第3章
常用命令与文件管理

第3章　常用命令与文件管理

3.1 Linux基本操作

3.2 Linux命令

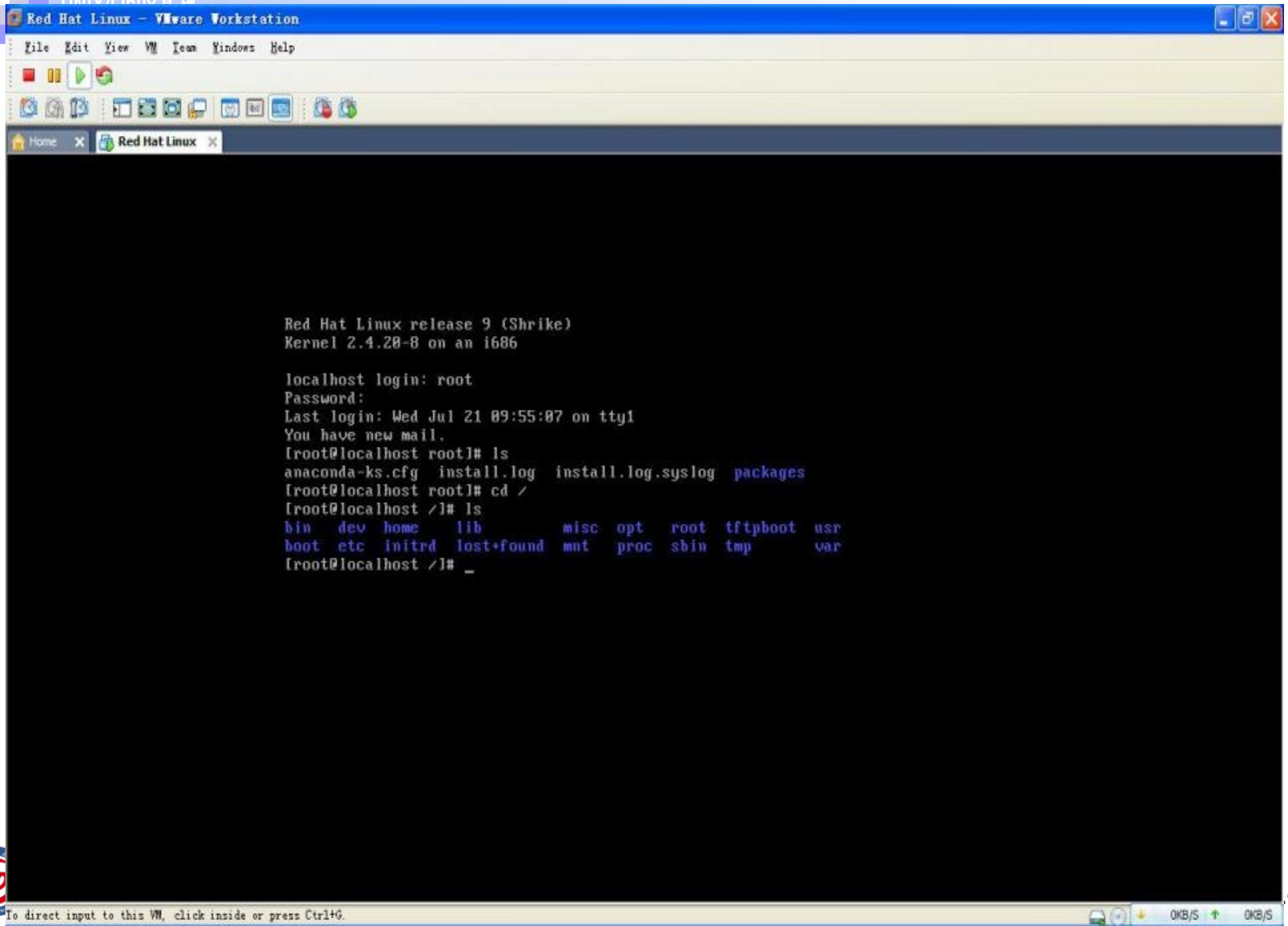
3.3 Linux文件操作

3.4 输入/输出重定向

关于字符界面

对Linux服务器进行管理时，经常需要进入字符界面进行操作，使用命令需要记住该命令的相关选项和参数。

Linux系统的命令行界面可以通过字符界面、图形界面下的终端窗口以及虚拟控制台等多种方式进入。


Red Hat Linux - VMware Workstation
File Edit View VM Team Windows Help
Home
Red Hat Linux
Red Hat Linux release 9 (Shrike)
Kernel 2.4.20-8 on an i686
localhost login: root
Password:
Last login: Wed Jul 21 09:55:07 on tty1
You have new mail.
[root@localhost root]# ls
anaconda-ks.cfg install.log install.log.syslog packages
[root@localhost root]# cd /
[root@localhost /]# ls
bin dev home lib misc opt root tftpboot usr
boot etc initrd lost+found mnt proc sbin tmp var
[root@localhost /]# _
To direct input to this VM, click inside or press Ctrl+G.
0KB/S
0KB/S

3.1 Linux基本操作

在使用Linux系统前，首先需要了解和掌握一些基本的操作，包括如何登录和退出系统、如何修改口令以及关闭和重启系统。

3.1.1 登录

Linux系统是一个多用户操作系统，系统的每个合法用户都拥有一个用户账号，包括用户名和口令等信息。

任何用户在使用Linux系统前必须先登录系统。登录(login)过程就是系统对用户进行认证和授权的过程。登录时，用户须提供用户名和口令。

每个Linux系统都有一个特殊的用户，称为超级用户。超级用户的用户名是root。root具有对系统的完全控制权限，非必要时应避免使用root登录。

1. 终端

- 终端(terminal)是指用户用来与系统交互的设备，包括显示器、键盘和鼠标。每个用户都需要通过一个终端来使用系统。
- 根据显示模式的不同，终端分为字符终端和图形终端。
- 字符终端只能显示字符界面，接收键盘输入的命令；
- 图形终端可以支持图形界面显示和鼠标操作。

根据连接方式的不同，终端又可分为本地终端和远程终端。

本地终端是直接与系统相连的终端，也称为控制台(console)，是供系统本地用户使用的终端；

远程终端指用户通过网络或其他通信方式远程地使用系统时所用的终端，可能是专门的终端机，更多的是PC机终端。

虚拟控制台

Linux系统提供了虚拟控制台的访问方式，允许多个用户同时登陆，还允许用户进行多次登陆。

在字符界面下，可以用Alt+ F1至Alt +F6来访问；

在图形界面下，可以使用 Ctrl +Alt +F1至F6来切换到所需的虚拟控制台。

虚拟控制台

当用户从一个虚拟控制台切换到一个新的虚拟控制台以后，在原来 那个控制台运行的程序将继续运行。

虚拟控制台最大的好处是当一个进程出错锁住输入时可以切换到其他虚拟控制台来终止这个进程。

2. 登录方式

Linux系统的登录方式可分为本地登录和远程登录。

（1）本地登录

（2）远程登录

远程用户可以从远程终端登录到Linux系统上。远程登录的用户可以在自己所在的终端上像本地用户一样与系统交互，发布命令、运行程序并得到显示结果。

从PC机上远程登录Linux系统的方法是：使用Windows的telnet或其他虚拟终端软件（putty,VNC），通过网络、拨号或串口方式与Linux系统建立通信连接，连通后即可看到Linux系统的登录提示符“login”。

3.1.2 修改口令

用户在初次使用系统时，一般是用超级用户为其设置的初始口令登录。登录后应及时修改口令。此后，为安全起见，用户还应定期修改登录口令。

口令应具有一定的长度和复杂度，使其不易被破解。口令还应便于记忆，若忘记口令只能找超级用户重新设置。在桌面环境下，可以在系统菜单中找到修改口令的界面，在字符界面修改口令应使用passwd命令。

3.1.3 退出

- 退出(logout)就是终止用户与系统的当前交互过程。操作完成后及时应退出系统，即使是暂时离开也应如此。
- 在桌面上可以找到退出系统的按钮或菜单项。
- 在字符控制台界面，用exit命令或在命令提示符后按Ctrl+d键即可退出系统。(注意：可能需要多次使用exit命令或Ctrl+d键直至退出系统。)
- 退出后，系统回到登录界面，用户可以重新登录系统。

3.1.4 系统的关闭与重启

- 在Linux系统中，强烈不建议直接关机，一方面因为多用户操作系统，关闭和重启系统会影响到所有已登录的用户；另一方面Linux系统与Windows系统不同，其后台运行着许多进程，所以强制关机可能会导致进程的数据丢失，使系统处于不稳定的状态，甚至会损坏某些系统的硬件设备。
- 因而执行此操作需要有root权限。

3.1.4 系统的关闭与重启

- 关机前注意：

  查看系统的使用状态；

  通知在线用户关机时刻；

  正确使用关机命令。

- 在关机或重启前，建议执行数据同步写入磁盘命令sync，将内存中尚未被更新的数据写入磁盘。

3.1.4 系统的关闭与重启

- 为方便个人应用，Linux系统默认设置为允许登录图形桌面的普通用户关闭和重启系统。
- 在桌面环境下关机或重启很简单，只要点击相应的按钮即可。
- 按Ctrl+Alt+Del键也可以重启系统。

常用的关机命令“shutdown”

可以自由选择关机模式；

可以设置关机时间；

可以自定义关机消息；

可以仅发出警告消息，实际上不是真的关机；

可以选择是否进行磁盘检查；

……

命令语法：

shutdown［选项］［时间］［警告信息］

命令中各个选项的含义：

-k 只是送出信息给所有用户，但并不会真正关机。
-r 关闭系统后立即重新启动。
-h 关闭系统后不重新启动。
-f 快速关闭系统，重新启动时不进行磁盘检测。
-n 快速关闭系统，不调用init程序关机。
-c 中断关闭系统。
……

- 应用举例：
- 指定现在立即关机：

```
# shutdown -h now
```

- 指定固定时间关机：

```
# shutdown -h 20:25
```

- 30分钟后系统会重新启动，并发出警告信息：

```
# shutdown -r +30 “system will be reboot now.”
```

- 发出警告，系统不会关机

```
# shutdown –k now “system will be reboot now.”
```

- **halt 命令**
- **使用“halt”命令就是调用“shutdown –h”命令执行关闭系统。**
- **命令语法：　halt [选项]**

- **在关机命令的执行过程中，系统屏幕上会显示出关机操作的输出信息。要等到关机过程完成后方可切断电源。**

- -w：并不会真的关机，只是把记录写到 /var/log/wtmp 文件里
- -d：关机但不把记录写到 /var/log/wtmp 文件里
- -f：强迫关机，不调用 shutdown 这个指令
- -i：在关机之前先把所有网络相关的装置先停止
- -p：当关机的时候，顺便做关闭电源（poweroff）的动作

范例：

- halt -p 关闭系统后关闭电源。
- halt -d 关闭系统，但不留下纪录。

reboot命令

命令语法： **reboot [选项]**

“reboot”命令的工作过程与“halt”命令相似，不过“reboot”是引发主机重启，而“halt”是引发主机关闭系统。

- -w：并不会真的重开机，只是把记录写到 /var/log/wtmp 文件里
- -d：重启但不把记录写到 /var/log/wtmp 文件里
- -f：强迫重开机，不调用 shutdown 这个指令
- -i：在重开机之前先把所有网络相关的装置先停止

init命令

“init 0”为关闭系统。

init命令是Linux下的进程初始化工具，init进程是所有Linux进程的父进程，它的进程号为1。发送“TERM”信号给“init”会终止所有的用户进程和守护进程等。“shutdown”命令就是使用这种机制。

init命令是Linux操作系统中不可缺少的程序之一。

3.1.5 系统的运行级别

- Linux系统运行级别是一种状态
- Linux运行级别有如下7种。

0：停止运行，所有进程中止，关闭系统。

1：单用户模式，用于维护系统，只有少数进程运行。

2：多用户模式，除了NFS服务没有启动外，其他和运行级别3一样。

3：完整的多用户模式，进入Linux系统的字符界面。
4：没有使用（可由用户定义）。
5：完整的多用户模式（带有基于X Window的图形界面）。
6：重新引导计算机。

- 使用“runlevel”命令可查看系统的当前运行级别。
- 使用“init”命令可以切换运行级别

3.2 shell基础

- Linux系统为用户提供了一套完备的命令，使用这些命令可以有效地完成各种工作。
- Linux的命令由Shell程序解释执行，所以也常称其为Shell命令。在使用Linux命令前首先要启动Shell程序。
- 在Linux系统中，Shell是最常使用的程序，其主要作用是侦听用户指令、启动指令所指定的进程并将结果返回给用户。

Linux系统由4个重要的部分组成：内核；Shell；文件系统；应用程序。

内核部份，操作者不易和它直接沟通，因此，必须要有一个友善的界面，使得操作时能更为方便，这个界面便是Shell。通俗地讲，Shell就是位于核心和操作者之间的一层使用者界面。

Shell的本意是“**壳**”**的意思。在核心的外面，包覆着一层外壳，用来负责接收使用者输入的指令，然后将指令解译成核心能够了解的方式，传给核心去执行，再将结果传回至预设的输出周边设备。**

Shell类型和功能

Linux系统中的Shell，除了可作为命令编译器之外，它也是一种不错的编程语言，是系统管理维护时的重要工具。

由于Linux系统对Shell的处理，采用独立自由开放的方式，因此，Shell的种类相当多，目前流行的Shell有sh，csh，ksh，tcsh和bash等。

Bourne shell （通称为sh）是 UNIX 最初使用的shell，并且在每种 UNIX 上都可以使用, 在 shell编程方面相当优秀，但在处理与用户的交互方面做得不如其他几种shell。

Linux 操作系统缺省的 shell 是Bourne Again shell，它是 Bourne shell 的扩展，简称 Bash，与 Bourne shell 完全向后兼容，并且在Bourne shell 的基础上增加、增强了很多特性。

在Linux系统中的bash具有以下功能：

（1）兼容Bourne Shell（sh）；

（2）包含C Shell以及Korn Shell中最好的功能；

（3）具有命令列编写修改的能力；

（4）具有工作控制的能力，可控制前台和后台程序；

（5）具有Shell编程能力。

- 启动Shell的方式有多种，通常的方式是：在字符终端登录，登录成功后Shell将自动启动；
- 在图形桌面登录，可以选择“应用程序”→“系统工具”→“终端”来打开终端界面，也可以右键单击桌面并从菜单中选择“打开终端”进入字符终端。

图 3-2

开始使用bash

当登入系统或打开一个终端窗口时，首先看到的是Shell提示符。

以普通用户zhangsan登入名为PC-LINUX的主机，当前工作目录是/home/zhangsan，如下所示：

[zhangsan@PC-LINUX ~]$

以root用户登录系统的提示符如下所示：

[root@PC-LINUX ~]#

要运行命令的话，只需要在提示符后敲进命令，然后再按“回车”键。

一个Shell命令可能含有一些可选项和参数，其一般格式为：

[Shell命令] [可选项] [参数]

3.2.1 命令的格式

一条Shell命令是由一到多个项组成的命令行，命令各项之间用空格分隔。命令的一般格式如下：

命令名 [选项1] [选项2]…[参数1] [参数2]…

其中，命令名是命令的名称，表示要执行的操作，通常为小写。

选项是对命令的特别定义，指出怎么执行这个操作，一般以“-”开始。

参数则是要操作的对象或数据。方括号括起的部分表明该项是可选的。

例如：rm -i abc

rm是命令名（remove），表示删除文件操作；

- i是命令选项，表示删除前要提示用户确认；

abc是命令参数，表示要删除的文件的名称。

另外，当一个命令带多个选项时，可以只用一个“-”连起来，如rm -r -f abc可以写成rm -rf abc。

3.2.2 命令的输入与修改

Shell命令是通过终端键盘输入的。输入命令时可以使用一些编辑键来修改输入错误，控制命令的执行。

表 3-1 常用的 Shell 命令行编辑键

按 键	功 能
Backspace、Delete、Ctrl+h	删除字符
Ctrl+u	删除整行
\	续行符，用于跨行输入长命令
Tab	命令补齐
↑ ↓	翻找命令历史记录前后移动光标

- 在Linux系统中有太多的命令和配置文件，可以使用命令补全功能快速地写出文件名和命令名称。
- 　如果需要快速地从当前所在的目录跳转到/usr/src/redhat/目录，可以执行以下操作：

- [root@PC-LINUX ～]# cd /u<Tab>/sr<Tab>/r<Tab>

- 通过按向上方向键，可以向后遍历最近在该控制台下输入的命令。用向下方向键可以向前遍历命令。

编辑命令行

通过光标和功能键（Home，End等键），可以浏览并编辑命令行，还可以用键盘的快捷方式来完成一般的编辑，主要功能键如下。

[Ctrl+k]：删除从光标到行尾的部分。

[Ctrl+u]：删除从光标到行首的部分。

[Alt+d]：删除从光标到当前单词结尾的部分。

[Ctrl+w]：删除从光标到当前单词开头的部分。

[Ctrl+a]：将光标移到行首。

[Ctrl+e]：将光标移到行尾。
[Ctrl+y]：插入最近删除的单词。
[!$]：重复前一个命令最后的参数。

可用的Shell快捷方式

Linux系统带有不少快捷方式，其中一部分是bash原来就有的，还有一些则是预先设置的。由于主目录是每位用户的活动中心，许多Linux系统对此有特殊的快捷方式。

每个用户都有自己的主目录，“~”就是每个用户的主目录的简写形式。

例如：[zhangsan@PC-LINUX ~]$

cp /etc/pssswd /home/zhangsan/docs

[zhangsan@PC-LINUX ~]$cp /etc/pssswd ~/docs

shell实用功能：

命令的排列、替换和别名

在Shell中可以使用命令排列同时执行多个命令，可以使用命令替换将一个命令的输出当作另一个命令的参数，也可以将复杂命令定义别名。

1. 命令的排列

如果希望一次执行多个命令，Shell允许在不同的命令之间，放上特殊的排列字符。这里将介绍最常用的两种方法。

（1）使用“；”

使用“；”命令时先执行命令1，不管命令1是否出错，接下来就执行命令2。

命令语法：

命令1；命令2

【例】 使用排列命令“;”同时执行两个命令。

```
[zhangsan@PC-LINUX ～]$whoami;echo hello

[zhangsan@PC-LINUX ～]$wh;echo hello
```

（2）使用"&&"

使用"&&"命令时只有当命令1正确运行完毕后，才能执行命令2。

命令语法：

命令1&&命令2

【例】 使用排列命令"&&"同时执行两个命令。

```
[zhangsan@PC-LINUX ～]$whoami&&echo hello

[zhangsan@PC-LINUX ～]$wh&&echo hello
```

2. 命令替换

将一个命令的输出当做另一命令的参数

命令替换这种机制的语法如下：

命令1 $(命令2)

除了使用"$（）"之外，还可以使用后引号"`"，命令语法如下所示。

命令1 `命令2`

命令替换是一项很实用的功能。

例如：杀死进程less

```
[root@PC-LINUX ~]# kill -9 $(pidof less)
```

3. 命令别名

在需要执行某一非常长的命令时，所有的命令以及命令的选项、参数都要一一输入，很枯燥也容易出现错误。可以为常用命令定义快捷方式，这些快捷方式可以用较简单的命令别名来定义。

命令语法：

alias [别名]=[需要定义别名的命令]

- **例如：**
- 使用下面命令查看/boot目录内容
- [root@PC-LINUX ～]# ls -l /boot
- [root@PC-LINUX ～]# alias ok="ls -l /boot"

- 取消别名的定义使用unalias命令
  unalias [别名]
- [root@PC-LINUX ～]# unalias ok

- 注意：
- 当创建别名时，如果已经有同名的别名存在，则原有的别名设置将被替换。
- 在重启计算机或关闭终端后，定义的别名就会失效。
- 可以通过将设置别名的命令写进启动文件使别名持久生效。

shell实用功能：

文件名匹配和输出重定向

1. 文件名匹配

文件名匹配使得用户不必一一写出文件名称就可以指定多个文件。

“*”可匹配零个或多个字符。

第二个通配符是问号“?”。在匹配时，一个问号只能代表一个字符。

假如当前目录下，包含124.bak，346.bak，583.bak，和some.text 4个文件。

删除当前目录下所有以字符串“.bak”结尾的文件。

[root@PC-LINUX ～]#rm *.bak

若想保留583.bak文件，删除其它以“.bak”结尾的文件文件，则：

[root@PC-LINUX ～]#rm *4*.bak

若想列出所有在点号后有4个字符的文件，则：

[root@PC-LINUX ～]#ls *.????

“[]”
代表“[”和“] ”之间的某一个字符，比如[0-9]可以代表0-9之间的任意一个数字，[a-zA-Z]可以代表a-z和A-Z之间的任意一个字母，字母区分大小写。

“^”
表示匹配结果取反的意思，注意这个通配符必须要在[]中使用

“{} ”
表示符合括号内包含条件（可多个条件）的文件

[abcd]表示匹配中括号内任意一个字符就成立

```
[root@chengliang ~]# ls t[abcde]st
test
[root@chengliang ~]# ls t*t
test  tmpexit
[root@chengliang ~]#
```

```
[admin@localhost ~]$ touch {a..f}
[admin@localhost ~]$ ls
a  b  c  d  e  f
[admin@localhost ~]$ ls [a-c]
a  b  c
```

```
[admin@localhost ~]$ ls [!ac]
b  d  e  f
[admin@localhost ~]$ ls [^ac]
b  d  e  f
```

2. 管道

Linux系统的理念是汇集许多小程序，每个程序都有特殊的专长。复杂的任务不是由大型软件完成，而是运用Shell的机制，组合许多小程序共同完成。管道就在其中发挥着重要的作用，它可以将某个命令的输出信息当作某个命令的输入，由管道符号"|"来标识。

命令语法：

[命令1]|[命令2]|[命令3]

【例】 将命令ls /etc显示的内容分页显示。

[root@PC-LINUX ～]#ls /etc|more

3. 重定向文件

有时，希望将命令的输出结果保存到文件中，或以文件内容作为命令的参数，这时就需要用到重定向。

重定向不使用系统的标准输入端口、标准输出端口或标准错误端口，而进行重新的指定。

重定向有5种方式，分别是：输出重定向、输入重定向、错误重定向、追加重定向以及同时实现输出和错误的重定向。

（1）输出重定向

输出重定向，即将command命令的输出保存到文件file中，如果存在相同的文件，则覆盖file文件中的内容。

命令语法：

```
command > file
```

【例】 使用输出重定向将目录/boot的内容保存到文件/root/abc。

```
[root@PC-LINUX ～]#ls /boot > /root/abc
```

【例】 使用echo命令和输出重定向创建文本文件/root/mm，内容是hello。

```
[root@PC-LINUX ～]#echo hello > /root/mm
```

（2）输入重定向

输入重定向，即将文件file的内容作为command命令的输入。

命令语法；

command < file

【例】 使用输入重定向将文件/root/mm的内容作为输入让cat命令执行。

[root@PC-LINUX ～]#cat < /root/mm

（3）错误重定向

错误重定向，即将command命令执行的出错信息输出到文件file中。

命令语法：

```
command 2> file
```

【例】查看根本不存在的/root/kk文件，出现报错信息，将其保存到文件/root/b中。

```
[root@PC-LINUX ~]#cat /root/kk 2> /root/b
```

（4）追加重定向

追加重定向，即将command命令执行的输出添加到已存在的文件file中。

命令语法：

```
command >> file
```

【例4.22】 使用追加重定向将数据写入文件/root/a。

```
[root@PC-LINUX ～]#echo aaa > /root/a
[root@PC-LINUX ～]#echo bbb >> /root/a
```

（5）同时实现输出和错误的重定向

同时实现输出和错误的重定向，即可以同时实现输出重定向和错误重定向的功能。

命令语法：

```
command &> file
```

命令成功执行，使用输出重定向

命令执行不成功，使用错误重定向

为何要使用命令输出重导向

- 当屏幕输出的信息很重要，需要将他存下来的时候；
- 背景执行中的程序，不希望他干扰屏幕正常的输出结果时；
- 一些系统的例行命令（例如写在 /etc/crontab 中的文件）的执行结果，希望他可以存下来时；
- 一些执行命令，已经知道他可能的错误讯息，所以想以『2> /dev/null』将他丢掉时；
- 错误讯息与正确讯息需要分别输出时。

1. 通配符

表1　　bash中使用的通配符

[root@PC-LINUX ～]#cp test[1~5] /tmp
test1, test2, test3, test4, test5若存在，则复制到/tmp目录下

2. 特殊字符及引号

表2　　特殊字符及引号

3. 常用控制组合键

表3　　常用控制组合键

3.2.3 命令的执行

命令输入完成后，就可按Enter键提交给Shell运行。运行结果通常显示在屏幕上。

Enter，Ctrl+m，Ctrl+j提交命令运行

Ctrl+c键终止命令的运行。

3.2.4 命令的分类

按照实现方式的不同，Shell命令分为内部命令和外部命令两种。

内部命令指的是集成于Shell解释器程序内部的一些特殊指令，也成为内建（Built-IN)指令。
内部命令属于Shell的一部分，只要Shell解释器被运行，内部指令也就自动载入内存了，用户可以直接使用。
内部命令无需从硬盘中重新读取文件，因此执行效率更高。

3.2.4 命令的分类

按照实现方式的不同，Shell命令分为内部命令和外部命令两种。

外部命令指的是Linux系统中能够完成特定功能的脚本文件或二进制程序，完成比较复杂或耗时的功能。
每个外部命令对应系统中的一个文件，是属于Shell解释器程序之外的命令，是在bash之外额外安装的，所以称为外部命令。
Linux系统必须知道外部命令对应的文件位置，才能够由Shell加载并执行。

- 内部命令在系统启动时就调入内存，是常驻内存的，所以执行效率高。
- 外部命令是系统的软件功能，用户需要时才从硬盘中读入内存。
- 查看命令为内部命令还是外部命令?

- type可以用来判断一个命令是否为内置命令
- enable既可以查看内部命令，同时也可以判断是否为内部命令

```
[root@localhost Desktop]# type cd
cd 是 shell 内嵌
[root@localhost Desktop]# type pwd
pwd 是 shell 内嵌
[root@localhost Desktop]# type shutdown
shutdown 是 /sbin/shutdown
[root@localhost Desktop]# enable -a
enable .
enable :
enable [
enable alias
enable bg
enable bind
enable break
enable builtin
enable caller
enable cd
```

按命令功能分类，Shell命令可以大致分为以下几类：

文件、目录操作；

文本编辑与处理；

备份与压缩；

系统监控与管理；

网络通信；

其他

3.2.5 简单命令

who命令

【功能】显示已登录的用户。

【格式】who [选项] [am i]

【选项】

-H　　显示各列的标题。

-q　　显示登录的用户名和用户数。

【参数】带有参数am i时，显示自己的登录信息。

【说明】显示内容分为4列：用户名、登录的终端名、登录时间和备注。有-q选项时，只输出用户名和用户数。

例3.1 who命令用法示例：

```
$ who
  root       tty1     May 25  11:39                tty（teletype）
  zhao       tty2     May 25  09:12
  cherry     :0       May 25  08:45
  cherry     pts/0    May 25  08:45  (:0.0)        pts（Pseudo Terminal）
$ who -q
  root       zhao     cherry  cherry
  # users=4
 $ who am i
   cherry     pts/0    May 25  08:45  (:0.0)
 $
```

echo命令

【功能】显示命令行中的参数字符串。

【格式】echo [选项] [字符串]…

【选项】

-n 输出字符串后光标不换行。

-e 若字符串中出现“\a,\b,\t……”等字符，则特别加以处理，而不会将它当成一般字符原样输出。

例3.2 echo命令用法示例：

```
$ echo Hello
  Hello
[root@PC-LINUX ～]# echo -n Hello
Hello[root@PC-LINUX ～]#
$ echo "Hello\nworld"
  Hello\nworld
```

```
$ echo -e “Hello\nworld”
Hello
world
```

```
$ echo Hello; echo world
Hello
world
```

date命令

【功能】显示、设置系统日期和时间。

【格式】date [选项] [+格式]

【选项】

-d　显示描述的日期。字符串前后必须加上双引号；

-s　设置时间和日期。

-u　使用格林威治时间。

【参数】格式是由格式控制字符和其他字符构成的字符串，用于控制输出的格式。当格式字符串中有空格时，要用引号“”将格式字符串括起来。常用的格式控制字符如下：

%r 用hh:mm:ss AM/PM(□:分:秒 上午/下午)的形式□示12小□制□□。

%T 用hh:mm:ss(□:分:秒)的形式□示24小□制□□。

%a □示星期的□写，如Sun。

【说明】不带选项和格式参数时显示当前日期与本地当前时间。

显示格式是：

星期 月 日 时间 时区 年

其他格式控制字符

%A	口示星期的全称，如Sunday。
%b	口示月份的口写，如Jan。
%B	口示月份的全称，如January。
%m	用2位数字口示月份，如02。
%d	用2位数字口示日期，如27。
%D	用mm/dd/yy(月/日/年)的形式口示日期，如02/27/08。
%y	用2位数口示年份，如08。
%Y	用4位数口示年份，如2008。

例 date命令用法示例：

```
$ date
  Mon  May  26  20:04:34  CST  2008
```

月/日/年的形式显示

```
$ date "+Today is %D, now is %r"
  Today is 05/26/08, now is 08:14:36 PM
```

```
$ date "+%B %d, %Y"
   May 26, 2008
```

4位数字口示年份

月份的全称

2位数字口示日期

- date +"%Y-%m-%d"
- date -d "1 day ago" +"%Y-%m-%d“
- date -s #设置当前时间，只有root权限才能设置，其他只能查看

cal命令

【功能】显示月份和日历。

【格式】cal [[月份] 年份]

【参数】月份是1～12的数字，年份是1～9999的数字。

【说明】若带有一个参数，则该参数被解释为年份；若带有两个参数，则第1个参数表示月份，第2个参数表示年份。不带参数时，显示当年当月的日历。

例 cal命令用法示例：

```
$ cal           #显示当年当月的日历
$ cal 3 2016    #显示2016年3月的日历
```

cal命令的选项：

- -1：显示单月输出；
- -3：显示临近三个月的日历；
- -s：将星期日作为月的第一天；
- -m：将星期一作为月的第一天；
- -j：显示“julian”日期；
- -y：显示当前年的日历。

在Linux下获取帮助

Linux系统中的每个命令都具有众多的参数和选项，要一一记住比较困难，可借助Linux的**帮助功能**。

3.2.6 联机帮助

1. --help选项

许多Linux命令都提供了一个--help选项。

执行带有--help选项的命令将显示该命令的帮助信息。

例如：显示date命令的帮助信息。

[root@PC-LINUX ～]#date --help

3.2.6 联机帮助

2. man（manual）命令

Linux系统配有一个联机手册，每条Linux命令都对应有相关的手册页。手册页是对命令的解释，因此是学习和使用Linux命令的必不可少的工具。

man命令

【功能】显示联机手册页。

【格式】man [选项] [命令名称]

【说明】在浏览手册页时，用以下按键翻页、查找和退出：

PageUp、b	向上翻一页。
PageDown、Space	向下翻一页。
↑	向上滚一行。
↓、Enter	向下滚一行。
/string	在手册页中查找字符串string。
n	查找下一个字符串。
q	退出。

3．info命令

除了联机手册外，Linux系统还提供了大多数命令的超文本形式的联机文档，可用info命令浏览。info命令与man命令的用法类似，但浏览起来更方便。

- 敲？键，它就会显示info的常用快捷键。
- N键：显示（相对于本节点的）下一节点的文档内容
- P键：显示（相对于本节点的）前一节点的文档内容
- U键：进入当前命令所在的主题
- M键：敲M键后输入命令的名称就可以查看该命令的帮助文档了
- G键：敲G键后输入主题名称，进入该主题
- L键：回到上一个访问的页面
- SPACE键：向前滚动一页
- BACKUP或DEL键：向后滚动一页
- Q：退出info

--help , man , info

- --help 选项并不是一个“独立”的工具。作为一种命令的选项，它可以显示命令的一些信息。提供的是一种快捷、高效的帮助。
- man和info就像两个集合，它们有一个交集部分，但与man相比：info工具可显示更完整的最新的GNU工具信息。
- 若man页包含的某个工具的概要信息在info中也有介绍，那么man页中会有“请参考info页更详细内容”的字样。通常情况下，man工具显示的非GNU工具的信息是唯一的，而info工具显示的非GNU工具的信息是man页内容的副本补充。

思考题

- 简述进入命令行界面有哪些方式?
- 简述可以使用哪些命令重启或关闭计算机。
- 简述Linux系统的运行级别。
- 简述Linux系统下获取帮助信息的命令及其特点。
- 简述有哪些重定向的方式。
- 列举自己课内和课外学到的命令，并说明其功能及用法

3.3 Linux文件操作

用户使用Linux系统的最基本的操作就是对文件和目录的操作。与Windows系统相同，Linux系统提供了在图形窗口界面操作文件的便利手段。

在Linux系统中，最基本和最有效的操作文件的方法是使用命令。因此用户应该熟练掌握用命令方式操作文件的方法。

3.3 Linux文件操作

3.3.1 Linux系统的文件

3.3.2 Linux目录结构

3.3.3 常用的目录操作命令

3.3.4 常用的文件操作命令

3.3.1 Linux系统的文件

1. 文件的命名
2. 文件名通配符
3. Linux文件的扩展名
4. 文件的类型
5. 文件的归属关系
6. 文件的访问权限
7. 新建文件的默认权限
8. 文件的其他属性

1. 文件的命名

Linux文件名的最大长度是255个字符，通常由字母、数字、“.”、“_”和“—”字符组成。以“.”开头的文件名是隐含文件(通常的文件列表不显示)。

文件名中不能含有斜杠字符“/”和空字符“\0”，因为它们对Linux内核具有特殊含义，如“/”表示根目录或路径分隔符。文件名中也不应含有空格符、制表符、控制符以及以下字符：

; | < > ` “ ‘ $! % & * ? \ ( ) [ ]

因为它们对Shell具有特殊含义。

2. 文件名通配符

1) 模式与通配符

模式是对一类事物的一种概括性描述。需要指定具有某种特征的多个文件名时，可以用一个表示文件名的**字符串模式**来描述。

字符串模式由**普通字符**和一些具有**特殊**含义的**字符**组成，这些特殊字符称为**通配符**(wildcard)。 通配符不代表某个具体的字符，而是代表多种选择，就像中奖号码模式中“X”的作用一样。 这样，用一个模式来表示多个文件名，就不必在命令的参数中写出每个文件的名字了。

2) 基本的通配符与匹配规则

以下是在构造模式时常用的基本通配符：

(1) 问号“？”：匹配任意的单个字符。如模式“abc??”匹配所有以abc开始，后面是2个任意字符的字符串。

(2) 星号“*”：匹配0或多个任意字符(隐含文件的前缀“.”字符除外)。如模式“abc*”匹配所有以abc开始的字符串。模式“*abc”匹配所有以abc结尾的字符串，但不匹配隐含文件“.abc”。

(3) 方括号“[]”：匹配方括号中列出的字符集合中的任何单个字符。

方括号与问号相似，只匹配单个字符。不同的是，问号与任何一个字符匹配，而方括号只与括号内字符集合中的一个相匹配。字符集合的描述方法有以下几种：

列举：逐个列出各个字符，如[abc]表示由a、b、c三个字符构成的字符集合。

范围：用“-”描述字符范围，如[0-9]表示0-9构成的集合。注意，范围内的字符按升序排列，因而[9-0]是无效的。可以指定多个范围，如[A-Za-z]表示所有英文字母。

3. Linux文件的扩展名

文件扩展名是文件名的最后一个点（.）之后的部分，下面简单列出了一些在Linux系统中的文件扩展名和它们的含义。

3. Linux文件的扩展名

1）压缩的和归档的文件

.bz2 ------bzip2的压缩文件

.gz ------gzip的压缩文件

.tar ------tar打包文件（是包文件不是压缩文件）

.tbz------tar打包并用bzip压缩文件

.tgz-----tar打包并用gzip压缩的文件

.zip-----zip压缩的文件，Linux下使用gzip命令压缩的文件

2）文件格式

.au -----audio文件　　.gif -----gif图像文件

.html/.htm-----HTML文件　　.jpg-----JPEG图像文件

.pdf------电子文档（PDF格式的）

. png-----PNG图像文件

.ps------postscipt文件（打印格式文件）

.txt------纯文本文件　　.wav-----audio文件

.XPm-----图像文件

3）系统文件

.conf-------配置文件（有时也用.cfg）

.lock-------LOCK文件（用来判断一个文件或设备是否被使用）

.rpm------Linux系统中用来安装软件的软件包管理文件

4）编程和脚本文件

.c -------C源程序代码文件

.cpp------C++源程序代码文件

.h -------C或C++程序的头文件

.o------程序目标文件

.pl------perl脚本文件

.so-----类库文件

.sh：Shell脚本

.tcl：TCL脚本

文件扩展名不总是被使用或被一致地使用。
可以使用file命令查看该文件的类型就可以了

```
[root@PC-LINUX ～]# file mm
 mm: ASCII text
```

4.Linux文件类型

Linux文件类型和Linux文件的文件名所代表的意义是两个不同的概念。

通过一般应用程序创建的文件，比如file.txt，file.tar.gz，这些文件虽然要用不同的程序来打开，但放在Linux文件类型中衡量的话，大多称之为普通文件。

Linux系统定义了一些特殊类型的文件，它们在系统中具有特殊的用途。

Linux文件类型常见的有：普通文件、目录文件、设备文件、管道文件和符号链接文件等。

Linux系统支持以下文件类型(括号内是表示该类型的字符)：

普通文件(-)：普通意义上的文件，用于保存文本、数据或程序等信息。

目录文件(d)：一种特殊文件，用于构成文件系统的树型结构。

设备文件(c、b)：Linux系统将设备看做是一种特殊文件，设备文件分为字符设备文件(c)和块设备文件文件(b)两类。

符号链接文件(l)：一种特殊文件，它的内容是到另一个文件的链接，用于实现文件的共享。链接方式有两种：软链接和硬链接。

管道文件(p)：一种特殊文件，用于在进程间传递数据。

● 普通文件

[root@localhost ~]# ls -lh install.log

-rw-r--r-- 1 root root 53K 03-16 08:54 install.log

用 ls -lh 来查看某个文件的属性，可以看到有类似 -rw-r--r-- ，值得注意的是第一个符号是 - ，这样的文件在Linux中就是普通文件。

这些文件一般是用一些相关的应用程序创建，比如图像工具、文档工具、归档工具等。

● 目录文件

```
[root@localhost ~]# ls -lh
总计 14M
......
drwxr-xr-x 2 1000 users 4.0K 04-04 23:30 mkuml-
2004.07.17
drwxr-xr-x 2 root root  4.0K 04-19 10:53 mydir
......
```

在某个目录下执行，看到有类似 drwxr-xr-x ，这样的文件就是目录，目录在Linux是一个比较特殊的文件。注意它的第一个字符是d。

● 设备文件

Linux系统下的/dev目录中有大量的设备文件

（1）块设备文件

主要特点：可以随机读写，最常见的块设备就是硬盘。

（2）字符设备文件

最常见的字符设备是打印机和终端，可以接收字符流。/dev/null是个非常有用的字符设备文件，送入这个设备的所有内容都被忽略。

● 设备文件

进入/dev目录，列一下文件，会看到类似如下的；

```
[root@localhost ~]# ls -la /dev/tty
crw-rw-rw- 1 root tty 5, 0 04-19 08:29 /dev/tty
[root@localhost ~]# ls -la /dev/hda1
brw-r----- 1 root disk 3, 1 2006-04-19 /dev/hda1
```

看到/dev/tty的属性是 crw-rw-rw-， 注意前面第一个字符是 c， 这表示字符设备文件。比如猫等串口设备
我们看到 /dev/hda1 的属性是 brw-r-----， 注意前面的第一个字符是b， 这表示块设备，比如硬盘，光驱等设备；

● 管道文件

管道是Linux系统中一种进程通信的机制。通常，一个进程写一些数据到管道中，这些数据就可以被另一个进程从这个管道中读取出来。

管道可以分为两种类型：无名管道与命名管道。

>. 无名管道由进程在使用时创建，读写结束关闭文件后消失。因为它们并不存在于文件系统中，没有文件名称，所以称为无名管道。

● 管道文件

>. 命名管道在形式上就是文件系统中的一个文件，它不占用存储文件内容的磁盘空间，但有自己的文件名。命名管道通常称为FIFO文件。

```
[root@localhost ~]# ls -l /dev/initctl
prw-------. 1 root root 0 6月  3 05:24  /dev/initctl
```

● 链接文件

（1）软链接文件

软链接又称为符号链接，即某文件包含了另一个文件的路径名，可以是任意文件或目录，也可以链接不同文件系统的文件。可以链接不存在的文件，可以循环链接自己。在对符号文件进行读或写操作时，系统会自动把该操作转换为对源文件的操作，但删除链接文件时，系统只删除链接文件，而不删除源文件本身。

● 链接文件

（2）硬链接文件

硬链接只能引用同一文件系统中的文件。它引用的是文件在文件系统中的物理索引（也称为 inode）。当移动或删除原始文件时，硬链接不会被破坏，因为它所引用的是文件的物理数据而不是文件在文件结构中的位置。

如果删除的文件有相应的硬链接，那么这个文件依然会保留，直到所有对它的引用都被删除。

Linux系统支持以下文件类型：

普通文件(-)

目录文件(d)

设备文件(c、b)

符号链接文件(l)

管道文件(p)

5. 文件的归属关系

Linux是一个多用户的系统，每个用户都要在系统中存放自己的文件。为了管理的需要，系统要能够区分文件的归属关系。Linux系统中的每个文件都有两个描述其归属关系的属性，这就是属主(owner)和属组(group owner)。

5. 文件的归属关系

文件的属主就是文件的所有者，通常是建立文件的用户，用其用户名标识。例如：用户zhou建立的文件的属主就是zhou。

为便于管理，Linux系统将用户划分为用户组。文件的属组就是文件属主所在的用户组，用组名标识。例如：用户zhou所在的用户组是guest，则他所建立的文件的属组就是guest。

- 在多用户的系统，文件的保密和安全性至关重要。
- 为防止文件被非法地使用或破坏，系统使用权限来限制用户对文件的访问。

6. 文件的访问权限

1) 文件的访问权限

文件权限用于规定对于一个文件所能进行的操作。通常访问文件的操作分为读(显示浏览文件内容)、写(修改文件的内容)和执行(运行可执行文件)。相应的，Linux对文件定义了几种访问权限。

表 3-3 文件和访问权限及表示

访问权限	字符表示	含 义
读权限	r	可读取其内容
写权限	w	可修改其内容
执行权限	x	可执行其内容
无权限	-	不能做相应的操作

2) 文件的权限范围

Linux系统将每个文件的用户分为属主(user)、组用户(group)和其他人(other)3类，权限范围的划分及字符表示法如表3-4所示。在为文件设置访问权限时可以针对不同的权限范围分别设置。注意：root用户不受访问权限的限制。

表 3-4 文件的权限范围划分及表示

权限范围	针对的用户	字符表示
属主	文件的拥有者，通常是创建文件的用户	u
组用户	文件的属主所在的用户组中的其他用户	g
其他人	除文件属主和组用户外的其他系统用户	o
所有人	以上 3 类用户的总和	a

- 在Linux系统中，一个文件可能会被多个用户使用。
- 如果不加区分地对所有用户设置相同的文件访问权 限，则难以满足不同用户对此文件的不同需求和权 利。
- 因此，Linux系统采用了更加细致的权限分配方式，即允许对不同类型的用户赋予不同的文件访问权限。

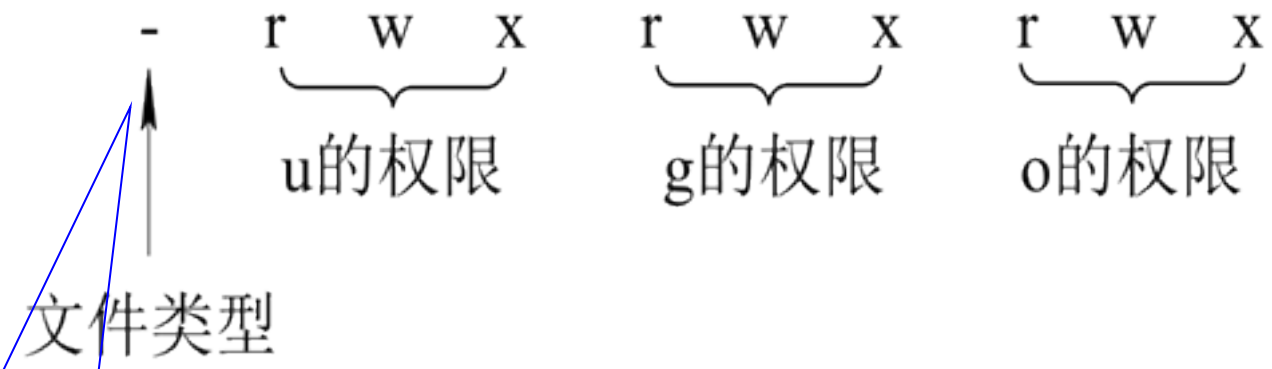
- 在Linux里面，任何一个文件都具有『User, Group及Others』三种身份的权限。

Linux 中用户身份与群组记录的文件

- 在Linux系统当中，默认的情况下，所有的系统上的账号与一般身份使用者，还有那个root的相关信息，都是记录在/etc/passwd这个文件内的。
- 个人的密码则是记录在/etc/shadow这个文件下。
- Linux所有的组名都纪录在/etc/group内！

3) 文件类型与权限表示法

文件的类型与权限是文件的重要属性，通常采用字符表示法表示，即用10个字符的字符串表示文件的类型和权限，规则如图3-3所示。



文件类型与访问权限的表示

文件的访问权限的另一种表示方法是数字表示法。其规则是：用数字1或0来表示权限字符，有相应权限的位为1，无权限的位为0，形成一个9位长的二进制数，用3位八进制数字来表示。

例如：字符表示是rwxr-x---，　数字表示就是750；

字符表示是rwx--x--x，　数字表示就是711。

4) 文件权限的作用

文件权限限制了对文件的访问操作。正确地设置文件权限可以允许正常的访问操作，同时阻止不期望的访问。表3-5显示了访问权限对普通文件和目录文件的限制作用。

表 3-5 对文件和目录的访问权限的作用

访问权限	字符表示	对文件的访问限制	对目录的访问限制
读权限	r	可读取其内容	可列出其中的文件列表
写权限	w	可修改其内容	可在其中建立、删除文件，或改文件名
执行权限	x	可执行其内容	可进入该目录，可访问该目录下的文件

- 访问权限对普通文件的作用容易理解，需要注意的是权限对目录的限制作用。
- 目录其实也是一个文件，只不过它的内容不是记录普通数据，而是其下的文件的列表数据。
- 因此，显示目录中的文件列表就是对目录文件的读操作，改变目录下的文件列表(新建文件、删除文件、修改文件名等)就是对目录文件的写操作，进入目录或其下级子目录就是对目录文件的执行操作。

- 因此，对文件的删除权由其所在的目录的w权限决定(当然还要有x和r权限)，而不是文件本身的w权限决定的。在这一点上，Linux系统是不同于Windows系统的。
- 另外，Linux系统规定非空目录不能删除。而空目录等同于文件，它的删除权取决于它的上一级目录的w权。
- 如果希望目录让其他人浏览，至少要给予r和x权限。

例： 设有如下3个目录及其各自下属的3个文件，这些文件的删除权如下(首先要对目录有权限，才能对文件和子目录有权限)。

目录1：drwxr-x--x

文件1：-rwxr-xr-x　目录的属主可删除

目录1的权限为rwxr-xr-x，则目录的属主可以完全控制这个目录，其他人只能进入目录和显示文件列表，只有目录属主有权删除文件1。

(首先要对目录有权限，才能对文件和子目录有权限)。

目录2：drwxrwxrwx

文件2：-rwx------　任何人可删除

目录2的权限为rwxrwxrwx，即所有人可完全控制该目录。即使它下面的文件2的权限为rwx------，阻止了除属主之外的人访问这个文件，但他们却可以删除它。他们还有权在此目录中建立新文件、删除目录下的任意文件(包括空目录)和更改目录下的任意文件的文件名。所以在Linux中存放文件小心谨慎，不要把重要文件放在所有人可完全控制的目录里，即使这个文件的权限是0。

(首先要对目录有权限，才能对文件和子目录有权限)。

目录3：dr-x------

文件3：-rwxr-xr-x　只有目录属主可看到，任何人不可删除

目录3的权限为r-x------，则只有目录属主可以进入目录和看到目录中的文件列表，所有人(包括属主)都不能在目录中建立、删除文件或改文件名。即使它下面的文件3赋予其他人读和执行的权限，他们因为无法进入和使用这个目录，也就无法读和执行这个文件。

这是用于保管重要文件的高安全度限制。

7. 新建文件的默认权限

当新建一个文件或目录时，系统会为其设置最初的权限。文件的初始权限由文件创建掩码(creation mask)决定。掩码是一个9位二进制数字，通常用八进制数字表示，如022。掩码中的位与权限字符串相对应，设置的是权限补码，即掩码中为1的位限制对应的权限位的权限。例如：掩码022表示组用户和其他人没有w权限，对其他权限不做限制。

文件创建时的默认权限有以下几种情况：

1) 可执行文件

通过编译程序生成的可执行文件，它的默认权限是777-掩码。例如：若掩码为022，则新文件的权限就是755。

2) 非可执行文件

对于非可执行文件(如文本文件、数据文件等)，在创建时默认是没有x权限的，因此新建文件的权限是(777-掩码)&666。

例如：

若掩码为022，则新文件的权限就是(777-022)&666 = 644；

若掩码为003，则新文件的权限就是(777-003)&666 = 664。

对非可执行文件可通过chmod命令将其改为可执行文件。

3) 目录

同可执行文件一样，新建目录的默认权限是777-掩码。若掩码为022，则新目录的权限就是755。

用户登录时，系统自动地为其设置了掩码，通常是022。用户可以用命令umask修改掩码，从而改变新建文件的默认权限，使之具有合适的安全性限制。

Linux文件权限的重要性

与Windows系统不一样的是，在Linux系统当中，每一个文件都多加了很多的属性进来，尤其是群组的概念，这样最大的用途是在“数据安全性”上面的。

- 系统保护的功能：
- 在Linux系统中，关于系统服务的文件通常只有root才能读写或者是执行，例如/etc/shadow这一个账号管理的文件，由于该文件记录了你系统中所有账号的数据， 因此是很重要的一个配置文件，只有root才能够来读取，所以该文件的权限就会是[-rw-------]。

- 团队开发软件或数据共享的功能
- 如果有一个软件开发团队，希望这个团队每个人都可以使用某一些目录下的文件， 而非这个团队的其他人则不予以开放呢?
- 例如，testgroup的团队共有三个人，分别是test1, test2, test3，那么就可以将团队所需的文件权限订为[-rwxrwx---]来提供给testgroup的工作团队使用。

- 权限设定不恰当的危害
- 如果你的目录权限没有作好的话，可能造成其他人都可以在你的系统上面捣乱！
- 例如本来只有root才能做的开关机、ADSL的拨接程序、新增或删除用户等等的指令，若被你改成任何人都可以执行的话， 那么如果使用者不小心给你重新启动、重新拨接等，而且万一你的用户的密码被其他不明人士取得的话，只要他登入你的系统就可以轻而易举的执行一些root的工作！

- 在shell中，使用chmod命令改变文件或目录的访问权限。用户用它控制文件或目录的访问权限。
- 在 shell 中，使用chown命令来改变文件所有者（属主）及用户组（属组）。
- 在shell中，使用chgrp命令来改变文件所在用户组（属组）。

- 增加文件所有用户组可执行权限
- 命令：chmod a+x 文件名

- 同时修改不同用户权限
- 命令：chmod ug+w,o-x 文件名

- 使用"="设置权限
- 命令：chmod u=r 文件名

- chmod 751 文件名

8. 文件的其他属性

除了文件名、文件类型、归属关系和存取权限外，文件还有其他一些属性，包括：

(1) 文件的时间标签，用于记录文件的时间属性，分为：

修改时间(modify time)：文件内容被修改的最后时间。

访问时间(access time)：文件最近一次被访问的时间。

变更时间(change time)：文件属性变更的最近时间。

(2) 文件的大小，即文件所占用的字节数。

(3) 文件的链接数，即此文件硬链接的数目。

root@localhost:~

文件(F) 编辑(E) 查看(V) 终端(T) 转到(G) 帮助(H)

```
-rw-------  1 root root   254 2014-03-23  a.c
-rw-------  1 root root   248 2014-03-23  a.c~
-rw-r--r--  1 root root  1219 2011-07-05  anaconda-ks.cfg
-rw-------  1 root root  2264  3月 23 21:24 .bash_history
-rw-r--r--  1 root root    24 2000-06-11  .bash_logout
-rw-r--r--  1 root root   234 2001-07-06  .bash_profile
-rw-r--r--  1 root root   176 1995-08-24  .bashrc
-rw-r--r--  1 root root   210 2000-06-11  .cshrc
drwx------  3 root root  4096 2014-03-23  .emacs.d
-rw-r--r--  1 root root 46996 2014-03-23  .fonts.cache-1
drwx------  5 root root  4096  3月 23 20:30 .gconf
drwx------  3 root root  4096  3月 23 22:02 .gconfd
drwx------  5 root root  4096 2014-03-23  .gnome
drwxr-xr-x  6 root root  4096  3月 23 21:56 .gnome2
drwx------  2 root root  4096 2014-03-23  .gnome2_private
drwxr-xr-x  2 root root  4096 2014-03-23  .gnome-desktop
drwxr-xr-x  2 root root  4096 2011-07-05  .gstreamer
-rw-r--r--  1 root root   120 2003-02-27  .gtkrc
-rw-r--r--  1 root root   130 2014-03-23  .gtkrc-1.2-gnome2
-rw-------  1 root root  1134  3月 23 20:30 .ICEauthority
-rw-r--r--  1 root root 18275 2011-07-05  install.log
-rw-r--r--  1 root root  3049 2011-07-05  install.log.syslog
drwx------  3 root root  4096 2014-03-23  .metacity
-rw-r--r--  1 root root     6  3月 23 20:37 mm
```