

一、shell命令

1、shell简介

1》命令

命令是用户向系统内核发出控制请求，与之交互的文本流。

2》shell

Shell是一个命令行解释器，将用户命令解析为操作系统所能理解的指令，实现用户与操作系统的交互。

2、shell格式与编辑

1》命令格式

通常一条命令包含三个要素：**命令名称、选项、参数**。命令名称是必须的，选项和参数都可能是可选项。命令格式如下所示：

\$：Shell提示符，如果当前用户为超级用户，提示符为“#”，其他用户的提示符均为“\$”；

Command：命令名称，Shell命令或程序，严格区分大小写，例如设置日期指令为date等；

Options：**命令选项**，用于改变命令执行动作的类型，由“-”引导，可以同时带有多**个选项**；

Argument：**命令参数**，指出命令作用的对象或目标，有的命令允许带多个参数。

2》命令行编辑

类别	组合键	含义	描述
移动	Ctrl+a	行首	到当前行首
	Ctrl+e	行尾	到当前行尾
	Ctrl+L	清屏	清除屏幕，并在屏幕顶端留下一行
剪切和粘贴	Ctrl+K	剪切行尾	剪切文到该行末尾
	Ctrl+U	剪切行首	剪切文到该行开头
	Ctrl+W	剪切前个单词	剪切光标前的一个单词
	Alt+D	剪切下个单词	剪切光标后的一个单词
	Ctrl+Y	粘贴最近的文本	粘贴最近剪切的文本
	Alt+Y	粘贴早期的文本	轮回到先前剪切的文本并粘贴它
	Ctrl+C	删除整行	删除一整行，中止当前命令

3》自动补齐：tab键，或者ESC键

在输入命令或文件名的**前几个字符**后，按TAB键或ESC键**自动补齐剩余没有输入的字符串**。

如果存在多个命令或文件有相同前缀，Shell将列出所有相同前缀的命令或文件。Shell给出的提示信息，帮助用户回忆和完成输入。之后等待用户输入足够的字符。

4》命令重命名

alias 给已有的命令取别名，例如：
alias c='clear'
unalias 取消命令别名，例如：
unalias c

3、shell特殊字符

1》通配符：

通配符	含义	实例
星号 (*)	匹配任意长度的字符串	用file_*.txt, 匹配file_wang.txt、file_Lee.txt、file3_Liu.txt
问号 (?)	匹配一个长度的字符	用file_?.txt, 匹配file_1.txt、file1_2.txt、file_3.txt
方括号 ([...])	匹配其中指定的一个字符	用file_[otr].txt, 匹配file_o.txt、file_r.txt和file_t.txt
方括号 ([-])	匹配指定的一个字符范围	用file_[a-z].txt, 匹配file_a.txt、file_b.txt, 直到file_z.txt
方括号 ([^ ...])	除了其中指定的字符, 均可匹配	用file_[^otr].txt, 除了file_o.txt、file_r.txt和file_t.txt其他文件

2》管道：|

命令A的输出 通过管道传递给 命名B
ls | wc -w //将ls的输出结构通过管道传给wc，统计ls输出的单词数并将其打印到屏幕。
3》输入输出重定向：

重定向符	含义	实例
> file	将file文件重定向为输出源，新建模式	ls /usr > Lsoutput.txt，将ls /usr的结果，写到Lsoutput.txt文件中，若文件存在则覆盖，不存在则创建，若文件将被删除
>> file	将file文件重定向为输出源，追加模式	ls /usr >> Lsoutput.txt，将ls /usr的结果，追加到Lsoutput.txt文件已有内容后面
< file	将file文件重定向为输入源	wc < file1，将file1中的内容作为输入给wc命令h
2> 或 &>	将由命令产生的错误信息输入到	ls noexistingfile.txt 2> err.log，使用ls命令，查看一个不存在的文件时，产生的错误提示保存在err.log文件中

在linux中：

存在三个标准设备文件：

0，标准输入 对应的是键盘
1，标准输出 对应的是屏幕
2，标准错误输出 对应的是屏幕

1) 标准输出重定向：> file
ls > 1.txt //将ls的正确输出结果写入到文件1. txt,如果文件1. txt不存在，则创建，存在则将其清空后写入
2) 标准输出重定向：>> file

ls > 1.txt //将ls的输出结果写入到文件1.txt, 如果文件1.txt不存在, 则创建, 存在则在文件末尾追加新的数据

3) 标准输入重定向: < file

wc < 1.txt //将wc的默认输入源重定向到文件1.txt, wc将会统计1.txt中的行数, 单词数和字符数。

4) 标准错误输出: 2> 或 &>

ls 2> 1.txt //将ls执行错误时的结果写入到文件1.txt, 如果文件1.txt不存在, 则创建, 存在则将其清空后写入

4、shell系统命令

1) man 找到特定的联机帮助页, 并提供简短的命令说明

通常联机帮助信息包括以下几个部分:

E: 命令的名称

OPSIS: 命令的常用参数

DESCRIPTION: 命令的一般描述以及用途

OPTIONS: 描述命令所有的参数或选项

ALSO: 列出联机帮助页中与该命令直接相关或功能相近的其他命令

S: 解释命令或其输出中存在的任何已知的问题或缺陷

EXAMPLES: 普通的用法示例

AUTHORS: 联机帮助页以及命令的作者

注意:

1) man可以获得shell命令的帮助, 也可以获得C函数的帮助, 例如:

man ls //获得命令ls的帮助信息

man scanf //获得函数scanf的帮助信息

2) man里面分不同的章节, 如下:

The table below shows the section numbers of the manual followed by the types of pages they contain.

- 1 Executable programs or shell commands
- 2 System calls (functions provided by the kernel)
- 3 Library calls (functions within program libraries)
- 4 Special files (usually found in /dev)
- 5 File formats and conventions eg /etc/passwd
- 6 Games
- 7 Miscellaneous (including macro packages and conventions), e.g. man(7), groff(7)
- 8 System administration commands (usually only for root)
- 9 Kernel routines [Non standard]

3) 使用:

```
peter@farsight:~/1604/linuxjc$ man printf
peter@farsight:~/1604/linuxjc$
peter@farsight:~/1604/linuxjc$ man 3 printf
peter@farsight:~/1604/linuxjc$
peter@farsight:~/1604/linuxjc$ man -f printf
printf (1)          - format and print data
printf (3)          - formatted output conversion
```

2) wall

wall命令的功能是对全部已登录的用户发送信息

3) write

write用于向系统中某一个用户发送消息

4) passwd: 修改密码

passwd peter //当前用户peter修改过自己的密码

`sudo passwd peter` //超级用户修改用户peter修改过自己的密码

5》su: 切换用户

普通用户切换到超级用户:

`sudo su`

超级用户切换到普通用户:

`su peter`

6》echo: 向屏幕打印字符串

`echo hello world` //将hello world 打印到屏幕上

`echo $PATH` //将环境变量PATH的值打印到屏幕上, 如下:

`/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/cross
tool/gcc-3.4.5-glibc-2.3.6/arm-linux-gnu/bin`

7》mount: 挂载某个分区或者镜像, 比如挂载u盘:

1) 找到设备名称

`cat /proc/partitions`

2) 设置挂载点

`mkdir mnt`

3) 挂载u盘的命令

`sudo mount /dev/sdb1 ./mnt`

4) 检查挂载是否成功:

`peter@farsight:~/1604/linuxjc$ ls mnt/
vimrc 课件`

8》umount: 卸载:

`sudo umount mnt`

5、用户管理:

1》用户属性:

用户名

口令

用户ID (UID)

用户主目录 (HOME)

用户shell

2》用户清单文件:

1) /etc/passwd

/etc/passwd文件是系统能够识别的用户清单, 用户登陆时, 系统查询这个文件, 确定用户的UID并验证用户口令

`peter:x:1000:1000:peter,,,:/home/peter:/bin/bash`

登陆名

经过加密的口令

UID

默认的GID

个人信息

主目录

登陆shell (/bin/bash) 自动启动shell

2) /etc/shadow

/etc/shadow文件是加密的用户清单, 只有超级用户可以访问这个文件, 用来保护加密口令的安全

`peter:6a94. TRKw$RMCCJ9aqToUqH. zmhNbenXT. aahMDWDiKrQ/pvpxcFm1dnj069odLC6CbMybZx3KkZ5SIi
fnsL2FL3gGL30L30:16906:0:99999:7:::`

登陆名

经过加密的口令

上次修改口令的时间

两次修改口令之间最少的天数

两次修改口令之间最多的天数

提前多少天警告用户口令即将过期
在口令过期之后多少天禁用账号
账号过期的日期
保留字段，目前为空

3》添加用户：adduser

```
peter@farsight:~/1604/linuxjc$ sudo adduser user
Adding user `user' ...
Adding new group `user' (1002) ...
Adding new user `user' (1002) with group `user' ...
Creating home directory `/home/user' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for user
Enter the new value, or press ENTER for the default
    Full Name []:
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n] y
peter@farsight:~/1604/linuxjc$
```

注意：此时上面添加的新用户不能使用sudo，所以需要在配置文件中添加该用户，

如下：

1) 打开配置文件：

```
sudo vi /etc/sudoers
```

2) 添加以下信息：

```
user    ALL=(ALL:ALL) ALL
```

3) 保存退出

4》删除用户：deluser

```
sudo deluser --remove-home user
```

```
peter@farsight:/home$ sudo deluser --remove-home user
Looking for files to backup/remove ...
Removing files ...
Removing user `user' ...
Warning: group `user' has no more members.
userdel: user user is currently logged in
/usr/sbin/deluser: `/usr/sbin/userdel user' returned error code 8. Exiting.
peter@farsight:/home$ ls
farsight  peter
```

5》修改主机名：

1) 修改文件：/etc/hosts

```
---》sudo vi /etc/hosts
```

```
---》将文件中的farsight改为ubuntu
```

```
:%s/farsight/ubuntu/g
```

2) 修改文件：/etc/hostname

---》 sudo vi /etc/hostname
---》 将文件中的farsight改为ubuntu

```
:%s/farsight/ubuntu/g
```

3) 重启系统即可

6》 修改用户名:

1) 修改文件: /etc/passwd

----》 sudo vim /etc/passwd

----》 将文件中的farsight改为user

```
:%s/farsight/user/a
```

2) 修改文件: /etc/shadow

----》 sudo vim /etc/shadow

----》 将文件中的farsight改为user

```
:%s/farsight/user/a
```

3) 修改文件: /etc/sudoers

----》 sudo vi /etc/sudoers

----》 将文件中的farsight改为user

```
:%s/farsight/user/a
```

4) 修改/home下的用户主目录:

sudo mv farsight/ user

5) 重启系统即可

6、进程管理:

1》 进程的概念:

程序的一次执行过程称为进程。

1) 进程控制块:PCB

进程名

特征信息

进程状态信息

调度优先权

通信信息

现场保护区

资源需求、分配和控制方面的信息

进程实体信息

族系关系

2) 进程与PCB的关系

---》 每个进程有唯一的PCB

---》 系统中所有进程都有自己的PCB

---》 操作系统依据PCB管理进程

---》 操作系统利用PCB实现进程的动态和并发

---》 PCB是进程存在的唯一标志

2》 进程相关命令:

1) 显示进程 (process) 的动态: ps

常见的参数:

-A 列出所有的行程

-w 显示加宽可以显示较多的资讯

-au 显示较详细的资讯

-aux 显示所有包含其他使用者的行程

例如:

```
ps -aux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START
TIME COMMAND
root         1  0.3  0.3   3592  2012 ?        Ss
23:11    0:01 /sbin/init
root         2  0.0  0.0      0     0 ?        S
```

23:11	0:00	[kthreadd]									
			root	3	0.0	0.0	0	0	?	S	
23:11	0:00	[ksoftirqd/0]									
			root	4	0.0	0.0	0	0	?	R	
23:11	0:00	[kworker/0:0]									
			root	6	0.0	0.0	0	0	?	S	
23:11	0:00	[migration/0]									
			root	7	0.0	0.0	0	0	?	S	
23:11	0:00	[watchdog/0]									

进程状态标志

- D: 不可中断的静止
- R: 正在执行中
- S: 阻塞状态
- T: 暂停执行
- Z: 不存在但暂时无法消除
- W: 没有足够的内存分页可分配
- <: 高优先级的进程
- N: 低优先级的进程
- L: 有内存分页分配并锁在内存中

2) 监视进程：top

```

top - 09:03:04 up 3:36, 1 user, load average: 0.00, 0.00, 0.00
Tasks: 48 total, 1 running, 47 sleeping, 0 stopped, 0 zombie
Cpu(s): 0.0% us, 1.7% sy, 0.0% ni, 98.3% id, 0.0% wa, 0.0% hi, 0.0% st
Mem: 189720k total, 124300k used, 65420k free, 10588k buffers
Swap: 915664k total, 0k used, 915664k free, 84832k cached

```

PID	USER	PR	NI	UIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
5816	root	16	0	2004	1012	804	R	1.3	0.5	0:00.14	top
1	root	16	0	1560	532	472	S	0.0	0.3	0:00.67	init
2	root	RT	0	0	0	0	S	0.0	0.0	0:00.00	migration/0
3	root	34	19	0	0	0	S	0.0	0.0	0:00.00	ksoftirqd/0
4	root	10	-5	0	0	0	S	0.0	0.0	0:06.11	events/0
5	root	10	-5	0	0	0	S	0.0	0.0	0:00.05	khelper
6	root	10	-5	0	0	0	S	0.0	0.0	0:00.00	kthread
8	root	20	-5	0	0	0	S	0.0	0.0	0:00.00	kacpid

3) 树状显示系统进程：pstree

```
peter@ubuntu:~$ pstree
init--NetworkManager--{NetworkManager}
    |--accounts-daemon--{accounts-daemon}
    |--acpid
    |--atd
    |--avahi-daemon--avahi-daemon
    |--bamfdaemon--2*[{bamfdaemon}]
    |--bluetoothd
    |--colord--2*[{colord}]
    |--console-kit-dae--64*[{console-kit-dae}]
    |--cron
    |--cupsd
    |--2*[dbus-daemon]
    |--dbus-launch
    |--dconf-service--2*[{dconf-service}]
    |--gconfd-2
    |--geoclue-master
    |--6*[getty]
```

4) 终止进程:kill

kill [-signal] PID

signal是信号，PID是进程号

kill 命令向指定的进程发出一个信号signal，在默认的情况下，kill 命令向指定进程发出信号15，正常情况下，将杀死那些不捕捉或不忽略这个信号的进程

例1：用默认方式杀死进程

杀死进程号为：2763的进程

```
peter      2705  0.0  0.8  35692  4060 ?        Sl    23:16   0:00 /usr/lib/dej
root       2716  0.0  0.0      0      0 ?        S     23:19   0:00 [kworker/0:1
peter      2763  5.1  0.0   1988    284 pts/1    S+    23:30   0:00 ./a.out
peter      2764  1.2  0.7   7200   3584 pts/3    Ss    23:30   0:00 bash
peter      2823  0.0  0.2   4928   1160 pts/3    R+    23:31   0:00 ps -aux
peter@ubuntu:~/1604/linuxjc$ kill 2763
```

例2：发送信号让进程自杀：

```
peter      2668  0.0  1.6  40972  8372 ?        Sl    23:14   0:00 gnome-screens
peter      2680  0.0  2.3  51500 11684 ?        Sl    23:15   0:00 update-notifi
peter      2705  0.0  0.8  35692  4060 ?        Sl    23:16   0:00 /usr/lib/deja
root       2716  0.0  0.0      0      0 ?        S     23:19   0:00 [kworker/0:1
peter      2764  0.1  0.7   7200   3584 pts/3    Ss    23:30   0:00 bash
peter      2824  5.3  0.0   1988    280 pts/1    S+    23:33   0:00 ./a.out
peter      2825  0.0  0.2   4928   1164 pts/3    R+    23:33   0:00 ps -aux
peter@ubuntu:~/1604/linuxjc$ kill -9 2824
```

二、文件系统

1、文件系统类型

1) 文件系统的概念：

文件系统用于组织和管理计算机存储设备上的大量文件，并提供用户交互接口。

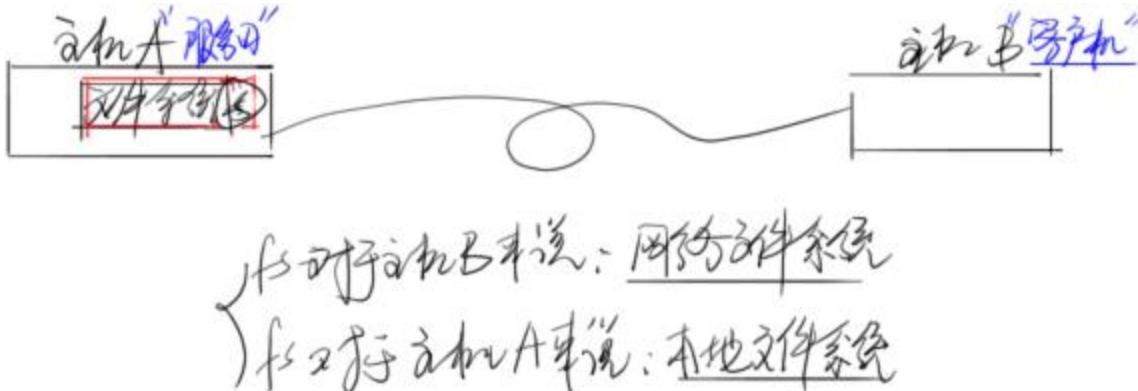
2) linux中文件系统的分类

--->磁盘文件系统：指本地主机中实际可以访问到的文件系统，包括硬盘、CD-ROM、DVD、USB存储器、磁盘阵列等。常见文件系统格式有：autofs、coda、Ext

(Extended File sytem, 扩展文件系统)、Ext2、Ext3、VFAT、ISO9660 (通常是CD-ROM)、UFS (Unix File System, Unix文件系统)、ReiserFS、XFS、JFS、FAT(File Allocation Table, 文件分配表)、FAT16、FAT32、NTFS (New Technology File System) 等；

--->网络文件系统：是可以远程访问的文件系统，这种文件系统在服务器端仍是本地的磁盘文件系统，客户机通过网络远程访问数据。常见文件系统格式有：NFS

(Network File System, 网络文件系统)、Samba (SMB/CIFS)、AFP (Apple Filling Protocol, Apple 文件归档协议) 和WebDAV等；



--->专有/虚拟文件系统：不驻留在磁盘上的文件系统。常见格式有：TMPFS (临时文件系统)、PROCFS (Process File System, 进程文件系统) 和LOOPBACKFS

(Loopback File System, 回送文件系统)。

3) SCSI与IDE设备命名

--->SCSI硬盘的设备名称是“/dev/sda

--->IDE硬盘的设备名称是“/dev/hda”

--->不同硬盘的表示：

前两个字母表示设备类型

“hd”代表IDE硬盘

“sd”表示SCSI硬盘

第三个字母说明具体的设备

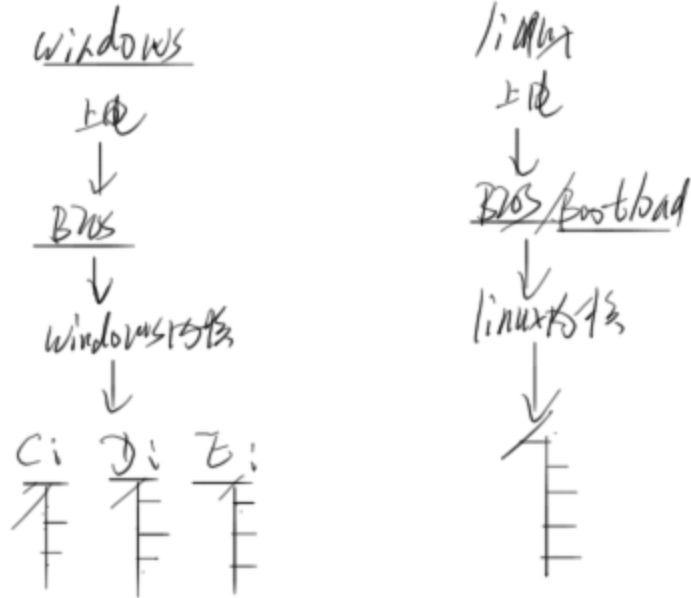
“/dev/hda”表示第一个IDE硬盘

“/dev/hdb”表示第二个IDE硬盘

2、文件系统结构

1) 概念：

Linux的文件组织模式犹如一颗倒置的树，这与Windows文件系统有很大差别。



从上图可以得出:




在Windows中目录从属于分区, 在linux中分区从属于目录
在Linux种文件名严格区分大小写, 但是在Windows中不区分。

linux中目录之间用正斜杠 '/' 间隔, Windows中用反斜杠 '\' 间隔

2) linux根目录下的子目录含义:

目录名	描述
/	Linux文件系统根目录
/bin	存放系统中最常用的可执行文件（二进制）
/boot	存放Linux内核和系统启动文件，包括Grub、lilo启动器程序
/dev	存放所有设备文件，包括硬盘、分区、键盘、鼠标、USB、tty等
/etc	存放系统的所有配置文件，例如passwd存放用户账户信息，hostname存放主机名等
/home	用户主目录的默认位置
/initrd	存放启动时挂载initrd.img映像文件的目录，以及载入所需设备模块的目录。
/lib	存放共享的库文件，包含许多被/bin和/sbin中程序使用的库文件
/lost+found	存放由fsck放置的零散文件
/media	Ubuntu系统自动挂载CD-ROM、软驱、USB存储器后，存放临时读入的文件
/mnt	该目录通常用于作为被挂载的文件系统的挂载点
/opt	作为可选文件和程序的存放目录，主要被第三方开发者用来简易地安装和卸载他们的软件包
/proc	存放所有标志为文件的进程，它们是通过进程号或其他的系统动态信息进行标识，例如cpuinfo文件存放CPU当前工作状态的数据
/root	根用户（超级用户）的主目录
/sbin	存放更多的可执行文件（二进制），包括系统管理、目录查询等关键命令文件
/srv	
/sys	
/tmp	存放用户和程序的临时文件，所有用户对该目录都有读写权限

/usr	用于存放与系统用户直接有关的文件和目录，例如应用程序及支持它们的库文件。以下罗列了/usr中部分重要的目录。	
	/usr/X11R6:	X Window系统
	/usr/bin	用户和管理员的标准命令
	/usr/include	c/c++等各种开发语言环境的标准include文件
	/usr/lib	应用程序及程序包的连接库
	/usr/local	系统管理员安装的应用程序目录
	/usr/local/share	系统管理员安装的共享文件
	/usr/sbin	用户和管理员的标准命令
	/usr/share	存放使用手册等共享文件的目录
	/usr/share/dict	存放词表的目录
	/usr/share/man	系统使用手册
	/usr/share/misc	一般数据
	/usr/share/sgml	SGML数据
	/usr/share/xml	XML数据
/var	通常用于存放长度可变的文件，例如日志文件和打印机文件。以下罗列了/var其中部分重要的目录。	
	/var/cache	应用程序缓存目录
	/var/crash	系统错误信息
	/var/games	游戏数据
	/var/lib	各种状态数据
	/var/lock	文件锁定纪录
	/var/log	日志记录
	/var/mail	电子邮件
	/var/opt	/opt目录的变量数据
	/var/run	进程的标示数据
	/var/spool	存放电子邮件，打印任务等的队列目录。
	/var/tmp	临时文件目录

	 Linux文件系统	 Windows文件系统
文件格式	使用的主要文件格式有：EXT2、EXT3、RerserFS、ISO9660、vfat等	使用的主要文件格式有：FAT16、FAT32、NTFS等
存储结构	逻辑结构犹如一颗倒置的树。将每个硬件设备视为一个文件，置于树形的文件系统层次结构中。因此，Linux系统的某一个文件就可能占有一块硬盘，甚至是远端设备，用户访问时非常自然	逻辑结构犹如多颗树（森林）。将若干个分区，与存储设备一起（例如USB存储器等），使用驱动器盘符来代表软驱、C：代表硬盘中的第一个分区
与硬盘分区的关系	分区在目录结构中	目录结构在分区中
文件命名	Linux文件系统中严格区分大小写，MyFile.txt与myfile.txt指不同的文件。区分文件类型不依赖于文件后缀，可以使用程序file命令判断文件类型。	windows文件系统中不区分大小写，MyFile.txt与myfile.txt是指同一个文件。使用文件后缀来标识文件类型。例如使用“.txt”表示文本文件
路径分隔符	Linux使用斜杠“/”分隔目录名，例如“/home/usr/share”，其中第一个斜杠是根目录（/），绝对路径都是以根目录作为起点	Windows使用反斜杠“\”分隔目录名，例如“C:\program\username”，绝对路径都是以驱动器盘符作为起点
文件与目录权限	Linux最初的定位是多用户的操作系统，因而有完善文件授权机制，所有的文件和目录都有相应的访问权限	Windows最初的定位是单用户的操作系统，系统时没有文件权限的概念，后期逐渐增加了这方面的功能

3、文件系统导航

1) 相关命令：

---》查看当前位置：pwd

---》查看目录中的文件：ls

----》只查看文件名，且不显示隐藏文件：

```
peter@ubuntu:~/1604/linuxjc$ ls
1.txt 2.txt 3.txt 4.txt 5.txt hello.c Hello.c mnt
```

----》只查看文件名，且显示隐藏文件：

隐藏文件：以.开始的文件名，该文件称为隐藏文件，例如：

如：.hello.c

```
peter@ubuntu:~/1604/linuxjc$ ls -a
. . . 1.txt 2.txt 3.txt 4.txt 5.txt .hello.c Hello.c mnt
```

----》查看文件的详细信息：

```
peter@ubuntu:~/1604/linuxjc$ ls -l
total 4
-rw-r--r-- 1 peter lpf      0 Apr 14 19:21 1.txt
-rw-r--r-- 1 peter lpf      0 Apr 14 19:04 2.txt
-rw-r--r-- 1 peter lpf      0 Apr 14 19:04 3.txt
-rw-r--r-- 1 peter lpf      0 Apr 14 19:04 4.txt
-rw-r--r-- 1 peter lpf      0 Apr 14 19:04 5.txt
-rw-r--r-- 1 peter lpf      0 Apr 15 00:10 Hello.c
drwxr-xr-x 2 peter lpf 4096 Apr 14 20:17 mnt
```

----》文件的属性信息：

-rw-r--r--

1

peter

lpf

0

Apr 15 00:10

Hello.c

权限和权限

↑

文件所有者

↑

大小

↑

时间

↑

文件名

硬链接数

↑

文件所属组

----》linux中文件的类型：有7中类型

文件类型	类型符	描述
普通文件	-	指ASCII文本文件、二进制可执行文件，以及硬件链接
块设备文件	b	块输入/输出设备文件
字符设备文件	c	原始输入/输出设备文件,每次I/O操作仅传送一个字符
目录文件	d	包含若干文件或子目录
符号链接文件	l	只保留了文件地址，而不是文件本身
命名管道	p	一种进程间通信的机制,用于进程间通讯的暂时文件
套接字	s	用于进程间通信,两台计算机间通讯的应用程序接口(API)

所以在linux中有句非常著名的话：一切皆文件

----》linux中文件的权限：

文件分为三种权限：可读，可写，可执行
对于不同的使用者权限可以不同，使用者分为三种：文

件拥有者，所属组，其他用户

<u>文件类型</u>	<u>所有者权限</u>	<u>所属组权限</u>	<u>其他用户权限</u>
	<u>-rw-</u>	<u>r--</u>	<u>r--</u>
	<u>110</u>	<u>100</u>	<u>100</u>
	<u>6</u>	<u>4</u>	<u>4</u> (二进制制)
	<u>Hello.c 的权限: 0644</u>		

r: 可读
 w: 可写
 x: 可执行
 -: 不可读
 -: 不可写
 -: 不可执行

----》如何修改权限: chmod

```

peter@ubuntu:~/1604/linuxjc$ ls -l Hello.c
-rw-r--r-- 1 peter lpf 0 Apr 15 00:10 Hello.c
peter@ubuntu:~/1604/linuxjc$ sudo chmod 0777 Hello.c
peter@ubuntu:~/1604/linuxjc$ ls -l Hello.c
-rwxrwxrwx 1 peter lpf 0 Apr 15 00:10 Hello.c
peter@ubuntu:~/1604/linuxjc$ sudo chmod 0664 Hello.c
peter@ubuntu:~/1604/linuxjc$ ls -l Hello.c
-rw-rw-r-- 1 peter lpf 0 Apr 15 00:10 Hello.c
peter@ubuntu:~/1604/linuxjc$
  
```

----》如何修改文件所有者和所属组: chown

```

peter@ubuntu:~/1604/linuxjc$ ls -l Hello.c
-rw-rw-r-- 1 peter lpf 0 Apr 15 00:10 Hello.c
peter@ubuntu:~/1604/linuxjc$ sudo chown root:lpf Hello.c
peter@ubuntu:~/1604/linuxjc$ ls -l Hello.c
-rw-rw-r-- 1 root lpf 0 Apr 15 00:10 Hello.c
peter@ubuntu:~/1604/linuxjc$ sudo chown root:root Hello.c
peter@ubuntu:~/1604/linuxjc$ ls -l Hello.c
-rw-rw-r-- 1 root root 0 Apr 15 00:10 Hello.c
peter@ubuntu:~/1604/linuxjc$ vi Hello.c
  
```

4、文件资源管理

- 1》创建文件: touch
- 2》查看文件属性: file:

```

peter@ubuntu:~/1604/linuxjc$ file hello
hello: ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), dynamically linked (uses shared libs), for GNU/Linux 2.6.24, BuildID[sha1]=0x09ae583738ad82e1182c1b79db8fc3, stripped
peter@ubuntu:~/1604/linuxjc$
  
```

- 3》复制文件或者目录: cp

选项	描述
-a	该选项通常在拷贝目录时使用，保留链接、文件属性，并递归地拷贝目录，其作用等于cp-r选项的组合
-d	拷贝时保留链接
-f	非交互模式的复制操作。覆盖已经存在的目标文件而不提示
-i	交互模式的复制操作，与-f选项相反。在覆盖目标文件之前将给出提示，要求用户确认，回答“y”表示目标文件将被覆盖
-p	除复制源文件内容外，还将其修改时间和访问权限也同时复制
-r	若源文件是一个目录，cp命令将递归地复制该目录下的所有子目录和文件
-l	不做拷贝，只是链接文件
-v	显示详细的复制过程

- 4》文件或者目录重命名：mv
5》移动文件或者目录到指定位置：mv
6》删除文件或者目录：rm

选项	描述
-f	非交互模式的删除操作，忽略不存在的文件
-i	交互模式的删除操作。在删除目标文件之前将给出提示，要求用户确认，回答“y”表示目标文件将被删除
-r	若源文件是一个目录，rm命令将递归地删除该目录下的所有子目录和文件
-v	显示详细的删除过程

- 7》创建目录：mkdir

5、文件内容显示

- 1》cat：用于依次读取指定文件的内容，并在标准输出设备上显示

选项	描述
-v	用一种特殊形式显示控制字符，LFD与TAB除外
-T	将TAB显示为“^T”，只与“-v”选项同时使用
-E	在每行的末尾显示一个“\$”符，只与“-v”选项同时使用
-u	输出不经过缓冲区

2》more

more命令用于在终端屏幕按屏显示文本文件。该命令依次显示一屏文本，显示满之后停下来，并在终端底部显示“--more--”，系统还将同时显示出已显示文本占全文的百分比。若要继续显示，按Enter键为一行一行显示，按空格键为一屏一屏显示

3》less

less命令的功能几乎与more命令相同，也是用于在终端屏幕按屏显示文本文件。该命令依次显示一屏文本。不同之处在于less命令在显示文本时，允许用户使用上下方向键前后翻阅文件。需要说明的是，使用less命令进入显示状态后，按“q”或“Q”退出显示状态。

4》head

如果用户通过查看文件前几行，就可以大体知晓文件内容，就不必浏览整个文件。head命令用于显示文件或标准输入的前几行。

5》tail

与head相反，tail命令用于显示文件或标准输入的最后几行

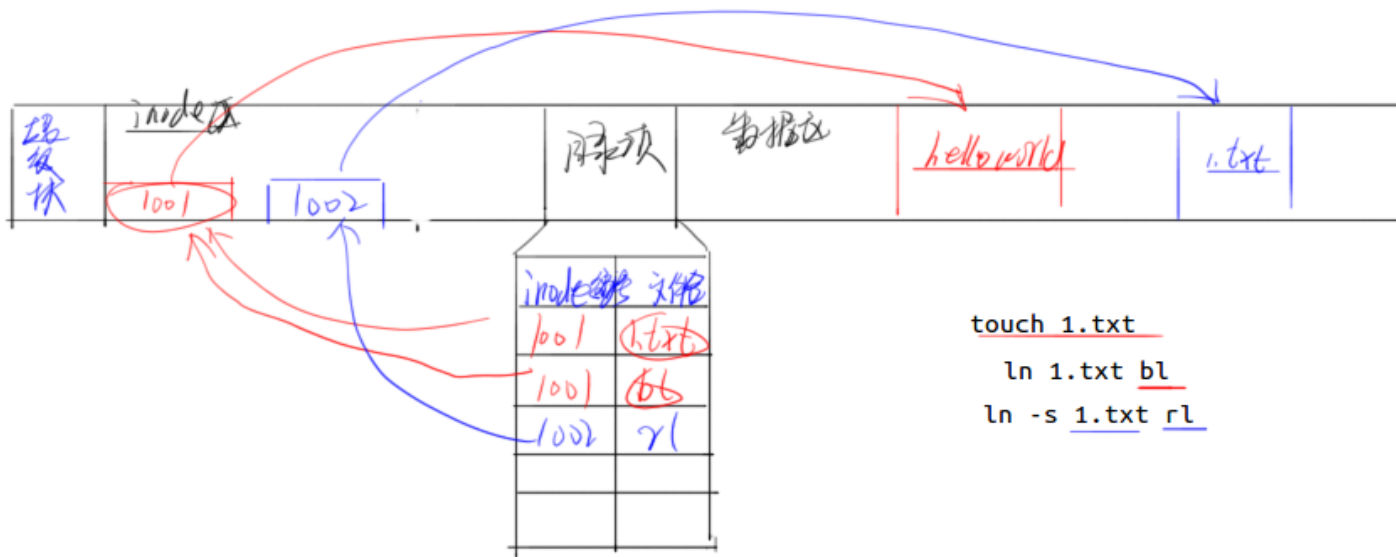
6、链接文件创建

1》概念：

链接文件就类似于微软Windows的快捷方式，只保留目标文件的地址，而不占用存储空间。使用链接文件与使用目标文件的效果是一样的。为链接文件指定不同的访问权限，以控制对文件的共享和安全性的问题。

2》在linux中链接的分类

- 1) 硬链接是利用Linux中为每个文件分配的物理编号——inode建立链接。因此，硬链接不能跨越文件系统。
- 2) 软链接（符号链接）是利用文件的路径名建立链接。通常建立软链接使用绝对路径而不是相对路径，以最大限度增加可移植性。



```
touch 1.txt
ln 1.txt bl
ln -s 1.txt rl
```

3》硬链接创建: ln

```
peter@ubuntu:~/1604/linuxjc$ ln 1.txt bl
peter@ubuntu:~/1604/linuxjc$ ls -l 1.txt bl
-rw-r--r-- 2 peter lpf 12 Apr 15 01:47 1.txt
-rw-r--r-- 2 peter lpf 12 Apr 15 01:47 bl
```

4》创建软链接: ln -s

```
ln -s /home/peter/1604/linuxjc/1.txt rrl
lrwxrwxrwx 1 peter lpf 30 Apr 15 01:57 rrl ->
/home/peter/1604/linuxjc/1.txt
```

```
peter@ubuntu:~/1604/linuxjc$ ln -s 1.txt rl
peter@ubuntu:~/1604/linuxjc$ ls -l 1.txt bl rl
-rw-r--r-- 2 peter lpf 12 Apr 15 01:47 1.txt
-rw-r--r-- 2 peter lpf 12 Apr 15 01:47 bl
lrwxrwxrwx 1 peter lpf 5 Apr 15 01:51 rl -> 1.txt
```

7、文件压缩与打包