

第2章

文件管理与常用命令

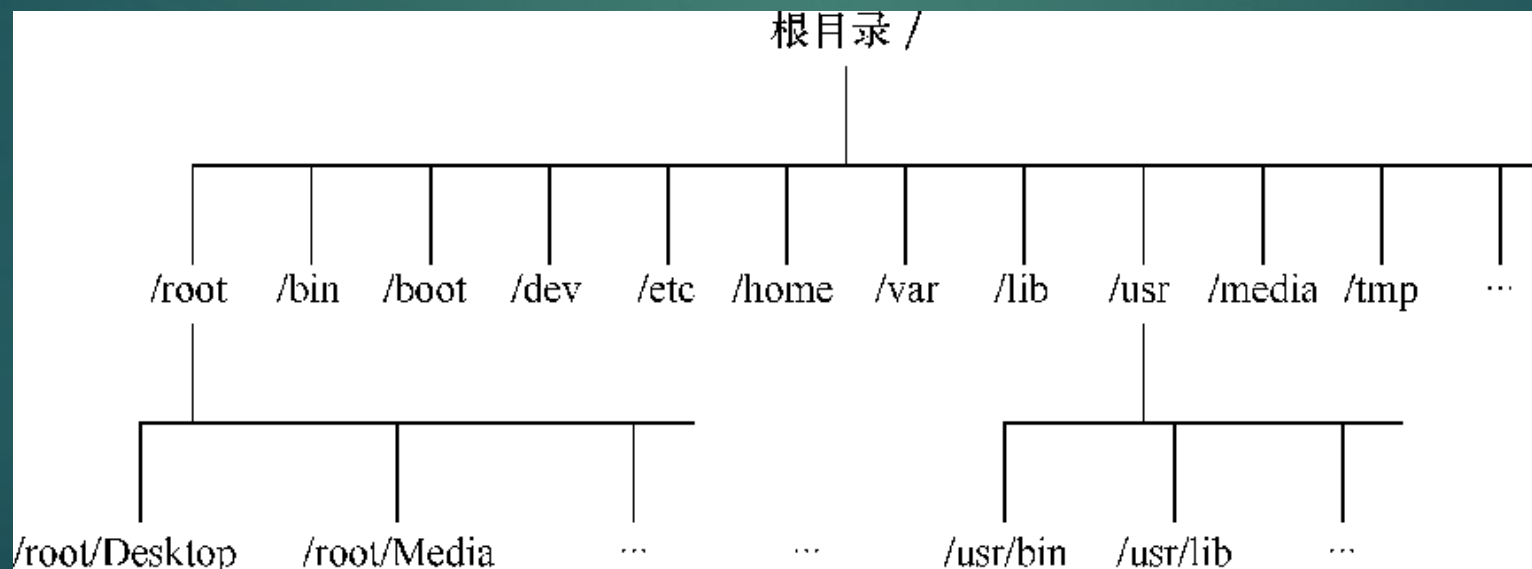
第2章 文件管理与常用命令

- ▶ 2.1 Linux目录配置
- ▶ 2.2 文件及文件管理
- ▶ 2.3 Linux文件权限
- ▶ 2.4 Linux文件搜索
- ▶ 2.5 文件的归档与压缩处理
- ▶ 2.6 系统调用

2.1 目录配置

目录树

在Linux系统中并不存在C/D/E/F等盘符，Linux系统中的一切文件都是从“根（/）”目录开始的，并按照文件系统层次化标准（Filesystem Hierarchy Standard, FHS）采用树形结构来存放文件（称为目录树），定义了常见目录的用途。例如/home/liu/test.txt需要由/home一直向下找到test.txt



目录树

目录树特点：

1. 目录树的起点始终为根目录（/）
2. 目录不仅能够使用本地的文件系统，还能使用网络上的文件系统
3. 每一个文件在此目录树中的文件名都是独一无二的

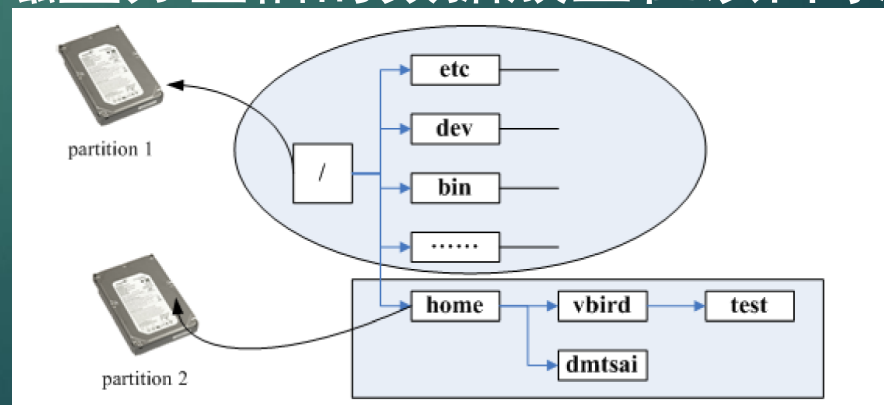
文件系统与目录树的关系

硬盘需要经过分区、格式化、挂载才能使用。

分区：硬盘分区是指将硬盘的整体存储空间划分成多个独立的区域，分别用来安装操作系统、安装应用程序以及存储数据文件等。

格式化：磁盘分区完毕后需要进行格式化，之后操作系统才能够使用这个文件系统。每种操作系统所设定的文件属性、权限并不相同，为了存放这些文件所需的数据，需要对分区槽进行格式化，已成为操作系统能够利用的文件系统类型。

挂载：挂载是利用一个目录当成进入点，将磁盘分区槽的数据放置在该目录下，进入该目录就可以读取该分区槽。



文件系统与目录树的关系

每个文件都会占用一个inode。

对于目录，inode存储属性、权限以及分配的data block号，而data block存储的是文件名与该文件对应的inode号。

对于普通文件，inode inode存储属性、权限以及分配的一组data block号。

文件系统与目录树的关系

目录树的读取方式：从根目录的inode逐层向下访问

例如：访问/etc/passwd

1. 通过挂载点信息找到根目录的inode，查看inode中的权限是否可读取data block。
2. 从data block中找到etc目录的inode，查看权限，读取etc的data block。
3. 从etc的data block中找到passwd的inode，查看权限，读取passwd内容

文件和目录的布局

Linux的发行版很多，如果每个版本下的目录都不同，将造成很多麻烦。因此，制定了FHS（Filesystem Hierachy Standard），目的是规范特定目录下应该放置什么内容。方便系统维护，FHS实际定义了根目录“/”，“/var”，“/usr”三个目录下的具体内容。

“/” 与开机有关

“/var” 与系统运行有关

“/usr” 与软件安装操作有关

文件和目录的布局

/etc 系统主要的配置文件几乎都放置在这个目录内，例如人员的帐号密码档、各种服务的配置文件等等。一般来说，这个目录下的各文件属性是可以让一般使用者查阅的，但是只有root有权力修改。FHS建议不要放置可执行文件(binary)在这个目录中。

/bin 系统有很多放置文件的目录，但/bin比较特殊。因为/bin放置的是在单人维护模式下还能够被操作的指令。在/bin底下的指令可以被root与一般帐号所使用，主要有：cat, chmod, chown, date, mv, mkdir, cp, bash等等常用的指令。

/boot 这个目录主要在放置开机会使用到的文件，包括Linux核心文件以及开机选单与开机所需设定档等等。Linux kernel常用的文件名为：vmlinuz，如果使用的是grub2这个开机管理程式，则还会存在/boot/grub2/这个目录！

/dev 在Linux系统上，任何装置与周边设备都是以文件的型态存在于这个目录当中的。你只要透过存取这个目录底下的某个文件，就等于存取某个设备～ 比较重要的文件有/dev/null, /dev/zero, /dev/tty, /dev/loop*, /dev/sd*等等

文件和目录的布局

`/lib` 系统的函数库非常多，而`/lib`放置的则是在开机时会用到的函数库，以及在`/bin`或`/sbin`底下的指令会呼叫的函数库而已。

`/media` `media` 放置的可移除的设备！包括软盘、光盘、DVD等等设备都暂时挂载于此。常见的档名有：`/media/floppy`，`/media/cdrom`等等。

`/mnt` 如果你想要暂时挂载某些额外的装置，一般建议你可以放置到这个目录中。

`/opt` 这个是给第三方软件放置的目录。比如：KDE这个桌面管理系统是一个独立的软件，不过他可以安装到Linux系统中，因此KDE的软体就建议放置到此目录下了。另外，如果你想要自行安装额外的软件（非原本的distribution提供的），也能够将你的软体安装到这个文件夹。不过，以前的Linux系统中，还是习惯放置在`/usr/local`目录下！

文件和目录的布局

/run 早期的 FHS 规定系统开机后所产生的各项信息应该要放置到 `/var/run` 目录下，新版的 FHS 则规范到 `/run` 底下。

/sbin Linux有非常多指令是用来设定系统环境的，这些指令只有root才能够利用来『设定』系统，其他使用者最多只能用来『查询』而已。放在/sbin底下的为开机过程中所需要的，包括了开机、修复、还原系统所需要的指令。

/srv srv可以视为『service』的缩写，是一些网路服务启动之后，这些服务所需要取用的资料目录。常见的服务例如WWW，FTP等等。举例来说，WWW服务器需要的网页资料就可以放置在/srv/www/里面。

/tmp 这是让一般使用者或者是正在执行的程序暂时放置文件的地方。这个目录是任何人都能够存取的，所以你需要定期的清理一下。当然，重要资料不可放置在此目录！因为FHS甚至建议在开机时，应该要将/tmp下的资料都删除！

/usr 第二层 FHS 设定，主要为放置不变动的内容

/var 第二层 FHS 设定，主要为放置变动性的内容

文件和目录的布局

/usr

/usr/bin: 所有一般用户能够使用的指令都在该目录里，Centos7已将全部的用户命令放置在该目录，/bin链接到此目录。FHS要求该目录下不应该有子目录。

/usr/lib: 与lib功能大致形同，/lib链接到此目录。

/usr/local: 系统管理员在本机自行下载安装的软件，建议安装到此目录。

/usr/sbin: 基本功能与/sbin相同，/sbin链接到此目录中

/usr/share: 放置共享数据文件，比如联机帮助文件、软件的说明等。

文件和目录的布局

`/var`

`/var/cache`: 程序运行中产生的缓存文件

`/var/lib`: 程序运行过程中需要的数据文件, 比如mysql、rpm的数据库

`/var/lock`: 防止同时使用一个设备或文件, 将该设备或文件上锁, 解锁后才能由后续使用者使用, 现在已放到`/run/lock`中

`/var/log`: 非常重要, 包括系统运行的日志文件, 比如message、dmessages等

`/var/run`: 存放程序运行的pid, 连接到run目录。

文件和目录的布局

FHS定义目录用途

目录名称	应放置文件的内容
/	Linux 文件的最上层根目录
/boot	开机所需文件—内核、开机菜单以及所需配置文件等
/dev	以文件形式存放任何设备与接口
/etc	配置文件
/home	用户家目录
/bin	Binary 的缩写，存放用户的可运行程序，如 ls 、 cp 等，也包含其他 shell ，如 bash 和 cs 等
/lib	开机时用到的函数库，以及/bin与/sbin下面的命令要调用的函数
/sbin	开机过程中需要的命令

文件和目录的布局

目录名称	应放置文件的内容
/media	用于挂载设备文件的目录
/opt	放置第三方的软件
/root	系统管理员的家目录
/srv	一些网络服务的数据文件目录
/tmp	任何人都可使用的“共享”临时目录
/proc	虚拟文件系统，如系统内核、进程、外部设备及网络状态等
/usr/local	用户自行安装的软件
/usr/sbin	Linux系统开机时不会使用到的软件/命令/脚本
/usr/share	帮助与说明文件，也可放置共享文件
/var	主要存放经常变化的文件，如日志
/lost+found	当文件系统发生错误时，将一些丢失的文件片段存放在这里

绝对路径和相对路径

绝对路径：由根目录（/）开始写起的文件名或目录名称，`/home/dmtsai/basher`。

相对路径：相对于目前路径的文件名写法，如`../home/dmtsai`或`home/dmtsai/`等。

相对路径是以当前所在路径的相对位置来表示的。举例来说，你目前在`/home`这个目录下，如果想要进入`/var/log`这个目录时，可以怎么写呢？有两种方法。

`cd /var/log`：绝对路径。

`cd ../var/log`：相对路径。

`.`：代表当前的目录，也可以使用 `./` 来表示；

`..`：代表上一层目录，也可以 `../` 来代表。



► 练习:

现在你正在你的家目录，分别用绝对路径和相对路径的方式进入” /tmp” 目录；

目录/tmp/abc/def是绝对路径还是相对路径，若当前正在abc目录中，如何进入def？

特殊目录

- . 代表当前目录
- .. 代表上一层目录
- 代表上一个工作目录
- ~ 代表当前用户的家目录

2.2 文件及文件管理

文件类型

在Linux系统中，将普通文件、目录、磁盘设备、打印机等都看成文件，所以文件类型十分丰富。

“一切皆文件”指的是对所有文件（目录、字符设备、块设备、套接字、打印机、进程、线程、管道等）操作，读写都可用`fopen()`/`fclose()`/`fwrite()`/`fread()`等函数进行处理。屏蔽了硬件的区别，所有设备都抽象成文件，提供统一的接口给用户。虽然类型各不相同，但是对其提供的却是同一套操作界面。

文件类型

▶ 普通文件

最常用的一类文件，其特点是不包含文件系统的结构信息。通常用户所接触到的文件，如图形文件、数据文件、文档文件等都属于这种文件。包括文本文件、数据文件、二进制文件

▶ 目录文件

目录文件是用于存放文件名及其相关信息的文件。是内核组织文件系统的基本节点。

▶ 符号链接文件

是一种特殊文件，实际上是指向一个真实存在的文件链接，类似于Windows下的快捷方式。

▶ 设备文件

字符设备文件：允许I/O传送任意大小的数据，代表终端、打印机及鼠标等，类型为 `c`

块设备文件：存储数据以供系统存取的设备，典型设备是硬盘，类型为 `b`

▶ 管道文件：一种特殊文件，主要用于不同进程间的信息传递，Linux对管道操作与文件相同。

类型	简称	描述
普通文件	-,Normal File	如mp4、pdf、html log; 用户可以根据访问权限对普通文件进行查看、更改和删除, 包括 纯文本文件(ASCII); 二进制文件(binary); 数据格式的文件(data);各种压缩文件.第一个属性为 [-]
目录文件	d,directory file	/usr/ /home/ 目录文件包含了各自目录下的文件名和指向这些文件的指针, 打开目录事实上就是打开目录文件, 只要有访问权限, 就可以随意访问这些目录下的文件。能用#cd命令进入的。第一个属性为[d], 例如 [drwxrwxrwx]
硬链接	-,hard links	若一个inode号对应多个文件名, 则称这些文件为硬链接。硬链接就是同一个文件使用了多个别名删除时,只会删除链接, 不会删除文件; 硬链接的局限性:1.不能引用自身文件系统以外的文件,即不能引用其他分区的文件;2.无法引用目录;
符号链接 (软链接)	l,symbolic link	若文件用户数据块中存放的内容是另一文件的路径名的指向, 则该文件就是软连接, 克服硬链接的局限性, 类似于快捷方式,使用与硬链接相同。
字符设备文件	c,char	文件一般隐藏在/dev目录下, 在进行设备读取和外设交互时会被使用到 即串行端口的接口设备, 例如键盘、鼠标等等。第一个属性为 [c]。 #/dev/tty的属性是 crw-rw-rw-, 注意前面第一个字 c, 这表示字符设备文件
块设备文件	b,block	存储数据以供系统存取的接口设备, 简单而言就是硬盘。 # /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 系统中的所有设备要么是块设备文件, 要么是字符设备文件, 无一例外
FIFO管道文件	p,pipe	管道文件主要用于进程间通讯。FIFO解决多个程序同时存取一个文件所造成的错误。比如使用mkfifo命令可以创建一个FIFO文件, 启用一个进程A从FIFO文件里读数据, 启动进程B往FIFO里写数据, 先进先出, 随写随读。 # pipe
套接字	s,socket	以启动一个程序来监听客户端的要求, 客户端就可以通过套接字来进行数据通信。用于进程间的网络通信, 也可以用于本机之间的非网络通信, 第一个属性为 [s], 这些文件一般隐藏在/var/run目录下, 证明着相关进程的存在

Linux文件扩展名

与Windows不同，Linux系统中文件的扩展名是没有意义的。一个文件能否执行取决于是否拥有“x”权限。扩展名仅仅用来了解文件是什么，比如.sh, .tar, .gz, .html

例：在Linux系统中abc.exe是什么类型的文件？

Linux文件命名规则

- ▶ 名字长度：允许1—256字符
- ▶ 文件名的字符包括：字母、数字、.（点）、_（下划线）和-（连字符）。

有些转意字符在 Linux 的命令解释器中有特殊的含义。?、*、\$、&、!、|、\、[]、{}、()等等。在文件名中应尽量避免使用这些字符。

以‘.’开头的文件为隐藏文件。/ 既可代表目录树的根也可作为路径名中的分隔符（类似DOS下的 \），因此 / 不能出现在文件名中。

- ▶ **文件名区分大小写**

abc.txt 和 ABC.txt是同一个文件么？

目录操作命令

1. 显示当前路径pwd命令

命令功能：显示当前工作目录的绝对路径

命令格式：pwd

```
[liu@localhost testmake]$ pwd  
/home/liu/testmake
```

目录操作命令

2. 更改当前目录cd命令

命令功能：改变当前所在的目录

命令格式：cd 目录名

```
[liu@localhost testmake]$ ls -al
总用量 20
drwxrwxr-x.  4 liu liu   59 12月  6 2020 .
drwx----- 51 liu liu 4096 3月 15 09:44 ..
drwxrwxr-x.  3 liu liu 4096 3月 15 09:44 ti
[liu@localhost testmake]$ cd ti
[liu@localhost ti]$ pwd
/home/liu/testmake/ti
```

目录操作命令

3. 创建目录mkdir命令

命令功能：创建目录

命令格式：mkdir [选项] 目录名

常用选项：

-m 数字 对新建目录设置存取权限；

-p在创建目录时，如果父目录不存在，则同时创建该目录及该目录的父目录。

目录操作命令

4. 删除目录`rmdir`命令

命令功能：删除目录（必须为空）

命令格式：`rmdir` [选项] 目录名

常用选项：

-p 在删除目录时，一同删除父目录，但父目录中必须没有其他目录及文件。

目录操作命令

5. 显示文件信息ls命令

命令功能：显示文件和目录列表

命令格式：ls [选项] [目录名或文件名]

```
[root@server1 ~] #ls          //列出当前目录下的文件及目录
[root@server1 ~] #ls -a        //列出包括以 "." 开始的隐藏文件在内的所有文件
[root@server1 ~] #ls -t        //依照文件最后修改时间的顺序列出文件
[root@server1 ~] #ls -l        //列出当前目录下所有文件的权限、所有者、文件大小、修改时间及名称
[root@server1 ~] #ls -R        //显示出目录下以及其所有子目录的文件名
```

ls命令参数列表

- a 列出目录下的所有文件, 包括以. 开头的隐含文件;
- l 列出文件的详细信息;
- k 以k字节的形式表示文件的大小;
- t 以时间排序
- b 把文件名中不可输出的字符用反斜杠加字符编号的形式列出;
- d 将目录像文件一样显示, 而不是显示其中所包含的文件;
- e 输出时间的全部信息, 而不是输出简略信息
- m 横向输出文件名, 并以 “,” 作分格符
- n 用数字的UID、GID代替名称在每个文件名后附上一个字符以说明该文件的类型, “*” 表示可执行的普通文
- p或-F件: “/” 表示目录; “@” 表示符号链接; “|” 表示FIFOs; “—” 表示套接字
- r 对目录反向排序
- s 在每个文件名后输出该文件的大小
- u 以文件上次被访问的时间排序版-x按列输出, 横向排序;
- l 一行只输出一个文件
- color=no不显示彩色文件名

目录操作命令

```
[liu@localhost etc]$ ls -al
```

总用量 1420

drwxr-xr-x.	145	root	root	12288	8月	31	10:00	.
dr-xr-xr-x.	19	root	root	4096	9月	2	11:31	..
drwxr-xr-x.	3	root	root	101	1月	11	2020	abrt
-rw-r--r--.	1	root	root	16	1月	11	2020	adjtime
-rw-r--r--.	1	root	root	1518	6月	7	2013	aliases
-rw-r--r--.	1	root	root	12288	1月	11	2020	aliases.db
drwxr-xr-x.	3	root	root	65	1月	11	2020	alsa
drwxr-xr-x.	2	root	root	4096	1月	11	2020	alternatives

目录操作命令

练习：在家目录下建立“abc”文件夹，并进入；
在“abc”文件夹下创建“ABC”和“DEF”文件夹；
删除“ABC文件夹”；
查看当前目录；

文件操作命令

1. 显示文件内容 cat命令

命令功能：显示一个或多个文件的内容

命令格式：cat [选项] 文件名

常用选项：

-n：由 1 开始对所有输出的行数编号。

-b：和 -n 相似，只不过对于空白行不编号。

-s：当遇到有连续两行以上的空白行，就代换为一行的空白行。

利用cat命令还可以合并多个文件。如把file1和file2文件的内容合并为file3，且file2文件的内容在file1文件的内容前面，则命令为

```
[root@server1 ~]# cat file2 file1>file3  
//如果file3文件存在，则此命令的执行结果会覆盖file3文件中原有内容
```

文件操作命令

2. head命令

命令功能：显示文件的开头部分，默认情况下只显示文件的前10行内容

命令格式：head [选项] 文件名

常用选项：

-n num：显示指定文件的前num行

-c num：显示指定文件的前num个字符。

-num：从倒数第num行开始显示指定文件的内容

```
[root@server1 ~]#head -n 20 /etc/httpd/conf/httpd.conf //显示 httpd.conf文件的前20行
```

文件操作命令

3. tail命令

命令功能：用于显示文件的末尾部分，默认显示文件的末尾10行内容

命令格式：tail [选项] 文件名

常用选项：

-n num：显示指定文件的末尾num行。

-c num：显示指定文件的末尾num个字符。

+num：从第num行开始显示指定文件的内容

```
[root@server1 ~]# tail -n 20 /etc/httpd/conf/httpd.conf  
//显示 httpd.conf文件的末尾20行
```

文件操作命令

4. 分屏显示文件内容more命令

命令功能：分屏显示文件内容

命令格式：more [选项] 文件名

常用指令：

“空格”：向下翻一页

“Enter”：向下翻一行

“q”：终止浏览文件

```
[root@server1 ~]#more file1
```

// 以分页方式查看file1文件的内容

文件操作命令

选项	说明
-num	指定屏幕上一次所显示的行数；
-d	提示使用者，在画面下方显示 <code>--More-- [Press space to continue, 'q' to quit.]</code> ，如果使用者按错键，则会显示 <code>[Press 'h' for instructions.]</code> ；
-f	计算行数时，以实际上的行数，而非自动换行过后的行数（有些单行字数太长的会被扩展为两行或两行以上）；
-p	不以卷动的方式显示每一页，而是先清除屏幕后再显示内容；
-c	跟 -p 相似，不同的是先显示内容再清除其他旧资料；
-s	当遇到有连续两行以上的空白行，就代换为一行的空白行；
-u	不显示下引号（根据环境变数 TERM 指定的 terminal 而有所不同）；
+/	在每个文件显示前搜寻该字串（ pattern ），然后从该字串之后开始显示；
+num	从第 num 行开始显示。

文件操作命令

5. 分屏显示文件内容less命令

命令功能：分屏显示文件内容

命令格式：less [选项] 文件名

常用指令：

“空格”：向下翻一页

“pagedown/pageup”：向下/上翻一行

“/”：向下搜寻字符串

“g/G”：到文件的第一行/最后一行

“q”：离开

```
[root@server1 ~]#less /etc/man_db.conf  
// 以分页方式查看httpd.conf文件的内容
```

文件操作命令

选项	说明
-a	在当前屏幕显示最后一页；
-c	从顶部（从上到下）刷新屏幕，并显示文件内容。而不是通过底部滚动完成刷新；
-f	强制打开文件，二进制文件显示时，不提示警告；
-i	搜索时忽略大小写；除非搜索串中包含大写字母；
-I	搜索时忽略大小写，除非搜索串中包含小写字母；
-m	显示读取文件的百分比；
-M	显示读取文件的百分比、行号及总行数；
-N	在每行前输出行号；
-p pattern	搜索pattern；
-s	把连续多个空白行作为一个空白行显示；

文件操作命令

6. 创建文件/修改文件时间标签touch命令

命令功能：生成新的空文件、更改现有文件的时间标签

命令格式：touch [选项] 文件名

常用参数选项如下。

-d yyyymmdd：把文件的存取和修改时间改为yyyy年mm月dd日。

-a：只把文件的存取时间改为当前时间。

-m：只把文件的修改时间改为当前时间。

```
[root@server1 ~]# touch aa
```

//如果当前目录下存在aa文件，则把aa文件的存取和修改时间改为当前时间

//如果不存在aa文件，则新建aa文件

```
[root@server1 ~]# touch -d 20180808 aa //将aa文件的存取和修改时间改为  
2018年8月8日
```

文件操作命令

选项	说明
-a	改变文件的读取时间标签；
-c	假如目的文件不存在，不会建立新的文件；
-m	改变文件的修改时间标签；
-d	设定时间与日期，可以使用各种不同的格式；
--no-create	不会建立新文件；
--version	列出版本信息。

文件操作命令

8. 文件内容排序sort命令

命令功能：对文本文件的各行进行排序

命令格式：sort [选项] 文件列表

文件操作命令

选项	说明
-b	忽略每行前面开始出的空格字符；
-c	检查文件是否已经按照顺序排序；
-d	排序时，处理英文字母、数字及空格字符外，忽略其他的字符；
-f	排序时，将小写字母视为大写字母；
-i	排序时，只排序可打印字符，忽略其他的字符；
-m	将几个排序好的文件进行合并；
-M	将前面3个字母依照月份的缩写进行排序；
-n	依照数值的大小排序；
-o<输出文件>	将排序后的结果存入指定的文件；
-r	以相反的顺序来排序；
-t<分隔字符>	指定排序时所用的栏位分隔字符；
+<起始栏位> -<结束栏位>	以指定的栏位来排序，范围由起始栏位到结束栏位的前一栏位。

文件操作命令

例1：将/etc/passwd中的内容以：来分隔，并按照第3栏排序

```
[liu@localhost yum.repos.d]$ cat /etc/passwd | sort -t ':' -k 3  
root:x:0:0:root:/root:/bin/bash  
liu:x:1000:1000:liu:/home/liu:/bin/bash  
zhangsan:x:1002:1002::/home/zhangsan:/bin/bash
```

例2：将/etc/passwd中的内容以：来分隔，并按照第3栏数字大小排序

```
[liu@localhost yum.repos.d]$ cat /etc/passwd | sort -t ":" -nk 3  
root:x:0:0:root:/root:/bin/bash  
bin:x:1:1:bin:/bin:/sbin/nologin  
daemon:x:2:2:daemon:/sbin:/sbin/nologin  
adm:x:3:4:adm:/var/adm:/sbin/nologin
```

文件操作命令

9. 文件比较diff命令

命令功能：用于比较两个文件内容的不同，主要是纯文本文件

命令格式：diff [参数] 源文件 目标文件

diff命令的常用参数选项如下。

- a：将所有的文件当作文本文件处理。
- 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。
- i：忽略大小写的变化。

```
[root@server1 ~]# diff aa.txt bb.txt //比较aa.txt文件和bb.txt文件的不同
```

文件操作命令

选项	说明
-b	忽略每行前面开始出的空格字符；
-c	检查文件是否已经按照顺序排序；
-d	排序时，处理英文字母、数字及空格字符外，忽略其他的字符；
-f	排序时，将小写字母视为大写字母；
-i	排序时，除了040至176之间的ASCII字符外，忽略其他的字符；
-m	将几个排序好的文件进行合并；
-M	将前面3个字母依照月份的缩写进行排序；
-n	依照数值的大小排序；
-o<输出文件>	将排序后的结果存入指定的文件；
-r	以相反的顺序来排序；
-t<分隔字符>	指定排序时所用的栏位分隔字符；
+<起始栏位> -<结束栏位>	以指定的栏位来排序，范围由起始栏位到结束栏位的前一栏位。

文件操作命令

例：

```
[dmtsai@study testpw]$ diff passwd.old passwd.new
4d3      <==左边第四行被删除 (d) 掉了，基准是右边的第三行
< adm:x:3:4:adm:/var/adm:/sbin/nologin  <==这边列出左边(<)文件被删除的那一行内容
6c5      <==左边文件的第六行被取代 (c) 成右边文件的第五行
< sync:x:5:0:sync:/sbin:/bin/sync  <==左边(<)文件第六行内容
---
> no six line      <==右边(>)文件第五行内容
# 很聪明吧！用 diff 就把我们刚刚的处理给比对完毕了！
```

10a10

> adduser

文件操作命令

10. 复制文件cp命令

命令功能：复制文件或目录

命令格式：cp [选项] 源文件或目录 目标文件或目录

常用参数选项如下。

- a：尽可能将文件状态、权限等属性照原状予以复制。
- f：如果目标文件或目录存在，先删除它们再进行复制（即覆盖），并且不提示用户。
- i：如果目标文件或目录存在，提示是否覆盖已有的文件。
- r：递归复制目录，即包含目录下的各级子目录。

```
[root@server1 ~]# cp ~/.bashrc /tmp/bashrc
[root@server1 ~]# cp -i ~/.bashrc /tmp/bashrc
cp: overwrite `/tmp/bashrc'? n 不覆盖, y为覆盖
# 重复做两次, 由于/tmp下已经存在bashrc了, 加上-i选项后,
# 则在覆盖前会询问使用者是否确定! 可以按下n或者y来二次确认
```

文件操作命令

选项	说明
-a	该选项通常在复制目录时使用。它递归地将源目录下的所有子目录及其文件都复制到目标目录中，并且保留文件链接和文件属性不变。它等效于 -dpR ；
-d	复制时保留文件链接；
-f	若目标文件或目录已存在，则覆盖已存在的目标文件或目录并且不提示；
-i	和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时目标文件将被覆盖，是交互式复制；
-p	此时cp除复制源文件的内容外，还将把其修改时间和访问权限也复制到新文件中；
-R, -r	若给出的源文件是一目录文件，此时cp将递归复制该目录下所有的子目录和文件至目的地。此时目标文件必须为一个目录名；
-l	不复制，而是创建指向源文件的链接文件，链接文件名由目标文件给出。

文件操作命令

11. 删除文件rm命令

命令功能：删除文件或目录

命令格式：rm [选项] 文件列表

常用参数选项如下。

-i：删除文件或目录时提示用户。

-f：删除文件或目录时不提示用户。

-R：递归删除目录，即包含目录下的文件和各级子目录。

```
//删除当前目录下的所有文件，但不删除子目录和隐藏文件
[root@server1 ~]# mkdir /dir1;cd /dir1
[root@server1 dir1]# touch aa.txt bb.txt; mkdir
subdir11;ll
[root@server1 dir1]# rm *
// 删除当前目录下的子目录subdir11，包含其下的所有文件和子目录，
并且提示用户确认
[root@server1 dir]# rm -iR subdir11
```

文件操作命令

12. 移动文件mv命令

命令功能：移动文件或目录、重命名文件或目录

命令格式：mv [选项] 源文件或目录 目标文件或目录

常用选项：

-i：如果目标文件或目录存在，则提示是否覆盖目标文件或目录。

-f：无论目标文件或目录是否存在，直接覆盖目标文件或目录，不提示。

```
//将当前目录下的testa文件移动到/usr/目录下，文件名不变  
[root@server1 ~]# mv testa /usr/  
//将/usr/testa文件移动到根目录下，移动后的文件名为tt  
[root@server1 ~]# mv /usr/testa /tt
```

文件操作命令

13. 统计文件字节wc命令

命令功能：统计指定文件的字节数、字数、行数

命令格式：wc [选项] 文件

常用选项：

-c： 统计字节数

-l： 统计行数

-m： 统计字符数

-w： 统计字数

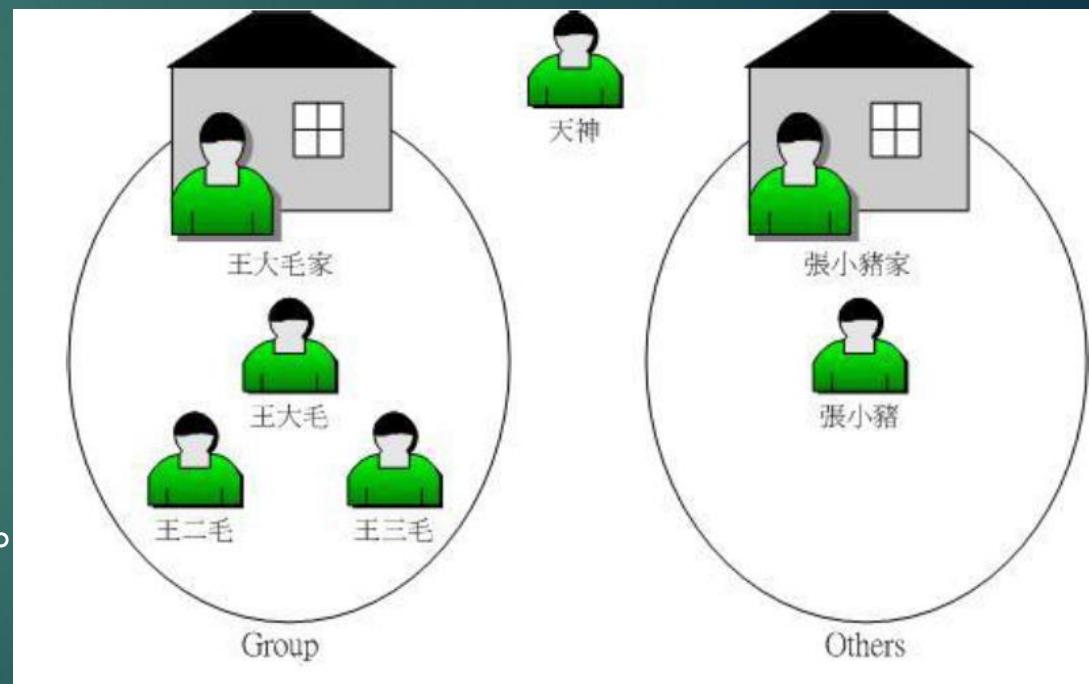
2.3 Linux文件权限

用户与用户组

Linux是多人多任务的操作系统，因此可能常常会有多人使用一台机器，为了考虑每个人的隐私、方便用户合作，每个文件都有三类用户，权限是基于这三类用户设定的：

- 1) 文件拥有者 (user)
- 2) 组用户 (group)
- 3) 其它 (others)

万能用户：root用户，不受任何权限限制。



文件权限概述

在Linux中的每一个文件或目录都包含有访问权限，这些访问权限**决定了谁能访问和如何访问这些文件和目录**。通过设定权限可以从以下3种访问方式限制访问权限。

根据赋予权限的不同，3种不同的用户（所有者、用户组或其他用户）能够访问不同的目录或者文件。所有者是创建文件的用户，文件的所有者能够授予所在用户组的其他成员以及系统中除所属组之外的其他用户的文件访问权限。每一个用户针对系统中的所有文件都有它自身的读、写和执行权限。

文件属性详解

“ls -l” 命令显示文件的详细信息，其中包括权限。如下所示：

文件类型权限	连接数	文件所属群组		文件最后被修改时间		
↑	↑	↑	↑	↑		
-rwxr-xr-x	2	root	root	4096	Aug 9 15:03	file1.sh
	↓		↓			↓
	文件拥有者		文件容量			文件名

文件属性示意图（一共7组或7列）

文件属性详解

1. 第1组为文件类型权限

每一行的第一个字符一般用来区分文件的类型，一般取值为d、-、l、b、c、s、p。具体含义如下。

d: 表示是一个目录，在ext文件系统中目录也是一种特殊的文件。

-: 表示该文件是一个普通的文件。

l: 表示该文件是一个符号链接文件，实际上它指向另一个文件。

b、c: 分别表示该文件为区块设备或其他的外围设备，是特殊类型的文件。

s、p: 这些文件关系到系统的数据结构和管道，通常很少见到。

```
[liu@localhost ~]$ ls -al nihao.txt
-rw-rw-r--. 1 liu liu 16 9月 16 16:50 nihao.txt
```

文件属性详解

每一行的第2~10个字符表示文件的访问权限。

这9个字符每3个为一组，左边3个字符表示所有者权限，中间3个字符表示与所有者同一组的用户的权限，右边3个字符是其他用户的权限。代表的意义如下。

字符2、3、4表示该文件所有者的权限，也简称为u（User）的权限。

字符5、6、7表示该文件所有者所属组的组成员的权限，简称为g（group）的权限。

字符8、9、10表示该文件所有者所属组群以外的权限，简称为o（Other）的权限。

```
[liu@localhost ~]$ ls -al nihao.txt
-rw-rw-r--. 1 liu liu 16 9月 16 16:50 nihao.txt
```

文件属性详解

r (Read, 读取) : 对文件而言, 具有读取文件内容的权限; 对目录来说, 具有浏览目录的权限。

w (Write, 写入) : 对文件而言, 具有新增、修改文件内容的权限; 对目录来说, 具有删除、移动目录内文件的权限。

x (execute, 执行) : 对文件而言, 具有执行文件的权限; 对目录来说, 具有**进入目录**的权限。

- : 表示不具有该项权限。

练习: -rw-rw-r-x、drwx--x--x、lrwxrwxrwx

```
[liu@localhost ~]$ ls -al nihao.txt
-rw-rw-r--. 1 liu liu 16 9月 16 16:50 nihao.txt
```


文件属性详解

2. 第2组表示有多少文件名连结到此节点（i-node）

每个文件都会将其权限与属性记录到文件系统的i-node中，不过，我们使用的目录树却是使用文件来记录，因此每个文件名就会连接到一个i-node。这个属性记录的就是有多少不同的文件名连接到相同的一个i-node。

3. 第3组表示这个文件（或目录）的拥有者账号

4. 第4组表示这个文件的所属群组

在Linux系统下，账号会附属于一个或多个的群组中。举例来说明：class1、class2、class3均属于projecta这个群组，假设某个文件所属的群组为projecta，且该文件的权限为（-rwxrwx---），则class1、class2、class3 3人对于该文件都具有可读、可写、可执行的权限（看群组权限）。但如果是不属于projecta的其他账号，对于此文件就不具有任何权限了。

文件属性详解

5. 第5组为这个文件的容量大小，默认单位为bytes

6. 第6组为这个文件的创建日期或者是最近的修改日期

这一栏的内容分别为日期（月/日）及时间。如果这个文件被修改的时间距离现在太久了，那么时间部分会仅显示年份而已。如果想要显示完整的时间格式，可以利用ls的选项，即ls -l --full-time就能够显示出完整的时间格式了。

7. 第7组为这个文件的文件名

比较特殊的是：如果文件名之前多一个“.”，则代表这个文件为隐藏文件。使用ls -a可以查看隐藏文件。

权限对文件的重要性：

r(read)：可读取文件的实际内容，如读取文本文件的文字内容

w(write)：可以编辑、新增或者是修改文件的内容（但不含删除文件）

x（execute）：该文件具有可以被系统执行的权限

权限对目录的重要性：

r(read)：能够查看目录结构，可以查询改了目录下的文件名。

w(write)：建立新的文件与目录、传出已存在的文件或目录、将已经存在的文件或目录进行更名；搬移该目录内的文件、目录位置。（不能随便给）

x（execute）：能否进入该目录工作

权限举例

```
[liu@localhost home]$ ls -al
```

总用量 8

```
drwxr-xr-x.  9 root  root   114  9月  21 11:28 .
dr-xr-xr-x. 19 root  root  4096  9月  10 12:04 ..
-rw-r--r--.  1 root  root     0  8月  13 10:53 abc-a.txt
drwx-----.  3 jone  jone    78  6月  14 16:19 jone
drwx-----. 32 liu   liu   4096  9月  21 11:27 liu
```

```
[liu@localhost ~]$ ls -al /etc/shadow /etc/passwd
```


```
-rw-r--r--. 1 root root 2531 9月  21 11:28 /etc/passwd
-----. 1 root root 1692 9月  21 11:28 /etc/shadow
```

```
[liu@localhost ~]$ ls -al nihao.txt
```

```
-rw-rw-r--. 1 liu liu 16 9月  16 16:50 nihao.txt
```

```
[liu@localhost ~]$ ls -al /usr/bin/ls
```

```
-rwxr-xr-x. 1 root root 117608 8月  20 2019 /usr/bin/ls
```

- 
- ▶ 用户分为三类—每类用户有三种权限，要看自己的身份属于对于文件属于哪类用户，再看自己具有的权限。
 - ▶ 权限不足：用root 万能（不推荐）， 命令前加sudo（需配置）。*修改文件的权限。

权限修改

通常在权限修改时可以用两种方式来表示权限类型：数字表示法和文字表示法。

chmod命令的格式是：

chmod 选项 文件

(1) 以数字表示法修改权限

所谓数字表示法是指将读取（r）、写入（w）和执行（x）分别以数字4、2、1来表示，没有授予的部分就表示为0，然后再把所授予的权限相加而成。

例：为文件abc设置权限：赋予拥有者和组群成员读取和写入的权限，而其他人只有读取权限。

权限应为rw-rw-r--，对应数字权限664，因此命令为：chmod 664 abc

例：若一个文件的权限为654，说明他的权限。

权限修改

(2) 文字表示法

使用权限的文字表示法时，系统用4种字母来表示不同的用户。

u: user, 表示所有者。

g: group, 表示属组。

o: others表示其他用户。

a: all, 表示以上3种用户。

使用下面3种字符的组合表示法设置操作权限。

r: read, 可读。

w: write, 写入。

x: execute, 执行。

操作符号包括以下几种。

+: 添加某种权限。

-: 减去某种权限。

=: 赋予给定权限并取消原来的权限。

对于上例: `chmod u=rw, g=rw, o=r abc`

权限修改

练习

1. 查看/etc 目录下的各文件权限
2. 在家目录创建testmod 文件夹，查看默认权限
3. 进入testmod目录，创建文件testfile, 查看默认权限
4. testfile文件的权限为664，为testfile的拥有者添加执行权限
5. 为群组用户去掉写权限
6. 为所有用户添加执行权限

分别用文字表示法、数字表示法。

文件所有者与属组修改

要修改文件的所有者可以使用chown命令。chown命令格式如下所示：

chown 选项 用户和属组 文件列表

***仅root用户可使用**

例：把/yy/file文件的所有者修改为test用户，命令如下：

```
chown test /yy/file
```

文件所有者与属组修改

chown命令可以同时修改文件的所有者和属组，用“:”分隔。

例：将/yy/file文件的所有者和属组都改为test的命令如下所示：

```
chown test:test /yy/file
```

如果只修改文件的属组可以使用下列命令：

```
chown :test /yy/file
```

修改文件的属组也可以使用chgrp命令。命令范例如下所示：

```
chgrp test /yy/file
```

文件默认权限修改

Umask 命令用于指定用户在建立文件和目录时的权限默认值，umask值称作权限掩码。

查看默认权限命令： `umask`

设置默认权限命令： `umask` 数字权限

文件、目录默认权限修改方法：

文件的预设权限为`-rw-rw-rw-`，目录的预设权限为`-rwxrwxrwx`，普通用户umask默认值为002，root用户为022， 创建文件时在预设权限中减掉umask值对应的权限。

例：umask为002的普通用户创建文件，user、group都没有变化，但others用户要减掉2对应的‘写权限’，变为`-rw-rw-r--`

练习：修改umask为003，普通用户创建文件的权限是什么。

默认权限

练习

1. `umask` 查看值
2. `testmod`目录下创建`aa.txt`文件, `aadir`目录, 查看默认权限
3. 修改`umask`为`003`, 创建`bb.txt`, `bbdir`目录, 查看默认权限
4. 修改`umask`为`004`, 创建`cc.txt`, `ccdir`, 查看默认权限
5. 把`aa.txt`文件的所有者变换为`root`
6. 把`aa.txt`文件的所述群组变换为`root`

特殊权限SUID、SGID、SBIT

1.Set UID 简称 SUID

限制与功能：

SUID权限仅对**二进制程序**有效；

执行者对于该程序需要具有x的执行权限；

本权限仅在执行该程序的过程中有效；

► **执行者将具有该程序拥有者的权限。**

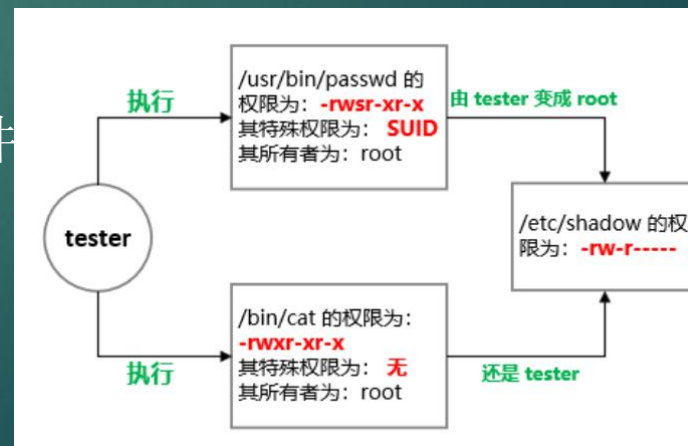
特殊权限SUID、SGID、SBIT

例：tester 用户是如何利用 SUID 权限完成密码修改的：

1. tester 用户对于 /usr/bin/passwd 这个程序具有执行权限，因此可以执行 passwd 程序
 2. passwd 程序的所有者为 root
 3. tester 用户执行 passwd 程序的过程中会暂时获得 root 权限
 4. 因此 tester 用户在执行 passwd 程序的过程中可以修改 /etc/shadow 文件
- 但是如果由 tester 用户执行 cat 命令去读取 /etc/shadow 文件确是不行的：

如何解决？？？

- 如果想让任意用户通过 cat 命令读取 /etc/shadow 文件权限就可以了：chmod 4755 /usr/bin/cat



设置 SUID

特殊权限SUID、SGID、SBIT

2.Set GID 简称 SGID

限制与功能：

- SGID对二进制程序有用；

- 程序执行者对于该程序来说，需具备x的权限；

- 执行者在执行的过程中将会获得该程序群组的支持。

目录功能：

- 用户若对于此目录具有r与x权限时，该用户能够进入目录；

- 用户在此目录下的有效群组，将会变成该目录的群组；

用途：若用户在此目录有w权限，则使用者所建立的新文件，该新文件的群组与此目录的群组相同。

特殊权限SUID、SGID、SBIT

例子：locate 命令 查询 locate.db 文件

locate `rwxr-sr-x` root locate

locate.db `rw-r-----`



特殊权限SUID、SGID、SBIT

例子：testdir目录，权限为drwxrwxr-x nick tester

建立nickfile文件， rw-rw-r-- nick nick

testdir 加sgid权限 新建nickfile2， 文件组为tester，不再是nick

总结：当用户对某一目录有写和执行权限时，该用户就可以在该目录下建立文件，如果该目录用 SGID 修饰，则该用户在这个目录下建立的文件都是属于这个目录所属的组。

特殊权限SUID、SGID、SBIT

3.Sticky Bit 简称 SBIT

限制与功能：

只针对目录有效，对于文件没有效果。

当用户对此目录具有w、x权限时

当用户在该目录下建立文件或目录时，仅有自己与root才有删除的权利。

SBIT 目前只对目录有效，用来阻止非文件的所有者删除文件。

特殊权限SUID、SGID、SBIT

SUID、SGID、SBIT设定

数字：4为SUID，2为SGID，1为SBIT

suid u+s, sgid g+s, SBIT o+t

suid rwsr-xr-x

suid/sgid rwsr-sr-x

sbit rwxr-xr-t

没有执行权限时：suid/sgid rwSr-Sr-T

练习

当一个文件的权限为 `-rwxrwxrwx` 则表示这个文件的意义为？

任何人都可以读、修改或编辑、执行，但未必能删除。

需要将文件的权限改为 `-rwxr-xr--`，如何下达指令？

`chmod u=rwx,g=rx,o=r filename` 或 `chmod 754 filename`

练习

假设当前登录用户为liu，在家目录/home/liu下有一个文件abc.exe，权限为：

```
- rwx - - - - -      1   root   root       abc.exe
```

你可以对这个文件做什么操作？

思考

► 有个目录的权限如下所示:

```
drwxr - -r - - 3  root root 4096Jun 25 08:35  .ssh
```

系统账号名称为liu，这个账号并没有支持 root 群组，请问liu对这个目录有何权限？是否可切换到此目录中？

思考

假设有个账号名称为liu，他的家目录在/home/liu/，liu对此目录具有[rwx]的权限。若在此目录下有个名为the_root.data的文件，该文件的权限如下：

```
-rwx- - - - - 1 root  root 4365 Sep 19 23:20 the_root .data
```

请问liu对此文件的权限为何？可否删除此文件？

总结

Linux的每个文件中，可分别给予拥有者、群组与其他人三种身份个别的 `rwX` 权限；

利用`ls -l`显示的文件属性中，第一个栏位是文件的权限，共有十个位元，第一个位元是文件类型， 接下来三个为一组共三组，为拥有者、群组、其他人的权限，权限有`r,w,x`三种；

如果档名之前多一个『`.`』，则代表这个文件为『隐藏文件』；

若需要`root`的权限时，可以使用 `su -` 这个指令来切换身份。处理完毕则使用 `exit` 离开 `su` 的指令环境。

更改文件的群组支援可用`chgrp`，修改文件的拥有者可用`chown`，修改文件的权限可用`chmod`

`chmod`修改权限的方法有两种，分别是符号法与数字法，数字法中`r,w,x`分数为`4,2,1`；

对文件来讲，权限的效能为：

`r`：可读取此一文件的实际内容，如读取文字档的文字内容等；

`w`：可以编辑、新增或者是修改该文件的内容(但不含删除该文件)；

`x`：该文件具有可以被系统执行的权限。

对目录来说，权限的效能为：

`r` (read contents in directory)

`w` (modify contents of directory)

`x` (access directory)

要开放目录给任何人浏览时，应该至少也要给予`r`及`x`的权限，但`w`权限不可随便给；

能否读取到某个文件内容，跟该文件所在的目录权限也有关系 (目录至少需要有 `x` 的权限)。

2.4 文件的搜索

文件通配符

▶ 星号* :

匹配任意长度的文件名字符串(包括空字符串)

▶ 问号? :

匹配任一单字符

▶ 点字符(.)

当它作为文件名或路径名分量的第一个字符时, 必须显式匹配

例: *file匹配file, makefile, 不匹配.profile文件

try*c 匹配try1.c try.c try.basic

```
[liu@localhost findtest]$ ls -al *file
-rw-rw-r--. 1 liu liu 0 9月 23 17:37 file
-rw-rw-r--. 1 liu liu 0 9月 23 17:37 makefile
-rw-rw-r--. 1 liu liu 0 9月 23 17:37 profile
```

文件通配符

► 方括号 []

匹配括号内任一字符，也可以用减号指定一个范围

例： [A-Z]* *. [ch] [Mm]akefile

► [^]

若中括号内的第一个字符为“^”，则表示反向选择

例： [^abc]表示一定有一个字符，只要是非a、b、c即可

```
[liu@localhost findtest]$ ls -al m[abc]kefile
-rw-rw-r--. 1 liu liu 0 9月 23 17:37 makefile
```

文件名通配符规则与正则表达式的规则不同，应用场合不同。不同种类shell通配符规则会略有些差别

shell 文件名通配符处理

- ▶ 文件名通配符的处理由shell完成，分以下三步
 - ▶ 在shell提示符下，从键盘输入命令，被shell接受
 - ▶ shell对所键入内容作若干加工处理，其中含有对文件通配符的展开工作(文件名生成)，生成结果命令
 - ▶ 执行前面生成的结果命令

文件名通配举例

1. 假设当前目录下只有try.c, zap.c, arc.c三文件

键入内容 `cat *.c` , 实际执行 `cat arc.c try.c zap.c` (按字典序)

2. 假设当前目录下有名为makefile的文件

键入 `rm m*e`, 实际执行 `rm makefile`

文件查找命令

1. 查找指定文件find命令

命令功能：查找满足条件的文件，功能强大。

命令格式：find [搜索的路径] [选项] [-print -exec -ok]

```
[root@server1 ~]# find . -type f -exec ls -l {} \;  
//在当前目录下查找普通文件，并以长格形式显示  
[root@server1 ~]# find /logs -type f -mtime 5 -exec rm {} \;  
//在/logs目录中查找修改时间为5天以前的普通文件，并删除。保证/logs目录存在  
[root@server1 ~]# find /etc -name "*.conf"  
//在/etc/目录下查找文件名以 ".conf" 结尾的文件  
[root@server1 ~]# find . -type f -perm 755 -exec ls {} \;  
//在当前目录下查找权限为755的普通文件并显示
```

文件查找命令

find命令的匹配表达式主要有以下几种类型。

- name filename: 查找指定名称的文件。
- user username: 查找属于指定用户的文件。
- group grpname: 查找属于指定组的文件。
- size n: 查找比n大或小的文件，符号“+n”表示查找大小大于n文件；符号“-n”表示查找大小小于n的文件；符号“nc”表示查找大小为n个byte的文件，“nk”表示n*1024bytes
- inum n: 查找索引节点号为n的文件。
- type: 查找指定类型的文件。文件类型有：b（块设备文件）、c（字符设备文件）、d（目录）、p（管道文件）、l（符号链接文件）、f（普通文件）。

文件查找命令

- atime n: 查找n天前被访问过的文件。“+n”表示超过n天前被访问的文件；“-n”表示n天内被访问的文件，n表示n天之前的[一天之内]被访问过的文件。
- mtime n: 类似于atime，但检查的是文件内容被修改的时间。
- ctime n: 类似于atime，但检查的是文件索引节点被改变的时间。
- perm mode: 查找与给定权限匹配的文件，必须以八进制的形式给出访问权限。mode表示权限为mode，-mode搜索权限包括mode的文件（范围比mode大），/mode表示搜索符合任何一个权限的文件（范围比mode小）。
- newer file: 查找比指定文件新的文件，即最后修改时间离现在较近。
- exec command {} \;; 对匹配指定条件的文件执行command命令。
- ok command {} \;; 与exec相同，但执行command命令时请求用户确认。
- print: 显示查找结果。

find使用举例(1)

- ▶ `find . -type d`

从当前目录开始查找，寻找所有目录，打印路径名

- ▶ `find ~ -name 'stud*' -type d`

指定了两个条件：名字与stud*匹配，类型为目录

两个条件逻辑“与”，必须同时符合这两个条件

- ▶ `find /var -type f -mtime -10`

从根目录开始检索最近10天之内曾经修改过的普通磁盘文件

- ▶ `find . -atime +30 -mtime +30`

从当前目录开始检索超过30天没有读过，也没有写过，而且也没有被当作命令执行过的文件，筛选出一个时间周期内不活跃的文件。

- ▶ `find . -newer file1` 从当前目录检索比file1新的文件

- ▶ `find / -perm 4755` 查找有4权限的

find使用举例(2)

► `find / -name make -exec ls -l {} \;`

-exec及随后的分号之间的内容作为一条命令执行

shell中分号有特殊含义，前面加反斜线\

{ }代表所查到的符合条件的路径名。注意，两花括号间无空格，之后的空格不可省略

对所有满足条件的文件或目录，依次执行print和exec命令

► `find ~ -name "*.txt" -ok rm {} \;`

找出用户主目录下所有的.txt文件并删除

► `find -name "*.conf" -maxdepth 1 -exec cp {} bak \;`

利用find的递归式遍历目录的功能

选项-maxdepth 1将find的搜索深度限制为最多1层，如果当前目录有子目录，就不再检索子目录

find使用举例(3)

- ▶ `find . -type f -name "*.txt" -exec cat {} \; > all.txt`

查找当前目录下所有.txt文件并把他们拼接起来写入到all.txt文件中

- ▶ `find . -type f -mtime -3 -name "*.txt" -exec cp {} old \;`

将3天内创建的.txt文件移动到old目录中

- ▶ `find / -size +100000c \(-name core -o -name '*.tmp' \) -ok rm {} \;`

-ok选项在执行指定的命令前等待用户确认

寻找大于100K的名叫core或有.tmp后缀，使用了两条件“或”（-o）及组合（圆括号）

不要遗漏了所必需的引号，反斜线和空格，尤其是圆括号前和圆括号后。圆括号是shell的特殊字符

文件查找命令

选项	说明
-name filename	查找与指定文件名相匹配的文件，支持通配符 “*” 和 “[]” 的使用；
-perm	按照文件权限来查找文件，支持完全指定和 “-” 符号、“+” 符号部分符合；
-user	按照文件属主来查找文件；
-group	按照文件所属的组来查找文件；
-mtime -n +n	按照文件的更改时间来查找文件；
-amin -n +n	按照文件的访问时间来查找文件；
-cmin -n +n	按照文件状态的更改时间来查找文件，其中：“-n”表示文件更改时间距现在n天以内，“+n”表示文件更改时间距现在n天以前；
-nogroup	查找无有效所属组的文件，即该文件所属的组在/etc/groups中不存在；
-nouser	查找无有效属主的文件，即该文件的属主在/etc/passwd中不存在；

文件查找命令

2. locate命令

命令功能：查找文件的命令，支持通配符。

命令格式：locate [选项] [文件名]

```
[root@server1 ~]# locate blockchian* //查找blockchian开头文件的位置
```

文件查找命令

3. whereis命令

命令功能：用来寻找命令的可执行文件所在的位置。

命令格式：whereis [选项] [文件名]

常用参数选项：

- b：只查找二进制文件。
- m：只查找命令的联机帮助手册部分。
- s：只查找源代码文件。

```
//查找命令rpm 的位置  
[root@server1 ~]# whereis rpm  
rpm: /bin/rpm /etc/rpm /usr/lib/rpm /usr/include/rpm /usr/share/man/man8/rpm.8.gz
```

文件查找命令

4. grep命令

命令功能：用于查找文件中包含有指定字符串的行

命令格式：

grep [选项] 查找模式 [文件名1, 文件名2, ...]

grep [选项] [-e 查找模式 | -f 文件] [文件名1, 文件名2, ...]

```
[root@server1 ~]# grep -2 root /etc/passwd
//在文件passwd中查找包含字符串“root”的行，如果找到，
显示该行及该行前后各2行的内容
```

```
[root@server1 ~]# grep "^root$" /etc/passwd
//在passwd文件中搜索只包含“root”4个字符的行
```

文件查找命令

grep命令的常用参数选项如下。

- v: 列出不匹配的行。
- c: 对匹配的行计数。
- l: 只显示包含匹配模式的文件名。
- n: 每个匹配行只按照相对的行号显示。
- i: 对匹配模式不区分大小写。

在grep命令中，字符“^”表示行的开始，字符“\$”表示行的结尾。如果要查找的字符串中带有空格，可以用单引号或双引号括起来。

文件操作命令

选项	说明
-c	只输出匹配行的计数;
-I	不区分大 小写(只适用于单字符);
-h	查询多文件时不显示文件名;
-l	查询多文件时只输出包含匹配字符的文件名;
-n	显示匹配行及 行号;
-s	不显示不存在或无匹配文本的错误信息;
-v	显示不包含匹配文本的所有行。

2.5 硬链接和符号链接

硬链接和符号链接

为了避免同样的文件在系统中被不必要地多次复制，可以通过创建文件链接，使得各个用户在自己方便的位置存取同一个文件，实现文件的共享。

硬链接和符号链接

► 硬链接

含义：是一个文件的一个或多个文件名，也就是把文件名和计算机文件系统使用的节点号链接起来。

格式：ln 源文件 创建的目标链接文件

限制：不能对目录文件做硬链接；不能在不同的文件系统之间做硬链接。

特点：创建的硬链接文件和源文件节点号相同。

创建的硬链接文件命名可以和源文件名不同。

源文件删除了，而链接文件仍具有原来特性。

创建的硬链接文件属性和源文件相同。

更改源文件或链接文件属性，则全部更改。

```
$ ln linktest linktest.hard
$ ls -l linktest linktest.hard
-rw-rw-rw- 2  kc kermit 17935 Dec 12 18:07 chapt0
-rw-rw-rw- 2  kc kermit 17935 Dec 12 18:07 intro
    (前面的几项必相同, 为什么? )
$ ls -li linktest linktest.hard
    13210  linktest
    13210  linktest.hard
```

`linktest`与`linktest.hard`同时存在时, 地位完全平等
删`linktest`文件, 则`linktest.hard`仍存在但link数减1

硬链接和符号链接

► 符号链接（软链接）

含义：软链接又叫符号链接，是将一个路径名链接到一个文件上，类似于Windows系统中的快捷方式。

格式： `ln -s 源文件 创建的目标链接文件`

特点：软链接是一个新文件，它与目标文件有不同的inode；

软链接可以对目录文件做符号链接，也可以在不同文件系统之间做符号链接；

删除源文件或目录时，不会删除链接，但链接失效；

软链接的大小是其链接文件的路径名中的字节数。

在目录长列表中，符号链接作为一种特殊的文件类型显示出来，其第一个字母是l。



```
$ ln -s linktest linktest.soft
```

```
$ ls -l linktest.soft
```

```
lrwxrwxrwx 1 guest other 8 Jul 26 16:57 linktest.soft->linktest
```

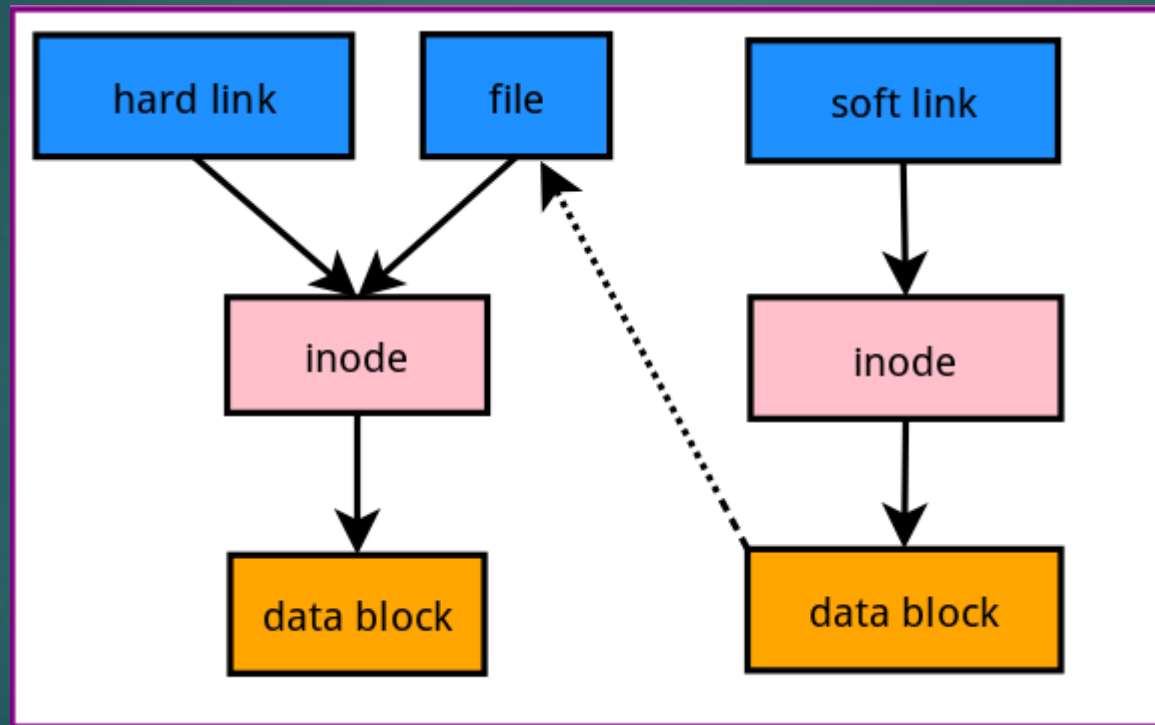
类型为l, 大小为8字节, 文件中只存放linktest字符串

文件的最后一次写时间以后不再变化

一旦建立了符号连接, 删除操作删除的是符号连接文件, 其它所有操作都将访问符号连接所引用的文件

目录的连接数

目录的link数=直属子目录数+2；新建目录为2， 上层+1



2.6 文件的归档与压缩处理

文件压缩

压缩命令 `gzip`

功能：文件压缩

使用方法：`gzip` [参数] 待压缩文件名，生成`.gz`文件

参数：

- `-c`：将压缩的数据输出到屏幕上
- `-d`：解压缩
- `-t`：检验压缩文件的完整性
- `-v`：显示出源文件/压缩文件的压缩比等信息
- `-num`：数字代表压缩等级-1最快，压缩比最差，预设值为-6.

```
[root@server1 ~]# gzip -v initial-setup-ks.cfg
initial-setup-ks.cfg:  53.4% -- replaced with initial-setup-ks.cfg.gz
[root@server1 ~]# gunzip -v initial-setup-ks.cfg.gz
initial-setup-ks.cfg.gz:  53.4% -- replaced with initial-setup-ks.cfg
```

文件压缩

压缩命令 `bzip2`

功能：文件压缩

使用方法： `bzip2` [参数] 待压缩文件名，生成 `.gz` 文件

参数：

- d：解压缩

- k：在压缩或解压缩时保留输入文件（不删除这些文件）。

- v：显示出源文件/压缩文件的压缩比等信息

- num：数字代表压缩等级-1最快，压缩比最差，预设6.

文件压缩

压缩命令 xz

功能：文件压缩

使用方法： xz [参数] 待压缩文件名，生成.gz文件

参数：

- d：解压缩
- v：显示出源文件/压缩文件的压缩比等信息
- num：数字代表压缩等级-1最快，压缩比最差，预设6.
- t：测试压缩文件的完整性
- l：列出压缩文件中的相关信息
- k：压缩但不删除，输入文件
- f：解压强制覆盖文件

文件压缩

查看纯文本文件压缩内容：

gzip压缩文件， `zcat` `zmore` `zless`

bzip2压缩文件， `bzcat` `bzmore` `bzless`

xz压缩文件， `xzcat` `xzmore` `xzcat`

文件压缩

Linux中有很多可以用于压缩的命令。Linux中常见的压缩文件扩展名如图：

*.Z	compress 程序压缩的文件；
*.zip	zip 程序压缩的文件；
*.gz	gzip 程序压缩的文件；
*.bz2	bzip2 程序压缩的文件；
*.xz	xz 程序压缩的文件；
*.tar	tar 程序打包的数据，并没有压缩过；
*.tar.gz	tar 程序打包的文件，其中并且经过 gzip 的压缩
*.tar.bz2	tar 程序打包的文件，其中并且经过 bzip2 的压缩
*.tar.xz	tar 程序打包的文件，其中并且经过 xz 的压缩

文件归档

文件归档命令tar

功能：将多个文件或目录打包在一个文件里，便于传输和保持。

命令用法：tar [options] 包名 file-list（待归档文件或目录列表）

主要参数：

- c：创建新的文件。如果用户想备份一个目录或是一些文件，就要选择这个选项。
- f：制定备份文件名。
- t：列出备份文件的内容，查看已经备份的文件。
- x：从备份文件中还原文件。
- v：显示指令执行过程。
- r：添加文件到归档包文件的尾部
- C：解压到指定的目录dir
- z：用gzip来压缩/解压缩文件，加上该选项后可以将文件进行压缩，但还原时也要用该选项进行解压缩。

文件归档

文件的归档与压缩通常一起使用

`tar -z`: 以gzip压缩或解压

`tar -j`: 以bzip2压缩或解压

`tar -J`: 以xz压缩或解压

文件归档

使用举例：

1. 对当前目录下的所有文件进行打包，生成example.tar

```
tar -cf example.tar *
```

2. 打包同时，列出包里的文件

```
tar -cvf example.tar *
```

3. 查看包example.tar内容

```
tar -tf example.tar
```

4. 还原example.tar包内容

```
tar -xvf example.tar
```

5. 将文件file5添加到包example.tar中

```
tar -rf example.tar file5
```

6. 对当前目录下的所有文件或目录进行gzip压缩，生成tmp.tar.gz

```
tar -zcvf tmp.tar.gz *
```

7. 解压缩tmp.tar.gz包

```
tar -zxvf tmp.tar.gz
```

2.7 系统调用

系统调用(System call)

- ▶ 系统调用的基本概念

- ▶ 是操作系统的内核提供的编程界面
- ▶ 是应用程序和操作系统进行交互的唯一手段
- ▶ 例如：文件操作的open, read, write, close
- ▶ 调用函数的名字，参数排列顺序，参数的类型定义，返回值的类型，以及实现的功能，都属于POSIX标准来规范的内容

系统调用(System call)

- 所有 C 函数库是相同的，而各个操作系统的系统调用是不同的。
- 函数库调用是调用函数库中的一个程序，而系统调用是调用系统内核的服务。
- 函数库调用是与用户程序相联系，而系统调用是操作系统的一个进入点
- 函数库调用是在用户地址空间执行，而系统调用是在内核地址空间执行
- 函数库调用的运行时间属于「用户」时间，而系统调用的运行时间属于「系统」时间
- 函数库调用属于过程调用，开销较小，而系统调用需要切换到内核上下文环境然后切换回来，开销较大
- 在C函数库libc中大约 300 个程序，在 UNIX 中大约有 90 个系统调用
- 函数库典型的 C 函数：system, fprintf, malloc，而典型的系统调用：chdir, fork, write, brk
- ▶ 库函数性能更高，速度大约70倍。库函数没有系统调用，性能好。有系统调用，如文件 IO 函数 fread、fwrite、fputc、fgetc 等，这些函数通常情况下性能确实比系统调用高，原因在于这些库函数使用了缓冲区，减少了系统调用的次数，因而显得性能比较高。

系统调用返回值

▶ 返回值

- ▶ UNIX的系统调用一般都返回一个整数值
- ▶ 返回值大于或等于零：成功
- ▶ 返回值为-1：失败

▶ 整型变量errno

- ▶ 标准库为errno保留存储空间，系统调用失败后填写错误代码，记录了失败原因
- ▶ `#include <errno.h>`之后，就可以直接使用errno
- ▶ `errno.h`头文件定义了许多有E前缀的宏，例如
 - ▶ `EACCESS`, `EIO`, `ENOMEM`, `EINTR`
 - ▶ 相关系统调用的手册页中有出错说明

strerror与perror

▶ strerror

`char *strerror(int errno);`

- ▶ `errno`便于程序识别错误原因，不便于操作员理解失败原因。库函数`strerror`将数字形式的错误代码，转换成一个可阅读的字符串

▶ perror

`void perror(char *string);`

- ▶ 在标准错误输出上产生一消息，描述`errno`的错误
- ▶ 格式：先打印字符串`string`，再打印出一个冒号和空格,然后再打印出这条消息和换行符

系统调用chmod

- 修改已存文件的权限

```
#include <sys/stat.h>
```

```
int chmod(char *path, mode_t mode);
```

- 功能：对文件访问权限进行修改
- 参数：
 - *path*：制定文件的路径及文件名
 - *mode*：10进制数
- 实例
 - `$ chmod 764 test.c`
 - `chmod("test.c",500);`

注： $(764)_8=500$

利用chmod系统调用实现自己的chmod命令

```
#include <stdio.h>
#include <stdlib.h>
#include <sys/stat.h>
#include <sys/types.h>
int main(int argc, char ** argv)
{ int mode, mode_u, mode_g, mode_o; char *path;
  if(argc<3)
  {printf("%s<mode num><target file>",argv[0]);exit(0);}
  mode=atoi(argv[1]); //8进制数转换为10进制数
  if(mode>777||mode<0) {printf("mode num error");exit(0);}
  mode_u=mode/100;
  mode_g=(mode-mode_u*100)/10;
  mode_o=mode-mode_u*100-mode_g*10;
  mode=mode_u*8*8+mode_g*8+mode_o;
  Path=argv[2];
```

利用chmod系统调用实现自己的chmod命令

```
int main(int argc, char ** argv)
{
    . . . . .
    mode_u=mode/100;
    mode_g=(mode-mode_u*100)/10;
    mode_o=mode-mode_u*100-mode_g*10;
    mode=mode_u*8*8+mode_g*8+mode_o;
    Path=argv[2];
    if(chmod(path, mode)==-1)
        { perror("chmod error");
          exit(1);
        }
    return 0;
}
```

假设上述c语言代码的文件名为my_chmod.c,编译生成的可执行文件为my_chmod.

```
$/my_chmod 744 a
```

习题

- 1、用户显示当前目录中内容的命令是（ ）
- 2、同MSDOS下的copy命令一样，Linux系统中的（ ）命令是将给出的文件或目录复制到另一文件或目录中，功能十分强大。
- 3、（ ）命令用于创建指定的目录名，要求创建目录的用户在当前目录中具有写权限。
- 4、能删除一目录中的一个或多个文件或目录，也可以将整个目录及其下的所有文件及目录均删除的命令是（ ）
A. mv B. rm C. mkdir D. rmdir
- 5、可以显示目录下的所有文件，包括以“.”开头的隐含文件的命令是（ ）
A. ls -a B. ls -b C. ls -c D. ls -d
- 6、（ ）命令可以用来压缩或解压缩扩展名维.z格式的文件。
A. zip B. gunzip C. tar D. compress
- 7、Linux下主要有哪些不同类型的文件？
- 8、用什么命令可以将两个文件合并成一文件？
- 9、试说明以下两个文件的拥有者与其相关的权限是什么？
(1) -rw-r--r-- 1 root root 238 Jun 18 17:22 test1.txt
(2) -rwxr-xr-- 1 user1 usergroup 5238 Jun 19 10:25 test2.txt

案例分析与操作

