

# 浙江大学实验报告

课程名称: Linux 程序设计 (课程综合实践 II) 实验类型: 验证型

实验项目名称: Linux shell 基本命令

学生姓名: 王子腾 专业: 软件工程 学号: 3180102173

电子邮件地址: 3180102173@zju.edu.cn

实验日期: 2020 年 7 月 21 日

## 一、实验环境

计算机配置:

处理器	Intel Core i7-8550 CPU @ 1.80GHz 1.99Ghz
内存	8GB DDR4 2400MHz
硬盘	128GB PCIe 固态硬盘+1TB 硬盘
显卡	NVIDIA GeForce 150MX

操作系统环境:

Windows 10 家庭中文版

Linux 版本:

Ubuntu 18.04

## 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。

目前国际上常用的软件许可证有: **GPL**、**BSD** 许可证、**MPL** 许可证等。作为一款软件许可证, 其主要目的便是维护和平衡开发者的利益, 如 **GPL** 的开创目的便是保障用户对软件的自由权, 同时给予开发者以版权收益。

由于 **GPL** 未很好地平衡开发者对源代码的需求和他们利用源代码获得

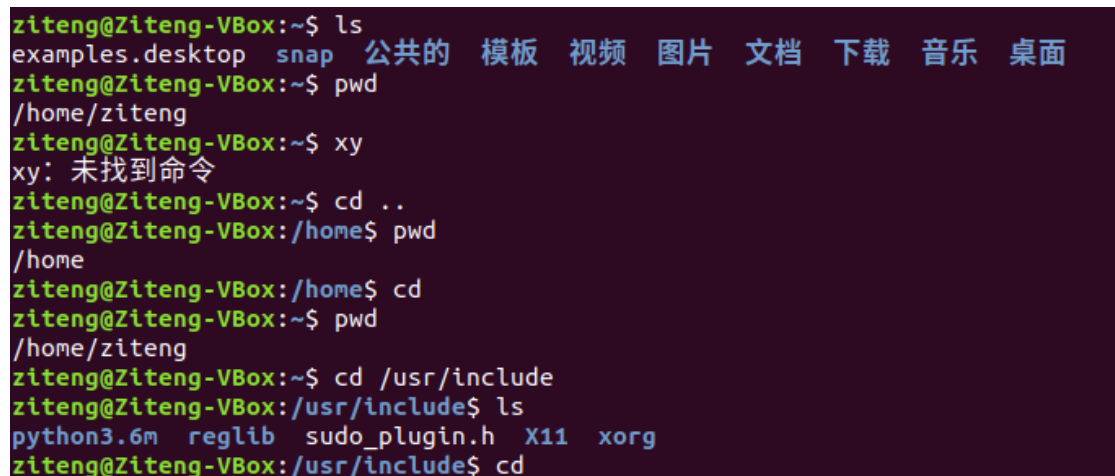
的利益，之后出现的 **MPL** 许可证在其基础上补充了对软件专利和源代码的定义，为后人借鉴源代码用于商业开发保留了可能。

**BSD** 许可证比前两者要更加宽松，要求附上许可证原文和开发者自己的版权资料，目前 **Apache** 和 **BSD** 操作系统也在采用这种许可证。

从软件许可证的变化历程可以看出，软件以及版权的限制正在逐步趋于灵活和宽松，也为开源软件和传统商业软件的融合提供了可能，为以后新型开发模式带来了制度上的保障和启发。

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



```
ziteng@Ziteng-VBox:~$ ls
examples.desktop  snap  公共的  模板  视频  图片  文档  下载  音乐  桌面
ziteng@Ziteng-VBox:~$ pwd
/home/ziteng
ziteng@Ziteng-VBox:~$ xy
xy: 未找到命令
ziteng@Ziteng-VBox:~$ cd ..
ziteng@Ziteng-VBox:/home$ pwd
/home
ziteng@Ziteng-VBox:/home$ cd
ziteng@Ziteng-VBox:~$ pwd
/home/ziteng
ziteng@Ziteng-VBox:~$ cd /usr/include
ziteng@Ziteng-VBox:/usr/include$ ls
python3.6m  reglib  sudo_plugin.h  X11  xorg
ziteng@Ziteng-VBox:/usr/include$ cd
```

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）

```
ziteng@Ziteng-VBox: ~
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)

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 speci-
fied.

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
    with -l, print the author of each file

帮助 -b, --escape
    print C-style escapes for nongraphic characters

--block-size=SIZE
    scale sizes by SIZE before printing them; e.g.,
Manual page ls(1) line 4 (press h for help or q to quit)
```

```
ziteng@Ziteng-VBox: ~
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)

PASSWD(1) 用户命令 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。

然后，测试密码的负责程度。一般来讲，密码应该包含 6 到 8 位字符，从下边
的一个或多个集合中选择：

Manual page passwd(1) line 1 (press h for help or q to quit)
```

```
ziteng@Ziteng-VBox: ~
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
Next: stty invocation, Up: Working context

19.1 'pwd': Print working directory
=====

'pwd' prints the name of the current directory. Synopsis:

    pwd [OPTION]...

The program accepts the following options. Also see *note Common options::.

'-L'
'--logical'
    If the contents of the environment variable 'PWD' provide an
    absolute name of the current directory with no '.' or '..'
    components, but possibly with symbolic links, then output those
    contents. Otherwise, fall back to default '-P' handling.

'-P'
'--physical'
    Print a fully resolved name for the current directory. That is,
    all components of the printed name will be actual directory
    names—none will be symbolic links.

If '-L' and '-P' are both given, the last one takes precedence. If
-----Info: (coreutils)pwd invocation, 37 lines --Top-----
Welcome to Info version 6.5. Type H for help, h for tutorial.
```

```
ziteng@Ziteng-VBox: ~
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)

ziteng@Ziteng-VBox:~$
ziteng@Ziteng-VBox:~$ who --help
用法: who [选项]... [ 文件 | 参数1 参数2 ]
显示当前已登录的用户信息。

-a, --all                等于-b -d --login -p -r -t -T -u 选项的组合
-b, --boot                上次系统启动时间
-d, --dead                显示已死的进程
-H, --heading            输出头部的标题列
--ips                    print ips instead of hostnames. with --lookup,
                        canonicalizes based on stored IP, if available,
                        rather than stored hostname
-l, --login              显示系统登录进程
--lookup                 尝试通过 DNS 规范化主机名
-m                       只针对和标准输入有直接交互的主机和用户
-p, --process            显示由 init 进程衍生的活动进程
-q, --count              列出所有已登录用户的登录名与用户数量
-r, --runlevel           显示当前的运行级别
-s, --short              只显示名称、线路和时间(默认)
-T, -w, --mesg           用+, - 或 ? 标注用户消息状态
-u, --users              列出已登录的用户
--message                等于-T
--writable               等于-T
--help                  显示此帮助信息并退出
--version                显示版本信息并退出

If 显示应用程序 not specified, use /var/run/utmp. /var/log/wtmp as FILE is common.
If ARG1 ARG2 given, -m presumed: 'am i' or 'mom likes' are usual.
```

```
ziteng@Ziteng-VBox: ~
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
ziteng@Ziteng-VBox:~$ 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
-F, --classify            append indicator (one of */=>@|) to entries
                        --file-type      likewise, except do not append '*'
显示应用程序 at=WORD      across -x, commas -m, horizontal -x, long -l,
                        single-column -1, verbose -l, vertical -C
                        full time style=full iso
```

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?

ziteng

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?

Ziteng-VBox

4) 你计算机系统的 CPU 名字是什么?

x86\_64

```
ziteng@Ziteng-VBox:~$ whoami
ziteng
ziteng@Ziteng-VBox:~$ uname
Linux
ziteng@Ziteng-VBox:~$ uname -n
Ziteng-VBox
ziteng@Ziteng-VBox:~$ 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

① cal 4      ② cal 1752      ③ cal 7 2012

```
ziteng@Ziteng-VBox:~$ cal 4
              4
      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  5             1  2             1
  6  7  8  9 10 11 12   3  4  5  6  7  8  9   2  3  4  5  6  7  8
13 14 15 16 17 18 19   10 11 12 13 14 15 16   9 10 11 12 13 14 15
20 21 22 23 24 25 26   17 18 19 20 21 22 23   16 17 18 19 20 21 22
27 28 29 30 31         24 25 26 27 28 29       23 24 25 26 27 28 29
                                   30 31
      四月          五月          六月
```

```
ziteng@Ziteng-VBox:~$ 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
```

```
ziteng@Ziteng-VBox:~$ cal 7 2012
      七月 2012
日 一 二 三 四 五 六
  1  2  3  4  5  6  7
  8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31
```

2) 1752 年有几天，为什么？

只有 355 天，因为 1752 年的 9 月只有 19 天，比正常的日期少了 11 天。原因在于地球公转周期 365.2422 天与 4 年闰年平均周期 365.25 天中间有 0.0078 天的差距，当这个数字扩大成百上千倍时，就产生了数天的偏差。因此为进行修正，罗马教廷将 1582 年的 10 月 5 日改为了 10 月 15 日，而这一改变因为宗教和殖民原因直到 1752 年 9 月才在英国及其美洲殖民地实行，变法后，1752 年的 9 月 3 日被改为 9 月 14 日。

```

ziteng@Ziteng-VBox: ~
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
ziteng@Ziteng-VBox:~$ 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
                   31

   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
 26 27 28 29 30 31  23 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。  
系统已启动运行了 1 小时 2 分钟，当前共有一个登录用户。

```

ziteng@Ziteng-VBox:~$ uptime
21:40:11 up 1:02, 1 user, load average: 1.17, 1.04, 0.94

```

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
<code>touch</code>	修改文件或目录的时间属性，若文件不存在则新建一个空文件	<code>touch hello.c</code>
<code>cp</code>	复制文件到指定文件或目录	<code>cp hello.c /abc/</code>
<code>mv</code>	修改文件或目录的名字，或将文件移入另一个目录	<code>mv hello.c world.c</code>
<code>rm</code>	删除文件或目录	<code>rm hello.c</code>
<code>mkdir</code>	创建文件夹	<code>mkdir 123</code>
<code>who</code>	查看系统当前有哪些登录中的使用者	<code>who -a</code> <code>who -q</code>
<code>ls</code>	列出指定工作目录下的内容	<code>ls -s</code> <code>ls -l hello.c</code>
<code>cd</code>	切换当前目录	<code>cd ..</code> <code>cd ~</code>
<code>pwd</code>	当前工作目录	<code>pwd</code>



open	打开文件
read	从已打开的文件中读取数据
write	向已打开的文件中写入数据
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：

- 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？

有 4 个文件，分别是：

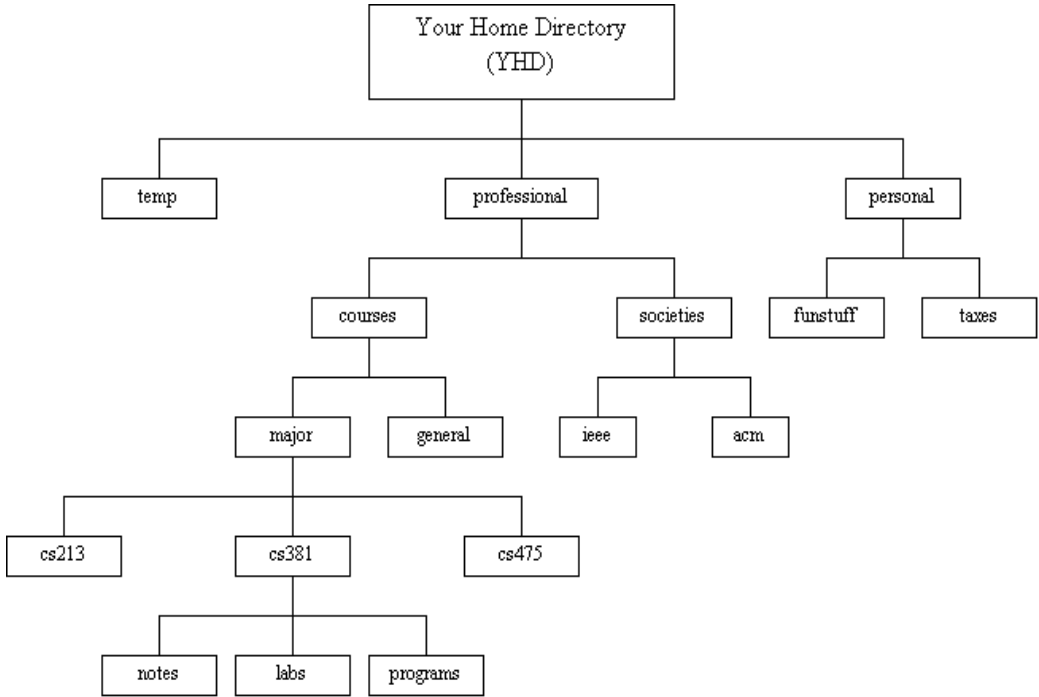
名称	大小(Byte)
----	----------



.	4096
..	4096
abc.txt	0
firscrip	16

只有 ziteng 在使用我的计算机系统，当前工作目录是：/home/ziteng/test

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）



会话如下：  
输入命令

```
ziteng@Ziteng-VBox: ~  
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)  
ziteng@Ziteng-VBox:~$ mkdir temp  
ziteng@Ziteng-VBox:~$ mkdir professional  
ziteng@Ziteng-VBox:~$ mkdir personal  
ziteng@Ziteng-VBox:~$ ls  
examples.desktop  personal      snap  test  模板  图片  下载  桌面  
fircrip           professional  temp  公共的  视频  文档  音乐  
ziteng@Ziteng-VBox:~$ cd personal  
ziteng@Ziteng-VBox:~/personal$ mkdir funstuff  
ziteng@Ziteng-VBox:~/personal$ mkdir taxes  
ziteng@Ziteng-VBox:~/personal$ ls  
funstuff  taxes  
ziteng@Ziteng-VBox:~/personal$ cd ..  
ziteng@Ziteng-VBox:~$ cd professional  
ziteng@Ziteng-VBox:~/professional$ mkdir societies  
ziteng@Ziteng-VBox:~/professional$ mkdir courses  
ziteng@Ziteng-VBox:~/professional$ cd societies  
ziteng@Ziteng-VBox:~/professional/societies$ mkdir ieee  
ziteng@Ziteng-VBox:~/professional/societies$ mkdir acm  
ziteng@Ziteng-VBox:~/professional/societies$ cd ..  
ziteng@Ziteng-VBox:~/professional$ cd courses  
ziteng@Ziteng-VBox:~/professional/courses$ mkdir major  
ziteng@Ziteng-VBox:~/professional/courses$ mkdir general  
ziteng@Ziteng-VBox:~/professional/courses$ cd major  
ziteng@Ziteng-VBox:~/professional/courses/major$ mkdir cs213  
ziteng@Ziteng-VBox:~/professional/courses/major$ mkdir cs381  
ziteng@Ziteng-VBox:~/professional/courses/major$ mkdir cs475  
  
ziteng@Ziteng-VBox:~/professional/courses/major$ cd cs381  
ziteng@Ziteng-VBox:~/professional/courses/major/cs381$ mkdir notes  
ziteng@Ziteng-VBox:~/professional/courses/major/cs381$ mkdir labs  
ziteng@Ziteng-VBox:~/professional/courses/major/cs381$ mkdir programs
```

输出结果

```
ziteng@Ziteng-VBox: ~
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
ziteng@Ziteng-VBox:~$ tree
.
├── examples.desktop
├── firscrip
├── personal
│   ├── funstuff
│   └── taxes
├── professional
│   ├── courses
│   │   ├── general
│   │   └── major
│   │       ├── cs213
│   │       ├── cs381
│   │       │   ├── labs
│   │       │   ├── notes
│   │       │   └── programs
│   │       └── cs475
│   └── societies
│       ├── acm
│       └── ieee
├── snap
│   └── gnome-system-monitor
│       ├── 127
│       ├── 145
│       ├── common
│       └── current -> 127
└── temp
```

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/ziteng`      命令： `echo ~`

输出： `ziteng@Ziteng-VBox:~$ echo ~`  
`/home/ziteng`

- 2) `acm` 目录的绝对路径是什么？

`/home/ziteng/professional/societies/acm`

- 3) 给出 `acm` 目录的两个相对路径。

相对于 `societies`： `./acm`

相对于 `professional`： `./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。

```
ziteng@Ziteng-VBox:~/professional/courses$ cd major/cs381/labs
ziteng@Ziteng-VBox:~/professional/courses/major/cs381/labs$ pwd
/home/ziteng/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 `echo ~`      ③ `cd;pwd`

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？

```
ziteng@Ziteng-VBox:~$ ls -d .*
.      .bashrc  .ICEauthority  .sudo_as_admin_successful
..     .cache   .lesshtst
.bash_history  .config  .local
.bash_logout   .gnupg   .profile
```

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。

```
ziteng@Ziteng-VBox:/$ ls -ai
 2      .      15 initrd.img      1 proc      1310722 tmp
 2      ..     14 initrd.img.old 1048577 root     262146 usr
1179649 bin      655361 lib          2 run      1441793 var
1310721 boot     393217 lib64        786434 sbin      17 vmlinuz
1714020 cdrom    11 lost+found  1179650 snap   16 vmlinuz.old
 2      dev      131073 media      917506 srv
1572865 etc       524289 mnt          12 swapfile
262145  home      786433 opt          1 sys
```

```
ziteng@Ziteng-VBox:~$ ls -ai
280385 .      262672 .ICEauthority  265682 公共的
262145 ..     269907 .lesshtst     265681 模板
269812 .bash_history  262296 .local        265686 视频
280403 .bash_logout  789989 personal     265685 图片
280402 .bashrc      789988 professional 265683 文档
262277 .cache       280401 .profile      265273 下载
263025 .config      269127 snap         265684 音乐
280400 examples.desktop 264251 .sudo_as_admin_successful 265268 桌面
264254 firscrip     661989 temp
262157 .gnupg        661647 test
```

/目录: 2      主目录:280385      temp: 661989  
professional: 789988      personal: 789989

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

```

ziteng@Ziteng-VBox:~/professional/courses/major/cs381/labs$ vi lab1
ziteng@Ziteng-VBox:~/professional/courses/major/cs381/labs$ ls -l lab1
-rw-r--r-- 1 ziteng ziteng 152 7月 21 15:56 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

```

ziteng@Ziteng-VBox:~/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。

```

ziteng@Ziteng-VBox:/usr/include$ ls [t]*.h
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

```
ziteng@Ziteng-VBox: ~  
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)  
ziteng@Ziteng-VBox:~$ ls  
examples.desktop  mediumFile  smallFile  test  视频  下载  
firscrip          personal    snap       公共的  图片  音乐  
largeFile         professional temp       模板   文档  桌面  
ziteng@Ziteng-VBox:~$
```

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`

```
ziteng@Ziteng-VBox:~$ 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`

```
ziteng@Ziteng-VBox:~$ 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`

```
ziteng@Ziteng-VBox:~$ 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？

*cp ./smallFile ./dataFile 修改时间不同；*

```
ziteng@Ziteng-VBox:~$ cp ./smallFile ./dataFile
ziteng@Ziteng-VBox:~$ ls -l smallFile dataFile
-rw-r--r-- 1 ziteng ziteng 963 7月 21 15:38 dataFile
-rw-r--r-- 1 ziteng ziteng 963 7月 21 15:36 smallFile
```

*cp -p ./smallFile ./dataFile 修改时间相同；*

```
ziteng@Ziteng-VBox:~$ cp -p ./smallFile ./dataFile
ziteng@Ziteng-VBox:~$ ls -l smallFile dataFile
-rw-r--r-- 1 ziteng ziteng 963 7月 21 15:36 dataFile
-rw-r--r-- 1 ziteng ziteng 963 7月 21 15:36 smallFile
```

*inode 号: smallFile——262440      dataFile——272982*

```
ziteng@Ziteng-VBox:~$ ls -li
272982 dataFile          789989 personal
280400 examples.desktop  789988 professional
264254 firscrip         262440 smallFile
```

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？

*newDataFile 与 dataFile 的 inode 相同，都为 272982；*

*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。*

```
ziteng@Ziteng-VBox:~$ mv dataFile newDataFile
ziteng@Ziteng-VBox:~$ ls -li
280400 examples.desktop  789989 personal
264254 firscrip         789988 professional
272935 largeFile        262440 smallFile
272934 mediumFile       269127 snap
272982 newDataFile      661989 temp
```

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。

*/tmp/newDataFile 的 inode 号仍然为 272982，因为移动文件不影响 inode 指向的存放文件地址。*



```

ziteng@Ziteng-VBox:~$ mv newDataFile /tmp
ziteng@Ziteng-VBox:~$ cd /tmp
ziteng@Ziteng-VBox:/tmp$ ls -li
1314963 config-err-Zj5Dy5
272982 newDataFile

```

18. 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。

```

ziteng@Ziteng-VBox:~$ wc -cml smallFile mediumFile largeFile /tmp/newDataFile
 20   963   963 smallFile
 71  1953  1954 mediumFile
6024 355394 356570 largeFile
 20   963   963 /tmp/newDataFile
6135 359273 360450 总用量

```

文件	字节数	字数	行数
smallFile	963	963	20
mediumFile	1954	1953	71
largeFile	356570	355394	6024
newDataFile	963	963	20

得到字节数：ls -li smallFile mediumFile largeFile /tmp/newDataFile

```

ziteng@Ziteng-VBox:~$ ls -li smallFile mediumFile largeFile /tmp/newDataFile
-rw-r--r-- 1 ziteng ziteng 356570 7月 21 14:27 largeFile
-rw-r--r-- 1 ziteng ziteng  1954 7月 21 14:25 mediumFile
-rw-r--r-- 1 ziteng ziteng   963 7月 21 15:36 smallFile
-rw-r--r-- 1 ziteng ziteng   963 7月 21 15:36 /tmp/newDataFile

```

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

```

ziteng@Ziteng-VBox:~$ find ~ \( -name '*.c' -o -name '*.html' -o -name '*.htm' \)
/home/ziteng/web.html
/home/ziteng/test.c
/home/ziteng/web1.htm

```

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 ~ \( -cnewer smallFile \)

```

ziteng@Ziteng-VBox:~$ find ~ \( -cnewer smallFile \)
/home/ziteng
/home/ziteng/web.html
/home/ziteng/test.c
/home/ziteng/.local/share/gnome-shell
/home/ziteng/.local/share/gnome-shell/application_state
/home/ziteng/.local/share/keyrings
/home/ziteng/.local/share/gvfs-metadata
/home/ziteng/.local/share/gvfs-metadata/root-d4d122f0.log
/home/ziteng/.local/share/gvfs-metadata/root
/home/ziteng/.local/share/nautilus/scripts
/home/ziteng/.config/dconf
/home/ziteng/.config/dconf/user
/home/ziteng/.config/nautilus
/home/ziteng/.config/nautilus/desktop-metadata
/home/ziteng/web1.htm
/home/ziteng/smallFile
/home/ziteng/tmp
/home/ziteng/.bash_history

```

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*
```

```

ziteng@Ziteng-VBox:~$ find ~ -name *Linux*
/home/ziteng/my_Linux.txt

```

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。

```

ziteng@Ziteng-VBox:~$ ls -ld /
drwxr-xr-x 24 root root 4096 7月 21 16:11 /
ziteng@Ziteng-VBox:~$ ls -ld /etc/passwd
-rw-r--r-- 1 root root 2457 7月 21 16:09 /etc/passwd
ziteng@Ziteng-VBox:~$ ls -ld /bin/df
-rwxr-xr-x 1 root root 84776 1月 18 2018 /bin/df
ziteng@Ziteng-VBox:~$ ls -ld ~
drwxr-xr-x 26 ziteng ziteng 4096 7月 21 15:48 /home/ziteng

```

文件	文件类型	存取权限	链接数	所有者	组	文件大小(B)
/	目录	rw- r-x r-x	24	root	root	4096
/etc/passwd	普通	rw- r-- r--	1	root	root	2457
/bin/df	普通	rw- r-x r-x	1	root	root	84776
~	目录	rw- r-x r-x	26	ziteng	ziteng	4096

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。

```
ziteng@Ziteng-VBox:~$ chmod u=rwx,o=rx personal professional temp
ziteng@Ziteng-VBox:~$ ls -l
总用量 460
```

```
drwxr-xr-x 4 ziteng ziteng 4096 7月 21 15:35 personal
drwxr-xr-x 4 ziteng ziteng 4096 7月 21 15:35 professional
-rw-r--r-- 1 ziteng ziteng 963 7月 21 15:36 smallFile
drwxr-xr-x 3 ziteng ziteng 4096 4月 12 16:58 snap
drwxrwxr-x 5 ziteng ziteng 4096 7月 21 15:35 temp
```

```
d1, d2, d3      u: rwx  g: rx   o: rx
f1              u: rw   g: r    o: r
```

```
ziteng@Ziteng-VBox:~/temp$ mkdir d1 d2 d3
ziteng@Ziteng-VBox:~/temp$ cd d1
ziteng@Ziteng-VBox:~/temp/d1$ touch f1
ziteng@Ziteng-VBox:~/temp/d1$ ls -l
总用量 0
-rw-r--r-- 1 ziteng ziteng 0 7月 21 16:20 f1
ziteng@Ziteng-VBox:~/temp/d1$ cd ..
ziteng@Ziteng-VBox:~/temp$ ls -l
总用量 12
drwxr-xr-x 2 ziteng ziteng 4096 7月 21 16:20 d1
drwxr-xr-x 2 ziteng ziteng 4096 7月 21 16:20 d2
drwxr-xr-x 2 ziteng ziteng 4096 7月 21 16:20 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。

```
ziteng@Ziteng-VBox:~/temp$ cd
ziteng@Ziteng-VBox:~$ chmod a=x temp
ziteng@Ziteng-VBox:~$ ls -ld temp
d--x--x--x 5 ziteng ziteng 4096 7月 21 16:20 temp
ziteng@Ziteng-VBox:~$ ls -l temp
ls: 无法打开目录'temp': 权限不够
ziteng@Ziteng-VBox:~$ chmod a=r temp
ziteng@Ziteng-VBox:~$ ls -l temp
ls: 无法访问'temp/d1': 权限不够
ls: 无法访问'temp/d3': 权限不够
ls: 无法访问'temp/d2': 权限不够
总用量 0
d????????? ? ? ? ?      ? d1
d????????? ? ? ? ?      ? d2
d????????? ? ? ? ?      ? d3
ziteng@Ziteng-VBox:~$ chmod a=rx temp
ziteng@Ziteng-VBox:~$ ls -l temp
总用量 12
drwxr-xr-x 2 ziteng ziteng 4096 7月 21 16:20 d1
drwxr-xr-x 2 ziteng ziteng 4096 7月 21 16:20 d2
drwxr-xr-x 2 ziteng ziteng 4096 7月 21 16:20 d3
```

仅为执行权限无法打开目录，ls -l 所需最小权限为 rx，因为 r 代表可以读出目录下的文件名，x 代表能够进入该目录，只有 rx 都具备时才能够使用 ls -l temp 命令进入并浏览该目录。

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。

```
ziteng@Ziteng-VBox:~$ umask
0022
```

26. 删除 ~/temp 目录下的所有文件和目录。给出会话过程。

```
ziteng@Ziteng-VBox:~/temp$ rm -r ./ *
ziteng@Ziteng-VBox:~/temp$ ls
ziteng@Ziteng-VBox:~/temp$
```

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。

```
ziteng@Ziteng-VBox:~/temp$ mkdir d1 d2 d3
ziteng@Ziteng-VBox:~/temp$ cp ~/smallFile ./d1
ziteng@Ziteng-VBox:~/temp$ cd d1
ziteng@Ziteng-VBox:~/temp/d1$ ls -il
总用量 4
661579 -rw-r--r-- 1 ziteng ziteng 963 7月 21 16:26 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。

```

ziteng@Ziteng-VBox:~/temp$ cd d2
ziteng@Ziteng-VBox:~/temp/d2$ ln ../d1/smallFile newFile.hard
ziteng@Ziteng-VBox:~/temp/d2$ ls -l
总用量 4
-rw-r--r-- 2 ziteng ziteng 963 7月 21 16:26 newFile.hard
ziteng@Ziteng-VBox:~/temp/d2$ ls -li ../d1 ./
./:
总用量 4
661579 -rw-r--r-- 2 ziteng ziteng 963 7月 21 16:26 newFile.hard
../d1:
总用量 4
661579 -rw-r--r-- 2 ziteng ziteng 963 7月 21 16:26 smallFile

```

两文件属性除名字外完全一样,确定两文件相同的因素是inode 号完全相同,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。

```

ziteng@Ziteng-VBox:~/temp/d2$ cat newFile.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 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

```

取消权限

```

ziteng@Ziteng-VBox:~/temp/d2$ chmod u-r ../d1/smallFile
ziteng@Ziteng-VBox:~/temp/d2$ cat newFile.hard
cat: newFile.hard: 权限不够

```

发现打开权限不够,可推测出硬链接是一个指向原文件索引的指针,原文件的改变能够引起硬链接指向文件的改变。

增加读权限后,可以完全正确显示。

删除后发现仍然可以打开,说明硬链接相当于存储了文件的一份副本,并在其基础上同步修改。



30. 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。

```
ziteng@Ziteng-VBox:~/temp/d1$ ln -s smallFile ../d2/smallFile.soft
ziteng@Ziteng-VBox:~/temp/d1$ ls -li smallFile ../d2/smallFile.soft
661600 lrwxrwxrwx 1 ziteng ziteng 9 7月 21 16:40 ../d2/smallFile.soft -> smallFile
661579 -rw-r--r-- 2 ziteng ziteng 963 7月 21 16:26 smallFile
```

这两个文件大多数信息都不相同，如大小、inode、修改时间、名称等，因此可判断两个不同文件。但决定是两个不同文件的因素是 inode 号不同，表明是分别存储的两个文件。

31. 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。

取消 r 权限前能够正常显示文件，但取消后系统提示权限不够。

结合两文件的大小差异，可以推断出软链接只是一个指向原文件的快捷方式，由于原文件的权限被修改了，导致快捷方式也无法打开。

增加读权限后又可以再次显示原文件内容。

删除源文件后，又不能够通过软链接访问了，说明其本质仍是快捷方式，在指向原文件被删除后，软链接同样失效。

```
ziteng@Ziteng-VBox:~/temp/d1$ rm smallFile
ziteng@Ziteng-VBox:~/temp/d1$ ls -li ../d2/smallFile.soft
661600 lrwxrwxrwx 1 ziteng ziteng 9 7月 21 16:40 ../d2/smallFile.soft -> smallFile
ziteng@Ziteng-VBox:~/temp/d1$ 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。

有两个进程在运行，其中 bash 的 PID 为 2804，ps 的 PID 为 2903

ps 的父进程 ID 为 2804 (bash)，bash 的父进程为 2795

```
ziteng@Ziteng-VBox:~$ ps -l
F S  UID  PID  PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
0 S  1000  2804  2795  0  80   0 -  7457 wait  pts/0      00:00:00 bash
0 R  1000  2903  2804  0  80   0 -  9006 -      pts/0      00:00:00 ps
ziteng@Ziteng-VBox:~$
```

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。

```

ziteng@Ziteng-VBox:~$ (sleep 60; echo "Time for Lunch!");) &
[1] 3011
ziteng@Ziteng-VBox:~$ ps
  PID TTY          TIME CMD
 2804 pts/0        00:00:00 bash
 3011 pts/0        00:00:00 bash
 3012 pts/0        00:00:00 sleep
 3013 pts/0        00:00:00 ps
ziteng@Ziteng-VBox:~$ 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。

```

ziteng@Ziteng-VBox:~$ date & uname -a & who & ps
[1] 5579
[2] 5580
[3] 5581
2020年 07月 21日 星期二 16:47:09 CST
ziteng :0 2020-07-21 15:18 (:0)
Linux Ziteng-VBox 5.3.0-62-generic #56~18.04.1-Ubuntu SMP Wed Jun 24 16:17:03 UTC 2020
x86_64 x86_64 x86_64 GNU/Linux
  PID TTY          TIME CMD
 2064 pts/0        00:00:00 bash
 5582 pts/0        00:00:00 ps
[1] 已完成 date
[2]- 已完成 uname -a
[3]+ 已完成 who

```

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。

```

ziteng@Ziteng-VBox:~$ date && uname -a && who && ps
2020年 07月 21日 星期二 16:48:27 CST
Linux Ziteng-VBox 5.3.0-62-generic #56~18.04.1-Ubuntu SMP Wed Jun 24 16:17:03 UTC 2020
x86_64 x86_64 x86_64 GNU/Linux
ziteng :0 2020-07-21 15:18 (:0)
  PID TTY          TIME CMD
 2064 pts/0        00:00:00 bash
 5588 pts/0        00:00:00 ps

```

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



```

ziteng@Ziteng-VBox:~$ pwd
/home/ziteng
ziteng@Ziteng-VBox:~$ bash
ziteng@Ziteng-VBox:~$ cd /usr
ziteng@Ziteng-VBox:/usr$ pwd
/usr
ziteng@Ziteng-VBox:/usr$ exit
ziteng@Ziteng-VBox:~$ pwd
/home/ziteng

```

- ① /home/ziteng    ② /usr    ③ /home/ziteng

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）

```

ziteng@Ziteng-VBox:~$ (find ~ -name foobar) 2> /dev/null
/home/ziteng/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。

```

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

```

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
```

```
打开(O)  fout ~/
-n "The time now is:"
2020年 07月 21日 星期二 16:52:02 CST
-n "The users presently logged on are:"
ziteng :0 2020-07-21 15:18 (: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。

行数：28      单词数：245      字符数：1568

```
ziteng@Ziteng-VBox:~$ ls -l | wc -clw
28      245    1568
```

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？

```
ziteng@Ziteng-VBox:~$ find /bin -type f | wc -l
139
ziteng@Ziteng-VBox:~$ find /bin -type d | wc -l
1
ziteng@Ziteng-VBox:~$ find /bin -type l | wc -l
23
```

### 三、讨论、心得（必填）

本次实验主要对 2~5 章的内容做了复习和实际操作，同时为了更好的展示效果需要安装额外的配置包，做到了在复习课件的同时，还需要上网查找相关资料，让我们对于命令的格式和结果的成因有了更深层次的了解。以下是我在实验过程中遇到的一些问题：

1. 实验 9 中，需要我们展示出当前文件夹的层次架构，但在此多次使用 `list` 会效率很低，而且效果不好，而 `ubuntu` 又没有自带的文件夹层次展示命令，如 `tree`，因此需要首先在系统中安装相应的包。通过上网查找相关解决方案，并根据系统提示，我通过输入 `sudo apt-get install tree` 完成了安装 `tree` 指令包，并顺利完成了实验。
2. 在实验 15 中，我遇到了本次实验最难以解决的问题：即 `windows` 与 `ubuntu` 的剪贴板共享。由于默认无法将数据拷贝至 `ubuntu` 中，我上网查询了相关方案，但在按照教程配置好 `virtualbox` 的高级配置、安装增强功能以及使用主机 I/O 缓存后，仍然无法共享剪贴板。但在问题面前，我想到了共享文件夹的方式，通过将需要复制粘贴的内容存储在共享文件夹的 `txt` 文件内，我实现了内容的转移，并顺利完成了该实验题。
3. `linux` 有许多快捷键的使用方式与 `windows` 不同，比如复制粘贴还需要按下 `shift` 键，而 `ctrl+d` 则是结束输入，`ctrl+z` 是挂起进程等。

4. 在实验过程中想要启动 root 权限时，输入 `su root` 和密码却提示认证失败，在查阅资料后发现 ubuntu 在安装后默认锁定 root，因此需要先输入 `sudo passwd` 解锁 root 用户，进而进入 root 权限。之后通过 `ctrl+d` 或 `exit` 即可回到用户模式。

本次实验还让我发现了 linux 系统的强大之处，比如环境配置上，只需要在终端输入一行代码就可以实现 gcc 环境的配置，相比于 windows 下载软件、添加环境变量、添加配置性文件等操作简单了太多，另一方面，linux 一切皆文件的思想也使得我们能够更近距离接触到操作的过程性文件，拉近了我们与更底层操作系统的距离，对我们日后的学习有着启发式的作用。