动作，比如忽略那两个信号等动作）。
SIGTERM	15	程序结束(terminate)信号，与 SIGKILL 不同的是该信号可以被阻塞和处理。通常用来要求程序自己正常退出。

一般情况下使用最多的是强杀：

kill -9 2629

8. 服务管理命令

Service 命令用于启动、停止、重启服务程序。

启动服务命令：**service 服务名 start**

停止服务命令：**service 服务名 stop**

重启服务命令：**service 服务名 restart**

例如要启动 **VSFTP** 服务器：

```
[root@localhost ~]# service vsftpd start
```

例如要停止 **VSFTP** 服务器：

```
[root@localhost ~]# service vsftpd stop
```