



第3章 Linux文件管理操作





本章内容

- Linux的文件概念，文件类型
- 目录的概念，创建、删除等操作
- 讨论如何显示文件的内容
- 解释文件的复制、追加、转移/重命名和删除
- 描述如何确定文件大小
- 讨论比较文件的命令
- 描述如何合并文件
- 命令：cat、more、cd、echo、file、ls、mkdir、mv、nl、pwd、rmdir、cp、rm、tac、diff、head、less、uniq和wc





本章内容

- 讨论正则表达式的构成和使用;
- 解释文件压缩, 解释如何进行压缩文件;
- 解释排序过程、解释如何对文件排序;
- 讨论在Linux文件结构中查找命令和文件的方法;
- 表述在文件中查找表达式, 字符串和模式;
- Linux与Windows文件格式转化
- 涉及到的命令和原语有: gzip, gunzip, sort, find, egrep, fgrep, greg, tar, rpm, cut, paste, history, dos2unix, uinx2dos





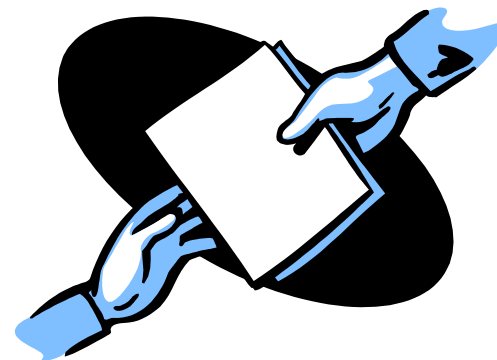
3.1 Linux的文件概念、文件类型、文件系统





Linux 文件概念

- Linux中, 文件是一个字节序列
- Linux 把所有东西看作文件来处理, 如:
 - *A simple file (text/executable)* 普通文件
 - *A directory*
 - *A keyboard*
 - *A printer*
 - *A disk drive*
 - *A network interface card, etc..*





Linux 文件类型

- Linux操作系统文件分为：普通文件、目录文件、字符设备文件、块设备文件、符号链接文件、管道文件、socket文件
- 普通文件
 - 文件名最长不能超过255个字符
 - 可以用除保留字符以外的任何字符给文件命名
 - 强烈建议不要使用非打印字符、空白字符（空格和制表符）和shell 命令保留字符
 - 扩展名对LINUX系统来说没有任何意义
 - 可以任意给文件名加上你自己或应用程序定义的扩展名 (e.g. .c file extension is required by C compilers)
- 目录文件：是文件系统中一个目录所包含的目录项组成的文件。目录文件只允许系统进行修改。用户进程可以读取目录文件，但不能对它们进行修改。两个特殊的目录项“.”代表目录本身，“..”表示父目录。





Linux 文件类型(cont.)

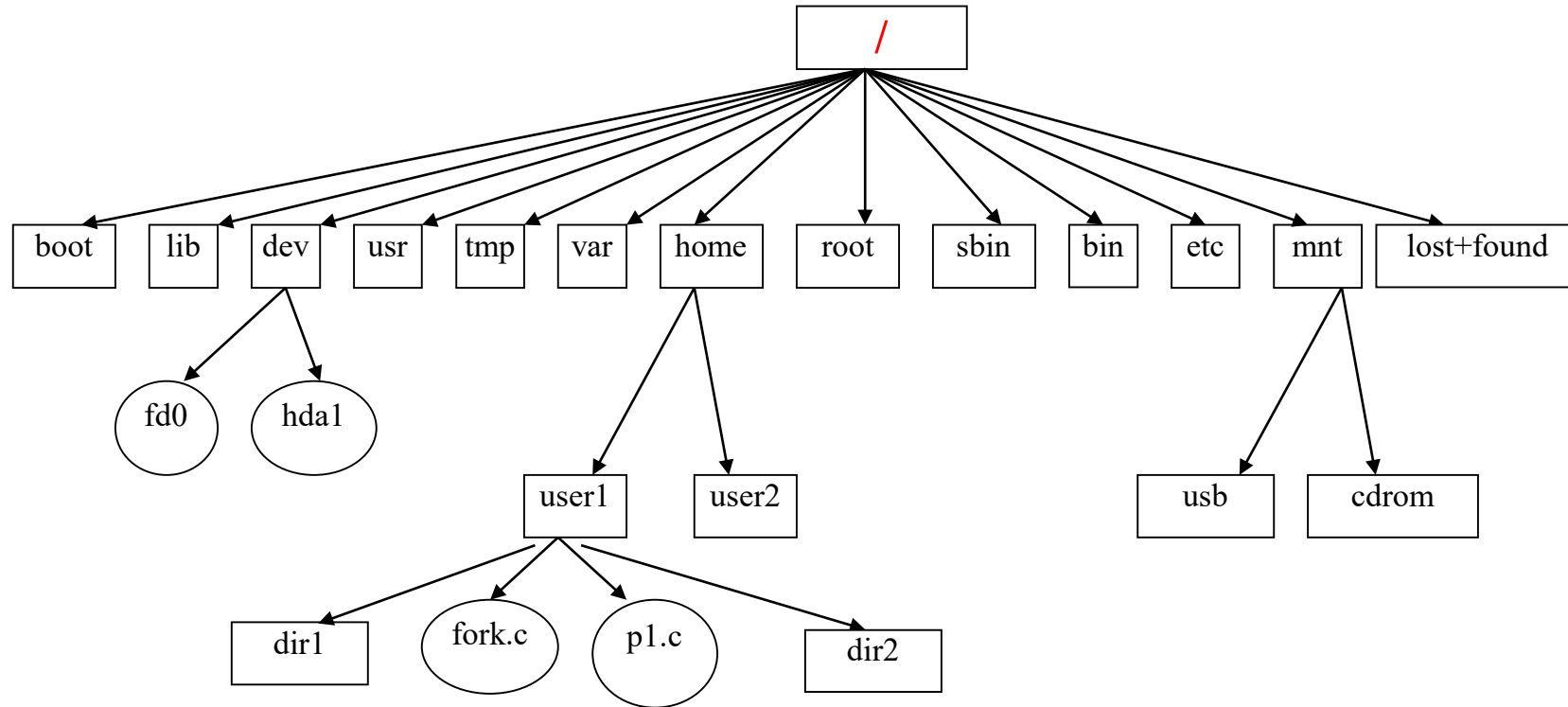
- **设备文件**：字符设备文件和块设备文件。Linux把对设备的I/O作为对文件的读取/写入操作内核提供了对设备处理和对文件处理的统一接口。
 - *sda* (SCSI硬盘, USB硬盘)
 - *hda* (for harddisk a)
 - *lp0* (for line printer 0)
 - *tty*(for teletype terminal)
- **链接文件**：又称符号链接文件，它提供了共享文件的一种方法。
- **管道(FIFO)文件**：用于在进程间传递数据。Linux对管道的操作与文件操作相同，它把管道做为文件进行处理。（创建管道文件**mkfifo**）
- **Socket（套接字）文件**

设备文件没有文件长度，它有两个值分别是主设备号和次设备号；两者共同构成一个唯一的号码，内核根据它找到对应的设备驱动程序





文件系统结构File System Structure



Linux文件系统结构





一些标准目录和文件

Linux系统主要目录说明

- **根目录 (/)**：根目录位于分层文件系统的最顶层，用斜线 (/) 表示。它包含一些标准文件和目录，因此可以说它包含了所有的目录和文件。
- **/bin**：也称二进制 (binary) 目录，包含了那些供系统管理员和普通用户使用的重要的Linux命令的可执行文件。一些常用的命令有：bash、cat、chmod、cp、date、echo、kill、ln、mail、mkdir、more、mv、ps、pwd、rm、rmdir、sh、stty、su、tcsh、uname和vi。一些用于系统恢复的命令如：tar、gzip、gunzip和zcat。还有一些网络命令如：domainname、hostname、netstat和ping。目录/usr/bin下存放了大部分的用户命令。
- **/boot**：在这个目录下存放系统启动时要用到的程序。包括Linux内核的二进制映像。内核文件名是vmlinuz加上版本和发布信息。
- **/dev**：dev 是设备 (device) 的英文缩写。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系统最重要的目录之一。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。





一些标准目录和文件(cont.)

- **/sbin** : 这个目录是用来存放系统管理员的系统管理程序。
- **/home** : 存放用户的主目录。如果建立一个用户, 用户名是“ji”, 那么在/home目录下就有一个对应的/home/ji路径, 用来存放用户的主目录。
- **/lib** : lib是库(library)英文缩写。这个目录是用来存放系统动态连接共享库的。几乎所有的应用程序都会用到这个目录下的共享库。
- **/media** 媒体目录, 可移动媒体设备的常用挂载点
- **/mnt** : 这个目录主要用来临时装载文件系统, 系统管理员运行mount命令完成装载工作。
- **/opt** : 该目录用来安附加软件包
- **/proc** : 目录存放了进程和系统得信息, 可以在这个目录下获取系统信息。这些信息是在内存中, 由系统自己产生的。
- **/root** : 根(root)用户的主目录。如果用户是以超级用户的身份登录的, 这个就是超级用户的主目录。





一些标准目录和文件(cont.)

- **/sbin, /usr/sbin, /usr/root/sbin**: 存放了系统管理的工具、应用软件和通用的root用户权限的命令。
- **/tmp**: 用来存放不同程序执行时产生的临时文件。
- **/usr**: 是linux文件系统中最大的目录之一。它存放了可以在不同主机间共享的只读数据。
- **/lost+found**: 目录中存放所有和其他目录没有关联的文件, 这些文件可以用Linux工具fsck查找得到。
- **/var**: 用来存放易变的数据, 这些数据在系统运行过程中会不断变化。/var/spool/mail 存放收到的电子邮件, /var/log 存放系统的日志, /var/ftp 。





文件系统结构

■ 主目录(登录目录) Home Directories

- When you logon, you are placed in a special directory, called your home directory (To see enter “**echo \$HOME**”)
- 字符 “~”来表示你的主目录

■ 当前工作目录 (当前目录) Present Working Directories

- 当前目录 . (dot)
- 父目录 .. (dot dot)

■ 绝对路径 Absolute Pathnames

- 从根目录开始的路径
- Examples: /home/faculty/ji/courses/ee446/exams/mid1

■ 相对路径 Relative Pathnames

- 从当前工作目录或用户主目录开始的路径
- Examples: exams/mid1

什么是当前目录?





文件系统概念

什么是文件系统？



- **文件系统**是操作系统中以文件方式管理计算机软件资源的软件和被管理的文件和数据结构（如目录和索引表等）的集合。





File System Types

- **ext** (minix文件系统)
- **ext2、ext3、ext4** (linux文件系统、Android)
- **FAT** (MS-DOS文件系统)
- **FAT32** (VFAT) (win98文件系统)
- **NTFS** (NT文件系统)
- **S51K/S52K** (AT&T UNIX sysv)
- **proc、sysfs** (linux虚拟文件系统)
- **yaffs** (Yet Another Flash File System)
- **ReiserFS** (Linux一种日志文件系统)
- **HFS+** (Mac OS , iOS文件系统)
- **HPFS** (OS/2高性能文件系统)
- **UFS**(BSD UNIX的一种文件系统)





File System Types

- **iso9660** (通用的光盘文件系统)

- **NFS** (网络文件系统)

- **VFS** (Linux虚拟文件系统)

VFS是物理文件系统与服务之间的一个接口，它屏蔽各类文件系统的差异，给用户和程序提供一个统一的接口

- **ZFS** (Open Solaris文件系统)

- **LTFS**(Liner Tape File System, 线性磁带文件系统) 为磁带提供了一种通用、开放的文件系统。

- **APFS** (Apple File System) 苹果新一代的文件系统，MAC os、iOS

-





3.2 文件系统命令





文件系统挂载mount

- Linux所有文件系统都挂载在根目录 (/) 下的某个子目录中
- 系统已经启动了X-windows图形用户界面时，光盘、软盘、U盘或移动硬盘的挂载是自动完成，称为即插即用的。
- Linux 使用 **mount** 命令挂载文件系统，
通常需要有 **root** 权限

现在较少使用这个命令

- 例： `mount -t vfat /dev/sda5 /mnt/c`

文件系统
类型

设备文件

挂载目录

Linux/Unix
环境下没
有盘符概
念

- `mount -a` 命令挂载在/etc/fstab中列出的所有设备





mount

■ 命令语法:

- **mount** [-t fstype] [-o options] device dirname

■ 常用参数:

- **fstype**: 文件系统类型, 如:
 - ▶ **iso9660** cd-rom使用的标准文件系统
 - ▶ **vfat** WINDOWS操作系统的fat32文件系统
 - ▶ **ntfs** WINDOWS的NTFS文件系统
 - ▶ **msdos** MS-DOS的fat文件系统






mount

■ **device**: 设备文件, 格式: `/dev/xxyN`

- `/dev` 保存所有设备文件的目录
- **xx** 设备类型, 如IDE硬盘为**hd**、SCSI硬盘和usb盘为**sd**、软盘为 **fd**
- **y** 同种设备的顺序号, 如第一个硬盘为**a**
- **N** 同一个设备编号, 如硬盘的第一个分区为1, 硬盘1-4为前面四个主分区, 5开始为逻辑分区。
- 常用设备文件名称:
 - ▶ `/dev/hda1` (第一个硬盘的第一个分区)
 - ▶ `/dev/hda5`、`/dev/hda6` (逻辑分区)
 - ▶ `/dev/fd0` (软盘)
 - ▶ `/dev/hdc` (光盘)
 - ▶ `/dev/sda1` (通常为移动硬盘或U盘的第一个分区)

V2.6.19后的内核使用**sd**



Linux/Unix
每个设备都
有一个文件
名与之对应

□ 设备文件没有文件长度, 而增加了另外的两个值, 分别是主设备号和从设备号。二者共同形成一个唯一的号码, 内核可由此查找对应的设备驱动程序。





mount

- **dirname**: 挂载目录，可以挂载在mnt，也可以挂载在你的主目录下，fedora core 6挂载在media目录下。如：
 - 软驱: /mnt/floppy
 - 光驱: /mnt/cdrom
- **options**: 设备或文件的挂接方式。常用的参数有：
 - loop: 把一个文件当成块设备挂接（环回设备）
 - ro: 采用只读方式挂接设备
 - rw: 采用读写方式挂接设备
 - iocharset: 指定访问文件系统所用字符集





挂载 Linux 分区

- `mount -t ext3 /dev/hda8 /mnt/linuxext`
- 这里假设 /dev/hda8 是 ext3 的 Linux分区





挂载 Windows 分区

- Fedora core 会自动检测你电脑上可能安装的 Windows 分区 (FAT32,NTFS), 然后在桌面上放上Windows 分区的小图标。除此之外, ntfs-3g 被默认安装, 这意味你对 NTFS 直接有读/写权限。

- 假设 /dev/hda5 是 Windows 下的 fat32 分区
(通常为 D: 盘)

mount -t vfat /dev/hda5 /mnt/win

- 假设 /dev/hda6 是 Windows 下的 NTFS 分区
(通常为 E: 盘)

mount -t ntfs /dev/hda6 /mnt/ntfs

注意: 挂载 ntfs 分区需要下载并安装与你机器上linux的 内核版本相一致的 linux-ntfs 模块或 ntfs-3g。





文件系统的卸载

- 使用 **umount** 命令卸载文件系统

umount 设备名或挂载点

umount /dev/hda5

或

umount /dosd





磁盘管理命令

■ **df** : 查看系统中文件系统的空间占用情况

● 命令语法: **df** -[选项]

● 参数选项:

- **a** 显示所有的文件系统
- **k** 以k字节为单位
- **t** 显示指定类型文件系统
- **l** 仅显示本地的文件系统
- **i** 显示inode的使用量而非块的使用量
- **x** 显示指定类型之外文件系统的情况
- **t** 显示文件系统类型
- **h** 用人们习惯方式显示, 输出包括: 已经使用了的空间量、可用空间、空间使用的百分比, 以及每个磁盘连接到系统的挂载点

例:

df -h

df -i

df -lh

df -h -t ext3

df -h -x ext3





磁盘管理命令

\$ df -h

文件系统	容量	已用	可用	已用%	挂载点
C:/cygwin64	204G	146G	59G	72%	/
D:	252G	192G	61G	77%	/cygdrive/d





磁盘管理命令du

■ du : 统计目录或文件占用空间

- 命令语法: `du -[选项] 目录名`

- 参数选项:

- s 只显示总数
- a 递归地显示目录及其子目录各文件的大小
- b 以字节为单位列出磁盘空间的使用情况
- k 以k字节为单位列出磁盘空间的使用情况
- c 最后加一总计
- h 用人们习惯方式显示

■ 例:

`du`

`du -h`

`du -sc /root /home`

```
$ du -h
12K    ./linux-4.13.3/arch/alpha/boot/tools
73K    ./linux-4.13.3/arch/alpha/boot
550K   ./linux-4.13.3/arch/alpha/include/asm
159K   ./linux-4.13.3/arch/alpha/include/uapi/asm
```





free命令

- **free**: 显示内存使用状态。
 - 命令语法: **free** [参数]
 - 参数选项: 常用参数:
 - ▶ **-b**: 以Byte为单位显示内存使用情况。
 - ▶ **-k**: 以KB为单位显示内存使用情况。
 - ▶ **-m**: 以MB为单位显示内存使用情况。
 - ▶ **-s <sec>**: 持续观察内存使用状况, <sec>为时间间隔秒数。
 - ▶ **-t**: 显示内存总和。
- 例:
 - free**
 - free -mt**
 - free -s 5**





fdisk命令

- **fdisk**: 查看硬盘分区情况及对硬盘进行分区管理

fdisk -l





3.3 文件系统操作





文件结构导航

■ echo命令

- 语法: **echo [options] [string]**
- 常用选项/功能:
 - ▶ -E 不解析转义字符
 - ▶ -e 解析转义字符, 常用的转义字符有:
 - \c 回车不换行
 - \t 制表符
 - \\ 反斜线
 - ▶ -n 不输出行尾的换行符



■ 显示主目录的绝对路径

- **echo \$HOME**
- **pwd** /*若当前目录为主目录*/
- **echo ~** /* “~” 表示主目录*/





文件结构导航

■ 改变当前目录 cd命令

语法: **cd [directory]**

- **cd** [directy]
- **cd** 不加任何参数, 返回到主目录
- **cd -** 目录之间切换

```
ji@ji-virtual-machine:~$ cd /dev
```

```
ji@ji-virtual-machine:/dev$ cd
```

```
ji@ji-virtual-machine:~$ cd -  
/dev
```

```
ji@ji-virtual-machine:/dev$ cd -  
/home/ji
```

```
ji@ji-virtual-machine:~$
```





文件结构导航(cont.)

ls命令:

- 语法: `ls [options [pathname-list]]`
- 用途: 显示指定目录下内容。
- 常用选项 (共有40多个选项):
 - `-a` 显示所有文件及子目录 (ls内定将文件名或目录名称开头为 “.”的视为隐藏, 不会列出)
 - `-l` 除文件名称外, 亦将文件类型、权限、拥有者、文件大小等详细列出
 - `-r` 将文件以相反次序显示(原定依英文字母次序)
 - `-t` 将文件依建立时间之先后次序列出
 - `-A` 同 `-a`, 但不列出 “.”(当前目录) 及 “..”(父目录)
 - `-F` 在列出的文件名称后加一符号; 例如可执行文件则加 “*”, 目录则加 “/”
 - `-R` 若目录下有文件, 则以下的文件亦皆依序列出
- 例:
 - 列出目前工作目录下所有名称是 s 开头的文件, 越新的排越后面:
`ls -ltr s*`
 - 将 /bin 目录以下所有子目录及文件详细信息列出:
`ls -lR /bin`
 - 列出目前工作目录下所有文件及子目录; 子目录于名称后加 “/”, 可执行文件名称后加 “*”:
`ls -AF`





文件结构导航(cont.)

- **ls**命令的**-l**选项显示详细信息，包括访问权限、连接数、所有者、组、文件大小（以字节计）和修改时间
- **-rwxr--r-- 1 user1 root 163 Apr 11 14:34 mid2**，含义如下：

表 命令ls -l的输出（字段按从左到右排列）	
字段	含义
第一个字段的第一个字母	文件类型： - 普通文件 d 目录(文件) b 块设备文件 c 字符设备文件 l 符号连接文件 p 命名管道（FIFO）
第一个字段的其它字母	所有者、组和其它用户的访问权限
第二个字段	链接数
第三个字段	所有者的登录名
第四个字段	所有者的组名
第五个字段	文件大小，以字节为单位
第六，七，八字段	最近一次修改的日期、时间
第九个字段	文件名





文件结构导航(cont.)

- 命令ls加上-i选项可以确定文件的inode号

- `ls -i`

12329 courses 22876 memos 12487 personal

- 命令ls -al显示目录下的所有文件的详细列表。如下所示

- `ls -al ~/courses/ee446/exams`

```
drwxr-x--- 1 user1 root 512 Mar 16 08:24 .
drwxr-x--- 1 user1 root 512 Jan 29 13:27 ..
-rwxr---r-- 1 user1 root 163 Mar 16 11:10 mid1
-rwxr---r-- 1 user1 root 163 Apr 11 14:34 mid2
drwxrwxrwx 1 user1 root 163 May 12 23:44 solutions
```





文件结构导航(cont.)

■ 显示指定目录的内容 ls

ls -a (to display all files)

ls -l (to display in long format)

ls -i (to display the inode of a file or dir)

ls -F (to display the file types)

■ ls 命令中使用元字符:

ls -l course*

ls -l ???.c

ls -l {pa,st}?.c

ls -l ~/courses/ee446/lab[!5]*.c

ls -i [a-zA-Z]??[1-5].html

ls ~/ [!0-9]*.[cC]

以pa或st开头，后跟一个字符，
且以.c结尾的所有文件

不以5字母开头

不以数字开头





stat命令

■ stat命令列出元数据信息

■ stat test.c

文件: "test.c"

大小: 145 块: 8 IO 块: 4096 普通文件

设备: 801h/2049d Inode: 690767 硬链接: 1

权限: (0644/-rw-r--r--) Uid: (0/ root) Gid: (0/ root)

最近访问: 2018-11-21 13:02:25.609288710 +0800

最近更改: 2018-11-13 18:09:27.315490413 +0800

最近改动: 2018-11-13 18:09:27.323490413 +0800

创建时间: -





文件结构导航(cont.)

创建目录 mkdir

- mkdir可以建立目录同时还可以给目录设置权限。
- 语法: **mkdir [options] dirnames**
 - 创建“dirname”制定的目录
 - -p 若所要建立目录的上层目录目前尚未建立, 则创建父目录
 - -m 建立目录时, 同时设置目录的权限。权限的设置法与chmod 指令相同
- 建立目录m02,并让全部人都有rwx的权限

ls

m01

mkdir -m 777 m02

ls

m01 m02





文件结构导航(cont.)

- 建立/home/m03/dir1 目录:

ls

m01 m02

mkdir -p /home/m03/dir1

ls

m01 m02 m03

cd m03

ls

dir1





文件结构导航(cont.)

删除目录rmdir

- 当有**空目录**要删除时，可使用rmdir指令。若所给予的目录非空目录，则会出现错误信息。
- 语法： **rmdir [options] dirnames**
 - -p 删除指定目录之后，若该目录的上层目录已变成空目录，则将其一并删除





文件结构导航(cont.)

- 目录m03下只有dir1目录，在删除dir1的同时也删除m03

```
rm -rf team03/dir1
```

```
ls
```

```
m01 m02
```





文件结构导航(cont.)

建立文件touch

- 使用 touch命令来建立新的空文件，touch命令也可用来将文件的访问时间或修改时间设置为当前时间或指定时间。
- 语法：touch [options] file-list

touch sample.h

- 若文件不存在，则建立一个空的文件
- 若文件已存在，则修改文件日期、时间等信息为当前日期、时间

显示文件内容类型命令

- 语法：file [options] file-list

file *





3.4 查看文件内容





查看文本文件的内容

- Linux提供了多个命令用来在显示屏上显示文本文件的内容。cat、tac、nl、pr、more、head、tail等
- cat命令

语法:	cat [options] [file-list]		
用途:	在标准输出（缺省情况下为显示屏）上连接、显示文件列表file-list里的文件		
输出:	显示屏上显示出来的文件内容，一次一个文件		
常用选项/功能:	-E	在每一行的末尾显示符号\$	
	-n	显示每一行的行号	
	--help	显示这个命令的用途，简要解释每一个选项的作用	

cat student_records

```
John Doe ECE 3.54
Pam Meyer CS 3.61
Jim Davis CS 2.71
Jason Kim ECE 3.97
Amy Nash ECE 2.38
```





查看文本文件的内容cat（续）

- 下面这个命令显示目录~/courses/ee446/labs下的两个文件lab1和lab2。在显示了文件lab1后，命令并不中止，而是马上接下去显示文件lab2。

```
cat ~/courses/ee446/labs/lab1 ~/courses/ee446/labs/lab2
```

```
[ contents of lab1 and lab2 ]
```

```
$
```

- 可以结合多个文件，并将它们的内容输出到标准输出设备。





查看文本文件的内容cat（续）

- 让cat指令从标准输入设备（如键盘）读取数据，转而输出至标准输出设备（如显示器）

cat 执行指令，不加任何参数

123 键入任何文字后，按下回车键

123 系统回应一模一样的文字

- 利用特殊字符“>”将名称为file1与file2的文件合并成一个文件file3:

cat file1 file2 > file3

- 若文件file3已经存在，则其内容会被覆盖过去；欲避免这种状况发生，可用“>>”代替“>”，新的内容就会附加在原有内容之后，而不会覆盖它。





查看文本文件的内容

■ 命令 **tac** (把命令 **cat** 倒过来) 来逆序显示一个文件。

■ **nl [options] [file-list]**

- 显示文本文件的内容, 同时显示行号

■ **nl student_records** //与 **cat -n student_records** 等价

1	John	Doe	ECE	3.54
2	Pam	Meyer	CS	3.61
3	Jim	Davis	CS	2.71
4	Jason	Kim	ECE	3.97
5	Amy	Nash	ECE	2.38





查看文本文件的内容 more

- 命令 **more** 可将文件内容显示于屏幕上，每次只显示一页。可以往下浏览，但无法向上浏览，**less** 指令可以上下浏览。

语法:	more [options] [file-list]		
用途:	在标准输出上连接、分页显示文件列表 file-list 里的文件		
输出:	显示屏上显示文件内容，每次一页		
常用选项/功能:	+/str	从包含 str 那行的前两行开始显示	
	-nN	每屏/页显示 N 行	
	+N	从第 N 行开始显示文件内容	





查看文本文件的内容 more（续）

- 在文件file1中查找“123”字符串，然后从该页开始显示文件的内容：

`more +/123 file1`

- 显示文件file1的内容，每10行显示一次，而且在显示之前先清屏。

`more -c -10 file1`





查看文本文件的内容less

- **less**命令类似 **more**, 效率更高, 使用vi的很多命令

语法:	less [options] [file-list]
用途:	分页显示文件列表 file-list 中的文件
输出:	列表 file-list 中文件的内容
常用选项/功能:	<ul style="list-style-type: none">-N 显示行号-o file 当命令输入是管道的时候, 除了显示屏, 还把输出写到文件file里; 如果这个文件已经存在, 会提示是否覆盖。-p pattern 在文件中查找匹配 “pattern”的第一处位置。





查看文本文件的内容

■ 查看文件的头和尾命令：head和tail

语法：	head [options] [file-list]
用途：	显示文件列表file-list中的文件的起始部分（文件头）；默认显示10行。
输出：	显示屏上显示的文件列表file-list中的文件的头部。
常用选项/功能：	-N 显示开始的N行

语法：	tail [options] [file-list]
用途：	显示文件列表file-list中的文件的尾部；默认显示10行。
输出：	显示屏上显示的file-list中文件的尾部。
常用选项/功能：	-f 显示完文件的最后一行后，如果文件正在被追加，会继续显示追加的行，直到键入<Ctrl-C>。 -n N 若N前加“+”号表示显示从文件第N行开始的所有行；否则显示文件的最后N行





查看文件的头head

■ 例：

head -5 sample

[first 5 lines of sample]

tail -5 otto

8 f3

() show

Pagelevel restore

Showpage

%%EOF

```
$ head sample
```

Ann

Ben

Chen

David

Erik

Fahim

George

Hamid

Ira

Jamal

```
$ tail sample
```

Queen

Rashid

Srini

Tang

Ursula

Vinny

Wang

X-Window-System

Yen

Zen





3.5文件的复制、转移、删除、大小





文件的复制、转移和删除

1、文件的复制命令cp

■ 语法: **cp** [options] *file1 file2*

- 复制文件 *file1*到*file2*; 如果*file2*是一个目录, 则把 *file1*复制到目录下

常用选项

- a 在备份中保持尽可能多的源文件结构和属性。
- d 拷贝时保留链接。
- f 删除已存在的目标文件不提示。
- 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 递归复制目录, 把所有非目录文件当普通文件复制。
- R 递归复制目录。





文件的复制、转移和删除

例：

■ `cp -i exam1.c /usr/s01/y1.c`

- 该命令将文件exam1.c拷贝到/usr/s01这个目录下，并改名为 y1.c。
提示是否覆盖已存在的目标文件

■ 若不希望重新命名，可以使用下面的命令：

- `cp exam1.c /usr/`

■ `cp -r /usr/s01/ /usr/s02/`

- 将/usr/s01目录中的所有文件及其子目录拷贝到目录/usr/s02中。





文件的复制、转移和删除

2、文件转移命令mv

- 可以使用mv命令来为文件或目录改名或将文件由一个目录移入另一个目录中。

- **mv [options] file1 file2**

- 重命名文件或移动文件

- **mv [options] file-list directory**

- 把文件列表 file-list中的所有文件移动到目录 directory下

- 常用选项

- -i 在覆盖目的文件前提示用户
 - -f 强制转移，覆盖某已有的目标文件时不给任何指示

- 例：将文件edc.txt重命名为fork.c

```
$ mv edc.txt fork.c
```

- 例:将/usr/s01中的所有文件移到当前目录（用“.”表示）中

```
$ mv /usr/s01/ * .
```





文件的复制、转移和删除

3、文件删除命令rm

■ **rm [options] file-list**

删除文件列表file-list中的文件

- -f 强制删除
- -i 删除前提示
- -r 递归删除

■ 例：当前目录下删除文件 temp

```
$ rm temp
```

■ 例：删除当前目录下子目录backups中的文件temp.old

```
$ rm backups/temp.old
```

■ 例：强制删除文件phones、grades和~/letters/letter.john

```
$ rm -f phones grades ~/letters/letter.john
```





文件的大小

4、文件大小确定

■ `ls -l file-list`

● 显示格式:

- ▶ Access permissions,
- ▶ File owner
- ▶ File owner group
- ▶ **File size**
- ▶ Date and time created or last modified

■ `ls -l lab2`

```
-r-xr--r-- 1 ji faculty 1063 may 16 23:46 lab2
```

文件lab2的字节数





文件的大小

■ `wc [options] file-list`

- 显示文件大小，可显示行数、单词数和字符数（lines, words, and characters）
- 常用选项：
 - c 只显示字符数
 - l 只显示行数
 - w 只显示单词数

■ `wc letter sample test`

44 250 1687 letter

4 44 227 sample

2 12 90 test

50 306 2004 total

Line count

Word count

Byte count

File name





文件的追加 (Appending to files)

在文件的末尾添加新的数据

■ `cat [file-list] >> destination-file`

- 把文件列表 *file-list* 中的所有文件的内容追加到文件 *destination-file* 中
- 如果不带参数 *file-list*, 通过键盘输入到文件 *destination-file* 中
- 如果 *destination-file* 不存在, 则自动创建





文件的合并

- `cat [file-list] > destination-file`
 - 合并文件列表file-list中的文件到destination-file中
 - 如果destination-file不存在,则自动创建
 - 如果destination-file已存在,它被覆盖
- `cat memo1 memo2 memo3>memo.new`





文件的比较

- 命令 **diff** 比较两个文件，按照某种可以把其中一个文件转化成另一个的命令的形式来显示这两个文件之间的区别。

语法:	diff [options] [file1] [file2]
用途:	一行一行地比较文件 file1 和文件 file2 ，以某种命令的形式显示它们之间的区别 这种命令可以用来把文件 file1 转化成 file2 ，或把 file2 转化成 file1 。如果用 “-” 代替 file1 或 file2 ，将从标准输入读取输入
常用选项/功能:	-b 忽略行尾的空白（包括空格和制表符），把空白字符串当作相同的字符串 -e 生成并且显示一个脚本，编辑器 ed 可以用它来把文件 file1 转化成 file2 -h 快速比较（此时不能使用 -e 选项）

- 如果被比较的两个文件内容相同，**diff** 不会产生任何输出。
- 若使用 **diff** 的时候不带任何选项，它会生成一系列指令，如果两个文件不同的话就可以使用这些指令把文件 **file1** 转化成 **file2**。这些指令是 **a**（增加）、**c**（替换）和 **d**（删除），如表所示。





文件的比较

表	diff生成的文件转化指令
指令	描述
L1aL2,L3 > lines L2 through L3	在file1第L1行后添加file2的L2行到L3行
L1,L2cL3,L4 < lines L1 through L2 in file1 ... > lines L3 through L4 in file2	把file1中的L1行到L2行替换成file2中的L3行到L4行
L1,L2dL3 < lines L1 through L2 in file1	删除file1中的L1行到L2行

■ 命令 **zdiff** 和 **zcmp** 可以用来比较两个压缩文件





删除重复行

■ `uniq [options] [+N][input-file] [output-file]`

- 删除已经排序好的文件input-file中的所有重复行，并把处理后不重复的行输出到output-file中
- 如果未指定output-file，输出到标准输出设备上（显示器）
- 如果未指定input-file，从标准输入设备中输入
- 常用选项
 - ▶ -c 在每行之前显示他们出现的次数
 - ▶ -d 只显示重复行
 - ▶ -u 只显示未重复行





3.6 文件的压缩、排序、查找





正则表达式

- **正则表达式**：它通过一系列的规则，用一个字符串来匹配多个字符串
- 支持正则表达式最常用的工具：
 - awk、ed、egrep、grep、sed 和vi
 - 解释型的程序设计语言:python...





正则表达式(cont.)

表	正则表达式操作符以及Linux工具对它们的支持			
名称	操作符	使用举例	意义	支持的工具
析取		x y z	x,y或者z	awk, egrep
任意单个字符	.	/L..e/	Love, Live, Lose, ...	所有工具
字符串首字符	^	^x	以字符x开始的串	所有工具
连接		xyz	xyz	所有工具
字符串尾字符	\$	x\$	以字符x结束的串	所有工具
转义字符	\	*	*	ed, sed, vi
组合	() 或者 \()	(xy)+	xy, xyxy, xyxyxy, ...	所有工具
可选	?	xy?	x, xy	awk, egrep
重复 (0次或多次)	*	xy*	x, xy, xyy, xyyy, ...	所有工具
重复 (1次或多次)	+	xy+	xy, xyy, xyyy, ...	所有工具
集合	[] [^]	/[Hh]ello/ /[^A-KM-Z]ove/	Hello, hello Love	所有工具





正则表达式(cont.)

正则表达式示例和含义		
正则表达式	意义	
$\{n\}$	匹配 n 次	
$\{n,\}$	匹配 n 次或 n 次以上	
$\{n,m\}$	匹配次数在 n 到 m 之间，包括 n 和 m	





文件压缩

- Linux 操作系统不仅有一些压缩、解压缩命令，而且还支持对压缩文件的多种操作。
- 这些命令中包括：
 - UNIX系统中的文件压缩命令和文件解压缩命令：zip、unzip、compress、uncompress。
 - GNU下的工具gzip、bzip2、7-zip、Lrzip、PeaZip、gzexe, zcat, zcmp, zforce, zmore 和 zgrep。

压缩文件工具	解压缩文件工具	扩展名	描述
compress	uncompress	.Z	最初的Unix文件压缩工具，已经快没人用了
tar+gzip	tar+gzip	.tgz	归档（打包）与压缩
gzip	gzip -d (或 gunzip)	.gz	GNU压缩工具，用Lempel-Ziv编码
zip	unzip	.zip	Windows上PKZIP工具的Unix实现
bzip2	bzip2 -d	.bz2	采用Burrows-Wheeler块排序文本压缩算法和霍夫曼编码

文件压缩工具





文件压缩

1. gzip命令

■ gzip命令用于压缩单个文件,没有打包功能。压缩后留下一个压缩文件,而原始文件则没有

语法:	gzip [选项][文件列表]
<p>用途: 压缩文件列表中的每一个文件并保存为“filename.gz”的形式, filename为原文件的名字。如果未指定文件名, 或者指定了一个名为“-”的文件名, 则从标准输入读取数据, 压缩的结果输出到标准输出。gzip 只尝试压缩常规文件, 特别地, 它将忽略符号连接。</p> <p>常用的选项/功能</p> <p>-N N取1到9 (并不是参数-N --no-name), 根据N的值控制压缩的速度 (压缩比率)。1代表最快的压缩, 但压缩率不高。9代表最慢的压缩, 但压缩率是最高的</p> <p>-c 将结果写到标准输出; 原文件保持不变</p> <p>-d 解压缩文件</p> <p>-f 强制压缩, 即使指定文件具有多重连接, 或相应的压缩文件已经存在, 或压缩数据来自标准输入</p> <p>-l 压缩文件使用的参数, 列出每个压缩文件如下内容:</p> <p> ◆compressed size: 压缩文件的长度</p> <p> ◆uncompressed size: 压缩前文件的长度</p> <p> ◆ratio: 压缩率(如果未知则为0.0%)</p> <p> ◆uncompressed_name: 压缩前的文件名</p> <p>-r 递归地压缩命令参数指定的目录下的文件</p> <p>-t 检查命令参数中的压缩文件的完整性</p> <p>-v 显示每个压缩文件的名字和压缩率</p>	





文件压缩(cont.)

2、gunzip命令

- 可以用gunzip命令执行解压缩的操作，把压缩文件还原成原始文件。
- 语法：gunzip [选项] 压缩文件名
 - gunzip命令选项与gzip相似，也使用-N, -c, -f, -l 和 -r等参数完成相应的操作。





文件压缩(cont.)

gunzip命令的常用选项

常用选项	说明
-x	文件列表 解压缩文件，但不包括指定的file文件
-v	查看压缩文件目录，但不解压。
-t	测试文件有无损坏，但不解压。
-d	目录把压缩文件解到指定目录下。
-z	只显示压缩文件的注解。
-n	不覆盖已经存在的文件。
-o	覆盖已存在的文件且不要求用户确认。
-j	不重建文档的目录结构，把所有文件解压到同一目录下。





文件压缩(cont.)

```
$ man bash > bash.man
$ ls -l bash.man
-rw----- 1 sarwar faculty 284064 Nov 20 12:24 bash.man
$ gzip bash.man
$ ls -l bash.man.gz
-rw----- 1 sarwar faculty 72501 Nov 20 12:24 bash.man.gz
$ gzip bash.man tcsh.man
$ gzip -l bash.man.gz tcsh.man.gz
compressed      uncompr.      ratio      uncompressed_name
72501            284064        74.4%      bash.man
73790            261316        71.7%      tcsh.man
146291           545380        73.1%      (totals)
$ gzip bash.man.gz
gzip: bash.man.gz already has .gz suffix -- unchanged
$ gunzip bash.man.gz
$ ls -l bash.man
-rw----- 1 sarwar faculty 284064 Nov 20 12:24 bash.man
$
```





文件压缩(cont.)

3、bzip2 命令

■ 安装bzip2包：

- `sudo apt-get install bzip2` \\On Debian/Ubuntu
- `sudo yum install bzip2` \\On CentOS/RHEL
- `sudo dnf install bzip2` \\On Fedora 22+

■ 压缩文件，压缩后生成 .bz2 文件

- `bzip2 filename` 或 `bzip2 -z filename`
- `bzip2 -z backup.tar` \\要压缩一个.tar文件

■ 解压文件，解压 .bz2 文件

- `bzip2 -d filename.bz2`





文件压缩(cont.)

■ 4、gzexe 压缩可执行文件

● gzexe [options] [file-list]

ELF (Executable and Linking Format)

Linux Standard Base

```
$ file banner
banner: ELF 32-bit LSB executable, Intel 80386, version 1, dynamically
linked (uses shared libs), not stripped
$ banner datafile 10
[ output of the banner command ]
$ gzexe banner
banner:  58.0%
$ ls -l banner*
-rwx----- 1 sarwar faculty 5239 Nov 19 11:45 banner
-rwx----- 1 sarwar faculty 10881 Nov 19 11:44 banner~
$ banner datafile 10
[ output of the banner command ]
$ gzexe -d banner
$ ls -l banner*
-rwx----- 1 sarwar faculty 10881 Nov 19 11:48 banner
-rwx----- 1 sarwar faculty 5239 Nov 19 11:45 banner~
$
```





文件压缩(cont.)

5、zcat、zmore 命令

- zcat命令显示用gzip或compress压缩的文件内容
- zmore命令分屏显示压缩文件的内容
- Example:

`zcat bash.man.gz`

..... \\bash.man.gz的内容

`zmore bash.man.gz`

..... \\bash.man.gz的内容





tar命令

- tar命令将多个文件打包成一个备份文件或从备份文件中取出文件
- 命令语法: **tar [options][filename-list]**
- 常用选项:
 - c 建立新的备份文件(压缩)。
 - r 将文件附加在备份文件后面。
 - f archname 用archname作为存档或恢复文件的备份文件名;默认是/dev/mto。如果archname是-,从标准输入读(对解压文件),或写到标准输出(对建立档案文件),这是当tar用作管道时的一个特性。
 - t 以类似ls -l格式列出磁带上的内容(备份在磁带上的文件名)。
 - u 将把比备份文件中更新的文件加入到备份文件中。
 - x 从备份文件中取出文件(解压)。
 - z 使用gzip来压缩或解压tar文件。
 - j 使用bzip2来压缩或解压tar文件
 - v 详细显示文件处理过程,用x选项解压文件的过程或存档文件的过程。





tar命令

- 例：下面命令用 tar程序将所有*.help文件打包成bash.help.tar的备份文件。

```
$ tar -cvf bash.help.tar *.help
```

- 例：下面命令用 tar程序来解开 linux内核包文件linux-3.9.6.tar.xz。（.xz文件采用LZMA SDK压缩）

```
$ tar -xvf linux-3.9.6.tar.xz
```

- 常用格式的命令

```
tar -czvf filename.tar.gz d/filename # gzip程序压缩
```

```
tar -xzvf filename.tar.gz # gzip程序解压
```

```
tar -cjvf filename.tar.bz2 d/filename # bz2程序压缩
```

```
tar -xjvf filename.tar.bz2 # bz2程序解压
```





软件包管理命令

- 下载并安装、更新等软件包命令, 自动从互联网的软件仓库中搜索、安装、升级、卸载软件或操作系统。
 - **apt-get** : Ubuntu/Debian等
 - **yum** : RHEL/CentOS等
 - **dnf** : dnf是新一代的RPM软件包管理器, Fedora 22+





apt-get

- **apt-get (或apt)**命令是debian、ubuntu发行版的包管理工具，主要用于自动从互联网的软件仓库中搜索、安装、升级、卸载软件或操作系统。apt-get命令需要root权限
 - apt-get install packagename //安装一个新软件包
 - apt-get remove packagename //卸载一个已安装的软件包
 - apt-get autoremove packagename //删除包及其依赖的软件包
 - **apt-get dist-upgrade** //将系统升级到新版本





文件排序

- **sort**命令是对文件中的各行进行排序。**sort**命令有许多非常实用的选项，这些选项最初是用来对数据库格式的文件内容进行各种排序操作的。**sort**命令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。
- **sort**排序是根据从输入行抽取的一个或多个关键字进行比较来完成的。排序关键字定义了用来排序的最小的字符序列。缺省情况下以整行为关键字按**ASCII**字符顺序进行排序。
- 语法：**sort** [选项] 文件列表
- 用途：**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。





文件排序(cont.)

■ 常用选项：

- -b 在每行中寻找排序关键字时忽略前导的空白（空格和制表符）。
- -d 按字典顺序排序，比较时仅字母、数字、空格和制表符有意义。
- -f 将小写字母与大写字母同等对待。
- -k pos1,pos2 指定一个或几个字段作为排序关键字，字段位置从pos1开始，到pos2为止。如不指定pos2，则关键字为从pos1到行尾。字段和字符的位置从1开始。
- -r 按逆序输出排序结果
- -c 检查给定文件是否已排好序，如果它们没有都排好序，则打印一个出错信息，并以状态值1退出。
- -o 输出文件 将排序输出写到输出文件中而不是标准输出，如果输出文件是输入文件之一，sort先将该文件的内容写入一个临时文件，然后再排序和写输出结果。





文件排序(cont.)

- 用**sort**命令对**ftext**文件中各行排序后输出其结果。请注意，在原文件的第二、三行上的第一个单词完全相同，该命令将从它们的第二个单词**vegetables**与**fruit**的首字符处继续进行比较。
- **\$ cat ftext**
vegetable soup
fresh vegetables
fresh fruit
lowfat milk
- **\$ sort ftext**
fresh fruit
fresh vegetables
lowfat milk
vegetable soup





文件排序(cont.)

\$ cat students

John Johnsen john.johnsen@tp.com 503.555.1111
Hassaan ji hji@k12.st.or 503.444.2132
David Kendall d_kendall@msnbc.org 229.111.2013
John Johnsen jjohnsen@psu.net 301.999.8888
Kelly Kimberly kellyk@umich.gov 555.123.9999
Maham ji mji@k12.st.or 713.888.0000
Jamie Davidson j.davidson@uet.edu 515.001.2932
Nabeel ji nji@xyz.net 434.555.1212

\$ sort students

David Kendall d_kendall@msnbc.org 229.111.2013
Hassaan ji hji@k12.st.or 503.444.2132
Jamie Davidson j.davidson@uet.edu 515.001.2932
John Johnsen jjohnsen@psu.net 301.999.8888
John Johnsen john.johnsen@tp.com 503.555.1111
Kelly Kimberly kellyk@umich.gov 555.123.9999
Maham ji mji@k12.st.or 713.888.0000
Nabeel ji nji@xyz.net 434.555.1212

\$





文件排序(cont.)

\$ sort -k 2 students #以姓氏——第二个字段开始(字段序为1)作为比较的键值。

Jamie	Davidson	j.davidson@uet.edu	515.001.2932
John	Johnsen	jjohnsen@psu.net	301.999.8888
John	Johnsen	john.johnsen@tp.com	503.555.1111
David	Kendall	d_kendall@msnbc.org	229.111.2013
Kelly	Kimberly	kellyk@umich.gov	555.123.9999
Hassaan	ji	hji@k12.st.or	503.444.2132
Maham	ji	mji@k12.st.or	713.888.0000
Nabeel	ji	nji@xyz.net	434.555.1212

■ 用电话号码作为比较键值，忽略行首的空格和tab键，按逆序排列结果。

\$ sort -k 4 -r -b students

Maham	ji	mji@k12.st.or	713.888.0000
Kelly	Kimberly	kellyk@umich.gov	555.123.9999
Jamie	Davidson	j.davidson@uet.edu	515.001.2932
John	Johnsen	john.johnsen@tp.com	503.555.1111
Hassaan	ji	hji@k12.st.or	503.444.2132
Nabeel	ji	nji@xyz.net	434.555.1212
John	Johnsen	jjohnsen@psu.net	301.999.8888
David	Kendall	d_kendall@msnbc.org	229.111.2013





文件排序(cont.)

- 可以保存排序后的文件内容，或把排序后的文件内容输出至打印机。下例中用户把排序后的文件内容保存到名为**result**的文件中。

```
$ sort ftext>result
```

- 以第1,2个字段作为排序关键字对文件**example**的内容进行排序。

```
$ sort -k 1,2 example
```





搜索命令和文件

- 可以用**find**命令查找目录列表中匹配表达式标准的文件，该命令递归搜索目录列表中的目录。

语法:	find [目录列表][表达式]
用途:	搜索目录列表中的目录，找出符合表达式（第二个参数）描述的文件。表达式可以由一个或者多个标准组成。
输出:	没有输出，除非在“表达式”中有要求
表达式中常用的标准:	
-exec CMD	如果命令(CMD)返回0（即该命令返回值为真），则该文件符合要求； 命令必须以 \; 结束
-inum N	搜索inode 为N的文件
-links N	搜索有N个链接的文件
-name pattern	搜索文件名匹配pattern的文件
-newer file	搜索修改时间在file之后的文件（即比file新的文件）
-ok CMD	和-exec相同，执行命令需要确认
-perm octal	搜索访问权限等于octal（八进制数字，如777）的文件
-print	显示符合要求的文件路径和文件名
-size ±N[c]	搜索文件大小为N块。字符c用来确定块的大小，默认为512个 字。+N表示大小超过N块的，-N就是小于N块的
-user name	搜索所有权为name的文件
\(expr \)	当表达式为真结果为真；表达式可以用OR和AND组合
! expr	取反，当表达式为假时结果为真





搜索命令和文件(cont.)

■ Examples of find command:

- 在主目录搜索Pakistan.gif，并显示文件的路径。如果在多个目录中找到目标文件，那么每一个包含文件的目录都会显示出来。

```
find ~ -name Pakistan.gif -print
```

```
/home/faculty/ji/myweb/Pakistan.html
```

- 在 /usr/include 目录中递归寻找socket.h文件，并且打印出它的绝对路径。

```
find /usr/include -name socket.h -print
```

```
/usr/include/sys/socket.h
```

- 如果想知道一个文件的所有硬链接，你可以用下面这个命令，找到/usr和.(当前目录)下所有文件inode=258072，并把他们打印出来。

```
find /usr . -inum 258072 -print
```

```
/home/faculty/ji/myweb/LinuxTcpIP
```

- 搜索当前目录下文件名为core或者扩展名为.ps 或者.p的文件，显示他们的绝对路径，并将它们删除。括号用来标明需要匹配的表达式，在\ (和-o前后必须要有空格。这个命令没有提示直接删除匹配的文件；如果想要提示，用参数-ok替换-exec。

```
$ find . \( -name core -o -name '*.ps' -o -name '*.p' \) -print -exec rm {} \;
```

.....命令的输出.....





搜索命令和文件(cont.)

- **whereis** 命令查明你系统上是否存在特定的一个命令，如果存在，给出该命令的路径。
whereis命令用于程序名的搜索，且只能搜索 { 二进制文件，man说明文件，源代码文件 }。

- **whereis [options] [file-list]**

- 用途: 搜索文件列表中的文件，给出可执行文件，源代码，帮助文件的位置
- 输出: 输出文件列表中命令的可执行文件，源代码，帮助文件的绝对路径
- 常用的选项/功能:

- b 只搜索可执行文件
- m 只搜索帮助文件
- s 只搜索源代码

```
$ whereis -b cat  
cat: /usr/bin/cat
```

- **Example:**

```
$ whereis ftp
```

```
ftp: /usr/bin/ftp.expect /usr/bin/ftp /usr/bin/ftp.sh /usr/man/man1/ftp.1.bz2
```

```
$ whereis -b cat
```

```
cat: /bin/cat
```





搜索命令和文件(cont.)

- 如果一个系统中有多版本命令，which命令告诉当输入某个命令执行时，shell到底调用了哪个版本的命令。
- which命令的作用是在PATH变量制定的路径中，查找系统命令的位置。
- **which [command-list]**
- Example:

```
$ which gzip  
/bin/gzip
```





搜索文件的内容

- LINUX有功能强大的搜索文件内容的工具，可以查找文本文件中包含特定的表达式、字符串或者模式的行。命令有 **grep**, **egrep** 和 **fgrep**
- **grep [选项] 模式 [文件列表]**
egrep [选项] [字符串] [文件列表]
fgrep [选项] [表达式] [文件列表]
 - **用途:** 按照给定的模式、字符串或表达式搜索文件列表中的文件。如果没有文件列表，则从标准输入读入数据
 - **输出:** 标准输出中包含给定模式、字符串或表达式的行
 - **常用的选项/功能:**
 - c 仅输出匹配的行的个数
 - i 在匹配的过程中忽略字母的大小写
 - l 仅输出有匹配行的文件名
 - n 匹配时同时输出行号
 - s 对shell脚本有用，避免出现错误信息(如果成功返回0，失败返回非零值)
 - v 打印出不匹配的行
 - w 全字匹配





搜索文件的内容(cont.)

■ 三个命令中：

- fgrep命令是最快的，不过有较多的限制。
- **egrep** 是最慢的，不过是最灵活的，完全支持正则表达式。
- grep 具有合理的速度和相当地对正则表达式的支持。

■ Examples:

● cat students

```
John Johnsen  john.johnsen@tp.com  503.555.1111
Hassaan ji  hji@k12.st.or    503.444.2132
David Kendall  d_kendall@msnbc.org  229.111.2013
John Johnsen  jjohnsen@psu.net      301.999.8888
Kelly Kimberly kellyk@umich.gov     555.123.9999
Maham ji  mji@k12.st.or  713.888.0000
Jamie Davidson j.davidson@uet.edu  515.001.2932
Nabeel ji  nji@xyz.net    434.555.1212
```





搜索文件的内容(cont.)

- 将student文件中包含ji的行都显示出来，它们按照在文件中的顺序依次显示。

`grep ji students`

```
Hassaan ji  hji@k12.st.or    503.444.2132
Maham ji    mji@k12.st.or    713.888.0000
Nabeel ji   nji@xyz.net      434.555.1212
```

- 显示students文件中包含ji的行和行号。

`grep -n ji students`

```
2:Hassaan ji  hji@k12.st.or    503.444.2132
6:Maham ji    mji@k12.st.or    713.888.0000
8:Nabeel ji   nji@xyz.net      434.555.1212
```





搜索文件的内容(cont.)

- 显示students文件中行首字母为A到H间的内容。^表示行首。

```
grep '^ [A-H]' students
```

```
Hassaan ji hji@k12.st.or 503.444.2132
```

```
David Kendall d_kendall@msnbc.org 229.111.2013
```

- 显示文件students中包含至少连续8个小写字母单词的行。

```
grep '[a-z]\{8\}' students
```

```
John Johnsen jjohnsen@psu.net 301.999.8888
```

```
Jamie Davidson j.davidson@uet.edu 515.001.2932
```





搜索文件的内容(cont.)

- 显示student文件中以J或者K开头的行的内容

`egrep "^J|^K" students`

Jamie Davidson j.davidson@uet.edu 515.001.2932

John Johnsen jjohnsen@psu.net 301.999.8888

John Johnsen john.johnsen@tp.com 503.555.1111

Kelly Kimberly kellyk@umich.gov 555.123.9999

- **grep应用:**

- 查找/dev目录下有多少个字符设备文件

`ls -l|grep "^c"|wc -l`





历史命令列表

■ history [options] [filename]

- 显示或操作历史命令列表
- 历史命令保存在~/.bash_history文件中
- !! 执行最近一个命令
- !n 执行历史命令列表中的第n个命令
- !-n 当前之前的第n个命令
- !string 最近用到的以string开始的命令





流编辑器sed

- 在字符串替换方面，用sed 比用vim更方便。sed 是流编辑器（streameditor）的缩写。sed 的常用格式为：

- sed [选项] '命令' 文件名

- 替换命令s

- sed 最常用的命令是s/regexp/replacement/flag。该命令的作用是，当某部分内容匹配了正则表达式regexp 时，用replacement 替换它。

```
$ echo "I study Linux, I need a Linux book" | sed 's/Linux/UNIX/'
```

```
I study UNIX, I need a Linux book
```

```
$ echo "I study Linux, I need a Linux book" | sed 's/Linux/UNIX/g'
```

```
I study UNIX, I need a UNIX book
```

```
$ echo "Linux Linux Linux" | sed 's/Linux/UBUNTU/1'
```

```
UBUNTU Linux Linux
```





Linux与Windows文件格式转化

■ 文本文件：

- Windows文件的换行符为\r\n
- Linux文件换行符为\n

■ dos2unix

- Windows转换为Linux文本文件命令

`dos2unix wintoLinuxF.txt`

■ unix2dos

- Linux转换为Windows文本文件命令

`unix2dos LinuxtowinF.txt`





小结

■ 本章需要掌握：

- 命令：cat、more、cd、echo、file、ls、mkdir、pwd、rmdir、mount、cp、head、more、mv、nl、rm、tac、tail和wc、gzip、gunzip、find、whereis、which、grep、tar、apt、yum
- 知识：文件概念，Linux文件类型，Linux文件属性，当前目录，主目录，绝对路径，相对路径、

■ 本章需要了解：

- 命令：diff、uniq、zcat、zmore、sort、history、cut、paste、rpm、dpkg、yum、sed
- 知识：文件描述符、正则表达式





End of chapter

