

Linux操作系统

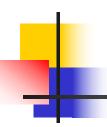
单位: 杭州电子科技大学

通信工程学院



第五章 Ubuntu文件管理

- 5.1 文件系统概述
- 5.2文件操作命令
- 5.3输入、输出重定向
- 5.4管道
- 5.5 链接



- ▶5.1.1几种文件系统介绍
- ▶5.1.2文件系统概念
- ▶5.1.3文件与目录的定义
- ▶5.1.4文件的结构、类型和属性



5.1.1几种文件系统介绍

Linux 操作系统支持的文件系统主要有ext2、ext3 和ext4。

- ▶ ext2 文件系统是伴随着 linux 一起发展起来的第二代扩展文件系统 (second extended filesystem, 缩写为ext2;
- ▶ 在ex2 的基础上增加日志就是ext3;
- ▶ ext4是第四代扩展文件系统,是Linux系统下的日志文件系统,是ext3文件系统的后继版本。



df-T-l可显示磁盘的文件系统与使用情形(有多少空闲的磁盘空间、被挂载文件系统的磁盘利用率)。或者cat/etc/fstab

```
CIIDIS
                                                  1% /run/user/1000
                        200190
                                         200084
lex@lex-virtual-machine:/devS sudo df -T -l
[sudo] password for lex:
Sorry, try again.
[sudo] password for lex:
              Type 1K-blocks
                                 Used Available Use% Mounted on
Filesystem
tmpfs
              tmpfs
                       200200
                                 1600
                                         198600
                                                  1% /run
/dev/sda3
              ext4
                     19991152 9063896
                                        9888716
                                                 48% /
tmpfs
                                                  0% /dev/shm
            tmpfs
                      1000988
                                        1000988
tmpfs
              tmpfs
                         5120
                                                  1% /run/lock
                                           5116
                                    4
tmpfs
              tmpfs
                         4096
                                           4096
                                                  0% /sys/fs/cgroup
/dev/sda2
              vfat
                                         516284
                                                  2% /boot/efi
                       524272
                                 7988
tmpfs
               tmpfs
                                                  1% /run/user/1000
                       200196
                                  112
                                         200084
 avalay victual machina / days
```

tmpfs是Linux/Unix系统上的一种基于内存的虚拟文件系统。 tmpfs可以使用内存或swap分区来存储文件。



5.1.2 文件系统概念

- ■在LINUX系统中有一个重要的概念:一切都是文件,实现了设备无关性。
- 机械硬盘:机械硬盘使用磁性存储技术,其主要组成部分是一个旋转的金属盘片,数据通过读写头在盘片上读写。机械硬盘的工作原理类似于传统的磁带机,数据的读写需要机械部件的运动。
- ▶ 固态硬盘:固态硬盘则采用闪存芯片来存储数据,数据的读写是通过电子存储单元的状态改变来实现的,不需要任何机械运动。这使得固态硬盘相比机械硬盘更加耐用,因为它没有机械部分会因摩擦而产生磨损。



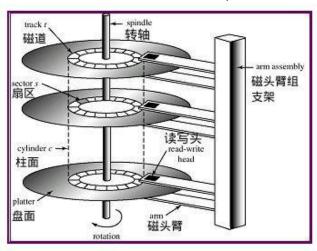


5.1.2 文件系统概念

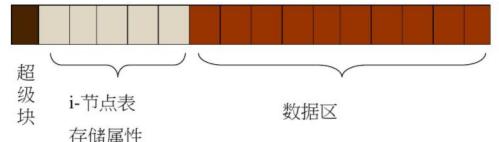
■在LINUX系统中有一个重要的概念:一切都是文件,实现了设备无关性。

Linux文件系统的要点:

- 1、物理磁盘到文件系统
 - (1) 硬盘有数个盘片, 每盘片两个面, 每个面一个磁头
 - (2) 盘片被划分为多个扇形区域即扇区
 - (3) 同一盘片不同半径的同心圆为磁道
 - (4) 不同盘片相同半径构成的圆柱面即柱面
- (5) 公式:存储容量=磁头数×磁道(柱面)数×每道扇区数×每扇区字节数
- (6) 信息记录可表示为: ××磁道(柱面), ××磁头, ××扇区







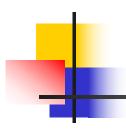
硬盘空间如何管理呢? (重点)

磁盘空间的管理, Linux将磁盘空间分为三部分进行管理:

- (1)超级块,文件系统中第一个块被称为超级块。这个块存放文件系统本身的结构信息,包括(文件系统名称、文件系统的大小和状态)。
- (2) 索引节点(inode,也叫i节点)

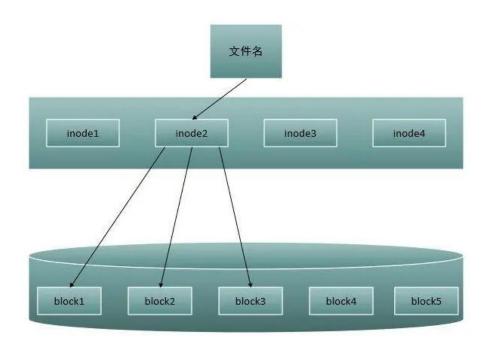
超级块的下一个部分就是i节点表。每个i节点对应一个文件/目录的结构,这个结构包含了一个文件的长度、创建及修改时间、权限、所属关系、磁盘中的位置等信息。一个文件系统维护了一个索引节点的数组,每个文件或目录都与索引节点数组中的唯一一个元素对应。系统给每个索引节点分配了一个号码,也就是该节点在数组中的索引号,称为索引节点号

(3) 数据区。文件系统的第3个部分是数据区。文件的内容保存在这个区域。 磁盘上所有块(block)的大小都一样。如果文件包含了超过一个块的内容,则 文件内容会存放在多个磁盘块中。



Linux打开文件,分为以下三步:

- ▶ 查找文件名对应的inode号
- ▶ 通过inode号获取inode信息
- ▶ 根据inode信息, 找到文件数据所在的block, 读出数据





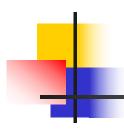
Disks from the Perspective of a File System

Dr. Marshall Kirk McKusick

Disks lie. And the controllers that run them are partners in crime.

ACM Queue, September 6, 2012, Volume 10, issue 9.

https://queue.acm.org/detail.cfm?id=2367378



2、存储介质

用以存储数据的物理设备,如软盘、硬盘、光盘、Flash盘、磁带、网络存储设备等。

3、磁盘分区

对于容量较大的存储介质来说(通常指硬盘),在使用时需要合理的规划分区,因而牵涉到磁盘的分区。常用的Linux磁盘分区命令有fdisk等。还有一些工具不是操作系统自带的,称为第三方工具,如PQ等。

4、格式化

创建新的文件系统是一个过程,通常称为初始化或格式化,这个过程是针对存储介质进行的。Ubuntu下格式化分区的命令是mkfs,有时候也可以借助第三方工具来完成此过程,而此过程是建立在磁盘分区的基础之上。



5、挂载

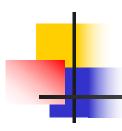
在使用磁盘分区前, 需要挂载该分区, 这相当于激活一个文件系统。

- ▶ Windows将磁盘分为若干个逻辑分区,如C盘、D盘,在各个分区中挂载文件系统。这个过程是使用内部机制完成的,用户无法探知其过程。
- ▶ Linux系统中,没有磁盘的逻辑分区(即没有C盘、D盘等),任何一种文件系统被创建后都需要挂载到某个特定的目录使用。Linux使用mount和umount命令来对文件系统进行挂载和卸载,挂载文件系统时需要明确挂载点。



显示分区的挂载点

| THE RESERVE OF THE PARTY OF THE | | ine:/dev\$ | | | | CONTRACTOR CONTRACTOR CONTRACTOR |
|--|-------|------------|---------|-----------|------|----------------------------------|
| Filesystem | Туре | 1K-blocks | Used | Available | Use% | Mounted on |
| tmpfs | tmpfs | 200200 | 1588 | 198612 | 1% | /run |
| /dev/sda3 | ext4 | 19991152 | 9063876 | 9888736 | 48% | 1 |
| tmpfs | tmpfs | 1000988 | 0 | 1000988 | 0% | /dev/shm |
| tmpfs | tmpfs | 5120 | 4 | 5116 | 1% | /run/lock |
| tmpfs | tmpfs | 4096 | 0 | 4096 | 0% | /sys/fs/cgroup |
| /dev/sda2 | vfat | 524272 | 7988 | 516284 | 2% | /boot/efi |
| tmpfs | tmpfs | 200196 | 108 | 200088 | 1% | /run/user/1000 |



外接硬件设备以文件的形式存在,在命令行模式下需要挂载再使用(较新的ubuntu系统在图形界面下均能自动挂载,字符模式下手动挂载)。

U盘插上后会以/dev/sdb1或者sdc1的形式存在。

1) 把U盘/dev/sdc1挂载到 /mnt/usb 下

mount: /mnt/usb: must be superuser to use mount. lex@lex–virtual–machine:~\$ sudo mount /dev/sdc1 /mnt/usb lex@lex–virtual–machine:~\$ _

2) 把U盘卸载(需指出挂载路径)

lex@lex–virtual–machine:~\$ sudo umount /dev/sdc1 lex@lex–virtual–machine:~\$

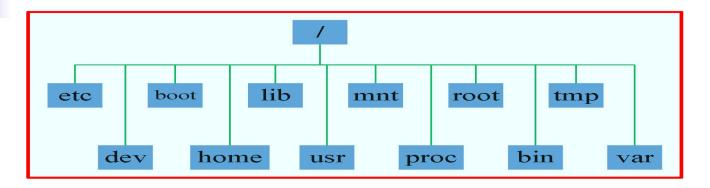


5.1.3 文件与目录的定义

- ■Linux操作系统中,以文件来表示所有的逻辑实体与非逻辑实体。逻辑实体 指文件与目录;非逻辑实体泛指硬盘、终端机、打印机等;
- ■Linux文件名由字母、标点符号、数字等构成;中间不能有空格符、路径名称符号"/"或"#*%&{}[]"等与Shell有关的特殊字符;
- ■Linux文件系统中,结构上以根文件系统(root file system)最为重要,所谓 root file system是开机时将root partition挂载在根目录(/),若无法挂载根目录,开机时就无法进入Linux系统中.



5.1.3 文件与目录的定义



- etc(etcetera):存放系统配置文件
- dev(devices):存放设备文件
- · boot:存放用于系统引导时使用的各种文件
- · home:存放用户文件的根目录
- lib(library):存放和文件系统中的程序运行所需要的共享库及内核模块
- usr(unix shared resources):存放共享的系统资源
- mnt(mount):系统管理员安装临时文件系统的安装点
- proc(processes):存放于内核/进程相关的文件
- root:超级用户目录
- bin(binaries):存放二进制可执行文件
- tmp(temporary):存放各种临时文件
- var(variable):存放系统数据文件



一级子目录:

1. /etc

本目录下存放着许多系统所需的重要配置与管理文件

- ▶ /etc/hostname是存放配置主机名字的文件;
- ▶ /etc/network/interfaces是配置修改网络接口的ip地址、子网掩码、网关的文件;
- ▶ /etc/resolv.conf 是配置DNS服务器的文件等;
- ▶ 通常在修改/etc目录下的配置文件内容后,只需重新启动相关服务,一般不用 重启系统。

```
malimei@malimei-virtual-machine:/home$ sudo /etc/init.d/networking restart
[ ok ] Restarting networking (via systemctl): networking.service.
malimei@malimei-virtual-machine:/home$
```



2. /dev

- ▶ device file (设备文件),使用者可以经由内核用来存取系统中的硬件设备,当使用设备文件时内核会辨识出输入输出请求,并传递到相应设备的驱动程序以便完成特定的动作。
- ▶该文件包含所有在linux 系统中使用的**外部设备**,每个设备在/dev目录下均有一个相应的项目。
- ▶注意Linux与windows和dos不同,不是存放外部设备的驱动程序,而是一个访问这些外部设备的端口。



3. /boot

本目录下存放与系统激活的相关文件,是系统启动时用到的程序。

例: 查看boot目录下文件, 其中

initrd.img为系统激活时最先加载的文件;

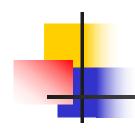
Vmlinuz*为kernel的镜像文件;

System.map*包括了kernel的功能及位置,均为重要文件,不可任意删除,内核版本不同,文件名不同。

4. /home

登录用户的主目录(\$HOME)就放在此目录下,以用户的名称作为/home 目录下各个子目录的名称。

例:建立一个用户,用户名是"lex",那么在/home目录下就有一个对应的/home/lex路径,当用户lex登录时,其所在的默认目录就是/home/lex。



5. /lib

- ▶本目录存放了许多系统激活时所需要的重要的共 享函数库, lib是库(library)的英文缩写。
- ▶几乎所有的应用程序都会用到这个目录下的共享库。例如文件名为library.so.version的共享函数库就放在/lib目录下,该函数库包含很多像GNU C library (C编译程序) 这样的重要部分。
- 例: 用命令ls查看了该目录下的库文件。
- 》说明: Linux下的库分为动态库和静态库,一般情况下.so 为共享库,用于动态连接的,.a为静态库,用于静态连接。
- ▶ . o是目标文件,相当于windows中的.obj文件
- ▶.so 为共享库,是shared object,用于动态连接的,和 dll差不多
- ▶.a为静态库,是好多个.o合在一起,用于静态连接

```
kerne1
klibc
klibc-bEuMZH6f626R-Ofd4T5tlRFtMQc.so
language-selector
libDeployPkg.so.0
libDeployPkg.so.0.0.0
libguestlib.so.0
libguestlib.so.0.0.0
libhgfs.so.0
libhgfs.so.0.0.0
libimmclient.a
libimm server.so
libimm server.so.0
libimm_server.so.0.0
libnetpbm.so.10
libnetpbm.so.10.0
libreoffice
libvgauth.so.0
libvgauth.so.0.0.0
libvmtools.so.0
libvmtools.so.0.0.0
linux
linux-boot-probes
linux-sound-base
locale
lp_solve
1sb
man-db
memtest86+
mime
modprobe.d
```



6. /usr/lib

本目录存放一些应用程序的共享函数库,其中,最重要的函数库为libc或glibc(glibc 2.x便是libc 6.x版本,标准C语言函数库),几乎所有的程序都会用到libc或glibc,因为这两个程序提供了对于Linux kernel的标准接口。

7. /mnt

这个目录在一般情况下是空的,是系统默认的挂载点(mount point),可以临时将别的文件系统挂在这个目录下,可以将硬盘、光驱、软驱分别挂载到这个目录。

在Ubuntu中使用mount命令将文件系统挂载到该目录,如果要挂载额外的文件系统到/mnt目录,需要在该目录下建立任一目录做为挂载目录。例如新建/mnt/usb目录,作为USB移动设备的挂载点。



8. /proc

- ▶目录为虚拟文件系统,不占用硬盘空间,该目录下的文件均放置于内存中。
- ▶/proc会记录系统正在运行的进程、硬件状态、内存使用的多少等信息,这些信息是在内存中,由系统自己产生的。

9. /root

系统管理用户root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。使用su命令从登录时的普通用户切换到了超级用户(这个过程是需要输入超级用户密码的)。

10. /bin

- ▶本目录存放一些系统启动时所需要的普通程序和系统程序,及一些经常被 其它程序调用的程序,是linux常用的外部命令存放的目录。例如,cat、 cp、mkdir、rm、su、tar等等,相当于DOS的外部命令,没有显示的如cd 内部命令:
- ▶ 注意内部命令和外部命令的区别; 内部命令只要系统启动就可以使用, 外部命令如果bin目录下没有, 就不能使用。

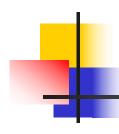


11. /tmp

本目录存放系统启动时产生的临时文件。有时某些应用程序执行中产生的临时文件也会暂放在此目录。

12. /var

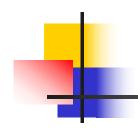
本目录存放被系统修改过的数据。在这个目录下的重要目录有 /var/log、/var/spool、/var/run等,它们分别用于存放记录文件、新闻邮件、运行时信息。



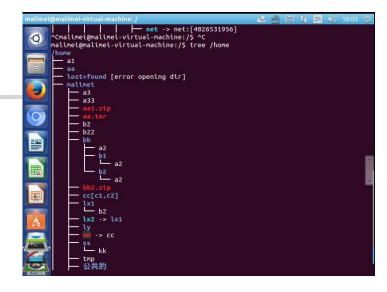
5.1.4 文件的结构、类型和属性

1. 文件结构

- ▶文件结构是文件存放在磁盘等存贮设备上的组织方法,主要体现在对文件和目录的组织上。
- ▶ Linux使用标准的目录结构,在安装的时候,安装程序就已经为用户创建了文件系统和完整而固定的目录组成形式,并指定了每个目录的作用和其中的文件类型。
- ■Linux采用的是**树型结构**。最上层是根目录,其他的所有目录都是从根目录出 发而生成的。
- ▶ 在Linux中,无论操作系统管理几个磁盘分区,这样的目录树只有一个。从结构上讲,各个磁盘分区上的树型目录不一定是并列的,因为Linux是一个多用户系统,制定一个固定的目录规划有助于对系统文件和不同的用户文件进行统一管理。



用 tree 工具命令显示/home目录结构 tree 使用之前需要用apt安装



- Linux中对文件路径的表达有两种方法——绝对路径和相对路径。
- ▶ **绝对路径:** 从根目录/开始的路径。比如/home/lex/test1, 这一路径与当前处于那个目录没有关系,表达式是固定的。
- ▶ 相对路径: 以.或..开始的,.表示用户当前操作所处的位置,而..表示上级目录,与当前目录相关。
- ▶ 当前路径: 当前工作路径, 当前用户所属的位置。
- ■常用特殊符号: '。', '。。', '~', '/'; ('。'路径时表示当前,文件时表示隐藏文件)



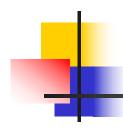
2.文件类型

1) 普通文件

普通文件就是用户通常访问的文件,由ls-l命令显示出来的属性中,第一个属性为"-"。

```
| TWXT XT X 1 TOOL TOOL | 173 20 20:21 apport Ctt | 18824 12月 7 19:02 apport Ctt | 19307 | 173 20 20:21 apport Ctt | 19307 | 173 20 20:21 apport Ctt | 19307 | 173 20 20:21 apport Ctt | 2068 1月 26 20:21 apport Ctt |
```

- 2) 纯文本文件
- ▶普通文件中,有些文件内容我们可以直接读取,如文本文件,文件的内容一般是字母、数字以及一些符号等。
- > 有些文件是为系统准备的, 如二进制文件, 可执行的文件就是这种格式。
- ▶还有些文件是为运行中的程序准备的,如数据格式的文件,Linux用户在登录系统时,会将登录数据记录在/var/log/wtmp文件内,这个文件就是一个数据文件,不能使用cat浏览。



- 3) 目录文件
- ▶目录文件就是目录,相当于windows中的文件夹。
- ▶可以使用Is-I命令显示文件的属性, 其中第一个属性为"d"的是目录文件。

```
lex@lex-virtual-machine:/home$ ls -l
total 4
drwxr-xr-x 17 lex lex 4096 4月 7 21:59 lex
lex@lex-virtual-machine:/home$
```

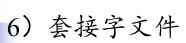
- 4) 链接文件
- > 在Linux中有两种链接方式, 符号链接和硬链接。
- ▶ 链接文件的第一个属性用"I"表示。

```
lex@lex-virtual-machine:~/lesson/lesson5$ ls -l
total 92
-rwx--x-w- 1 lex lex 47 4月 10 15:04 file1
-rw-rw-r-- 1 lex lex 66270 4月 10 16:34 test5_1.dat
drwxr-xr-x 2 root root 4096 4月 10 13:00 test5_2.dat
drwxr-xr-x 2 root root 4096 4月 10 13:00 test5_3.dat
---x-w--wx 1 root lex 39 4月 10 13:02 test5_4.dat
-rw-rw-r-- 2 lex lex 71 4月 10 20:47 xh.txt
lrwxrwxrwx 1 lex lex 5 4月 10 20:44 xs.txt -> x.txt
-rw-rw-r-- 2 lex lex 71 4月 10 20:47_x.txt
```

5) 设备文件

- ▶设备文件是Linux系统中最特殊的文件。Linux系统为外部设备提供一种标准接口,将外部设备视为一种特殊的文件,即设备文件。
- ▶ 它能够在系统设备初始化时动态的在/dev目录下创建好各种设备的文件节点, 在设备卸载后自动删除/dev下对应的文件节点。
- ▶ 在Linux系统中设备文件分为字符设备文件和块设备文件。字符设备文件是指设备发送和接收数据以字符的形式进行;而块设备文件则以整个数据缓冲区的形式进行。
- ▶ 由ls -l /dev命令显示出来的属性中,字符设备文件的第一个属性是"c",块设备文件的第一个属性是"b"。

```
|brw-rw---- 1 root disk 8, 0 4月
brw-rw---- 1 root disk 8, 1
                               4月
                                    9 20:55 sda1
brw-rw---- 1 root disk 8, 2
                               4月
                                    9 20:55 sda2
brw-rw---- 1 root disk 8, 3 4月
                                    9 20:55 sda
                                                            1 root ttv
                                                                            7, 70
                                                                                         9 20:55 VCSU6
brw-rw---- 1 root disk 8, 32 4月
                                    9 21:14 sdc
                                                                                60
                                                            2 root root
                                                                                         9 20:55 VIII
brw-rw---- 1 root disk 8, 33 4月
                                    9 21:14 sdc
                                                                            10, 63
                                                            1 root root
                                                                                         9 20:55 vga arbiter
                                                                            10, 137
                                                            1 root root
                                                                                         9 20:55 vhci
                                                                            10. 238
                                                            1 root root
                                                            1 root root
                                                                            10, 241
                                                                            10. 58
                                                            1 root root
```



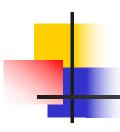
套接字文件通常用于网络数据连接。

由Is-I命令显示出来的属性中, 套接字文件的第一个属性用"s"表示。

例: 查看文件属性, 其中 "acpid.socket" 为套接字文件。

```
root
                                                 9 20:55 acpid.pid
            1 root
                                                 9 20:55 acpid socket
                                root
            1 root
                                                 9 21:13 agetty.reload
            1 root
                                root
           2 root
                                root
                                                 9 20:55
drwxr-xr-x 2 avahi
                                avahi
                                                 9 20:55 avahi-daemon
drwxr-xr-x 2 root
                                        60
                                                 9 20:55 blktd
                                root
                                        80
                                                 9 20:55 console-setup
drwxr-xr-x 2 root
                                root
                                         4
                                                9 20:55 crond.pid
-rw-r--r-- 1 root
                                root
                                                 9 20:55 crond.reboot
           1 root
                                root
                                lp
                                       100
                                                 9 20:55 cups
drwxr-xr-x 3 root
                                        60
                                                 9 20:55 dbus
drwxr-xr-x 2 root
                                root
                                        60
                                                 9 20:55 fsck
drwxr-xr-x 2 root
                                root
                                                 9 21:12 gdm3
drwx--x--x 3 root
                                adm
                                        60
           1 root
                                root
                                                 9 20:55 gdm3.pid
                                                 9 20:55 initctl
            1 root
                                root
```

- 1、源IP地址和目的IP地址以及源端口号和目的端口号的组合称为套接字。其用于标识客户端请求的服务器和服务。
- 2、套接字,是支持TCP/IP的网络通信的基本操作单元,可以看做是不同主机之间的进程进行双向通信的端点,简单的说就是通信的两方的一种约定,用套接字中的相关函数来完成通信过程。



7) 管道文件

- ▶ 管道是Linux中很重要的一种通信方式,是把一个程序的输出直接连接到另一个程序的输入,用"|"表示。
- ▶由ls-l命令显示出来的属性中,管道文件的第一个属性用"p"表示。管道一般的权限是:所属者有读写权限,而所属组与其他用户都只有读的权限。
- ▶ mkfifo则可以创建命名管道文件(在程序中使用)

```
lex@lex-virtual-machine:/dev$ sudo mkfifo 1.pipe 2.pipe
lex@lex-virtual-machine:/devS ls -l
total 0
                                0 4月 9 22:07 1 pipe
prw-r--r-- 1 root root
prw-r--r-- 1 root root
                                      9 22:07 2 pipe
                        10,175 4月
crw----- 1 root root
                                        9 20:55 agpgart
lex@lex-virtual-machine:/dev$ ls -l /dev/|grep "^p"
                                0 4月 9 22:07 1.pipe
 rw-r--r-- 1 root root
                                0 4月
 rw-r--r-- 1 root root
                                       9 22:07 2.pipe
```



3.文件属性

- 对于Linux系统的文件来说,其基本的属性有三种:读(r/4)、写(w/2)、执行(x/1)。
- ▶ 读权限:具有读取目录结构的权限,可以查看和阅读文件,禁止对其做任何的更改操作。
- ▶ 写权限:可以新建、删除、重命名、移动目录或文件(不过写权限受父目录权限控制)。
- ▶ 执行权限: 文件拥有执行权限, 才可以运行。比如二进制文件和脚本文件。 有执行权限才可以进入目录文件。
- ▶ 文件被创建时,文件所有者自动拥有对该文件的读、写和可执行权限,以便于对文件的阅读和修改。用户也可根据需要把访问权限设置为需要的任何组合。



5.2 文件操作命令

- ▶5.2.1文件内容显示命令
- ▶5.2.2目录内容显示及更改命令
- ▶5.2.3文件创建、删除命令
- ▶5.2.4目录创建、删除命令
- ▶5.2.5复制、移动命令
- ▶5.2.6压缩、备份命令
- ▶5.2.7权限管理命令
- >5.2.8文件查找命令
- ▶5.2.9统计命令



5.2.1 文件内容显示命令

- 1. cat: 串接或显示文件,不存在则新建。
- (1)cat -n /etc/network/interfaces, 带行号查看
- (2)cat a1.txt a1.txt>a3.txt, 输出(复制并覆盖a3中的原内容)
- (3)cat a1.txt a2.txt>>a3.txt, 追加(复制不覆盖a3中的原内容)
- 2.more:显示输出的内容,然后根据窗口的大小进行分页显示,在终端底部打印出"--More--"及已显示文本占全部文本的百分比。当文件较大时,文本内容会在屏幕上快速显示,more命令解决了这个问题,一次只显示一屏的文本。输入命令后显示的是文本内容的第一页,按enter显示下一行,按f/空格/下箭头显示下一页,按b/上箭头返回上一屏,按q退出显示。

例: more ***.**

- 3.less:显示输出的内容, 然后根据窗口的大小进行分页显示, 和more 相似,功能键稍有区别。
- 4.head:显示文件的前n行/n个字节,不带选项时,默认显示文件的前10行。
- 5.tail:显示文件的最后n行, 默认是10行。
- 6.echo:输出字符串到基本输出,通常就是在显示器上输出,shell编程中常用。
- 7.0d:用于输出文件的八进制、十六进制或其它格式编码的字节,通常用于显示或查看文件中不能直接显示在终端的字符。

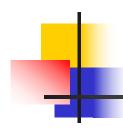


5.2.2目录内容显示及更改命令

- 1. ls
- 2. pwd
- 3. cd

5.2.3文件创建、删除命令

- 1. touch:生成空文件和修改文件存取时间。当执行了touch命令后,文件的创建时间或修改时间会更新为当前系统的时间,如果文件不存在,就会自动添加一个空文件。
- 2. rm:删除一个目录中的若干个文件或子目录, 在默认情况下, rm命令只能删除指定的文件, 而不能删除目录, 如果删除目录必须加参数-r。



5.2.4 创建目录、删除目录命令

1. mkdir:创建指定名称的目录,要求创建目录的用户在当前目录中具有写权限,并且指定的目录名不能是当前目录中已有的目录。

| 选项 | 作用 |
|----|---------------------------|
| -р | 依次创建目录,需要时创建目标目录的上级目录。 |
| -m | 设置权限模式,在建立目录时按模式指定设置目录权限。 |
| -V | 每次创建新目录都显示执行过程信息。 |

-m设置目录的权限。读、写、执行权限分别为4、2、1, 三个数的和表示对该目录的权限。 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。

| 参数 | 含义 |
|-----|---------------------|
| 600 | 只有所有者有读和写的权限 |
| 644 | 所有者有读和写的权限,组用户有读的权限 |
| 666 | 每个人都有读和写的权限 |
| 700 | 只有所有者有读和写以及执行的权限 |
| 777 | 每个人都有读和写以及执行的权限 |

2.rmdir:删除空目录,如果有文件需要先删除文件。可使用空格来分隔多个目录名(成为目录列表),同时删除多个目录。

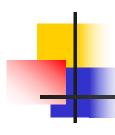


5.2.5复制、移动命令

- **√**1. cp
- ✓2. mv

5.2.6 压缩、备份命令

- ✓ 1. tar
- ✓2. gzip
- ✓3. gunzip
- **√**4. zip
- ✓ 5. unzip



5.2.7 权限管理命令

1.chgrp: change group,改变文件或目录的所属组。要被改变的组名必须在/etc/group文件内存在才可以,默认情况下只有root权限才能执行。

2.chown: change owner,将文件或目录的所有者改变为指定用户,还可以修改文件所属组群。

3.chmod:change mode,改变文件或目录的访问权。



1.chgrp

| 选项 | 作用 |
|----------------|------------------------------|
| –R | 处理指定目录以及其子目录下的所有文件。 |
| -с | 当发生改变时输出调试信息。 |
| -f | 不显示错误信息。 |
| -v | 运行时显示详细的处理信息。(过程) |
| -dereference | 作用于符号链接的指向,而不是符号链接本身。 |
| no-dereference | 作用于符号链接本身。 |
| reference | =文件/目录1 文件/目录2,改变文件2所属群组,使其与 |
| | 文件1相同 |

例:改变文件的群组属性并显示过程,test5_4的组由lex改为root.

```
lex@lex-virtual-machine:~/lesson/lesson5$ ll
total 88
                             4月 10 13:02 /
drwxrwxr-x 4 lex lex
                        4096
                             4月 10 10:48 /
drwxrwxr-x 18 lex lex
                        4096
                             4月 10 10:58 test5_1.dat
-rw-rw-r-- 1 lex lex
                      66263
                             4月 10 13:00 test5_2.dat/
drwxr-xr-x 2 root root
                       4096
                             4月 10 13:00 test5 3.dat/
drwxr-xr-x 2 root root 4096
                             4月 10 13:02 test5 4.dat
-rw-r--r-- 1 root root
```

```
lex@lex-virtual-machine:~/lesson/lesson5$ sudo chgrp -v lex test5_4.dat
[sudo] password for lex:
changed group of 'test5_4.dat' from root to lex
lex@lex-virtual-machine:~/lesson/lesson5$ ll
total 88
drwxrwxr-x 4 lex lex 4096 4月 10 13:02 /
                             4月 10 10:48 /
drwxrwxr-x 18 lex lex
                        4096
-rw-rw-r-- 1 lex lex 66263 4月 10 10:58 test5_1.dat
                            4月 10 13:00 test5 2 dat/
drwxr-xr-x 2 root root 4096
                             4月 10 13:00 test5 3.dat/
drwxr-xr-x 2 root root 4096
rw-r--r-- 1 root lex
                          39 4月 10 13:02 test5 4.dat
 ex@lex-virtual-machine:~/lesson/lesson5S
```

更改前

sudo chgrp –v lex test5 4.dat

更改后



2.chown

| 选项 | 作用 |
|----------------|-------------------------------|
| -c | 显示更改的部分信息。 |
| -f | 忽略错误信息。 |
| –R | 处理指定目录以及其子目录下的所有文件,递归式地改变指定 |
| | 目录及其下的所有子目录和文件的拥有者。 |
| _v | 显示详细的处理信息。 |
| reference | 把指定的目录/文件作为参考,把操作的目录/文件设置成参考文 |
| =目标目录或文件原文件或目录 | 件/目录相同所有者和群组。 |
| | |

例:改变文件的所属人, test5_1_lex2.dat由lex1改为lex

```
lex1@lex-virtual-machine:-S ll
total 32
drwxr-xr-x 2 lex1 lex1 4096 4月 10 13:19 /
drwxr-xr-x 4 root root 4096
                           4月 10 13:13 /
-rw-r--r-- 1 lex1 lex1 220 4月 10 13:13 .bash logout
                           4月 10 13:13 .bashrc
-rw-r--r-- 1 lex1 lex1 3771
-rw-r--r-- 1 lex1 lex1 807 4月 10 13:13 .profile
                               10 13:17 test5_1_lex1.dat
 rw-r--r-- 1 root root
                        45 4月 10 13:19 test5 1 lex2.dat
 rw-rw-r-- 1 lex1 lex1
 rw----- 1 lex1 lex1
                           4月 10 13:19 .viminfo
                       844
```

```
lex@lex-virtual-machine:/home/lexi$ sudo chown lex:lex test5_1_lex2.dat lex@lex-virtual-machine:/home/lexi$ ll total 32 drwxr-xr-x 2 lex1 lex1 4096 4月 10 13:19 // drwxr-xr-x 4 root root 4096 4月 10 13:13 ../
-rw-r--r-- 1 lex1 lex1 220 4月 10 13:13 .bash_logout
-rw-r--r-- 1 lex1 lex1 3771 4月 10 13:13 .bashrc
-rw-r--r-- 1 lex1 lex1 807 4月 10 13:13 .profile
-rw-r--r-- 1 root root 38 4月 10 13:17 test5_1_lex1.dat
-rw-rw-r-- 1 lex lex 45 4月 10 13:19 test5_1_lex2.dat
-rw----- 1 lex1 lex1 844 4月 10 13:19 .viminfo
```

切换lex后执行:

更改前

sudo chown lex:lex test5_1_lex2.dat

更改后



3.chmod

| 选项 | 作用 |
|---------------------|-----------------------|
| -v | 运行时显示详细的处理信息。 |
| -с | 显示改变部分的命令执行过程。 |
| - f | 不显示错误信息。 |
| –R | 将指定目录下的所有文件和子目录作递归处理。 |
| -reference=<目录或者文件> | 设置成与指定目录或者文件具有相同的权限。 |

(1)符号模式:chmod [选项] [who] operator [permission] files

| | 选项 | 作用 |
|-----|----|-------------|
| | a | 所有用户均具有的权限 |
| who | 0 | 其他人 |
| | u | 文件或目录的当前所有者 |
| | g | 文件或者目录的当前群组 |

| + | 增加权限 |
|---|----------|
| _ | 取消权限 |
| = | 设定权限 |
| | operator |

| permission | | | | |
|------------|------|-------|---------|---|
| nermission | | | | |
| nermiggion | | • • • | . ~ : . | |
| | pern | 118 | SS10 | m |

| 选项 | 作用 |
|----|------|
| r | 读权限 |
| w | 写权限 |
| X | 执行权限 |



(2)绝对模式:chmod [选项] mode files

其中mode代表权限等级,由3个八进制数表示。

0 [000] 无任何权限 1 [001] 执行权限

2[010]写权限 3[011]写、执行权限

4[100] 只读权限 5[101] 读、执行权限

6[110]读、写权限 7[111]读、写、执行权限

| 参数 | 含义 |
|-----|---------------------|
| 600 | 只有所有者有读和写的权限 |
| 644 | 所有者有读和写的权限,组用户有读的权限 |
| 666 | 每个人都有读和写的权限 |
| 700 | 只有所有者有读和写以及执行的权限 |
| 777 | 每个人都有读和写以及执行的权限 |



```
$ls -1 或者11 查看当前目录下所有文件及子目录的详细信息
```

```
1 root disk
CFW-FW----
                           21. 1
                                       10 10:43 sq1
                                    4月 10 10:43
drwxrwxrwt
            2 root root
                                    4月
                                       10 10:43 snapshot
            1 root root
                           10, 231
                                    4月
drwxr-xr-x
           3 root root
                               200
                                       10 10:43 snd/
                                  4月
brw-rw----+ 1 root cdrom
                                       10 10:43 STO
                           11,
                                 0
                                    4月 10 10:43 stderr -> /proc/self/fd/2
           1 root root
lrwxrwxrwx
                                15
              owner group
                                        修改时间
文件类型及权限
                           文件字节数
```

目录:包含文件数文件:文件链接数

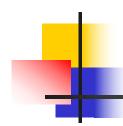
说明: 第1列(文件权限)的第2位开始表示文件的访问权限:

• r表示文件可以被读 (read)

第1列第1位表示?

- w表示文件可以被写(write)
- x表示文件可以被执行(execute)
- -表示相应的权限还没有被授予。
- 每3个字母为一组,分别表示所有者(user),所有者所在组(group)以及除上述两种外任何用户/组(others)对文件的操作权限。

So, brw-rw----+:b表示?,rw-表示?,rw-表示?,---表示?,+表示该文件/目录开启了ACL(Access Control Lists)权限



字符串权限和八进制权限对应关系

其中mode代表权限等级,由3个八进制数表示。

0 [000] 无任何权限 1 [001] 执行权限

2 [010] 写权限 3 [011] 写、执行权限

4[100] 只读权限 5[101] 读、执行权限

6[110] 读、写权限 7[111] 读、写、执行权限

| 参数 | 含义 |
|-----|---------------------|
| 600 | 只有所有者有读和写的权限 |
| 644 | 所有者有读和写的权限,组用户有读的权限 |
| 666 | 每个人都有读和写的权限 |
| 700 | 只有所有者有读和写以及执行的权限 |
| 777 | 每个人都有读和写以及执行的权限 |

例:八进制权限为675,即

| | 6 | | | 7 | | | 5 | |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 1 |
| r | W | _ | r | W | X | r | _ | X |

字符串 rwxr-xr--对应的八进制权限为?



例1: 符号模式下添加可执行权限

#chmod a+x file1 给file1文件所有用户增加可执行权限

```
lex@lex-virtual-machine:~/lesson/lesson5$ ls -al file1
-rw-rw-r-- 1 lex lex 47 4月 10 15:04 file1
lex@lex-virtual-machine:~/lesson/lesson5$ chmod a+x file1
lex@lex-virtual-machine:~/lesson/lesson5$ ls -al file1
-rwxrwxr-x 1 lex lex 47 4月 10 15:04 file1
```

例2: 符号模式下设置文件仅仅可执行

#chmod u=x file1 , 为当前所有者限定执行权限

说明:使用"=x"选项时,文件file1所有者的权限从"rwx"变为"--x",即原有的权限被撤销,重新设置为仅可执行。

```
.ex@lex-virtual-machine:~/lesson/lesson5$ ls -al file1
rwxrwxr-x 1 lex lex 47 4月 10 15:04 file1
.ex@lex-virtual-machine:~/lesson/lesson5$ chmod u=x file1
.ex@lex-virtual-machine:~/lesson/lesson5$ ls -al file1
--xrwxr-x 1 lex lex 47 4月 10 15:04 file1
```

例3:设置多重权限: chmod u+r,g-wx,o-x file1

执行后,结果变为:

```
lex@lex-virtual-machine:~/lesson/lesson5$ chmod u+r,g-wx,o-x file1
lex@lex-virtual-machine:~/lesson/lesson5$ ls -al file1
-r-xr--r-- 1 lex lex 47 4月 10 15:04 file1
```



例4:绝对模式下设置文件权限

\$ chmod 712 file1

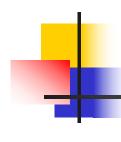
设置a的权限:所有者-读、写、执行权限,同组人-可执行权限,其它人-写权限说明:符号模式和绝对模式不能混着用,要遵循各自的格式。

```
lex@lex-virtual-machine:~/lesson/lesson5$ ls -al file1
-r-xr--r-- 1 lex lex 47 4月 10 15:04 file1
lex@lex-virtual-machine:~/lesson/lesson5$ chmod 712 file1
lex@lex-virtual-machine:~/lesson/lesson5$ ls -al file1
-rwx--x-w- 1 lex lex 47 4月 10 15:04 file1
```

思考题

1、某文件执行某命令后权限发生了如下变化,请写出该命令(符号模式和绝对模式)以及该文件权限变化前后的八进制权限。

```
命令前 -rw-r--r-- 1 root lex 39 4月 10 13:02 test5_4.dat 命令后 ---x-w--wx 1 root lex 39 4月 10 13:02 test5_4.dat
```



查找命令

- 1. whereis:寻找命令的二进制文件,同时也会找到其帮助文件。寻找一个外部命令所在的位置。但是whereis命令只能用于程序名的搜索,而且只搜索二进制文件(参数-b)、搜索说明文件(参数-m)和源代码文件(参数-s)。
- 2. man:查看命令的帮助手册, 其内容包括命令语法、各选项的意义以及相关命令等。
- 3. find:寻找文件或目录的位置。可以按文件名、部分文件名、文件特征(权限、所有者、时间、类型、长度等)查找。
- 4. locate:寻找文件和目录。在系统内通过搜寻数据库查找指定档案,相对find命令查找速度快。
- 5. grep:用来查找、替换那些符合某个模式(规则)的文本。



grep

- ■功能描述: **正则表达式**,又称规则表达式。计算机科学的一个概念。正则表达式通常被用来查找、替换那些符合某个模式(规则)的文本。
- ■语法: grep [选项] 匹配字符串文件列表
- ■选项:

| 选项 | 作用 |
|----|--------------------------------|
| -V | 列出不匹配串或正则表达式的行, 即显示不包含匹配文本的所有行 |
| -c | 对匹配的行计数 |
| -l | 只显示包含匹配的文件的文件名 |
| -h | 查询多文件时不显示文件名,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 |



例1:搜索文件包含's'的内容

\$grep -n "s" 1.txt 搜索文件1.txt中包含s的行,并显示行号。

例2:搜索文件的内容

\$grep -n '-' a

搜索文件a中,包含"-"的行,并显示行号

\$grep -vn '-' a

搜索文件a中,不包含"-"的行,并显示行号

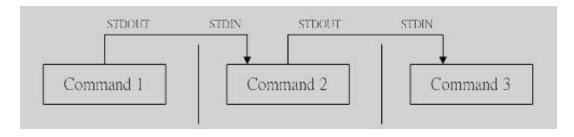
\$grep -n '*' a

搜索文件a中,包含"*"的行,并显示行号



5.4管道

- ■管道 (Pipeline): 一个由标准输入输出链接起来的进程集合, 是一个连接两个进程的连接器
- ■管道的命令操作符是: "|",它将操作符左侧命令的输出信息(stdout)作为操作符右侧命令的输入信息(stdin)。
- ■例: command1正确输出,作为command2的输入,然后comand2的输出作为comand3的输入,comand3输出则会直接显示在屏幕上。



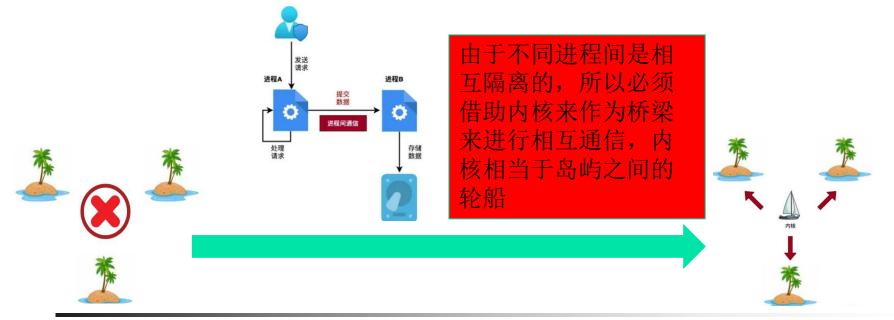
- ■管道类似于输入输出重定向,但是管道触发的是两个子进程,而重定向执行的是一个进程。如果是命令间传递参数使用管道,如果输出结果需要重定向到文件,使用输出重定向。
- 管道使用时需要注意以下几点:
- 1) 管道是单向的,一端只能输入,另一端只能用于输出,遵循"先进先出"原则。
- 2) 管道命令只处理前一个命令的正确输出,如果输出的是错误信息,则不处理。
- 3) 管道分为普管道操作符右侧命令通管道和命名管道两种。这里所讲的管道是普通管道。



5.4管道

实际上: 出于安全的考虑,不同进程之间的内存空间是相互隔离的,也就是说 进程A 是不能访问 进程B 的内存空间,反之亦然。如果不同进程间能够相互访问和修改对方的内存,那么当前进程的内存就有可能被其他进程非法修改,从而导致安全隐患。不同的进程就像是大海上孤立的岛屿,它们之间不能直接相互通信

但是:某些场景下,不同进程间需要相互通信,比如:进程A负责处理用户的请求,而进程B负责保存处理后的数据。那么当进程A处理完请求后,就需要把处理后的数据提交给进程B进行存储。此时,进程A就需要与进程B进行通信。





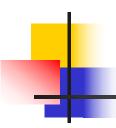
■例1: 查找文件内容并显示、统计

\$cat a1.txt|grep "a" 查找文件a1.txt中包含a的字符并显示 \$cat a1.txt|grep "a" |wc -1

查找文件al.txt中包含a的字符串并统计行数, 随后显示统计结果

可以看出系统将前一个命令的输出也就是al.txt的内容输入到后一个命令,作为grep的输入,最终的查询结果会显示在屏幕上。

```
lex@lex-virtual-machine:~/lesson/lesson5$ vi test5_1.dat
lex@lex-virtual-machine:~/lesson/lesson5$ cat test5_1.dat|grep "fasdf1"|wc -l
2
lex@lex-virtual-machine:~/lesson/lesson5$ cat test5_1.dat|grep "fasdf1"
fasdf1
fasdf1
```



■例2: 查找文件内容进行统计后显示

\$cat 1.txt|grep "s" |wc-l 统计文件1.txt中包含s的字符行数,并显示

\$cat 1.txt|grep "s" |wc-w 统计文件1.txt中包含s的字数,并显示 \$cat 1.txt|grep "s" |wc-c 统计文件1.txt中包含s的字节数,并显示

```
malimei@malimei-virtual-machine:~$ wc -lcw 1.txt
4 4 9 1.txt
malimei@malimei-virtual-machine:~$ cat 1.txt
ss
s
d
malimei@malimei-virtual-machine:~$ cat 1.txt|grep "s"|wc -l
3
malimei@malimei-virtual-machine:~$ cat 1.txt|grep "s"|wc -w
3
malimei@malimei-virtual-machine:~$ cat 1.txt|grep "s"|wc -c
7
```



5.5 链接

- ▶5.5.1链接
- ■链接是一种在共享文件和访问它的用户的若干目录项之间建立联系的方法。
- ■例如: 当我们需要在不同的目录, 用到相同的文件时, 我们不需要在每一个需要的目录下都放一个必须相同的文件, 我们只要在某个固定的目录, 放上该文件, 然后在其它的目录下链接(link)它就可以, 不必重复的占用磁盘空间。

▶5.5.2索引节点

索引节点(inode)。在Linux系统中,内核为每一个新创建的文件分配一个inode(索引结点),每个文件都有一个唯一的inode号,我们可以将inode简单理解成一个指针,它永远指向本文件的具体存储位置。文件属性保存在索引结点里,在访问文件时,索引结点被复制到内存里,从而实现文件的快速访问。系统是通过索引节点(而不是文件名)来定位每一个文件。

[exclex-virtual-machine://www.lesson.s.s.s.ls-il]

▶5.5.3两种链接

(1)硬链接(Hard Link):硬链接就是一个指针,指向文件索引节点。但系统并不为它重新分配inode,不占用实际空间,硬链接不能链接到不同目录和不同文件系统的文件。

524342 -rwx--x-w- 1 lex lex 47 4月 10 15:04 file1

In命令来建立硬链接

(2)软链接(Soft Link): 软链接又叫符号链接,这个文件包含了另一个文件的路径名,系统会为其重新分配inode,类似于Windows中的快捷方式。软链接可以是任意文件或目录。

In-s命令来建立软链接



5.5.4 ln 命令

- ■功能描述: 为某一个文件在另外一个位置建立一个同步的链接。In命令会保持每一处 链接文件的同步性。
- ■语法: ln [参数][源文件或目录][目标文件或目录]
- ■选项:

| 选项 | 作用 |
|----|--------------------------------------|
| -s | 软链接 (符号链接) |
| -b | 删除,覆盖以前建立的链接 |
| -d | 允许超级用户制作目录的硬链接 |
| -f | 强制执行 |
| -i | 交互模式,文件存在则提示用户是否覆盖 |
| -n | 把符号链接视为一般目录 |
| -v | 显示详细的处理过程 |
| -S | "-S<字尾备份字符串>"或 "suffix=<字尾备份字符串>" |
| -V | "-V<备份方式>"或 "version-control=<备份方式>" |



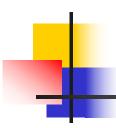
■例1: 在同一目录下, 建立文件的符号链接和硬链接 \$ln -s x.txt xs.txt 建立x.txt 文件的符号链接xs.txt \$ln x.txt xh.txt 建立x.txt 文件的硬链接xh.txt

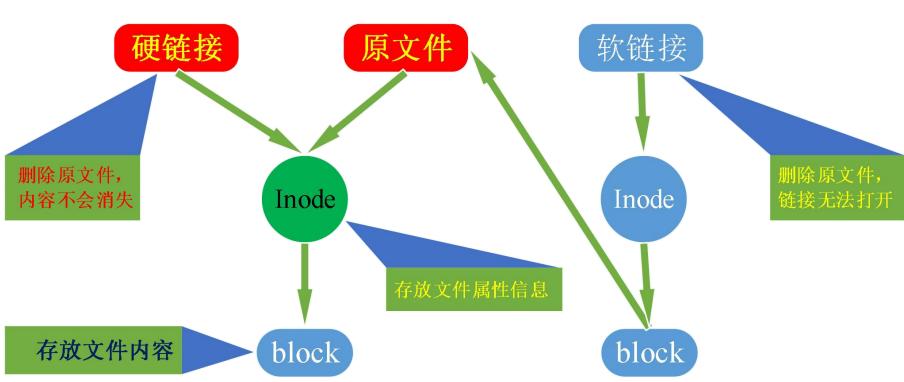
```
lex@lex-virtual-machine:~/lesson/lesson5$ ls -il
total 92
                               47 4月 10 15:04 file1
524342 -rwx--x-w- 1 lex lex
                            66270 4月 10 16:34 test5 1.dat
524345 -rw-rw-r-- 1 lex lex
524328 drwxr-xr-x 2 root root
                                   4月
                             4096
                                      10 13:00 test5 2 dat
                                   4月 10 13:00 test5 3.dat
524329 drwxr-xr-x 2 root root
                             4096
                                   4月
524333 ---x-w--wx 1 root lex
                                      10 13:02 test5 4.dat
                               39
                                  4月
524331 -rw-rw-r-- 2 lex lex
                               7
                                      10 20:44 xh.txt
                               5
                                  4月 10 20:44 xs.txt -> x.txt
524309 lrwxrwxrwx 1 lex lex
                                  4月 10 20:44 x.txt
524331 -rw-rw-r-- 2 lex lex
```



■硬链接和软链接的区别:

- (1) 硬链接原文件/链接文件共用一个inode号,说明他们是同一个文件,而软链接原文件/链接文件拥有不同的inode号,表明他们是两个不同的文件;
- (2) 在文件属性上软链接是链接文件,而硬链接没有写出来,在本质上硬链接文件和原文件是完全平等关系;
- (3) 链接数目是不一样的, 软链接的链接数目不会增加; 每添加一个硬链接, 文件的链接数就加1。
- (4) 文件大小是不一样的, 硬链接文件显示的大小是跟原文件是一样的。 而这里软链接显示的大小与原文件就不同。
- (5) 软链接没有任何文件系统的限制,任何用户可以创建指向目录的符号链接。





建立文件的硬链接, 删除原文件后, 链接文件没有影响。 建立文件的软链接, 删除源文件后, 链接的文件不能使用。



课本第168页

- 1、填空题
- 2、问答题