

浙江大学实验报告

课程名称: Linux 应用技术基础 实验类型: 验证型

实验项目名称: Linux shell 基本命令

学生姓名: 应承峻 专业: 软件工程 学号: 3170103456

电子邮件地址: 3170103456@zju.edu.cn

实验日期: 2019 年 5 月 3 日

一、实验环境

计算机配置: 处理器 英特尔 Core i7-8750H @ 2.20GHz 六核
内 存 16 GB (三星 DDR4 2667MHz)
主硬盘 PeM280240GP4C15B (240 GB/固态硬盘)
显 卡 Nvidia GeForce GTX 1060 (6 GB)

操作系统环境: Windows 10 64 位 (DirectX 12)

Linux 版本: ubuntu-17.04

二、实验内容和结果及分析

1. **查阅资料, 回答问题:** 什么是软件许可证 (Software License)? 通过互联网收集国际上常用的各种软件许可证, 并作简单的讨论和分析?

软件许可证是一种格式合同, 由软件作者与用户签订, 用以规定和限制软件用户使用软件 (或其源代码) 的权利, 以及作者应尽的义务。

国际中常用的软件许可证包括 GPL 许可证、BSD 许可证、MPL 许可证等。

GPL 是 GNU 通用公共许可协议, 适用于大多数自由基金会的软件, 但是它的缺点是许可证的要求太过严苛。为了更好地平衡开发者对源代码的需求和他们利用源代码获得的利益, MPL 许可证随之出现。MPL 许可证允许一个企业在自己已有的源代码库上加一个接口, 除了接口程序的源代码以 MPL 许可证的形式对外许可外, 源代码库中的源代码就可以不用 MPL 许可证的方式强制对外许可。这也允许了借鉴别人的源代码进行商业软件开发的行为。而相较于 GPL 许可证和 MPL 许可证的严格性, BSD 许可证较为宽松, 只需要附上许可证的原文和开发者自己的版权资料即可。

2. 进入 Linux 系统，在终端或命令行窗口中，输入如下 Linux 命令，记录下输出结果（\$为命令行提示符，您的 Linux 系统可能是其它的提示符）。

- a. \$ ls
- b. \$ pwd
- c. \$ xy #只看结果，不要问为什么没有 xy 命令
- d. \$ cd ..
- e. \$ pwd
- f. \$ cd
- g. \$ pwd
- h. \$ cd /usr/include
- i. \$ ls
- j. \$ cd

```
ying@ying:~$ ls
examples.desktop  公共的  模板  视频  图片  文档  下载  音乐  桌面
ying@ying:~$ pwd
/home/ying
ying@ying:~$ xy
xy: 未找到命令
ying@ying:~$ cd ..
ying@ying:/home$ pwd
/home
ying@ying:/home$ cd
ying@ying:~$ pwd
/home/ying

ying@ying:~$ cd /usr/include
ying@ying:/usr/include$ ls
aio.h          fnmatch.h      mntent.h       rdma            sysexits.h
aliases.h      fstab.h         monetary.h     re_comp.h      syslog.h
alloca.h       fts.h          mqueue.h      regex.h        tar.h
argp.h         ftw.h          mtd            regexp.h       termio.h
argz.h         _G_config.h    net            reglib         termios.h
ar.h           gconv.h        netash         resolv.h       tgmath.h
arpa           getopt.h       netatalk       rpc            thread_db.h
asm-generic    glob.h         netax25        rpcsvc         time.h
assert.h       gnumake.h      netdb.h        sched.h        ttyent.h
byteswap.h     gnu-versions.h neteconet      scsi           uapi
c++            grp.h          netinet        search.h       uchar.h
complex.h      gshadow.h      netipx         semaphore.h    ucontext.h
cpio.h         iconv.h        netiucv        setjmp.h       ulimit.h
crypt.h        ifaddrs.h      netpacket      sgtty.h        unistd.h
ctype.h        inttypes.h     netrom         shadow.h       ustat.h
dirent.h       langinfo.h     netrose        signal.h       utime.h
dlfcn.h        lastlog.h      nfs            sound           utmp.h
drm            libgen.h       nl_types.h     spawn.h        utmpx.h
elf.h          libintl.h      nss.h          stab.h         values.h
endian.h       libio.h        obstack.h      stdc-predef.h  video
envz.h         limits.h       openvpn        stdint.h       wait.h
err.h          link.h         paths.h        stdio_ext.h    wchar.h
errno.h        linux          poll.h         stdio.h        wctype.h
error.h        locale.h       printf.h       stdlib.h       wordexp.h
execinfo.h     malloc.h       pthread.h      string.h        X11
fcntl.h        math.h         pty.h          strings.h       x86_64-linux-gnu
features.h     mcheck.h      pwd.h          stropts.h      xen
fenv.h         memory.h       python3.5m    sudo_plugin.h  xlocale.h
fmtmsg.h       misc           syscall.h     xorg
```

3. 可以使用 man 和 info 命令来获得每个 Linux 命令的帮助手册，用 man ls, man passwd, info pwd 命令得到 ls、passwd、pwd 三个命令的帮助手册。也

可以使用：命令名 --help 格式来显示该命令的帮助信息，如 who --help，试一下这些命令。（实验报告中获取部分截图）

```
ying@ying: ~
LS(1)
NAME
ls - list directory contents
SYNOPSIS
ls [OPTION]... [FILE]...
DESCRIPTION
List information about the FILES (the current directory by default).
Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.
Mandatory arguments to long options are mandatory for short options too.
-a, --all
do not ignore entries starting with .
-A, --almost-all
do not list implied . and ..
--author
Manual page ls(1) line 1 (press h for help or q to quit)
```

```
ying@ying: ~
PASSWD(1)
名称
passwd - 更改用户密码
大纲
passwd [选项] [登录]
描述
passwd 命令用来更改用户账户的密码。普通用户通常只更改其自己账户的密码，而超级用户可以更改任何账户的密码。passwd 也更改账户或相关的密码有效期。
密码更改
如果有旧密码，首先提示用户输入旧密码。加密这个密码然后和存储的密码进行比较。用户只有一次机会输入正确密码。允许超级用户略过这个步骤，以便更改忘记了的密码。
输入了密码之后，会检查密码年龄信息，以查看此时是否允许用户更改密码。如果不允许，passwd 拒绝更改密码，然后退出。
提示用户输入两次新密码。第二次和第一次进行比较，并且需要相同才能更改密码。
Manual page passwd(1) line 1 (press h for help or q to quit)
```

```
ying@ying: ~
PWD(1)
NAME
pwd - print name of current/working directory
SYNOPSIS
pwd [OPTION]...
DESCRIPTION
Print the full filename of the current working directory.
-L, --logical
use PWD from environment, even if it contains symlinks
-P, --physical
avoid all symlinks
--help display this help and exit
--version
output version information and exit
If no option is specified, -P is assumed.
Manual page pwd(1) line 1 (press h for help or q to quit)
```

```
ying@ying:~$ ls --help
用法: ls [选项]... [文件]...
List information about the FILES (the current directory by default).
Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.
必选参数对长短选项同时适用。
-a, --all
不隐藏任何以 . 开始的项目
-A, --almost-all
列出除 . 及 .. 以外的任何项目
--author
与 -l 同时使用时列出每个文件的作者
-b, --escape
以八进制溢出序列表示不可打印的字符
--block-size=SIZE
scale sizes by SIZE before printing them; e.g.,
"--block-size=M" prints sizes in units of
1,048,576 bytes; see SIZE format below
-B, --ignore-backups
do not list implied entries ending with ~
-c
with -lt: sort by, and show, ctime (time of last
modification of file status information);
with -l: show ctime and sort by name;
otherwise: sort by ctime, newest first
-C
list entries by columns
--color[=WHEN]
colorize the output; WHEN can be 'always' (default
if omitted), 'auto', or 'never'; more info below
-d, --directory
list directories themselves, not their contents
-D, --dired
generate output designed for Emacs' dired mode
-f
do not sort, enable -aU, disable -ls --color
```

```
ying@ying:~$ passwd --help
用法: passwd [选项] [登录]

选项:
-a, --all          报告所有帐户的密码状态
-d, --delete       删除指定帐户的密码
-e, --expire       强制使指定帐户的密码过期
-h, --help         显示此帮助信息并推出
-k, --keep-tokens  仅在过期后修改密码
-i, --inactive INACTIVE 密码过期后设置密码不活动为 INACTIVE
-l, --lock         锁定指定的帐户
-n, --mindays MIN_DAYS 设置到下次修改密码所需等待的最短天数
为 MIN_DAYS
-q, --quiet        安静模式
-r, --repository REPOSITORY 在 REPOSITORY 库中改变密码
-R, --root CHROOT_DIR  chroot 到的目录
-S, --status       报告指定帐户密码的状态
-u, --unlock       解锁被指定帐户
-W, --warndays WARN_DAYS 设置过期警告天数为 WARN_DAYS
-x, --maxdays MAX_DAYS 设置到下次修改密码所需等待的最多天数
为 MAX_DAYS

ying@ying:~$ pwd --help
pwd: pwd [-LP]
Print the name of the current working directory.

Options:
-L          print the value of $PWD if it names the current working
            directory
-P          print the physical directory, without any symbolic links

By default, 'pwd' behaves as if '-L' were specified
```

4. 使用 whoami 命令找到用户名。使用下面的命令显示有关你计算机系统信息：uname（显示操作系统的名称），uname -n（显示系统域名），uname -p（显示系统的 CPU 名称）

- 1) 您的用户名是什么? **ying**
- 2) 你的操作系统名字是什么? **Linux**
- 3) 你计算机系统的域名是什么? **ying**
- 4) 你计算机系统的 CPU 名字是什么? **x86_64**

```
ying@ying: ~
ying@ying:~$ whoami
ying
ying@ying:~$ uname
Linux
ying@ying:~$ uname -n
ying
ying@ying:~$ uname -p
x86_64
```

5. 用 cal 命令显示下列年份的日历: 4、1752、2012 年 7 月。

- 1) 给出你显示以上年份年历的命令

答: ① cal 1752 ② cal 7 2012

- 2) 1752 年有几天, 为什么?

答: 从表中可以看出 1752 年 9 月只有 19 天, 故少了 11 天, 因而只有 355 天。原因是 1752 年 9 月大英帝国极其所属美洲殖民地的恺撒历法被格里高利教皇历法所取代。由于恺撒历法比格里高利历法迟 11 天, 因此 9 月 2 日当天改历法后, 次日须为 9 月 14 日。

```
ying@ying:~$ cal 1752

      1752
  一月  二月  三月
日 一 二 三 四 五 六 日 一 二 三 四 五 六 日 一 二 三 四 五 六
   1  2  3  4           1           1  2  3  4  5  6  7
  5  6  7  8  9 10 11  2  3  4  5  6  7  8  8  9 10 11 12 13 14
 12 13 14 15 16 17 18  9 10 11 12 13 14 15 15 16 17 18 19 20 21
 19 20 21 22 23 24 25 16 17 18 19 20 21 22 22 23 24 25 26 27 28
 26 27 28 29 30 31  23 24 25 26 27 28 29 29 30 31

  四月  五月  六月
日 一 二 三 四 五 六 日 一 二 三 四 五 六 日 一 二 三 四 五 六
   1  2  3  4           1  2           1  2  3  4  5  6
  5  6  7  8  9 10 11  3  4  5  6  7  8  9  7  8  9 10 11 12 13
 12 13 14 15 16 17 18 10 11 12 13 14 15 16 14 15 16 17 18 19 20
 19 20 21 22 23 24 25 17 18 19 20 21 22 23 21 22 23 24 25 26 27
 26 27 28 29 30      24 25 26 27 28 29 30 28 29 30

  七月  八月  九月
日 一 二 三 四 五 六 日 一 二 三 四 五 六 日 一 二 三 四 五 六
   1  2  3  4           1           1  2 14 15 16
  5  6  7  8  9 10 11  2  3  4  5  6  7  8 17 18 19 20 21 22 23
 12 13 14 15 16 17 18  9 10 11 12 13 14 15 24 25 26 27 28 29 30
 19 20 21 22 23 24 25 16 17 18 19 20 21 22 23 24 25 26 27 28 29
 26 27 28 29 30 31  23 24 25 26 27 28 29 30 31

  十月  十一月  十二月
日 一 二 三 四 五 六 日 一 二 三 四 五 六 日 一 二 三 四 五 六
   1  2  3  4  5  6  7           1  2  3  4           1  2
  8  9 10 11 12 13 14  5  6  7  8  9 10 11  3  4  5  6  7  8  9
 15 16 17 18 19 20 21 12 13 14 15 16 17 18 10 11 12 13 14 15 16
 22 23 24 25 26 27 28 19 20 21 22 23 24 25 17 18 19 20 21 22 23
 29 30 31  26 27 28 29 30  24 25 26 27 28 29 30 31
```

6. 使用 uptime 命令判断系统已启动运行的时间和当前系统中有多少登录用户, 给出显示的结果。

答: 系统已经启动了 39 分钟, 当前系统中共有 1 个登录用户


```

ying@ying: ~
ying@ying:~$ uptime
23:57:51 up 39 min, 1 user, load average: 0.00, 0.00, 0.00
ying@ying:~$

```

7. 通过 Linux 的 man、info 命令或互联网得到下面的 shell 命令、系统调用和库函数功能描述及每个命令使用例子：

命令	命令功能的简要描述	实例
touch	修改文件或目录的时间属性，若文件不存在则会新建一个文件	touch 123.txt
cp	复制文件或者目录	cp -r sdir/ targetdir
mv	为文件或目录改名、或将文件或目录移入其它位置	mv a.txt a.c mv -i cur/* ./
rm	删除文件或者目录	rm 123.txt rm -r abc
mkdir	创建一个文件夹	mkdir abc
who	显示系统中有哪些使用者正在上面	who who -count who -q
ls	显示指定工作目录下的内容	ls -alrt
cd	切换当前工作目录	cd ../ cd ~ cd -
pwd	显示工作目录	pwd
open	打开文件	
read	由已打开的文件读取数据	
write	将数据写入已打开的文件内	
close	关闭文件	
pipe	建立管道	
socket	建立一个 socket 通信	
mkfifo	建立具名管道	
system	执行 shell 命令	
printf	格式化输出数据	

8. vi 编辑器的使用：

- 1) 在 shell 提示符下，输入 vi firscrip 并按<Enter>键。vi 的界面将出现在显示屏上；
- 2) 按<A>键，输入 ls -la，并按<Enter>键；
- 3) 输入 who，并按<Enter>键；
- 4) 输入 pwd，再按<Enter>键。这时屏幕将如下图所示：

```
ying@ying: ~/桌面
ls -la
who
pwd
```

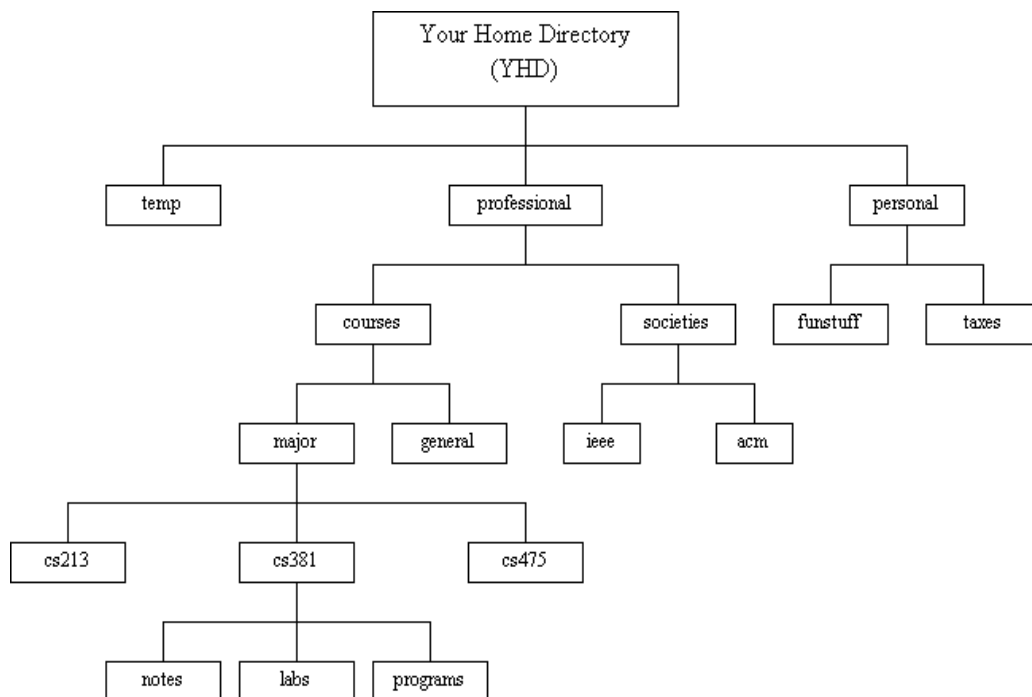
- 5) 在按<Esc>键;
- 6) 再输入: **wq**, 并按<Enter>键;
- 7) 在 shell 提示符下, 输入 **bash** **./firscrip** 并按<Enter>键;
- 8) 观察结果。当前的工作目录中有多少个文件? 他们的名称和大小? 还有谁在使用你的计算机系统? 当前的工作目录是什么?

```
ying@ying:~/桌面$ bash ./firscrip
总用量 16
drwxr-xr-x  2 ying ying 4096 4月 20 15:50 .
drwxr-xr-x 16 ying ying 4096 4月 20 15:48 ..
-rw-r--r--  1 ying ying   9 4月 17 20:32 1.txt
-rw-r--r--  1 ying ying  16 4月 20 15:50 firscrip
-rw-----  1 ying ying   0 4月 14 22:17 .firscrip.swp
ying      tty7          2019-04-20 15:45 (:0)
/home/ying/桌面
```

共有 5 个文件, 分别是:

名称	大小
.	4096 Byte
..	4096 Byte
1.txt	9 Byte
firscrip	16 Byte
.firscrip.swp	0 Byte

- 9. 在你的主目录下建立如下图所示的目录树。“Your Home Directory”表示你的主目录, 不需要再建立。给出完成这项工作的所有会话。(会话是指你命令的输入和结果的输出, 你提交的作业应包含这些内容)



输入命令如下:

```

ying@ying:~$ mkdir temp
ying@ying:~$ mkdir professional
ying@ying:~$ mkdir personal
ying@ying:~$ ls
examples.desktop  professional  公共的  视频  文档  音乐
personal         temp        模板  图片  下载  桌面
ying@ying:~$ cd personal
ying@ying:~/personal$ mkdir funstuff
ying@ying:~/personal$ mkdir taxes
ying@ying:~/personal$ cd ../
ying@ying:~$ cd professional
ying@ying:~/professional$ mkdir societies
ying@ying:~/professional$ mkdir courses
ying@ying:~/professional$ cd societies
ying@ying:~/professional/societies$ mkdir ieee
ying@ying:~/professional/societies$ mkdir acm
ying@ying:~/professional/societies$ cd ../
ying@ying:~/professional$ cd courses
ying@ying:~/professional/courses$ mkdir major
ying@ying:~/professional/courses$ mkdir general
ying@ying:~/professional/courses$ cd major
ying@ying:~/professional/courses/major$ mkdir cs213
ying@ying:~/professional/courses/major$ mkdir cs381
ying@ying:~/professional/courses/major$ mkdir cs475
ying@ying:~/professional/courses/major$ cd cs381
ying@ying:~/professional/courses/major/cs381$ mkdir notes
ying@ying:~/professional/courses/major/cs381$ mkdir labs
ying@ying:~/professional/courses/major/cs381$ mkdir programs
  
```

结果输出:

笔记: -a 输出全部 -d 仅输出目录 -f 显示完整路径 -L n 显示 n 层

```
ying@ying: ~  
ying@ying:~$ tree -d  
├── personal  
│   ├── funstuff  
│   └── taxes  
├── professional  
│   ├── courses  
│   │   ├── general  
│   │   └── major  
│   │       ├── cs213  
│   │       ├── cs381  
│   │       │   ├── labs  
│   │       │   ├── notes  
│   │       │   └── programs  
│   │       └── cs475  
│   └── societies  
│       ├── acm  
│       └── ieee  
├── temp  
├── 公共的  
├── 模板  
├── 视频  
├── 图片  
├── 文档  
├── 下载  
├── tree-1.8.0  
│   └── doc  
├── 音乐  
└── 桌面  
27 directories
```

10. 在系统中, 执行 `cd professional/courses` 命令, 回答下列问题:

1) 你的主目录的绝对路径是什么? 给出获得该绝对路径的命令及命令输出。

绝对路径: `/home/ying` 命令: `echo ~`

输出:

```
ying@ying:~/professional/courses/major/cs381/labs$ echo ~  
/home/ying
```

2) `acm` 目录的绝对路径是什么?

`/home/ying/professional/societies/acm`

3) 给出 `acm` 目录的两个相对路径。

①对于 `societies` 目录下的文件而言: `./acm`

②对于 `professional` 目录下的文件而言: `./societies/acm`

4) 执行 `cd major/cs381/labs` 命令。然后执行一个命令显示当前目录的绝对路径, 给出这个会话过程。

```
ying@ying:~/professional/courses$ cd major/cs381/labs  
ying@ying:~/professional/courses/major/cs381/labs$ pwd  
/home/ying/professional/courses/major/cs381/labs
```

5) 给出获得你的主目录三个不同的命令。

①`echo $HOME` ②`pwd` ③`echo ~`

11. Linux 系统规定, 隐含文件是首字符为“.”的文件, 如`.profile`。在你的系统中的主目录下查找隐含文件, 它们分别是哪些?


```
ying@ying:~$ ls -d .*
.
..
.bash_history
.bash_logout
.bashrc
.cache
.config
.dbus
.ICEauthority
.local
.mozilla
.presage
.profile
.sudo_as_admin_successful
.Xauthority
.xsession-errors
.xsession-errors.old
```

12. 下面这些目录的 inode 号是多少: /、你的主目录(home directory)、~/temp、~/professional、和~/personal? 写出会话过程。

```

ying@ying: ~
ying@ying:~$ ls -ali
总用量 128
1467650 drwxr-xr-x 19 ying ying 4096 4月 20 16:08 .
1441793 drwxr-xr-x  3 root root 4096 3月 28 16:05 ..
1467683 -rw-----  1 ying ying 1311 4月 20 16:15 .bash_history
1467653 -rw-r--r--  1 ying ying 220 3月 28 16:05 .bash_logout
1467651 -rw-r--r--  1 ying ying 3771 3月 28 16:05 .bashrc
1441807 drwx----- 13 ying ying 4096 4月 19 20:47 .cache
1442807 drwx----- 17 ying ying 4096 3月 28 15:23 .config
1444947 drwx-----  3 ying ying 4096 3月 28 17:13 .dbus
1467654 -rw-r--r--  1 ying ying 8980 3月 28 16:05 examples.desktop
1445274 -rw-----  1 ying ying 2480 4月 20 15:45 .ICEauthority
1444973 drwx-----  3 ying ying 4096 3月 28 17:13 .local
1467579 drwx-----  4 ying ying 4096 4月 10 20:47 .mozilla
1467700 drwxr-xr-x  4 ying ying 4096 4月 20 15:59 personal
1444957 drwx-----  2 ying ying 4096 3月 28 17:13 .presage
1467692 drwxr-xr-x  4 ying ying 4096 4月 20 16:00 professional
1467652 -rw-r--r--  1 ying ying  675 3月 28 16:05 .profile
1467546 -rw-r--r--  1 ying ying  0 4月 20 16:08 .sudo_as_admin_successful
1467699 drwxr-xr-x  2 ying ying 4096 4月 20 15:58 tempo

```

主目录: 1467650 temp: 1467699 professional: 1467692 personal: 1467700

【注】 也可以通过 `ls -di professional` 来查看 inode

13. 在 labs 目录下，用文本编辑器创建一个名字为 lab1 的文件，文件的内容为：“Use a text editor to create a file called lab1 under the labs directory in your directory hierarchy. The file should contain the text of this problem.”。回答下列问题：

1) lab1 文件的类型，用 Linux 命令回答这个问题，给出会话过程。

普通文件

```
ying@ying:~$ cd professional/courses/major/cs381/labs
ying@ying:~/professional/courses/major/cs381/labs$ touch lab1
ying@ying:~/professional/courses/major/cs381/labs$ vi lab1
ying@ying:~/professional/courses/major/cs381/labs$ ls -dl lab1
-rw-r--r-- 1 ying ying 154 4月 20 16:57 lab1
```

2) lab1 文件内容的类型，用 Linux 命令回答这个问题，给出会话过程。

ASCII text

```

y1ng@y1ng:~/professional/courses/major/cs381/labs$ file lab1
lab1: ASCII text

```

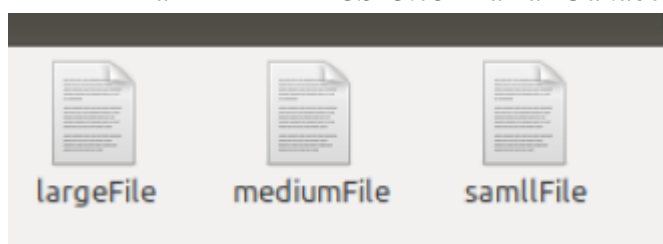
14. 在 linux 系统中，头文件以.h 为扩展名。在/usr/include 目录中，显示所有以 t 字母开头的头文件的名字。给出会话过程。

ls t*.h 或 ls t*

```
ying@ying: /usr/include
ying@ying: /usr/include$ ls t*
tar.h  termio.h  termios.h  tgmth.h  thread_db.h  time.h  ttyent.h
```

15. 创建几个大小不等的文本文件，供本实验和后面几个实验用：用 `man cat > mediumFile` 命令创建中等大小的文件；用 `man bash > largeFile` 命令创建一个文件；再创建一个名字为 `smallFile` 关于学生数据的小文件，文件每行内容如下，第一行为各自段的含义，注意字段之间用 `tab` 符隔开：

FirstName	LastName	Major	GPA	Email	Phone
John	Doe	ECE	3.54	doe@jd.home.org	111.222.3333
James	Davis	ECE	3.71	davis@jd.work.org	111.222.1111
Al	Davis	CS	2.63	davis@a.lakers.org	111.222.2222
Ahmad	Rashid	MBA	3.04	ahmad@mba.org	111.222.4444
Sam	Chu	ECE	3.68	chu@sam.ab.com	111.222.5555
Arun	Roy	SS	3.86	roy@ss.arts.edu	111.222.8888
Rick	Marsh	CS	2.34	marsh@a.b.org	111.222.6666
James	Adam	CS	2.77	jadam@a.b.org	111.222.7777
Art	Pohm	ECE	4.00	pohm@ap.a.org	111.222.9999
John	Clark	ECE	2.68	clark@xyz.ab.com	111.111.5555
Nabeel	Ali	EE	3.56	ali@ee.eng.edu	111.111.8888
Tom	Nelson	ECE	3.81	nelson@tn.abc.org	111.111.6666
Pat	King	SS	3.77	king@pk.xyz.org	111.111.7777
Jake	Zulu	CS	3.00	zulu@jz.sa.org	111.111.9999
John	Lee	EE	3.64	jlee@j.lee.com	111.111.2222
Sunil	Raj	ECE	3.86	raj@sr.cs.edu	111.111.3333
Charles	Right	EECS	3.31	right@cr.abc.edu	111.111.4444
Diane	Rover	ECE	3.87	rover@dr.xyz.edu	111.111.5555
Aziz	Inan	EECS	3.75	ainan@ai.abc.edu	111.111.1111



16. 显示 `largeFile` 文件的开始 12 行内容，显示 `smallFile` 文件的最后 5 行内容，要使用什么命令？使用命令 `tail -n +6 smallFile` 显示 `smallFile` 文件从第 6 行开始到结束全部行？给出你的会话。

`head -12 largeFile`

```
ying@ying: ~/桌面
ying@ying:~/桌面$ head -12 largeFile
BASH(1)                                General Commands Manual                                BASH(1)

NAME
    bash - GNU Bourne-Again SHell

SYNOPSIS
    bash [options] [command_string | file]

COPYRIGHT
    Bash is Copyright (C) 1989-2016 by the Free Software Foundation, Inc.

DESCRIPTION
```

tail -5 smallFile

```
ying@ying:~/桌面$ tail -5 smallFile
John Lee EE 3.64 jlee@j.lee.com 111.111.2222
Sunil Raj ECE 3.86 raj@sr.cs.edu 111.111.3333
Charles Right EECS 3.31 right@cr.abc.edu 111.111.4444
Diane Rover ECE 3.87 rover@dr.xyz.edu 111.111.5555
Aziz Inan EECS 3.75 ainan@ai.abc.edu 111.111.1111
```

tail -n +6 smallFile

```
ying@ying:~/桌面$ tail -n +6 smallFile
Sam Chu ECE 3.68 chu@sam.ab.com 111.222.5555
Arun Roy SS 3.86 roy@ss.arts.edu 111.222.8888
Rick Marsh CS 2.34 marsh@a.b.org 111.222.6666
James Adam CS 2.77 jadam@a.b.org 111.222.7777
Art Pohm ECE 4.00 pohm@ap.a.org 111.222.9999
John Clark ECE 2.68 clark@xyz.ab.com 111.111.5555
Nabeel Ali EE 3.56 ali@ee.eng.edu 111.111.8888
Tom Nelson ECE 3.81 nelson@tn.abc.org 111.111.6666
Pat King SS 3.77 king@pk.xyz.org 111.111.7777
Jake Zulu CS 3.00 zulu@jz.sa.org 111.111.9999
John Lee EE 3.64 jlee@j.lee.com 111.111.2222
Sunil Raj ECE 3.86 raj@sr.cs.edu 111.111.3333
Charles Right EECS 3.31 right@cr.abc.edu 111.111.4444
Diane Rover ECE 3.87 rover@dr.xyz.edu 111.111.5555
Aziz Inan EECS 3.75 ainan@ai.abc.edu 111.111.1111
```

17. 复制 smallFile 文件的拷贝，文件名为 dataFile。用 ls -l 命令观察这两个文件的修改时间是否一样。它们是不同的，dataFile 文件的修改时间应该是这个文件的创建时间。什么命令能够保留这个修改时间不变呢？这两个文件的 inode 号是多少？

再把文件名 dataFile 改成（移动）newDataFile，文件 newDataFile 的 inode 多少？与 dataFile 文件的 inode 号是否相同，若相同，为什么？

然后再把文件 newDataFile 移动到/tmp 目录下，文件/tmp/newDataFile 的 inode 号是多少？比较结果如何，为什么？给出完成上述工作的会话过程。

cp ./smallFile ./dataFile 两个文件的修改时间不一样

```
ying@ying:~/桌面$ cp ./smallFile ./dataFile
ying@ying:~/桌面$ ls -l
总用量 372
-rw-r--r-- 1 ying ying 9 4月 17 20:32 1.txt
-rw-r--r-- 1 ying ying 963 4月 20 17:32 dataFile
-rw-r--r-- 1 ying ying 981 4月 21 2019 data.txt
-rw-r--r-- 1 ying ying 16 4月 20 15:50 firscrip
-rw-r--r-- 1 ying ying 355236 4月 20 17:17 largeFile
-rw-r--r-- 1 ying ying 1958 4月 20 17:16 mediumFile
-rw-r--r-- 1 ying ying 963 4月 20 17:19 smallFile
```

cp -p smallFile dataFile_new

```
ying@ying:~/桌面$ cp -p smallFile dataFile_new
ying@ying:~/桌面$ ls -l
总用量 376
-rw-r--r-- 1 ying ying 9 4月 17 20:32 1.txt
-rw-r--r-- 1 ying ying 963 4月 20 17:32 dataFile
-rw-r--r-- 1 ying ying 963 4月 20 17:19 dataFile_new
-rw-r--r-- 1 ying ying 981 4月 21 20:19 data.txt
-rw-r--r-- 1 ying ying 16 4月 20 15:50 firscrip
-rw-r--r-- 1 ying ying 355236 4月 20 17:17 largeFile
-rw-r--r-- 1 ying ying 1958 4月 20 17:16 mediumFile
-rw-r--r-- 1 ying ying 963 4月 20 17:19 smallFile
```

ls -li smallFile dataFile dataFile 的 inode 号为 1467695 smallFile 为 1468160

```
ying@ying:~/桌面$ ls -li smallFile dataFile
1467695 dataFile 1468160 smallFile
```

mv dataFile newDataFile 新文件的 inode 号仍然为 1467695。因为重命名文件不影响 inode 指向的文件地址，只是改变了对应文件的名字。

```
ying@ying:~/桌面$ mv dataFile newDataFile
ying@ying:~/桌面$ ls -li newDataFile
1467695 newDataFile
```

mv newDataFile /tmp 新文件的 inode 号仍然为 1467695。因为移动文件不影响 inode 指向的存放文件地址。

```
ying@ying:~/桌面$ cd /tmp
ying@ying:/tmp$ ls -li newDataFile
1467695 newDataFile
```

- 在屏幕上显示文件 smallFile、mediumFile、largeFile 和/tmp/newDataFile 的字节数、字数和行数。smallFile 和/tmp/newDataFile 文件应该是相同的。你能用其它命令给出这些文件的字节数的大小吗？什么命令。给出会话过程。

```
ying@ying:~/桌面$ ls -l smallFile mediumFile largeFile /tmp/newDataFile
-rw-r--r-- 1 ying ying 355236 4月 20 17:17 largeFile
-rw-r--r-- 1 ying ying 1958 4月 20 17:16 mediumFile
-rw-r--r-- 1 ying ying 963 4月 20 17:19 smallFile
-rw-r--r-- 1 ying ying 963 4月 20 17:32 /tmp/newDataFile
ying@ying:~/桌面$ wc -ml smallFile mediumFile largeFile /tmp/newDataFile
 20   963 smallFile
 71  1957 mediumFile
5951 354060 largeFile
 20   963 /tmp/newDataFile
6062 357943 总用量
```

文件	字数	行数	字节数
smallFile	963	20	963
mediumFile	1957	71	1958
largeFile	354060	5951	355236
newDataFile	963	20	963

显示字节数: wc -c smallFile mediumFile largeFile /tmp/newDataFile


```
ying@ying:~/桌面$ wc -c smallFile mediumFile largeFile /tmp/newDataFile
 963 smallFile
1958 mediumFile
355236 largeFile
 963 /tmp/newDataFile
359120 总用量
```

19. 搜索你的主目录，找到所有的 HTML 和 C 程序文件（文件有.html、.htm 或.c 扩展名。如果没有这类文件，可以用 touch 等命令生成一些这样的文件），显示符合要求的文件路径和文件名。给出你的会话。

```
find ~ \( -name '*.c' -o -name '*.html' -o -name '*.htm' \) -print
```

```
ying@ying:~$ find ~ \( -name '*.c' -o -name '*.html' -o -name '*.htm' \) -print
/home/ying/web3.html
/home/ying/桌面/web4.htm
/home/ying/cpro1.c
/home/ying/web2.html
/home/ying/cpro2.c
/home/ying/下载/tree-1.8.0/xml.c
/home/ying/下载/tree-1.8.0/color.c
/home/ying/下载/tree-1.8.0/file.c
/home/ying/下载/tree-1.8.0/unix.c
/home/ying/下载/tree-1.8.0/strverscmp.c
/home/ying/下载/tree-1.8.0/hash.c
/home/ying/下载/tree-1.8.0/json.c
/home/ying/下载/tree-1.8.0/html.c
/home/ying/下载/tree-1.8.0/tree.c
/home/ying/web1.html
```

20. 给出命令，搜索主目录，显示创建时间在~/smallFile 之后的文件及其路径。

```
find ~ \( -cnewer smallFile \) -print
```

```
ying@ying:~$ find ~ \( -cnewer smallFile \) -print
/home/ying
/home/ying/mediumFile
/home/ying/.config/dconf
/home/ying/.config/dconf/user
/home/ying/.config/nautilus
/home/ying/.config/nautilus/desktop-metadata
/home/ying/.config/flatpak/log/crash.log
/home/ying/.config/gedit/accels
/home/ying/largeFile
/home/ying/web3.html
/home/ying/桌面
/home/ying/桌面/web4.htm
/home/ying/桌面/LsCommand
/home/ying/cpro1.c
/home/ying/.cache/unity
/home/ying/.bash_history
/home/ying/web2.html
/home/ying/cpro2.c
/home/ying/smallFile
/home/ying/.local/share
/home/ying/.local/share/gvfs-metadata
/home/ying/.local/share/gvfs-metadata/home
/home/ying/.local/share/gvfs-metadata/home-bd95484d.log
/home/ying/.local/share/zeitgeist/activity.sqlite-wal
/home/ying/.local/share/zeitgeist/activity.sqlite-shm
/home/ying/.local/share/zeitgeist/fts.index
/home/ying/.local/share/zeitgeist/fts.index/ianglass
/home/ying/.local/share/zeitgeist/fts.index/termist.glass
/home/ying/.local/share/zeitgeist/fts.index/postlist.glass
/home/ying/.local/share/zeitgeist/fts.index/position.glass
/home/ying/.local/share/recently-used.xbel
/home/ying/web1.html
```

21. 给出一条命令，在主目录下显示所有文件中包含字符串“Linux”的文件名。

```
find ~ -name *Linux*
```

```
ying@ying:~$ find ~ -name *Linux*
/home/ying/.mozilla/firefox/yx4ds2ra.default/gmp/Linux_x86_64-gcc3
```

22. 在你的系统中有文件或目录分别是：/、/etc/passwd、/bin/df、~。用长列表格式显示这些文件或目录，并填写下列表格。

```
ying@ying:~$ ls -ld ~
drwxr-xr-x 19 ying ying 4096 4月 20 18:08 /home/ying
ying@ying:~$ ls -ld /
drwxr-xr-x 24 root root 4096 3月 28 16:05 /
ying@ying:~$ ls -ld /etc/passwd
-rw-r--r-- 1 root root 2385 3月 28 16:05 /etc/passwd
ying@ying:~$ ls -ld /bin/df
-rwxr-xr-x 1 root root 84776 3月 1 2017 /bin/df
```

文件	文件类型	存取权限	链接数	所有者	组	文件大小
/	目录	rwX r-X r-X	24	root	root	4096B
/etc/passwd	普通	rw- r-- r--	1	ying	ying	2385B
/bin/df	普通	rwX r-X r-X	1	root	root	84776B
~	目录	rwX r-X r-X	19	root	root	4096B

23. 在你的主目录中的 temp、professional 和 personal 三个子目录，设置使自己（owner）拥有读、写、执行 3 种访问权限，设置其它用户只有读和执行权限。在~/temp 目录下创建名为 d1、d2 和 d3 的目录。在 d1 目录下，用 touch 命令创建一个名为 f1 的空文件。给出 d1、d2、d3 和 f1 的访问权限。给出完成这些工作的会话。

```
ying@ying:~$ chmod u=rwx,o=rx personal professional temp
ying@ying:~$ ls -l
总用量 412
-rw-r--r-- 1 ying ying 0 4月 20 18:08 cpro1.c
-rw-r--r-- 1 ying ying 0 4月 20 18:08 cpro2.c
-rw-r--r-- 1 ying ying 8980 3月 28 16:05 examples.desktop
-rw-r--r-- 1 ying ying 355236 4月 20 17:17 largeFile
-rw-r--r-- 1 ying ying 1958 4月 20 17:16 mediumFile
drwxr-xr-x 4 ying ying 4096 4月 20 15:59 personal
drwxr-xr-x 4 ying ying 4096 4月 20 16:00 professional
-rw-r--r-- 1 ying ying 963 4月 20 17:19 smallFile
drwxr-xr-x 2 ying ying 4096 4月 20 15:58 temp
```

d1,d2,d3 所有者：读|写|执行，组和其他：读|执行

f1 所有者：读|写，组合其他：读

```
ying@ying:~/temp$ mkdir d1 d2 d3
ying@ying:~/temp$ cd d1
ying@ying:~/temp/d1$ touch f1
ying@ying:~/temp/d1$ ls -l
总用量 0
-rw-r--r-- 1 ying ying 0 4月 20 21:05 f1
ying@ying:~/temp/d1$ cd ../
ying@ying:~/temp$ ls -l
总用量 12
drwxr-xr-x 2 ying ying 4096 4月 20 21:05 d1
drwxr-xr-x 2 ying ying 4096 4月 20 21:05 d2
drwxr-xr-x 2 ying ying 4096 4月 20 21:05 d3
```

24. 设置当前目录为你的主目录，设置文件~/temp 仅为执行权限，然后执行 `ls -ld temp`，再执行 `ls -l temp` 命令。结果如何？成功执行 `ls -l temp` 命令需要的最小权限是什么？请设置 temp 目录的最小权限，然后再一次执行 `ls -l temp` 命令。给出这个过程的会话。注意：做这个实验不能使用 root 用户登录系统。

```
ying@ying:~$ chmod a=x temp
ying@ying:~$ ls -ld temp
ls: 无法打开目录'temp': 权限不够
ying@ying:~$ chmod a=r temp
ying@ying:~$ ls -ld temp
ls: 无法访问'temp/d3': 权限不够
ls: 无法访问'temp/d2': 权限不够
ls: 无法访问'temp/d1': 权限不够
总用量 0
d???????? ? ? ? ? ? ? d1
d???????? ? ? ? ? ? ? d2
d???????? ? ? ? ? ? ? d3
ying@ying:~$ chmod a=rx temp
ying@ying:~$ ls -ld temp
总用量 12
drwxr-xr-x 2 ying ying 4096 4月 20 21:05 d1
drwxr-xr-x 2 ying ying 4096 4月 20 21:05 d2
drwxr-xr-x 2 ying ying 4096 4月 20 21:05 d3
```

①结果是能够打开目录

②最小权限为 rx，因为 r 特权代表可以读出目录的内容，x 特权意味着可以检索这个目录，`ls -l temp` 需要检索这个目录并读出目录下的内容。

25. 用 `umask` 命令显示当前的掩码。把你的主目录设置为当前目录，然后在 ~/temp/d1 目录下，创建 d11 目录，用 `touch` 命令创建 f2 空文件。在 temp 目录下用编辑器创建 hello.c 文件，该文件的内容如下：

```
#include <stdio.h>

main(void)
{
    printf("Hello, world!\n");
}
```

再运行命令 `gcc -o greeting hello.c`，生成了可执行文件 `greeting`。长列表显示 f2、hello.c、greeting 和 d1 文件访问权限。

把掩码（mask）设置为 077，在目录~/temp/d2 下，创建 d21 目录，用 `touch` 命令创建 f2 空文件。长列表显示 f2、hello.c、greeting 和 d21 文件访问权限。最后根据掩码的不同填写下列表格。

umask 值	文件权限			
	f2	hello.c	greeting	d11 / d21
022	rw-r-r	rw-r-r	rwX-rX-rX	rwX-rX-rX
077	rw-/-/	rw-/-/	rwX-/-/	rwX-/-/

提示：gcc 是 c 语言的编译器。在 Linux 系统中执行文件和目录的缺省权限是 777，文本文件的缺省权限为 666。

```
ying@ying:~$ umask
0022
ying@ying:~$ mkdir temp
umask: 022
```

26. 删除~/temp 目录下的所有文件和目录。给出会话过程。

```
ying@ying: ~/temp
ying@ying:~/temp$ rm -r ./ *
ying@ying:~/temp$ ls
```

27. 在~/temp 目录下创建名为 d1、d2 和 d3 的目录。把文件 smallFile 拷贝到 d1 目录下，长列表格式显示文件 smallFile，显示的内容包括 inode 号、访问权限、硬链接数、文件大小。给出完成这些工作的会话。

```
ying@ying:~/temp$ mkdir d1 d2 d3
ying@ying:~/temp$ cp ~/smallFile ./d1
```

```
ying@ying:~/temp$ ls -il d1/smallFile
1595571 -rw-r--r-- 1 ying ying 963 4月 21 00:00 d1/smallFile
```

28. 在~/temp 目录下，把当前目录改变成 d2。创建一个名字为 newFile.hard 硬链接到 d1 目录下的 smallFile 文件。长列表格式显示 newFile.hard 文件，与 smallFile 文件的属性进行比较。你如何确定 smallFile 和 smallFile.hard 是同一文件的两个名字，是链接数吗？给出你的会话过程。

```
ying@ying:~/temp/d2$ ln ../d1/smallFile smallFile.hard
ying@ying:~/temp/d2$ ls -l
总用量 4
-rw-r--r-- 2 ying ying 963 4月 21 00:00 smallFile.hard
```

```
ying@ying:~/temp/d2$ ls -il ../d1 ./
./:
总用量 4
1595571 -rw-r--r-- 2 ying ying 963 4月 21 00:00 smallFile.hard
../d1:
总用量 4
1595571 -rw-r--r-- 2 ying ying 963 4月 21 00:00 smallFile
```

文件属性除了名字外一样。是同一个文件因为 inode 号相同而不是链接数。

29. 使用硬链接文件 smallFile.hard 显示 smallFile 文件的内容。然后取消你本人对 smallFile 文件读 (r) 权限，再显示文件的内容，发生了什么？根据以上练习，你能推断出什么？对 smallFile 文件增加读权限，再一次显示文件内容，发生了什么？最后作一个 smallFile 文件的备份，并删除 smallFile 文件，用 smallFile.hard 显示 smallFile 文件内容，又发生了什么？请你解释一下练习过程中的现象。


```
ying@ying:~/temp/d2$ vi smallFile.hard
ying@ying:~/temp/d2$ cat smallFile.hard
```

FirstName	LastName	Major	GPA	Email	Phone
John	Doe	ECE	3.54	doe@jd.home.org	111.222.3333
James	Davis	ECE	3.71	davis@jd.work.org	111.222.1111
Al	Davis	CS	2.63	davis@a.lakers.org	111.222.2222
Ahmad	Rashid	MBA	3.04	ahmad@mba.org	111.222.4444
Sam	Chu	ECE	3.68	chu@sam.ab.com	111.222.5555
Arun	Roy	SS	3.86	roy@ss.arts.edu	111.222.8888
Rick	Marsh	CS	2.34	marsh@a.b.org	111.222.6666
James	Adam	CS	2.77	jadam@a.b.org	111.222.7777
Art	Pohm	ECE	4.00	pohm@ap.a.org	111.222.9999
John	Clark	ECE	2.68	clark@xyz.ab.com	111.111.5555
Nabeel	Ali	EE	3.56	ali@ee.eng.edu	111.111.8888
Tom	Nelson	ECE	3.81	nelson@tn.abc.org	111.111.6666
Pat	King	SS	3.77	king@pk.xyz.org	111.111.7777
Jake	Zulu	CS	3.00	zulu@jz.sa.org	111.111.9999
John	Lee	EE	3.64	jlee@j.lee.com	111.111.2222
Sunil	Raj	ECE	3.86	raj@sr.cs.edu	111.111.3333
Charles	Right	ECS	3.31	right@cr.abc.edu	111.111.4444
Diane	Rover	ECE	3.87	rover@dr.xyz.edu	111.111.5555
Aziz	Inan	ECS	3.75	ainan@ai.abc.edu	111.111.1111

取消读权限后：系统提示权限不够

```
ying@ying: ~/temp/d2
ying@ying:~/temp/d2$ chmod u-r ../d1/smallFile
ying@ying:~/temp/d2$ cat smallFile.hard
cat: smallFile.hard: 权限不够
```

根据以上能够推断出硬链接是一个指向文件索引节点的指针，它的改变能够引起指向文件的同步改变。

增加读权限后，再一次显示内容能够完全显示，没有提示错误。

删除后发现还是能够打开。说明硬链接本质是对文件的一份副本加上能够同步修改的功能。

- 恢复/temp/d1/smallFile 文件。创建一个名字为~/temp/d2/smallFile.soft 软链接到~/temp/d1/smallFile 文件。长列表格式显示 smallFile.soft 文件，比较这两个文件的属性。你如何确定 smallFile 和 smallFile.soft 是两个不同的文件？是这两个文件的大小吗？给出你的会话过程。

```
ying@ying:~/temp$ ln -s d1/smallFile d2/smallFile.soft
ying@ying:~/temp$ ls -il d2 d1
```

d1:					
总用量 4					
1595572	-rw-r--r--	1	ying	ying	963 4月 20 17:19 smallFile

d2:					
总用量 4					
1595571	-rw-r--r--	1	ying	ying	963 4月 21 00:00 smallFile.hard
1595573	lrwxrwxrwx	1	ying	ying	12 4月 21 12:31 smallFile.soft -> d1/smallFile

两个文件是不同的文件 因为它们的 inode 号不相同，文件的大小在本例子中也可以作为是不同文件的依据（但是是必要不充分条件）。

- 使用软链接文件 smallFile.soft 显示 smallFile 文件的内容。然后取消你本人对 smallFile 文件读 (r) 权限，再显示文件的内容，发生了什么？根据以上练习，你能推断出什么？对 smallFile 文件增加读权限，再一次显示文件内容，发生了什么？最后作一个 smallFile 文件的备份，并删除 smallFile 文件，用 smallFile.soft 显示 smallFile 文件内容，又发生了什么？请你解释一下练习过程中的现象。

能够正常显示文件。

取消读权限后：系统提示权限不够

根据以上能够推断出软链接是一个快捷方式，它指向了源文件，因为源文件的权限被修改，所以不能够使用快捷方式访问它。

增加读权限后又能够显示。

删除源文件后，不能够通过软链接来访问。说明了软链接其本质是快捷方式，访问软链接相当于访问其指向的文件。而文件被删除后则找不到被访问的文件。

```
ying@ying:~/temp/d2$ rm ../d1/smallFile
ying@ying:~/temp/d2$ cat smallFile.soft
cat: smallFile.soft: 没有那个文件或目录
```

32. 在你使用的 Linux 系统中，有多少进程在运行？进程 init、bash、ps 的 PID 是多少？init、bash 和 ps 进程的父进程是哪一个？这些父进程的 ID 是什么？给出你得到这些信息的会话过程。

共有 2 个进程在运行，ps 的 PID 为 2119，bash 的 PID 是 2889

ps 的父进程的 ID 是 2089 (bash)，bash 的父进程 ID 是 2083 (gnome-terminal)

```
ying@ying: /
ying@ying:/$ ps -l
F S  UID  PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
0 S  1000  2089  2083  0  80   0 -  7425 wait  pts/0        00:00:00 bash
0 R  1000  2119  2089  0  80   0 -  9607 -    pts/0        00:00:00 ps
```

33. linux 系统中，进程可以在前台或后台运行。前台进程在运行结束前一直控制着终端。若干个命令用分号（；）分隔形成一个命令行，用圆括号把多个命令挂起来，他们就在一个进程里执行。使用“&”符作为命令分隔符，命令将并发执行。可以在命令行末尾加“&”使之成为后台命令。

请用一行命令实现以下功能：它 1 小时（实验中可以用 1 分钟代替）分钟后在屏幕上显示文字“Time for Lunch!”来提醒你去吃午餐。给出会话过程。

(sleep 60; echo “Time for Lunch!”;) &

```
ying@ying:~$ (sleep 60; echo "Time for Lunch!";) &
[1] 2090
ying@ying:~$ ps
  PID TTY          TIME CMD
 2025 pts/0        00:00:00 bash
 2090 pts/0        00:00:00 bash
 2091 pts/0        00:00:00 sleep
 2094 pts/0        00:00:00 ps
ying@ying:~$ Time for Lunch!
```

34. 写一命令行，使得 date 、uname -a 、who 和 ps 并发执行。给出会话过程。

```
ying@ying:~$ date & uname -a & who & ps
[1] 4382
[2] 4383
[3] 4384
2019年 04月 21日 星期日 20:56:44 CST
Linux ying 4.10.0-19-generic #21-Ubuntu SMP Thu Apr 6 17:04:57 UTC 2017 x86_64 x
86_64 x86_64 GNU/Linux
  PID TTY          TIME CMD
  4283 pts/0        00:00:00 bash
  4384 pts/0        00:00:00 bash
  4385 pts/0        00:00:00 ps
[1] 已完成          date
[2]- 已完成         uname -a
ying@ying:~$ ying      tty7      2019-04-21 13:45 (:0)
```

35. 写一命令行，先后执行 `date` 、 `uname -a` 、 `who` 和 `ps` 命令，后面 3 个命令的执行条件是：当只有前面一个命令执行成功后，才能执行后面一个命令。给出会话过程。

```
ying@ying:~$ date && uname -a && who && ps
2019年 04月 21日 星期日 20:58:30 CST
Linux ying 4.10.0-19-generic #21-Ubuntu SMP Thu Apr 6 17:04:57 UTC 2017 x86_64 x
86_64 x86_64 GNU/Linux
ying      tty7          2019-04-21 13:45 (:0)
  PID TTY          TIME CMD
  4283 pts/0        00:00:00 bash
  4394 pts/0        00:00:00 ps
ying@ying:~$
```

36. 在 shell 下执行下面的命令。3 个 `pwd` 命令的运行结果是什么？

`$ pwd`

`$ bash`

`$ cd /usr`

`$ pwd`

`$<Ctrl-D>` #终止 shell

`$ pwd`

“\$”为系统提示符

```
ying@ying:~$ pwd
/home/ying
ying@ying:~$ bash
ying@ying:~$ cd /usr
ying@ying:/usr$ pwd
/usr
ying@ying:/usr$ exit
ying@ying:~$ pwd
/home/ying
```

①/home/ying②/usr③/home/ying

37. 搜索并显示你主目录下 `foobar` 文件的绝对路径，错误信息重定向到 `/dev/null` 中。给出你的会话。（提示：若没有 `foobar` 文件，可以创建一个）

```
ying@ying:~$ (find ~ -name foobar) 2> /dev/null
/home/ying/foobar
```

38. 搜索你主目录下 `foobar` 文件，保存它的绝对路径到 `foobar.path` 文件中，错误信息写到 `/dev/null` 中，再显示 `foobar.path` 文件的内容。给出会话过程。

```
/home/ying/foobar
ying@ying:~$ (find ~ -name foobar) 1>foobar.path 2>/dev/null
ying@ying:~$ cat foobar.path
/home/ying/foobar
ying@ying:~$
```

39. `cat` 程序（命令），输入从标准输入设备中读入，输出送到标准输出设备中。现在运行这个程序，要求输入从 `student.records` 中读入，输出结果重定向到 `output.data` 文件中，错误重定向到 `error.log` 文件中。用一条命令来实现上述过程。

```
cat < student.records > output.data 2>error.log
```

40. 写出一个命令将 `shell` 的标准输入（`stdin`）更改到当前目录下名为 `fdata` 文件，标准输出（`stdout`）更改到当前目录下名为 `fout` 的文件。如果 `fdata` 文件包含下面的内容，那么在命令执行后会发生什么？

```
echo -n "The time now is:"
date
echo -n "The users presently logged on are:"
who
```

提示：使用 `exec` 命令

```
exec < fdata > fout
```

输出如下：输出当前时间和使用者

```
打开(O)  [icon]
The time now is:2019年 04月 21日 星期日 21:49:14 CST
The users presently logged on are:ying      tty7      2019-04-21 13:45 (:0)
```

41. 计算命令 `ls -l` 的输出中的字符数、单词数和行数，并把它显示在显示器上。给出命令和输出结果。

```
ying@ying:~$ ls -l | wc -clw
20      173     1028
```

行数：20 字数：173 字符数：128

42. 在 `/bin` 目录下有多少个普通文件、目录文件和符号链接文件？如何得到这个答案？

```
ying@ying:/$ find ./bin -type f | wc -l
139
ying@ying:/$ find ./bin -type d | wc -l
1
ying@ying:/$ find ./bin -type l | wc -l
23
```


普通文件：139 目录文件：1 符号链接文件：23

三、讨论、心得（必填）（10 分）

本次的实验主要针对于 linux 的基础命令、文件管理操作、文件安全与共享、进程、重定向和管道几方面知识做了一个复习回顾，在实现的过程中主要遇到如下几个问题：

①tree 命令和 at 命令在 ubuntu-17.04 中不能够马上使用，系统提示需要进行下载安装命令，下载安装命令的方式为 `sudo apt-get install at`，但是一开始会提示无法定位软件包，解决方案为更新软件源，具体教程如下：

https://blog.csdn.net/qq_42168575/article/details/81147944

②linux 中 find 和 ls 命令的作用又相似又不同，个人认为 find 命令功能更加强大。如果需要输出绝对路径，需要 find 命令来实现。

③可以通过 `ctrl+d` 结束输入，`ctrl+z` 挂起一个进程。

④ls 命令如果需要显示目录的信息，需要带上 `-ld`。如 `ls -ld /bin`，否则如果命令是 `ls -l /bin`，则显示的是 /bin 目录下的信息而不是 /bin 目录本身的信息。

⑤重定向时，0 为 stdin，1 为 stdout，2 为错误信息，写重定向的命令时，一定要注意要写成 `1> filename` 而不是 `1 > filename`，也就是 1 与 > 之间不能够有空格否则会出错。同时 & 的用法需要注意，如 `1>filename 2>&1` 代表标准输出和错误信息都重定向到 filename 中