

回顾

- 系统运行级别
- 如何注销、重启、关机
- 如何提高效率
 - 命令如何自动补全
 - 历史命令如何调出
 - **alias**
 - 通配符
- 如何获取帮助

第3章 Linux常用命令和文件操作

命令

宣江华

本章学习目标

- 文件（目录）的基本操作（列出、创建、复制、移动、删除）
- 文件权限的设定
- 文件的查找等操作
- 文件内容的查看

目录和文件名的命名规则

- 在Linux下可以使用长文件或目录名
 - 可以长达255个字符
- 可以给目录和文件取任何名字，但必须遵循下列的规则：
 - 除了/之外，所有的字符都合法
 - 有些字符最好不用，如空格符、制表符、退格符和字符：？，@ # \$ & () \ | ； ‘ ’ “ ” < > 等。
 - 避免使用+、-或.来作为普通文件名的第一个字符
 - 大小写敏感
 - 以“.”开头的文件或目录是隐含的

文件的类型

- 普通文件 (-)
- 目录 (d)
- 符号链接 (l)
- 字符设备文件 (c)
- 块设备文件 (b)
- 套接字 (s)
- 命名管道 (p)

普通文件

- 普通文件仅仅就是字节序列，**Linux** 并没有对其内容规定任何的结构。
- 普通文件可以是程序源代码（**c**、**c++**、**python**、**perl**等）、可执行文件（文件编辑器、数据库系统、出版工具、绘图工具等）、图片、声音、图像等。
- **Linux** 不会区别对待这些普通文件，只有处理这些文件的应用程序才会对根据文件的内容赋予相应的含义。
- 在**Linux**环境下，只要是可执行的文件并具有可执行属性它就能执行，不管其文件名后缀是什么。但是对一些数据文件一般也遵循一些文件名后缀规则。

目录

- 目录文件是由一组目录项组成，目录项可以是对其他文件的指向也可以是其下的子目录指向。
- 一个文件的名称是存储在他的父目录中的，而并非同文件内容本身存储在一起。
- 硬连接文件实际上就是在某目录中创建目录项，从而使不止一个目录可以引用到同一个文件。这种链接关系由 **ln** 命令行来建立。
- 硬链接并不是一种特殊类型的文件，只是因为文件系统中允许不止一个目录项指向同一个文件。

符号链接

- 符号链接又称软链接，是指将一个文件指向另外一个文件的文件名。
- 这种符号链接的关系由 **ln -s** 命令行来建立。

设备文件

- 设备是指计算机中的外围硬件装置，即除了**CPU**和内存以外的所有设备。通常，设备中含有数据寄存器或数据缓存器、设备控制器，它们用于完成设备同**CPU**或内存的数据交换。
- 在 **Linux** 下，为了屏蔽用户对设备访问的复杂性，采用了设备文件，即可以通过象访问普通文件一样的方式来对设备进行访问读写。
- 设备文件用来访问硬件设备，包括硬盘、光驱、打印机等。每个硬件设备至少与一个设备文件相关联。
- 设备文件分为：字符设备（如：键盘）和块设备（如：磁盘）。

Linux下设备的使用

- 设备的使用方法
- 用户可以用设备名来使用设备
- 用户可以用访问文件的方法来使用设备
- 设备名以文件系统中的设备文件的形式存在
- 所有的设备文件存放在/dev目录下
- 几个特殊的设备
 - /dev/null 一空设备
 - /dev/zero 一零设备

Linux 常用命令

cat	查看文件内容	more/less	查看文件内容
cd	切换工作目录	touch	改变文件的时间属性
chown	改变文件属权	mv	改名或移动文件
chmod	改变文件权限	pwd	显示当前所在的目录
clear	清除屏幕	rm	删除文件或目录
cp	拷贝文件	find	查找文件
ln	创建文件链接	which	寻找命令
ls	显示目录内容	tar	文件打包
mkdir rmdir	创建/删除目录	[g]zip/unzip 7za	文件压缩和解压

目录操作命令

命令	功能
ls	列出目录的属性信息
pwd	列出用户当前所处目录路径
du	列出目录占用磁盘空间大小
cd	改变用户的当前目录
mkdir	建立一个新目录
rmdir	删除一个目录及其中的文件
tree	显示目录树

文件操作命令

命令	功能
ls	列出文件的属性信息
find	在指定路径查找文件
whereis	在PATH环境变量指定路径查找文件
grep	在指定文件中查找指定的字符串
cat	连续显示、查看文件内容
more	分页查看文件内容
less	分页可控制查看文件内容

文件操作命令（Cont.）

- 复制、移动、删除文件

命令	功能
cp	复制文件或目录
mv	移动文件或目录
rm	删除文件或目录

文件概念及操作命令

文件显示命令

显示指定工作目录中所包含的内容的指令是ls，要说明的是
ls命令列出文件的名字，而不是文件的内容。ls [选项]
[文件目录列表]

ls命令中的常用选项如下：

- a**：显示所有文件及目录（ls规定将文件名或目录名中开头为“.”的视为隐藏档，不会列出）
- A**：同 -a，但不列出“.”（当前目录）及“..”（父目录）
- l**：除文件名外，也将文件状态、权限、拥有者、文件大小等信息详细列出
- R**：递归显示下层子目录
- F**：append indicator (one of */=@|) to entries
- color**：颜色显示

常见文件类型

- 目录
 - 蓝色
 - /
- 可执行文件
 - 绿色
 - *
- 符号链接
 - 青色
 - @

文件概念及操作命令

ls命令中的其他选项：

- c：按列输出，纵向排序
- x：按列输出，横向排序
- t：根据文件建立时间的先后次序列出
- X：按扩展名排序显示
- help：显示帮助信息
- version：显示版本信息

```
[root@localhost root]# ls
a.c          a.out  hello.c  install.log.syslog  z
anaconda-ks.cfg  hello  install.log  temp
```

```
[root@localhost root]# ls -a
.      .esd_auth      .gtkrc-1.2-gnome2  .recently-used
..     .fonts.cache-1  hello              .rhn-applet.conf
a.c    .gconf         hello.c            .tcshrc
anaconda-ks.cfg  .gconfd        .ICEauthority      temp
a.out   .gnome         install.log        .viminfo
.bash_history  .gnome2        install.log.syslog .vnc
.bash_logout  .gnome2_private .metacity          .Xauthority
.bash_profile .gnome-desktop .mozilla           .Xresources
.bashrc       .gstreamer     .nautilus          z
.cshrc        .gtkrc         .python
```

```
[root@localhost root]# ls -l
总用量 68
-rw-r--r--  1 root    root      54  3月 14 10:33 a.c
-rw-r--r--  1 root    root    1274 12月  9 16:58 anaconda-ks.cfg
-rwxr-xr-x  1 root    root   11530 3月 14 10:33 a.out
-rwxr-xr-x  1 root    root   11542 3月 21 10:33 hello
-rw-r--r--  1 root    root    60  3月 21 10:32 hello.c
-rwxrwxrwx  1 root    root   22005 12月  9 16:58 install.log
-rw-r--r--  1 root    root   3952 12月  9 16:57 install.log.syslog
drwxr-xr-x  2 root    root    4096 3月 21 10:26 temp
lrwxrwxrwx  1 root    root     3  3月 21 10:28 z -> a.c
```

```

[root@localhost root]# ls -F
a.c          a.out*  hello.c    install.log.syslog  z@
anaconda-ks.cfg hello*  install.log* temp/
[root@localhost root]# ls -R
.:
a.c          a.out  hello.c    install.log.syslog  z
anaconda-ks.cfg hello  install.
等价于ls -a -F

./temp:
ab.out.temp1
[root@localhost root]# ls -aF
./          .esd_auth      .gtkrc-1.2-gnome2  .recently-used
../         .fonts.cache-1 hello*              .rhn-applet.conf
a.c         .gconf/        hello.c            .tcshrc
anaconda-ks.cfg .gconfd/       .ICEauthority      temp/
a.out*      .gnome/        install.log*       .viminfo
.bash_history .gnome2/       install.log.syslog .vnc/
.bash_logout .gnome2_private/.metacity/
.bash_profile .gnome-desktop/.mozilla/
.bashrc       .gstreamer/    .nautilus/        z@
.cshrc        .gtkrc         .pyinput/

```

Linux支持多种文件类型，每一类用一个字符来表示，其说明如下：

Ø -：常规文件

Ø d：目录

Ø b：块特殊设备

Ø c：字符特殊设备

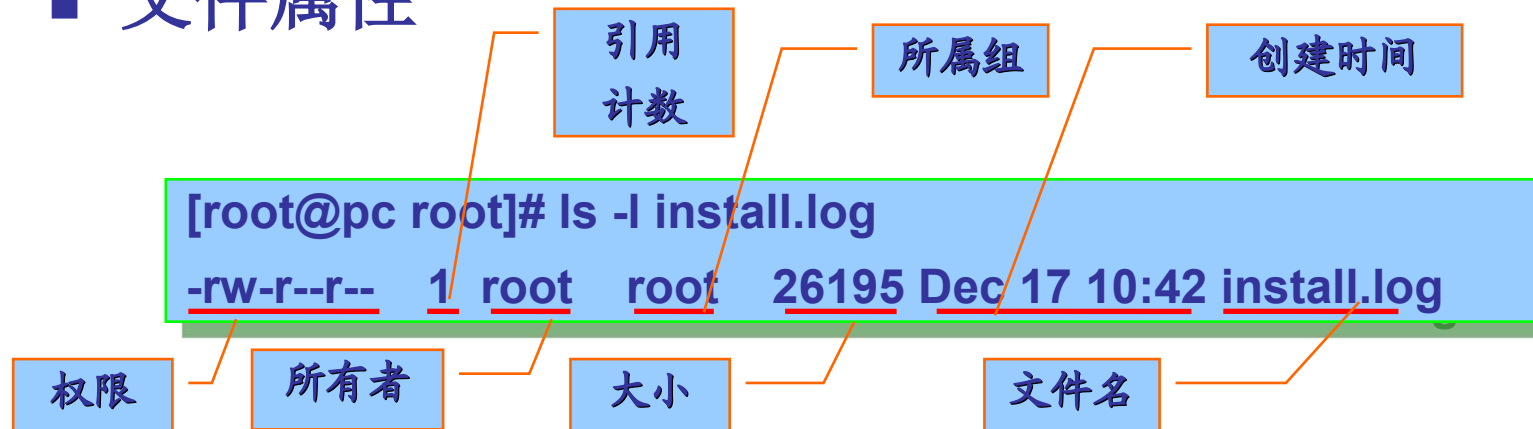
Ø p：有名管道

Ø s：信号灯

Ø m：共享存储器

文件操作命令（Cont.）

■ 文件属性



文件操作命令 (Cont.)

■ 文件权限 (-rw-r--r--)

权限项	文件类型	读	写	执行	读	写	执行	读	写	执行
字符表示	(d t c s)	(r)	(w)	(x)	(r)	(w)	(x)	(r)	(w)	(x)
数字表示		4	2	1	4	2	1	4	2	1
权限分配		文件所有者			文件所属组用户			其他用户		

例如：

`drwxr-x---`

表示的含义是：？

下面是使用命令来显示帮助和版本信息，命令如下：

```
#ls --version
```

```
#ls --help | more
```

改变文件或目录存取权限的命令

1. 改变文件或目录权限

在Linux系统中，用户设定文件权限控制其他用户不能访问、修改。

但在系统应用中，有时需要让其他用户使用某个原来其不能访问的文件或目录，这就需要重新设置文件的权限，使用的命令是chmod命令。

并不是谁都可改变文件和目录的访问权限，只有**文件和目录的所有者**才有权限修改其权限，另外**超级用户**可对所有文件或目录进行权限设置。

chmod命令的使用语法格式如下：

chmod [选项] 文件和目录列表

命令中的参数说明如下：

Ø 选项的常用取值如表2-7所示

Ø 文件和目录列表：准备修改权限的文件和目录

chmod命令的常用选项

常用选项	说 明
-c	只有在文件的权限确实改变时才进行详细说明。
-f	不打印权限不能改变的文件的错误信息。
-v	详细说明权限的变化。
-R	递归改变目录及其内容的权限。

chmod命令支持两种文件权限设定的方法：

(1) 使用字符串设置权限

(2) 使用八进制数设置权限

另外，文件和目录的权限还可用八进制数字模式来表示。三个八进制数字分别代表ugo的权限。执行权、读权和写权所对应的数值分别是1，2和4。

若要rwx属性则 $4+2+1=7$ ；

若要rw-属性则 $4+2=6$ ；

若要r-x属性则 $4+1=5$ 。

- 该命令用于改变或设置文件或目录的存取权限
- 根据表示权限的方式不同，该命令有两种用法：
- 以符号模式改变权限和以绝对方式改变权限。
- 1) a) 以符号模式改变权限
- 1) 一般格式 **chmod key 文件名**
- 2) 说明
- **key**有以下各项组成
- **[who] [操作符号] [mode]**
- 其中：**who** 是下列字母中的人一个或者他们的组合

- **u** 表示“用户”，即文件或目录的所有者
- **g** 表示“同组”，即文件属主有相同组**ID**的所有用户
- **o** 表示“其他用户”
- **a** 表示“所有用户”，其是系统默认值
- 操作符可以是
- **+** 添加某个权限
- **-** 取消某个权限
- **=** 赋予给定权限并取消其他所有权限

- **mode**所表示的权限可用下属字母的任意组合
- **r** 可读
- **w** 可写
- **x** 可执行
- 如**#chmod a+x lh** 将文件lh的权限改为所有用户都有执行
- 权限
- **2) 以绝对方式改变权限**
- **chmod mode 文件名**
- mode 是由3位八进制数字出现的，第一位表示文件
- 主权限，第二位表示组用户权限，第三位表示其他用户
- 权限

- 如#chmod 664 lh
 - 表示使文件lh的文件主和同组用户具有读写权限，但其他用户只可读。
- 例如，要将文件 **file1.txt** 设为所有人皆可读取
 - 命令：#chmod a+r file1.txt
- # chmod 777 file1.txt 的作用是什么？

```

[root@localhost root]# ls -l install.log
----- 1 root      root      22005 12月  9 16:58 install.log
[root@localhost root]# chmod a+r install.log
[root@localhost root]# ls -l install.log
-r--r--r-- 1 root      root      22005 12月  9 16:58 install.log
[root@localhost root]# chmod u+w install.log
[root@localhost root]# ls -l install.log
-rw-r--r-- 1 root      root      22005 12月  9 16:58 install.log
[root@localhost root]# chmod u-w install.log
[root@localhost root]# ls -l install.log
-r--r--r-- 1 root      root      22005 12月  9 16:58 install.log
[root@localhost root]# chmod 664 install.log
[root@localhost root]# ls -l install.log
-rw-rw-r-- 1 root      root      22005 12月  9 16:58 install.log
[root@localhost root]# chmod 777 install.log
[root@localhost root]# ls -l install.log
-rwxrwxrwx 1 root      root      22005 12月  9 16:58 install.log
[root@localhost root]# chmod 000 install.log
[root@localhost root]# ls -l install.log
----- 1 root      root      22005 12月  9 16:58 install.log

```

- **chown**

- 该命令用来改变文件或者子目录的所有权。用户可以被分配到不同的用户组，使用**chown**命令，可把所有权分配给不同的用户和组，允许他们对文件或子目录进行读、写或删除操作。
- 如**chown** 用户:用户组 文件

命令中的参数说明如下：

Ø 命令中的选择说明如表2-8所示。

Ø 用户：可能是用户号或文件/etc/passwd中出现的用户名。

文件和目录列表：用于重新指定所有权的用空格分隔的文件和目录列表。

表2-8 chown命令的常用选项

常用选项	说 明
-c	只有在文件的权限确实改变时才进行详细说明。
-f	不打印权限不能改变的文件的错误信息。
-v	详细说明权限的变化。
-R	递归改变目录及其内容的权限。

语法:

chown [-R] 帐号名称 档案或目录

chown [-R] 帐号名称:群组名称 档案或目录

范例:

```
[root@test root]# chown games test
```

```
[root@test root]# ls -l test
```

```
drw-r--r--  1 games  root    0 Jun 20 14:36 test
```

```
[root@test root]# chown -R root:root tmp
```

- 如果只想改变一个文件或目录的用户组的所有权，可用
- **chgrp**命令
- **chgrp** 组名 文件/目录

- 语法：
chgrp 群组名称档案或目录
- 范例：

```
[root@test root]# chgrp users test  
[root@test root]# ls -l
```

```
drw-r--r--  1 root  users    1 Jun 20 14:36 test
```

查找、排序及显示指定文件内容的命令

1. 查找文件

在Linux系统中查找文件的命令通常为find命令。find命令可在使用、管理Linux系统中方便地查找所需要的指定文件。

find命令的语法格式如下：

`#find [目录列表] [匹配标准]`

在命令格式中有两个参数，说明如下：

- Ø 目录列表：希望查询文件或文件集的目录列表，目录间用空格分隔。
- Ø 匹配标准：希望查询的文件的匹配标准或说明。详细的匹配标准如表2-2所示。

表达式	说 明
-name 文件	告诉 find 要找什么文件；要找的文件包括在引号中，可以使用通配符（*和?）
-perm 模式	匹配所有模式为指定数字型模式值的文件。不仅仅是读，写和执行，所有模式都必须匹配。如果在模式前是负号（-），表示采用除这个模式外的所有模式。
-type x	匹配所有类型为 x 的文件。 x 是 c （字符特殊）， b （块特殊）， d （目录）， p （有名管道）， l （符号连接）， s （套接文件）或 f （一般文件）。
-links n	匹配所有连接数为 n 的文件。
-user 用户号	匹配所有用户序列号是前面所指定的用户序列号的文件，可以是数字型的值或用户登录名。
-atime n	匹配所有在前 n 天内访问过的文件。
-mtime n	匹配所有在前 n 天内修改过的文件。
-newer 文件	匹配所有修改时间比 file 文件更新的文件。
-size n	匹配所有大小为 n 块的文件（ 512 字节块，若 k 在 n 后，则为 1K 字节块）。
-print	显示整个文件路径和名称。一般来说，都要用 -print ，如果没有这个参数，则 find 命令进行所要求的搜索是没有显示结果的。

1) 通过文件名查找

知道了某个文件的文件名，却不知道它存于哪个目录下，此时可通过查找命令找到该文件，命令

如下： `# find / -name httpd.conf -print`

2) 根据部分文件名查找

当要查找某个文件时，不知道该文件的全名，只知道这个文件包含几个特定的字母，此时用查找命令也是可找到相应文件的。这时在给字查找文件名时通配符“*”、“?”。例如，还是查找文件“httpd.conf”，但仅记得该文件名包含“http”字符串，可使用如下命令查找：

`#find / -name *http* -print`

3) 根据文件的特征查询

如果仅知道某个文件的大小、修改日期等特征也可使用find命令把该文件查找出来。例如，知道一个文件尺寸为2500bytes，可使用如下命令查找：

```
#find /etc -size -2500c -print
```

下面是以时间特征为查找条件的find命令函数的参数：

- Ø amin n 查找n分钟以前被访问过的所有文件.
- Ø atime n 查找n天以前被访问过的所有文件.
- Ø cmin n 查找n分钟以前文件状态被修改过的所有文件.
- Ø ctime n 查找n天以前文件状态被修改过的所有文件.
- Ø mmin n 查找n分钟以前文件内容被修改过的所有文件.
- Ø mtime n 查找n天以前文件内容被修改过的所有文件.


```
[root@localhost root]# find /root -name install.log -print
/root/install.log
/root/temp/install.log
[root@localhost root]#
```

显示文本文件内容

显示文本文件内容的命令是cat命令，用来将文件的内容显示到终端上，其命令语法如下：

cat [选项] 文件列表

命令中的参数说明如下：

Ø 常用选项如表2-3所示

Ø 文件列表：这是要连接文件的选项列表。如果没有指定文件或连字号（-），就从标准输入读取。

表2-3 cat命令常用选项

选 项	说 明
-b	计算所有非空输出行，开始为1。
-e	在每行末尾显示\$符号。
-n	计算所有输出行，开始为1。
-s	将相连的多个空行用单一空行代替。

3. 查找文件内容

查找文件内容的命令是grep命令。其使用的语法如下：

grep [选项] 匹配字符串 文件列表

命令中的参数说明如下：

选项如表2-4所示

Ø

文件列表：

匹配字符串：希望在文件中查到的串。

表2-4 grep命令的常用选项

选项	说明
-v	列出不匹配串或正则表达式的行。
-c	对匹配的行计数。
-l	只显示包含匹配的文件的文件名。
-h	抑制包含匹配文件的文件名的显示
-n	每个匹配行只按照相对的行号显示。
-i	产生不区分大小写的匹配，缺省状态是区分大小写。

```
[root@localhost root]# grep  ifdef test.txt -n  
8:#ifdef _DEBUG  
49:#ifdef _AFXDLL  
[root@localhost root]#
```

- **Question:** 如何只列出目录?

- `ls -l | grep ^d`
- `ls -F | grep /$`

```
[root@localhost root]# ls -l | grep ^d
drwxr-xr-x    2 root    root    4096  3月 21 10:26 temp
[root@localhost root]# ls -F | grep /$
temp/
[root@localhost root]#
```

4. 排序命令

sort命令的功能是对文件中的各行进行排序. **sort命令**可以被认为是**一个非常强大的数据管理工具**，用来管理内容类似数据库记录的文件。

Sort命令将逐行对文件中的内容进行排序，如果两行的首字符相同，该命令将继续比较这两行的下一字符，如果还相同，将继续进行比较。该命令的语法格式如下：

sort [选项] 文件

说明：**sort命令**对指定文件中所有的行进行排序，并将结果显示在标准输出上。如不指定输入文件或使用“-”，则表示排序内容来自标准输入。

比较文件内容的命令

1. comm命令

如果想对两个有序的文件进行比较，可以使用comm命令。该命令的使用语法如下：

```
comm [- 123 ] file1 file2
```

说明：该命令是对两个已经排好序的文件进行比较。其中file1和file2是已排序的文件。comm读取这两个文件，然后生成三列输出：仅在file1中出现的行；仅在file2中出现的行；在两个文件中都存在的行。如果文件名用“-”，则表示从标准输入读取。

2. diff命令

diff命令用于比较两个文件内容的不同，其命令语法格式如下：

diff [参数] 原文件 目标文件

其中“源文件”和“目标文件”是用户要比较的两个文件。

表2-4 diff命令的常用参数

参数	说明
a	将所有文件当做文本文件来处理
b	忽略空格造成的不同
B	忽略空行造成的不同
q	只报告什么地方不同，不报告具体的不同信息
H	利用试探法加速对大文件的搜索
i	忽略大小写的变化
l	用pr对输出进行分页
r	在比较目录时比较所有的子目录
s	两个文件相同时才报告
v	在标准输出上输出版本信息并退出

复制、删除和移动文件的命令

1. 文件复制命令

Linux下的cp命令用于复制文件或目录，该命令是最重要的文件操作命令，其命令格式如下：

```
cp [选项] 源文件 目标文件  
cp [选项] 源文件组 目标目录
```

参数说明如下：

- Ø 源文件：要拷贝的文件。
- Ø 目标文件：目标名。也可以是目录名，这种情况下，源文件名作为目标文件名，而文件放在该目录下。
- Ø 源文件组：要拷贝文件的由空格分隔的列表。
- Ø 目标目录：目标目录。

常用命令的选项列表如表2-5所示。

cp命令常用选项

常用选项	说 明
-a	在备份中保持尽可能多的源文件结构和属性。
-b	作将要覆盖或删除文件的备份。
-f	覆盖已存在的目标文件。
-i	提示是否覆盖已存在的目标文件。
-p	保持原先文件的所有者，组权限和时间标志
-r (-R)	递归拷贝目录

```
[root@localhost root]# cp install.log ins.txt
[root@localhost root]# ls ins.txt
ins.txt
[root@localhost root]# cp -i install.log ins.txt
cp: 是否覆盖 'ins.txt' ? y
[root@localhost root]# cp -f install.log ins.txt
[root@localhost root]# cp -r temp tmp
[root@localhost root]# ls temp
ab.out.tmp1  install.log
[root@localhost root]# ls tmp
ab.out.tmp1  install.log
[root@localhost root]# cp -rf temp tmp
[root@localhost root]#
```

2. 文件的删除命令

rm命令可从文件系统中删除文件及整个目录。rm所使用的命令格式如下：

rm [选项] 文件列表

参数说明如下：

Ø 常用命令的选项列表如表2-6所示。

Ø 文件列表：希望删除的用空格分隔的文件列表，可以包括目录名。

表2--6 rm命令常用选项

常用选项	说 明
-r	删除文件列表中指定的目录，若不用此标志则不删除目录
-i	指定交互模式。在执行删除前提示确认。任何以Y开始的响应都表示肯定；其他则表示否定。
-f	指定强行删除模式。通常，在删除文件权限可满足时rm提示。本标志强迫删除，不用提示。

- **-V** 在删除前回显文件名。
- **--** 指明所有选项结束。用于删除一个文件名与某一选项相同的文件。例如：假定偶然建立了名为-f的文件，又打算删除它，命令**rm -f**不起任何作用，因为-f被解释成标志而不是文件名；而命令**rm - -f**能成功地删除文件。
- Question: 如何删除特殊文件，例如：-f?
- 除了--参数，你还有什么方法？

3. 移动文件

在Linux系统中，移动文件可使用mv命令。mv命令还可改文件改名，即把源文件以一个新文件名移动到另一个新的目录中去。该命令所使用语法形式有：

```
mv [-f] [-i] 文件1 文件2
```

```
mv [-f] [-i] 目录1 目录2
```

```
mv [-f] [-i] 文件列表 目录
```

命令中的参数说明如下：

Ø -f: 通常情况下，目标文件存在但用户没有写权限时，mv会给出提示。本选项会使mv命令执行移动而不给出提示。

Ø -i: 交互模式，当移动的目录已存在同名的目标文件名时，用覆盖方式写文件，但在写入之前给出提示。

Ø 文件: 源文件名。

Ø 文件2: 目标文件名（新文件名）。

Ø 目录1: 源目录名。

目录2: 目标目录名（新目录名）。

Ø 文件列表: 用空格分隔的文件名列表。本选项用于文件保持它们的名称被移动到一个新目录。

Ø 目录: 目标目录。


```
[root@localhost root]# rm -i ins.txt
rm: 是否删除一般文件 'ins.txt'? n
[root@localhost root]# rm -f ins.txt
[root@localhost root]# rm -r tmp
rm: 是否进入目录 'tmp'? n
[root@localhost root]# rm -rf tmp
[root@localhost root]# ls tmp
ls: tmp: 没有那个文件或目录
[root@localhost root]# mv install.log install
[root@localhost root]# ls install.log
ls: install.log: 没有那个文件或目录
[root@localhost root]# ls install
install
[root@localhost root]# mv install ./temp/
[root@localhost root]# ls install
ls: install: 没有那个文件或目录
[root@localhost root]# ls ./temp/
ab.out.tmp1  install  install.log
[root@localhost root]#
```


文件内容统计命令

wc命令的功能为统计指定文件中的行数(line)、字数(word)、字节数(byte)、, 并将统计结果显示输出。**wc**命令使用的语法格式如下:

wc [选项] 文件列表

说明: 该命令统计给定文件中的字节数、字数、行数。如果没有给出文件名, 则从标准输入读取。**wc**同时也给出所有指定文件的总统计数。字是由空格字符区分的最大字符串。该命令各选项含义如下:

-c: 统计字节数。

Ø -l: 统计行数。

-w: 统计字数。

命令的输入和输出

在Linux系统中，执行一个shell命令行时通常会自动打开三个标准文件，即标准输入文件（stdin），通常对应终端的键盘；标准输出文件（stdout）和标准错误输出文件（stderr），这两个文件也都对应终端的屏幕。进程将从标准输入文件中得到输入数据，将正常输出数据输出到标准输出文件，而将错误信息送到标准错误文件中。

Linux系统为输入、输出的传送引入了另外两种机制，即输入/输出重定向和管道。

1. 输入重定向

输入重定向是指把命令（或可执行程序）的标准输入重定向到指定的文件中。输入重定向主要用于改变一个命令的输入源，特别是改变那些需要大量输入的输入源。

2. 输出重定向

输出重定向是指把命令（或可执行程序）的标准输出或标准错误输出重新定向到指定文件中。这样，该命令的输出就不显示在屏幕上，而是写入到指定文件中。

命令 > 文件名

命令 >> 文件名

■ 3. 管道

- 第一个命令的输出会作为第二个命令的输入通过管道传给第二个命令，第二个命令的输出又会作为第三个命令的输入。。。
- 命令1 | 命令2
- 不用管道的办法？临时文件

目录及其操作命令

- 树型目录结构
 - root
 - directory
 - file
- 跟Windows目录结构的区别

2. 工作目录、用户主目录与路径

(1) 工作目录与用户主目录

工作目录用“.”表示，其父目录用“..”表示。

用户主目录是系统管理员增加用户时建立起来的（以后也可以改变），每个用户都有自己的主目录，不同用户的主目录一般互不相同。用户刚登录到系统中时，其工作目录便是该用户主目录，通常与用户的登录名相同。

(2) 路径

路径是指从树型目录中的某个目录层次到某个文件的一条道路。此路径的主要构成是目录名称，中间用“/”分开。某个文件在文件系统中的位置都是由相应的路径决定的。

路径又分**相对路径**和**绝对路径**。绝对路径是指从“根”开始的路径，也称为完全路径；相对路径是从用户工作目录开始的路径。

应该注意到，在树型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。这一点对于以后使用某些命令如**cp**和**tar**等大有好处。

3. Linux系统主要目录说明

/bin：显而易见，**bin** 就是二进制（**binary**）英文缩写。

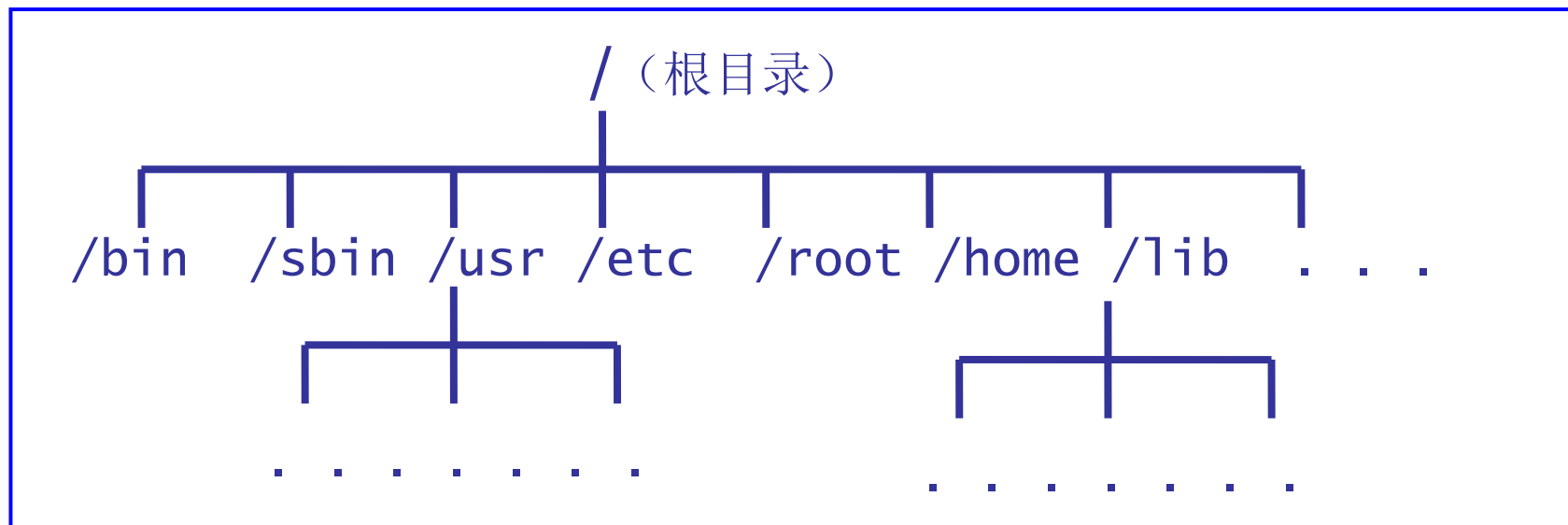
/boot：在这个目录下存放的都是系统启动时要用到的程序。在使用lilo引导linux的时候，会用到这里的一些信息。

/dev：dev 是设备（device）的英文缩写。这个目录对所有的用户都十分重要。因为在这个目录中包含了所有linux系统中使用的外部设备。但是这里并不是放的外部设备的驱动程序。

/etc：etc这个目录是linux系统中最重要目录之一。在这个目录下存放了系统管理时要用到的各种配置文件和子目录。我们要用到的网络配置文件，文件系统，x系统配置文件，设备配置信息，设置用户信息等都在这个目录下。

Linux 的目录结构

- **Linux** 文件系统是一个目录树的结构，文件系统结构从一个根目录开始，根目录下可以有任意多个文件和子目录，子目录中又可以有任意多个文件和子目录。
- **Linux** 的这种文件系统结构使得一个目录和它包含的文件/子目录之间形成一种层次关系。



/ 文件系统结构的起始位置，称为根

|-- bin 存放基本命令程序以调用

|-- boot 存放系统启动时所读取的文件

|-- dev 存放设备文件接口，如打印机、硬盘等

|-- etc 存放与系统设置和管理相关的文件，如用户帐号、密码等

| |...

|-- home 存放用户专属目录（用户主目录）

|-- lib 存放一些共享的函数库

|-- misc 一个空目录，供管理员存放公共杂物

|-- proc 存放系统核心和执行程序之间的信息

|-- root 系统管理员（超级用户）专用目录

|-- sbin 与/bin类似，存放用于系统引导和管理命令，通常供root使用。

|-- tmp 临时目录，供任何用户存放临时文件。

|-- usr 此目录包含许多子目录，用来存放系统命令和程序等信息

| |-- bin

| |-- sbin

| |-- man

| |...

|-- var 存放经常变动的文件，如日志文件、临时文件、电子邮件等

...

创建和删除目录的命令

1. 创建目录

在Linux系统中建立新目录的命令是mkdir。该命令的语法格式如下：

```
mkdir [-m模式] [-p目录名] 目录
```

命令中的参数说明如下：

m模式：在建立目录时把按模式指定设置目录权限。该目录的权限分为：目录所有者的权限、组中其他人对目录的权限和系统中其他人对目录的权限。这三个权限分别用三个数字之和来表示：对目录的读权限是4、写权限是2、执行权限是1。

-p 目录名：建立所有不存在父目录的目录。

目录：要建立的目录

2. 删除目录

`rmdir` 命令是用来删除目录的，一般情况下要删除的目录必须为空目录，如果所给的目录不为空，系统会报告错误。。该命令的语法格式如下：

```
rmdir -p 目录列表
```

命令中的参数说明如下：

- `-p`：在删除目录表指定的目录后，若父目录为空，则 `rmdir` 也删除父目录。状态信息显示什么被删除，什么没被删除。
- 目录列表：空格分隔的目录名列表。要删除的目录必须为空。

改变工作目录和显示目录内容的命令

1. 显示当前目录

显示当前目录的命令是**pwd**命令，该命令使用的语法格式如下：**pwd**

执行**pwd**命令形式如下：**# pwd**

- **2. 改变当前工作目录**
- **Linux**系统中改变当前工作目录在使用的命令是**cd**。该命令使用的语法格式如下：
- **cd [目录名]**
- 命令中的参数说明如下：
- 目录名：改变到所指定的目录名。如果没有指定目录，则返回到用户主目录（在**HOME**环境变量中指定）。
- **cd**命令还有以下几个使用技巧：
- 键入 **cd** 或**cd ~** 可进入用户的**home**目录；
- 键入 **cd -** 可进入上一个进入的目录；
- 键入 **cd /** 可进入根目录；
- 键入 **cd..** 可进入上一层目录。

链接文件的命令

链接文件命令是ln命令。该命令在文件之间创建链接。这种操作实际上是给系统中已有的某个文件指定另外一个可用于访问它的名称。

该命令的使用语法如下：

ln [选项] 目标 [链接名]

ln [选项] 目标 目录

链接有两种，一种被称为硬链接（Hard Link），另一种被称为符号链接（Symbolic Link）。建立硬链接时，链接文件和被链接文件必须位于同一个文件系统中，并且不能建立指向目录的硬链接。而对符号链接，则不存在这个问题。默认情况下，ln产生硬链接

链接文件的命令

■ 文件链接

■ 符号链接

不增加引用
计数

```
[root@pcl root]# ll x
-rw-r--r-- 1 root root 0 Jan 29 08:10 x
[root@pcl root]# ln -s x y
[root@pcl root]# ll x y
-rw-r--r-- 1 root root 0 Jan 29 08:10 x
lrwxrwxrwx 1 root root 1 Jan 29 08:10 y -> x
```

■ 硬链接

```
[root@pcl root]# ln x z
[root@pcl root]# ll x z
-rw-r--r-- 2 root root 0 Jan 29 08:10 x
-rw-r--r-- 2 root root 0 Jan 29 08:10 z
```

增加引
用计数

在硬链接的情况下，参数中的“目标”被链接至[链接名]。如果[链接名]是一个目录名，系统将在该目录之下建立一个或多个与“目标”同名的链接文件，链接文件和被链接文件的内容完全相同。如果[链接名]为一个文件，用户将被告知该文件已存在且不进行链接。如果指定了多个“目标”参数，那么最后一个参数必须为目录。

如果给ln命令加上-s选项，则建立符号链接。如果[链接名]已经存在但不是目录，将不做链接。[链接名]可以是任何一个文件名（可包含路径），也可以是一个目录，并且允许它与“目标”不在同一个文件系统中。如果[链接名]是一个已经存在的目录，系统将在该目录下建立一个或多个与“目标”同名的文件，此新建的文件实际上是指向原“目标”的符号链接文件。

例：\$ ln -s lunch /home/xu
用户为当前目录下的文件lunch创建了一个符号链接/home/xu。

硬链接和软链接的比较

■ 硬链接

- 链接文件和被链接文件必须位于同一个文件系统内
- 不能建立指向目录的硬链接

■ 软链接

- 链接文件和被链接文件可以位于不同文件系统
- 可以建立指向目录的软链接

文件内容查看命令

■ cat

- 合并文件命令**cat**用来把文件内容送到屏幕上显示。
- 也可一次查看多个文件内容。
- **-n** 在文件内容前面加上行号
- **Cat file1 file2 >file3** 把文件**file1**和**file2**合并到**file3**文件中
- **>**重定向操作符
- **Cat file1>>file2** 把文件**file1**放到**file2**后面

文件内容查看命令

■ more

- **Enter**键，只显示文本的下一行内容
- **H**键，显示帮助屏，该屏上有相关的帮助信息
- **B**键，显示上一屏内容
- **SPACE**，显示下一屏内容
- **Q**键，退出**MORE**命令

文件内容查看命令

■ less

- 也是页命令，但功能比**more**强，可以来回查看文件，
- 用**-M**参数可以显示行号或百分比；
- 在文档中，用**/***，可以实现文件的检索，
- 跟**more**的比较：
 - **more**的操作基本都可用
 - 支持上下光标键、**PageUp**、**PageDown**

文件内容查看命令

- **head**

- **-n** 定量显示文件的前n行，默认是前十行

- **tail**

- **-n** 定量显示文档的最后n行的内容，默认是最后10行

- **touch**

- **clear**

- **date**

- 功能：显示和设置日期时间

- 用法：

- **\$ date**

- **# date -s MM/DD/YYYY**

- **# date -s hh:mm:ss**

```
[root@localhost root]# date
六  4月  5 17:07:26 CST 2008
[root@localhost root]#
```

■ cal

- 功能：显示日历

- 用法：

- \$ cal

- \$ cal -y

- \$ cal year

- \$ cal month year

```
[root@localhost root]# cal
      四月 2008
日 一 二 三 四 五 六
      1  2  3  4  5
 6  7  8  9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30
```


■ **locale**

- 功能：显示当前语言环境
- 用法：
 - `$ locale`

■ **file**

- 功能：显示指定文件的类型
- 用法：
 - `$ file filename`

■ **stat**

- 功能：显示指定文件的各种相关信息
- 用法：
 - `$ stat filename`

■ dmesg

- 功能：显示系统启动信息
- 用法：
 - \$ dmesg

■ uname

- 功能：显示操作系统信息
- 用法：
 - \$ uname
 - \$ uname -r
 - \$ uname -a

```
[root@localhost root]# file install.log
install.log: UTF-8 Unicode text
[root@localhost root]# stat install.log
  File: 'install.log'
  Size: 22005          Blocks: 48          IO Block: 4096   Regular File
Device: 801h/2049d    Inode: 278530       Links: 1
Access: (0777/-rwxrwxrwx)  Uid: (  0/   root)   Gid: (  0/   root)
Access: 2008-04-05 17:11:27.000000000 +0800
Modify: 2007-12-09 16:58:06.000000000 +0800
Change: 2008-03-28 01:18:12.000000000 +0800
```

```
[root@localhost root]# uname
Linux
[root@localhost root]# uname -r
2.4.20-8
```