

浙江大学实验报告

课程名称： Linux 程序设计 实验类型： 验证型

实验项目名称： 实验 1 Linux shell 基本命令

学生姓名： 黄彦玮 专业： 混合班 学号： 3180102067

电子邮件地址： 3180102067@zju.edu.cn 手机（可选）： 15852758315

实验日期： 2020 年 07 月 14 日

一、 实验环境

配置：

内存：4GB

CPU：Intel® Core™ i7-8750H CPU @ 2.20GHz

处理器：2

硬盘：40GB

操作系统环境：Ubuntu 64 位

Linux 版本： Ubuntu 20.04

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？

软件许可证是一种具有法律性质的合同或指导，旨在规范受著作权保护的软件的使用或散布行为。通常的许可方式会允许用户来使用单一或多份该软件的复制，因为若无许可而径予使用该软件，将违反著作权法给予软件开发者的专属保护。效用上来说，软件许可是软件开发者与其用户之间的一份合约，用来保证在符合许可范围的情况下，用户将不会受到控告。

国际上常用的各种软件许可证包括 MIT License、Apache、GPL、AGPL 等等。总的来说，现在国际上的软件许可证非常多元，宽松程度也各不相同，即赋予用户的权利各不相同，可以满足各种类型的软件开发者的需要，同时也以

具有法律效力的形式规定了用户的使用权利，保障了软件开发者的合法权益。一些软件许可证的广泛应用，如 Apache、GPL 等，也反映了这些许可证能普遍地满足较多开发者的实际需求。

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

```
hyw@ubuntu:~$ ls
Desktop  Downloads  myout.txt  Public  Videos
Documents  Music      Pictures   Templates
```

解释：ls 命令用于显示指定工作目录下之内容，这里直接使用则列出了当前工作文件夹下的所有文件夹和文件。

b. \$ pwd

```
hyw@ubuntu:~$ pwd
/home/hyw
```

解释：pwd 命令的作用是显示当前工作目录的路径。这里显示的是主目录的路径。

c. \$ xy #只看结果，不要问为什么没有 xy 命令

```
hyw@ubuntu:~$ xy
xy: command not found
```

解释：由于 shell 在环境变量中的目录下找不到 xy 这个命令的执行程序，因此显示 command not found.

d. \$ cd ..

e. \$ pwd

```
hyw@ubuntu:~$ cd ..
hyw@ubuntu:/home$ pwd
/home
```

解释：cd .. 表示打开上一级目录，之前主目录的位置是/home/hyw，返回上一级即/home。这时 pwd 打印当前工作目录就是/home。

f. \$ cd

g. \$ pwd

```
hyw@ubuntu:/home$ cd
hyw@ubuntu:~$ pwd
/home/hyw
```

解释：cd 命令单独使用表示打开主目录，因此 pwd 打印当前工作目录就是 /home/hyw。

```
h. $ cd /usr/include
i. $ ls
j. $ cd
```

```
hyw@ubuntu:~$ cd /usr/include
hyw@ubuntu:/usr/include$ ls
gcalc-2 gci-2 iproute2 openvpn python3.8 reglib sudo_plugin.h X11 xorg
hyw@ubuntu:/usr/include$ cd
hyw@ubuntu:~$
```

解释：cd 后接具体的目录名称则切换当前工作目录至该目录，再输入 ls 则列出该目录下的子目录与文件名称。最后输入单独一条 cd 指令再次回到主目录，可以看到提示符中的当前目录变为了“~”即主目录。

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）

(1) man ls

命令输入：

```
hyw@ubuntu:~$ man ls
```

输出显示：

```
LS(1)                                User Commands                                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 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
    Manual page ls(1) line 1 (press h for help or q to quit)
```

按下 space 键可以显示下一屏的文本信息。输入 q 即退出此界面。

解释：man ls 显示 ls 的操作手册，可以看到手册给出了 ls 的全称（list directory contents）、语法（SYNOPSIS）、各个参数描述等等信息。

（2）man passwd

命令输入：

```
man passwd
```

输出显示：

```
PASSWD(1)                                User Commands                                PASSWD(1)

NAME
    passwd - change user password

SYNOPSIS
    passwd [options] [LOGIN]

DESCRIPTION
    The passwd command changes passwords for user accounts. A normal user
    may only change the password for their own account, while the superuser
    may change the password for any account. passwd also changes the
    account or associated password validity period.

    Password Changes
    The user is first prompted for their old password, if one is present.
    This password is then encrypted and compared against the stored
    password. The user has only one chance to enter the correct password.
    The superuser is permitted to bypass this step so that forgotten
    passwords may be changed.

    After the password has been entered, password aging information is
    checked to see if the user is permitted to change the password at this
    Manual page passwd(1) line 1 (press h for help or q to quit)
```

解释：man passwd 显示 passwd 的操作手册，可以看到手册给出了 passwd 的全称（change user password）、语法（SYNOPSIS）、修改密码的具体描述等等信息。

（3）info pwd

命令输入：

```
hyw@ubuntu:~$ info pwd
```

输出显示：

```

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,
    -----Info: (coreutils)pwd invocation, 37 lines --Top-----
Welcome to Info version 6.7.  Type H for help, h for tutorial.

```

按空格可上下翻页，按 q 可退出。

解释：info pwd 显示 pwd 的操作手册，给出了指令作用（打印当前目录的名称）、语法、参数等基本信息，格式与 man 略有不同。与 man 相比，info 工具可显示更完整的最新的 GNU 工具信息。

(4) who --help

```

hyw@ubuntu:~$ who --help
Usage: who [OPTION]... [ FILE | ARG1 ARG2 ]
Print information about users who are currently logged in.

-a, --all             same as -b -d --login -p -r -t -T -u
-b, --boot            time of last system boot
-d, --dead            print dead processes
-H, --heading         print line of column headings
--ips                 print ips instead of hostnames. with --lookup,
                     canonicalizes based on stored IP, if available,
                     rather than stored hostname
-l, --login           print system login processes
--lookup              attempt to canonicalize hostnames via DNS
-m                   only hostname and user associated with stdin
-p, --process         print active processes spawned by init
-q, --count           all login names and number of users logged on
-r, --runlevel        print current runlevel
-s, --short           print only name, line, and time (default)
-t, --time            print last system clock change
-T, -w, --mesg       add user's message status as +, - or ?
-u, --users           list users logged in
--message             same as -T
--writable            same as -T

```

解释：--help 命令也可以显示命令的用法，这里对 who 命令使用显示了其语法（usage）、各参数含义，显示的内容与使用 info、man 命令显示的大致相同，不同的是--help 本质上是一种命令的选项。

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？
 - 2) 你的操作系统名字是什么？
 - 3) 你计算机系统的域名是什么？
 - 4) 你计算机系统的 CPU 名字是什么？

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

由上图可见，我的用户名是 `hyw`，操作系统名字为 `Linux`，域名为 `ubuntu`，CPU 类型为 `x86_64`。

解释：`whoami` 可以用于打印用户名；`uname` 命令无参数等同于 `-s` 参数，即输出内核名；`uname -n` 命令打印网络域名；`uname -p` 命令打印 CPU 类型（不是 CPU 名字，题目略有错误）。以上指令的用法和作用可以用 `man uname` 简单查找得到。

5. 简述 Unix epoch（Unix 纪元）？用 `cal` 命令显示下列年份的日历：4、1752、2012 年 7 月。

unix 时间戳是从 1970 年 1 月 1 日（UTC/GMT 的午夜）开始所经过的秒数，不考虑闰秒。

- 1) 给出你显示以上年份年历的命令

`cal 4`

`cal 1752`

`cal Jul 2012`

```
hyw@ubuntu:~$ cal Jul 2012
      July 2012
Su Mo Tu We Th Fr Sa
 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
```

```
hyw@ubuntu:~$ cal 4
          4
   January February March
Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa
 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

   April May June
Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa
 1  2  3  4  5          1  2  3          1  2  3  4  5  6  7
 6  7  8  9 10 11 12    4  5  6  7  8  9 10    8  9 10 11 12 13 14
13 14 15 16 17 18 19   11 12 13 14 15 16 17   15 16 17 18 19 20 21
20 21 22 23 24 25 26   18 19 20 21 22 23 24   22 23 24 25 26 27 28
27 28 29 30          25 26 27 28 29 30 31   29 30

   July August September
Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa
 1  2  3  4  5          1  2          1  2  3  4  5  6
 6  7  8  9 10 11 12    3  4  5  6  7  8  9    7  8  9 10 11 12 13
13 14 15 16 17 18 19   10 11 12 13 14 15 16   14 15 16 17 18 19 20
20 21 22 23 24 25 26   17 18 19 20 21 22 23   21 22 23 24 25 26 27
27 28 29 30 31        24 25 26 27 28 29 30   28 29 30
                                   31

   October November December
Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa
 1  2  3  4          1          1  2  3  4  5  6
 5  6  7  8  9 10 11    2  3  4  5  6  7  8    7  8  9 10 11 12 13
12 13 14 15 16 17 18    9 10 11 12 13 14 15   14 15 16 17 18 19 20
19 20 21 22 23 24 25   16 17 18 19 20 21 22   21 22 23 24 25 26 27
26 27 28 29 30 31     23 24 25 26 27 28 29   28 29 30 31
                                   30
                                   31
```

```
hyw@ubuntu:~$ cal 1752
          1752
   January February March
Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa
 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

   April May June
Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa
 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

   July August September
Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa
 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

   October November December
Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa Su Mo Tu We Th Fr Sa
 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
```

2) 1752 年有几天，为什么？
 355 天，因为 9 月 2 日之后是 9 月 14 日，抹掉这 11 天是当时英国议会做出的决定。

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。

```
hyw@ubuntu:~$ uptime
10:26:08 up 4:37, 1 user, load average: 0.00, 0.00, 0.00
```

可以看到当前已运行了 4 小时 37 分钟，共有 1 个用户。

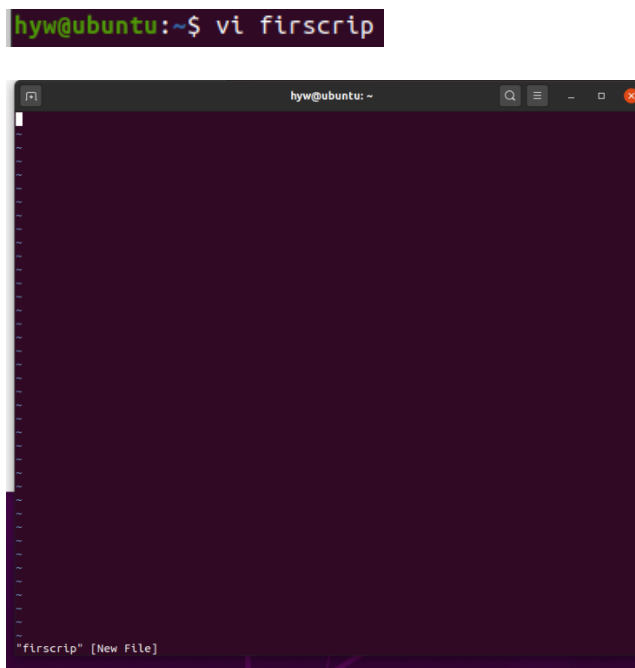
7. 通过 Linux 的 `man`、`info` 命令或互联网得到下面的 shell 命令、系统调用和库函数功能描述及每个命令使用例子：

命令	命令功能的简要描述	实例
<code>touch</code>	改变文件的时间戳	<code>touch a.txt</code> （若 <code>a.txt</code> 不存在新建该文件，否则更新修改日期）
<code>cp</code>	拷贝文件或目录	<code>cp a.txt b.txt</code> （将 <code>a.txt</code> 的内容复制到 <code>b.txt</code> ）
<code>mv</code>	移动文件	<code>mv a.txt b.txt</code> （将 <code>a.txt</code> 内容移动到 <code>b.txt</code> 并删除 <code>a.txt</code> ）
<code>rm</code>	删除文件或目录	<code>rm a.txt</code> （删除 <code>a.txt</code> ）
<code>mkdir</code>	创建文件夹	<code>mkdir aaa</code> （新建一个名为 <code>aaa</code> 的文件夹）
<code>who</code>	显示登陆的用户	<code>who</code> （显示当前登陆用户）
<code>ls</code>	列出目录下的信息	<code>ls -al</code> （显示当前目录所有文件的详细信息）
<code>cd</code>	改变工作目录	<code>cd ..</code> （打开上级目录）
<code>pwd</code>	打印当前工作目录	<code>pwd</code> （打印当前工作目录）
<code>open</code>	<code>open</code> 建立了一条到文件或设备的访问路径。如果调用成功，它将返回一个可以被 <code>read</code> 、 <code>write</code> 和其他系统调用使用的文件描述符。如： <code>open("aaa.txt", O_CREAT O_RDWR, S_IRUSR S_IWUSR);</code>	
<code>read</code>	<code>read(int fildes,void *buf,size_t nbytes)</code> 表示从文件描述符 <code>fildes</code> 相关联的文件里读入 <code>nbytes</code> 个字节的数据，并把它们放到数据区 <code>buf</code> 中。例： <code>char buffer[30]; size_t x = read(0,buffer,30);</code>	

write	write (int fildes,const void *buf,size_t nbytes) 把缓冲区 buf 的前 nbytes 个字节写入与文件描述符 fildes 关联的文件中。例： write(1,"hahaha!\n",20);
close	用来终止文件描述符 fildes 与其对应文件之间的关联，如 close(0)
pipe	创建管道文件，如 int fd_pipe[2] = {0};pipe(fd_pipe);创建一个空管道
socket	创建一个 socket 描述符（socket descriptor），它唯一标识一个 socket。如 socket(AF_INET,SOCK_STREAM,0)
mkfifo	创建一个有名管道，如 mkfifo(/tmp/fifo,O_CREAT O_NONBLOCK O_RDWR)
system	产生一个子进程执行命令，如 system("mkdir aaa")
printf	进行打印操作，如 printf("pid")

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；



解释：vi 命令打开了一个新的空文件 firscrip 并进入编辑界面。

- 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：

```
ls -la
who
pwd
~
~
~
```

- 5) 在按<Esc>键;
- 6) 再输入:wq, 并按<Enter>键;

```
:wq
```

按回车后回到 shell 界面:

```
hyw@ubuntu:~$
```

解释: 按 Esc 退出编辑模式, :wq 的意思是保存并退出 vim 界面。

- 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?

```
hyw@ubuntu:~$ bash ./firscrip
total 84
drwxr-xr-x 14 hyw  hyw  4096 Jul 14 15:42 .
drwxr-xr-x  3 root root  4096 Jul  6 09:54 ..
-rw-rw-r--  1 hyw  hyw    6 Jul 14 10:41 233
-rw-----  1 hyw  hyw   723 Jul 14 15:34 .bash_history
-rw-r--r--  1 hyw  hyw   220 Jul  6 09:54 .bash_logout
-rw-r--r--  1 hyw  hyw  3771 Jul  6 09:54 .bashrc
drwxr-xr-x 11 hyw  hyw  4096 Jul  6 10:36 .cache
drwxr-xr-x 13 hyw  hyw  4096 Jul 14 10:41 .config
drwxr-xr-x  2 hyw  hyw  4096 Jul  6 10:31 Desktop
drwxr-xr-x  2 hyw  hyw  4096 Jul  6 10:31 Documents
drwxr-xr-x  2 hyw  hyw  4096 Jul  6 10:31 Downloads
-rw-rw-r--  1 hyw  hyw   16 Jul 14 15:42 firscrip
drwx-----  3 hyw  hyw  4096 Jul  6 10:31 .gnupg
drwxr-xr-x  3 hyw  hyw  4096 Jul  6 10:31 .local
drwxr-xr-x  2 hyw  hyw  4096 Jul  6 10:31 Music
-rw-rw-r--  1 hyw  hyw    6 Jul 13 14:22 myout.txt
drwxr-xr-x  2 hyw  hyw  4096 Jul 14 15:30 Pictures
-rw-r--r--  1 hyw  hyw   807 Jul  6 09:54 .profile
drwxr-xr-x  2 hyw  hyw  4096 Jul  6 10:31 Public
drwxr-xr-x  2 hyw  hyw  4096 Jul  6 10:31 Templates
drwxr-xr-x  2 hyw  hyw  4096 Jul  6 10:31 Videos
hyw      :0                2020-07-06 10:31 (:0)
/home/hyw
hyw@ubuntu:~$
```

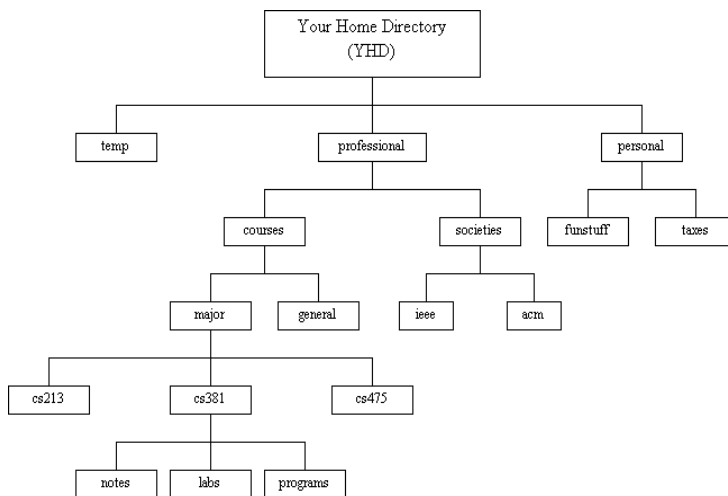
共有 84 个文件, 名称见上图的最后一列 (前三个文件分别为., ., ., 233), 大小见上图的第 5 列 (前三个文件分别为 4096, 4096, 6)

当前只有 hyw 一个用户在使用操作系统。

当前工作文件夹是 /home/hyw。

解释: `bash ./firscrip` 表示新建一个进程执行 `firscrip` 中的命令, 第一个命令 `ls -la` 列出所有文件的详细信息, 根据其输出格式即可知道当前目录下的文件数、文件名等信息; `who` 列出了所有登陆用户; `pwd` 打印了当前工作文件夹。

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)



```
hyw@ubuntu:~$ mkdir temp
hyw@ubuntu:~$ mkdir professional
hyw@ubuntu:~$ mkdir personal
hyw@ubuntu:~$ cd professional
hyw@ubuntu:~/professional$ mkdir courses
hyw@ubuntu:~/professional$ mkdir societies
hyw@ubuntu:~/professional$ cd courses
hyw@ubuntu:~/professional/courses$ mkdir major
hyw@ubuntu:~/professional/courses$ mkdir general
hyw@ubuntu:~/professional/courses$ cd major
hyw@ubuntu:~/professional/courses/major$ mkdir cs213
hyw@ubuntu:~/professional/courses/major$ mkdir cs381
hyw@ubuntu:~/professional/courses/major$ mkdir cs475
hyw@ubuntu:~/professional/courses/major$ cd cs381
hyw@ubuntu:~/professional/courses/major/cs381$ mkdir notes
hyw@ubuntu:~/professional/courses/major/cs381$ mkdir labs
hyw@ubuntu:~/professional/courses/major/cs381$ mkdir programs
hyw@ubuntu:~/professional/courses/major/cs381$ cd ..
hyw@ubuntu:~/professional/courses/major$ cd ..
hyw@ubuntu:~/professional/courses$ cd ..
hyw@ubuntu:~/professional$ cd societies
hyw@ubuntu:~/professional/societies$ mkdir ieee
hyw@ubuntu:~/professional/societies$ mkdir acm
hyw@ubuntu:~/professional/societies$ cd
hyw@ubuntu:~$ cd personal
hyw@ubuntu:~/personal$ mkdir funstuff
hyw@ubuntu:~/personal$ mkdir taxes
```

```

hyw@ubuntu:~$ tree -l
.
├── 233
├── Desktop
├── Documents
├── Downloads
├── firscrip
├── Music
├── myout.txt
├── personal
│   ├── funstaff
│   └── taxes
├── Pictures
│   ├── Screenshot from 2020-07-14 09-07-31.png
│   ├── Screenshot from 2020-07-14 09-10-00.png
│   ├── Screenshot from 2020-07-14 09-22-08.png
│   ├── Screenshot from 2020-07-14 09-34-03.png
│   ├── Screenshot from 2020-07-14 09-50-58.png
│   └── Screenshot from 2020-07-14 15-30-41.png
├── professional
│   ├── courses
│   │   ├── general
│   │   └── major
│   │       ├── cs213
│   │       ├── cs381
│   │       │   ├── labs
│   │       │   ├── notes
│   │       │   └── programs
│   │       └── cs475
│   └── societies
│       ├── acm
│       └── ieee
├── Public
├── temp
├── Templates
└── Videos

25 directories, 9 files

```

解释：通过 `cd` 切换目录再用 `mkdir` 命令新建目录即可新建目录树，安装 `tree` 命令后即可直接输出目录树。

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。

```

hyw@ubuntu:~$ cd professional/courses
hyw@ubuntu:~/professional/courses$ echo ~
/home/hyw

```

解释：`echo ~` 的意思是打印主目录路径。

- 2) `acm` 目录的绝对路径是什么？

`/home/hyw/professional/societies/acm`

- 3) 给出 `acm` 目录的两个相对路径。

`../societies/acm` `../../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。

```
hyw@ubuntu:~/professional/societies$ ../societies/acm
bash: ../societies/acm: Is a directory
hyw@ubuntu:~/professional/societies$ pwd
/home/hyw/professional/societies
```

- 5) 给出获得你的主目录三个不同的命令。

```
hyw@ubuntu:~$ cd ; pwd
/home/hyw
hyw@ubuntu:~$ echo ~
/home/hyw
hyw@ubuntu:~$ echo $HOME
/home/hyw
```

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？

```
hyw@ubuntu:~$ ls -a
.          .bashrc    Downloads  myout.txt   Public
..         .cache     firscrip   personal    .sudo_as_admin_successful
233        .config    .gnupg     Pictures     temp
.bash_history .Desktop   .local     professional Templates
.bash_logout .Documents Music       .profile     Videos
```

.bashrc .bash_history .bash_logout .profile .sudo_as_admin_successful
... .cache .config .gnupg .local

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。

```
hyw@ubuntu:~$ cd /
hyw@ubuntu:/$ ls -la
 2 .          1310721 etc          11 lost+found      2 run          655361 tmp
 2 ..         1048577 home         131073 media        18 sbin         1441793 usr
13 bin         14 lib        1703937 mnt          2359297 snap       917505 var
1572865 boot     15 lib32      524289 opt          393217 srv
1179649 cdrom    16 lib64      1 proc         12 swapfile
 2 dev         17 libx32     2097153 root         1 sys
hyw@ubuntu:/$ cd
hyw@ubuntu:~$ ls -la
1048578 .          1048893 Documents  1049351 professional
1048577 ..         1048890 Downloads 1048581 .profile
1049084 233        1048936 firscrip   1048892 Public
1048637 .bash_history 1048870 .gnupg     1049366 .sudo_as_admin_successful
1048580 .bash_logout 1048835 .local     1048952 temp
1048579 .bashrc      1048894 Music      1048891 Templates
1048848 .cache       1048968 myout.txt  1048896 Videos
1048839 .config      1049352 personal
1048889 Desktop    1048895 Pictures
```

/的 inode 号为 2，~的 inode 号为 1048578，~/temp 的 inode 号为 1048952，~/professional 的 inode 号为 1049351，~/personal 的 indoe 号为 1049352。

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。
- 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。

```
hyw@ubuntu:~$ cd professional
hyw@ubuntu:~/professional$ cd courses
hyw@ubuntu:~/professional/courses$ cd major
hyw@ubuntu:~/professional/courses/major$ cd cs381
hyw@ubuntu:~/professional/courses/major/cs381$ cd labs
hyw@ubuntu:~/professional/courses/major/cs381/labs$ vi lab1
hyw@ubuntu:~/professional/courses/major/cs381/labs$ ls -l
total 4
-rw-rw-r-- 1 hyw hyw 152 Jul 14 16:31 lab1
hyw@ubuntu:~/professional/courses/major/cs381/labs$ file lab1
lab1: ASCII text
```

文件类型为普通文件，文件内容为 ASCII 文本。

解释：ls -l 给出文件的详细信息，第一个字段的第一个字符表示文件类型，“-”表示普通文件。file 给出文件内容的详细信息。

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。

```
hyw@ubuntu:~$ cd /usr/include
hyw@ubuntu:/usr/include$ ls t*.h
tar.h  termio.h  termios.h  tgmath.h  thread_db.h  threads.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 命令创建一个大文件；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

```
hyw@ubuntu:~$ man cat > mediumFile
hyw@ubuntu:~$ man bash > largeFile
hyw@ubuntu:~$ gedit smallFile
```

在弹出的 gedit 文本编辑器中输入上述文本，点击保存后关闭即可。

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。

```
hyw@ubuntu:~$ head -n 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) 1989-2018 by the Free Software Foundation, Inc.

DESCRIPTION
hyw@ubuntu:~$ tail -n 5 smallFile
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
```

```
hyw@ubuntu:~$ 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？

```
hyw@ubuntu:~$ cp smallFile dataFile
hyw@ubuntu:~$ ls -l smallFile
-rw-rw-r-- 1 hyw hyw 964 Jul 14 17:24 smallFile
hyw@ubuntu:~$ ls -l dataFile
-rw-rw-r-- 1 hyw hyw 964 Jul 14 17:26 dataFile
hyw@ubuntu:~$ cp -p smallFile smallFile2
hyw@ubuntu:~$ ls -il smallFile
1048935 -rw-rw-r-- 1 hyw hyw 964 Jul 14 17:24 smallFile
hyw@ubuntu:~$ ls -il dataFile
1049267 -rw-rw-r-- 1 hyw hyw 964 Jul 14 17:26 dataFile
hyw@ubuntu:~$ ls -il smallFile2
1049342 -rw-rw-r-- 1 hyw hyw 964 Jul 14 17:24 smallFile2
```

答：用 cp -p 命令即可保留修改时间不变。smallFile 和 dataFile 的 Inode 号分别为 1048935 和 1049267。（使用 cp -p 命令创建的 smallFile2 文件的 inode 号为 1049342）

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？

```
hyw@ubuntu:~$ mv dataFile newDataFile
hyw@ubuntu:~$ ls -il newDataFile
1049267 -rw-rw-r-- 1 hyw hyw 964 Jul 14 17:26 newDataFile
```

答：文件 newDataFile 的 inode 为 1049267，与 dataFile 文件的 inode 号相同。这是因为执行 mv 指令时，系统会新建一个目录项，将新档案名称对应到 inode number，然后删除旧档案，所以 inode number 不变。

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。

```
hyw@ubuntu:~$ mv newDataFile /tmp
hyw@ubuntu:~$ ls -il /tmp/newDataFile
1049267 /tmp/newDataFile
```

文件/tmp/newDataFile 的 inode 号是 1049267，与之前的 dataFile、newDataFile 的 inode number 均相同。原因同样是因为 mv 指令只是将新档案名称对应到 inode number，并不创建新的 inode number，所以 inode 号不变。

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。

。


```
hyw@ubuntu:~$ wc smallFile
21 120 964 smallFile
hyw@ubuntu:~$ wc mediumFile
71 215 1962 mediumFile
hyw@ubuntu:~$ wc largeFile
6091 48769 370000 largeFile
hyw@ubuntu:~$ wc /tmp/newDataFile
21 120 964 /tmp/newDataFile
```

行数分别为 21、71、6091、21；字数分别为 120、215、48769、120；行数分别为 964、1962、370000、964。

其它命令给出字节数大小：

```
hyw@ubuntu:~$ ls -l *File
-rw-rw-r-- 1 hyw hyw 370000 Jul 14 17:04 largeFile
-rw-rw-r-- 1 hyw hyw 1962 Jul 14 17:04 mediumFile
-rw-rw-r-- 1 hyw hyw 964 Jul 14 17:24 smallFile
hyw@ubuntu:~$ ls -l /tmp/newDataFile
-rw-rw-r-- 1 hyw hyw 964 Jul 14 17:26 /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。

```
hyw@ubuntu:~$ touch a.html
hyw@ubuntu:~$ touch b.htm
hyw@ubuntu:~$ touch c.c
hyw@ubuntu:~$ touch d.c
```

```
hyw@ubuntu:~$ find . -name "*.html"
./a.html
hyw@ubuntu:~$ find . -name "*.htm"
./b.htm
hyw@ubuntu:~$ find . -name "*.c"
./d.c
./c.c
```

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。

```
hyw@ubuntu:~$ find ~ -cnewer ~/smallFile
/home/hyw
/home/hyw/.cache/tracker/meta.db-wal
/home/hyw/.cache/tracker/meta.db-shm
/home/hyw/.cache/tracker/meta.db
/home/hyw/d.c
/home/hyw/c.c
/home/hyw/a.html
/home/hyw/.config/gedit/accels
/home/hyw/.local/share
/home/hyw/.local/share/xorg/Xorg.0.log
/home/hyw/.local/share/recently-used.xbel
/home/hyw/.local/share/gvfs-metadata
/home/hyw/.local/share/gvfs-metadata/root
/home/hyw/.local/share/gvfs-metadata/root-ebc797aa.log
/home/hyw/.local/share/tracker/data
/home/hyw/.local/share/tracker/data/tracker-store.journal
/home/hyw/.local/share/gnome-shell
/home/hyw/.local/share/gnome-shell/application_state
/home/hyw/smallFile2
/home/hyw/b.htm
```

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。

```
hyw@ubuntu:~$ touch aLinuxb
hyw@ubuntu:~$ touch aLinux
hyw@ubuntu:~$ touch Linuxb
hyw@ubuntu:~$ ls *Linux*
aLinux  aLinuxb  Linuxb
```

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。

文件	文件类型	存取权限	链接数	所有者	组	文件大小
/	目录	rwxr-xr-x	20	root	root	4096
/etc/passwd	普通文件	rw-r--r--	1	root	root	2732
/bin/df	普通文件	rwxr-xr-x	1	root	root	93128
~	目录	rwxr-xr-x	17	hyw	hyw	4096

```
hyw@ubuntu:~$ ls -d -l /
drwxr-xr-x 20 root root 4096 Jul  6 09:53 /
hyw@ubuntu:~$ ls -d -l /etc/passwd
-rw-r--r-- 1 root root 2732 Jul  6 10:17 /etc/passwd
hyw@ubuntu:~$ ls -d -l /bin/df
-rwxr-xr-x 1 root root 93128 Sep  5 2019 /bin/df
hyw@ubuntu:~$ ls -d -l ~
drwxr-xr-x 17 hyw hyw 4096 Jul 14 17:59 /home/hyw
```

解释：ls -d 参数可以表示显示目录本身，再加上-l 即可显示详细信息。第一个字段第一个字符为文件类型，后面为权限；第二个字段为链接数，第三个字段为所有者，第四个字段为组，第五个字段为文件大小。

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。

```
hyw@ubuntu:~$ ls -ld professional
drwxrwxrw- 4 hyw hyw 4096 Jul 14 15:57 professional
hyw@ubuntu:~$ chmod 755 professional
hyw@ubuntu:~$ ls -ld professional
drwxr-xr-x 4 hyw hyw 4096 Jul 14 15:57 professional
hyw@ubuntu:~$ ls -ld temp
drwxrwxr-x 2 hyw hyw 4096 Jul 14 16:30 temp
hyw@ubuntu:~$ chmod 755 temp
hyw@ubuntu:~$ ls -ld temp
drwxr-xr-x 2 hyw hyw 4096 Jul 14 16:30 temp
hyw@ubuntu:~$ ls -ld personal
drwxrwxr-x 4 hyw hyw 4096 Jul 14 15:59 personal
hyw@ubuntu:~$ chmod 755 personal
hyw@ubuntu:~$ ls -ld personal
drwxr-xr-x 4 hyw hyw 4096 Jul 14 15:59 personal
```

```
hyw@ubuntu:~$ cd temp
hyw@ubuntu:~/temp$ mkdir d1
hyw@ubuntu:~/temp$ mkdir d2
hyw@ubuntu:~/temp$ mkdir d3
hyw@ubuntu:~/temp$ cd d1
hyw@ubuntu:~/temp/d1$ touch f1
hyw@ubuntu:~/temp/d1$ cd ..
hyw@ubuntu:~/temp$ ls -l
total 12
drwxrwxr-x 2 hyw hyw 4096 Jul 14 18:14 d1
drwxrwxr-x 2 hyw hyw 4096 Jul 14 18:14 d2
drwxrwxr-x 2 hyw hyw 4096 Jul 14 18:14 d3
hyw@ubuntu:~/temp$ ls -l d1
total 0
-rw-rw-r-- 1 hyw hyw 0 Jul 14 18:14 f1
```

解释：chmod 后的 755 分别对应自己、组、其他用户的三个权限的二进制表示。

d1、d2、d3 和 f1 的访问权限分别为 rwxrwxr-x、rwxrwxr-x、rwxrwxr-x、rw-rw-r--，九个字符的前三个对应自己，中间三个对应组，后三个对应其他用户，r、w、x 分别表示读取权限、写权限、执行权限，-表示无改权限。

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。

```
hyw@ubuntu:~$ chmod 111 ./temp
hyw@ubuntu:~$ ls -ld temp
d--x--x--x 5 hyw hyw 4096 Jul 14 18:14 temp
hyw@ubuntu:~$ ls -l temp
ls: cannot open directory 'temp': Permission denied
```

最小权限为仅当前用户可读、可执行文件

```
hyw@ubuntu:~$ chmod 500 ./temp
hyw@ubuntu:~$ ls -l temp
total 12
drwxrwxr-x 2 hyw hyw 4096 Jul 14 18:14 d1
drwxrwxr-x 2 hyw hyw 4096 Jul 14 18:14 d2
drwxrwxr-x 2 hyw hyw 4096 Jul 14 18:14 d3
hyw@ubuntu:~$
```

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。

```
hyw@ubuntu:~$ umask
0002
hyw@ubuntu:~$ cd temp
hyw@ubuntu:~/temp$ cd d1
hyw@ubuntu:~/temp/d1$ mkdir d11
hyw@ubuntu:~/temp/d1$ touch f2
hyw@ubuntu:~/temp/d1$ gedit hello.c
hyw@ubuntu:~/temp/d1$ gcc -o greeting hello.c
hello.c:2:1: warning: return type defaults to 'int' [-Wimplicit-int]
  2 | main(void)
    | ^~~~~~
hyw@ubuntu:~/temp/d1$ ls -l
total 28
drwxrwxr-x 2 hyw hyw 4096 Jul 14 22:13 d11
-rw-rw-r-- 1 hyw hyw 0 Jul 14 18:14 f1
-rw-rw-r-- 1 hyw hyw 0 Jul 14 22:14 f2
-rwxrwxr-x 1 hyw hyw 16688 Jul 14 22:14 greeting
-rw-rw-r-- 1 hyw hyw 61 Jul 14 22:14 hello.c
hyw@ubuntu:~/temp/d1$ ls -ld .
drwxrwxr-x 3 hyw hyw 4096 Jul 14 22:14 .
```

（解释：最后一行是 `d1` 的权限，`.`表示当前文件夹）

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
0002	rw-rw-r--	rw-rw-r--	rwxrwxr-x	rwxrwxr-x
0077	rw-----	rw-----	rwX-----	rwX-----
0066	rw-----	rw-----	rwX--X--X	rwX--X--X

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。

```
hyw@ubuntu:~/temp$ umask 077
hyw@ubuntu:~/temp$ cd d2
hyw@ubuntu:~/temp/d2$ mkdir d21
hyw@ubuntu:~/temp/d2$ touch f2
hyw@ubuntu:~/temp/d2$ gedit hello.c
hyw@ubuntu:~/temp/d2$ gcc -o greeting hello.c
hello.c:2:1: warning: return type defaults to 'int' [-Wimplicit-int]
  2 | main(void)
    | ^~~~~~
hyw@ubuntu:~/temp/d2$ ls -l
total 28
drwx----- 2 hyw hyw 4096 Jul 14 22:26 d21
-rw----- 1 hyw hyw    0 Jul 14 22:26 f2
-rwx----- 1 hyw hyw 16688 Jul 14 22:27 greeting
-rw----- 1 hyw hyw    60 Jul 14 22:26 hello.c
```

```
hyw@ubuntu:~/temp$ umask 066
hyw@ubuntu:~/temp$ umask
0066
hyw@ubuntu:~/temp$ cd d3
hyw@ubuntu:~/temp/d3$ mkdir d31
hyw@ubuntu:~/temp/d3$ touch f2
hyw@ubuntu:~/temp/d3$ gedit hello.c
hyw@ubuntu:~/temp/d3$ gcc -o greeting hello.c
hello.c:2:1: warning: return type defaults to 'int' [-Wimplicit-int]
  2 | main(void)
    | ^~~~~~
hyw@ubuntu:~/temp/d3$ ls -l
total 28
drwx--x--x 2 hyw hyw 4096 Jul 14 22:28 d31
-rw----- 1 hyw hyw    0 Jul 14 22:28 f2
-rwx--x--x 1 hyw hyw 16688 Jul 14 22:29 greeting
-rw----- 1 hyw hyw    60 Jul 14 22:29 hello.c
```

解释：由于在 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，所以上表中 f2 和 hello.c 两个文本文件的访问权限就是 666-掩码，greeting 可执行文件和 d11/d21/d31 文件夹的访问权限就是 777-掩码。

26. 删除~/temp 目录下的所有文件和目录。给出会话过程。

```
hyw@ubuntu:~/temp$ rm -rf *
hyw@ubuntu:~/temp$ ls
```

(ls 后没有文件了，说明已删除)

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。

```
hyw@ubuntu:~$ cd temp
hyw@ubuntu:~/temp$ mkdir d1
hyw@ubuntu:~/temp$ mkdir d2
hyw@ubuntu:~/temp$ mkdir d3
hyw@ubuntu:~/temp$ cp ../smallFile ./d1/smallFile
hyw@ubuntu:~/temp$ ls -il ./d1/smallFile
1048896 -rw----- 1 hyw hyw 964 Jul 14 23:20 ./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。

```
hyw@ubuntu:~/temp$ cd d2
hyw@ubuntu:~/temp/d2$ ln ../d1/smallFile newFile.hard
hyw@ubuntu:~/temp/d2$ ls -il newFile.hard
1048896 -rw----- 2 hyw hyw 964 Jul 14 23:20 newFile.hard
```

确定 smallFile 和 smallFile.hard 是同一文件的两个名字的方式：根本上是两者的 inode number 相同。链接数增加 1、字节数等信息都相等只能作为验证。

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。

```
hyw@ubuntu:~/temp/d2$ mv newFile.hard smallFile.hard
hyw@ubuntu:~/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 Pohn ECE 4.00 pohn@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
```

```
hyw@ubuntu:~/temp/d2$ chmod 222 ../d1/smallFile
hyw@ubuntu:~/temp/d2$ cat smallFile.hard
cat: smallFile.hard: Permission denied
```

```
hyw@ubuntu:~/temp/d1$ ls -l smallFile
--w--w--w- 2 hyw hyw 964 Jul 14 23:28 smallFile
hyw@ubuntu:~/temp/d1$ ls -l ../d2/smallFile.hard
--w--w--w- 2 hyw hyw 964 Jul 14 23:28 ../d2/smallFile.hard
```

推断：smallFile.hard 和 smallFile 本质上是一个文件的两个名字，当一个文件读权限被取消时另一个也被取消，因此访问被拒绝。


```

hyw@ubuntu:~/temp/d2$ chmod 666 ../d1/smallFile
hyw@ubuntu:~/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	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

解释：增加读权限后，可以正常输出文件内容，说明硬链接与链接对象的权限是一致的，验证了之前的推断。

```

hyw@ubuntu:~/temp/d2$ cp ../d1/smallFile ../d1/smallFile2
hyw@ubuntu:~/temp/d2$ rm ../d1/smallFile
hyw@ubuntu:~/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	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

解释：删除 smallFile 后，该文件的链接数-1，smallFile.hard 作为硬链接仍然指向原来的 inode，inode 和文件数据都没有改变，只是硬链接减少了 1，因此依然能够输出文件内容。

- 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。

```

hyw@ubuntu:~/temp/d1$ mv smallFile2 smallFile
hyw@ubuntu:~/temp/d1$ ls
smallFile
hyw@ubuntu:~/temp/d1$ ln -s ~/temp/d1/smallFile ~/temp/d2/smallFile.soft
hyw@ubuntu:~/temp/d1$ ls -il smallFile
1048936 -rw----- 1 hyw hyw 964 Jul 14 23:28 smallFile
hyw@ubuntu:~/temp/d1$ ls -il ../d2/smallFile.soft
1048952 lrwxrwxrwx 1 hyw hyw 27 Jul 14 23:36 ../d2/smallFile.soft -> /home/hyw/temp/d1/smallFile

```

解释：确定 smallFile 和 smallFile.soft 是两个不同的文件根本上是因为两者的 inode number 不同，而不是文件大小。

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。

```

hyw@ubuntu:~/temp/d1$ cat ../d2/smallFile.soft

```

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

```

hyw@ubuntu:~/temp/d1$ chmod 222 smallFile
hyw@ubuntu:~/temp/d1$ cat ../d2/smallFile.soft
cat: ../d2/smallFile.soft: Permission denied
hyw@ubuntu:~/temp/d1$ ls -il ../d2/smallFile.soft
1048952 lrwxrwxrwx 1 hyw hyw 27 Jul 14 23:36 ../d2/smallFile.soft -> /home/hyw/temp/d1/smallFile
hyw@ubuntu:~/temp/d1$ ls -il smallFile
1048936 --w--w--w- 1 hyw hyw 964 Jul 14 23:28 smallFile

```

推断：smallFile 和 smallFile.soft 是两个文件，且前者是后者的文件内容，因为修改前者权限的时候后者的权限并没有改变。


```

hyw@ubuntu:~/temp/d1$ chmod 666 smallFile
hyw@ubuntu:~/temp/d1$ cat ../d2/smallFile.soft
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

hyw@ubuntu:~/temp/d1$ ls -l smallFile
-rw-rw-rw- 2 hyw hyw 964 Jul 14 23:28 smallFile
hyw@ubuntu:~/temp/d1$ ls -l ../d2/smallFile.soft
lrwxrwxrwx 1 hyw hyw 27 Jul 14 23:36 ../d2/smallFile.soft -> /home/hyw/temp/d1/smallFile

```

解释：增加 smallFile 的读权限后，访问软链接可以正常打印出文件内容，验证了之前的结论即软链接的文件内容是链接对象。

```

hyw@ubuntu:~/temp/d1$ cp smallFile smallFile2
hyw@ubuntu:~/temp/d1$ rm smallFile
hyw@ubuntu:~/temp/d1$ cat ../d2/smallFile.soft
cat: ../d2/smallFile.soft: No such file or directory

```

解释：删除了 smallFile 后软链接无法访问并提示找不到软链接，说明删除 smallFile 即删除了软链接的文件内容，因此软链接也被删除了，软链接和连接对象是不同的两个文件。

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。

（注：此处用 gedit 进程代替 init）

```

hyw@ubuntu:~$ ps -j
  PID  PGID  SID  TTY  TIME CMD
  51520 51520 51520 pts/0 00:00:02 bash
  62745 62745 51520 pts/0 00:00:00 gedit
  62756 62756 51520 pts/0 00:00:00 ps

```

共有 3 个进程，gedit、bash、ps 的 PID 分别为 51520、62745、62756，其父进程的 ID 分别为 51520、62745、62756。

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。

```
hyw@ubuntu:~$ (sleep 60s ; echo "Time for Lunch!") &
[2] 62821
hyw@ubuntu:~$ 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。

```
hyw@ubuntu:~$ date & uname -a & who & ps &
[2] 62829
[3] 62830
[4] 62831
[5] 62832
hyw@ubuntu:~$ Linux ubuntu 5.4.0-40-generic #44-Ubuntu SMP Tue Jun 23 00:01:04 UTC
2020 x86_64 x86_64 x86_64 GNU/Linux
Wed 15 Jul 2020 12:03:58 AM CST
hyw      :0                2020-07-06 10:31 (:0)
  PID TTY          TIME CMD
  51520 pts/0        00:00:02 bash
  62745 pts/0        00:00:00 gedit
  62832 pts/0        00:00:00 ps

[2] Done                date
[3] Done                uname -a
[4]- Done              who
[5]+ Done              ps
```

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。

```
hyw@ubuntu:~$ date && uname -a && who && ps
Wed 15 Jul 2020 12:07:13 AM CST
Linux ubuntu 5.4.0-40-generic #44-Ubuntu SMP Tue Jun 23 00:01:04 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
hyw      :0                2020-07-06 10:31 (:0)
  PID TTY          TIME CMD
  51520 pts/0        00:00:02 bash
  62745 pts/0        00:00:00 gedit
  62838 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

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

分别为/home/hyw（主目录）、/usr、/home/hyw（主目录）。

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）

```
hyw@ubuntu:~$ cd temp
hyw@ubuntu:~/temp$ cd d1
hyw@ubuntu:~/temp/d1$ touch foobar
hyw@ubuntu:~/temp/d1$ cd
hyw@ubuntu:~$ find . -name foobar 2>/dev/null
./temp/d1/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。

```
hyw@ubuntu:~$ cd temp
hyw@ubuntu:~/temp$ cd d1
hyw@ubuntu:~/temp/d1$ touch foobar
hyw@ubuntu:~/temp/d1$ cd
hyw@ubuntu:~$ find . -name foobar 1>foobar.path 2>/dev/null
hyw@ubuntu:~$ cat foobar.path
./temp/d1/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。

```
hyw@ubuntu:~$ touch student.records
hyw@ubuntu:~$ gedit student.records
hyw@ubuntu:~$ cat student.records
aaa
bbb
ccc
hyw@ubuntu:~$ cat < student.records 1>output.data 2>error.log
hyw@ubuntu:~$ cat output.data
aaa
bbb
ccc
```

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

```
hyw@ubuntu:~$ cat fdata  
echo -n "The time now is:"  
date  
echo -n "The users presently logged on are:"  
who  
  
hyw@ubuntu:~$ bash  
hyw@ubuntu:~$ exec <fdata 1>fout  
hyw@ubuntu:~$ echo -n "The time now is:"  
hyw@ubuntu:~$ date  
hyw@ubuntu:~$ echo -n "The users presently logged on are:"  
hyw@ubuntu:~$ who  
hyw@ubuntu:~$  
hyw@ubuntu:~$ exit  
hyw@ubuntu:~$ cat fout  
The time now is:Wed 15 Jul 2020 12:25:51 AM CST  
The users presently logged on are:hyw      :0                2020-07-06 10:31 (:0)
```

命令执行后，生成了一个新文件 fout，内容为：

The time now is:Wed 15 Jul 2020 12:25:51 AM CST

The users presently logged on are:hyw :0 2020-07-06 10:31 (: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。

```
hyw@ubuntu:~$ ls -l | wc  
   12    101   566
```

行数为 12，字符数为 101，单词数为 566。

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？

```
hyw@ubuntu:~$ ls -l /usr/bin | grep "^-" | wc -l  
1197  
hyw@ubuntu:~$ ls -l /usr/bin | grep "^d" | wc -l  
0  
hyw@ubuntu:~$ ls -l /usr/bin | grep "^l" | wc -l  
252
```

（主要文件都储存在 `/usr/bin` 中，所以选择 `/usr/bin` 目录，`/bin` 目录下只有一个符号链接文件指向 `/usr/bin`）

共有 1197 个普通文件，0 个目录文件，252 个符号链接文件。

得到答案的方式是使用 Linux 管道，先 `ls -l` 列出文件详细信息，根据第一个字段第一个字符可以区分普通类型的文件，用 `grep` 按第一个字符区分后用 `wc -l` 统计行数即可。如统计普通文件的命令是 `ls -l /usr/bin | grep "^-" | wc -l`。

三、 讨论、心得（必填）

这次实验过程主要依靠查找资料与自己摸索完成，在实现的过程中也遇到了一些错误，有的错误非常危险，稍有不慎可能造成严重后果，积累了非常宝贵的经验教训。比如在做第 26 题的 `rm` 指令操作时，最开始我在主目录下用 `rm -rf temp *` 删除 `temp` 这个文件夹时删不掉，显示有一些文件没有删除权限，后来我在前面加上了 `sudo`，结果把整个主目录删除了（好在最后恢复了，没有造成很严重的后果）。这里至少犯了三个错误：第一个是对 `rm` 指令的危险性没有认识，第二是没有进到 `temp` 目录下去删，而是在主目录下删除，并且 `temp` 没有写成 `./temp`，第三是 `rm` 的时候没有加上参数 `-i`，直接删除无疑增加了危险性。这些都是在以后的操作中应该避免的。最后的解决方法是进入 `temp` 目录后再直接用 `rm -rf` 删除，就没有问题了。

在实验过程中还遇到一些细小的错误，比如 `/include` 文件下没有文件，是因为没有预装 `gcc` 等包造成的，用 `apt-get` 装上了就好了；再比如一些指令经常运行不正常，通常是复制粘贴的脚本中存在特殊字符造成的，重新打一遍通常就能好。这类问题通过自己摸索和查阅资料都能较为轻松地解决。

这次实验给我最大的体会是，做 Linux 实验需要非常细心，因为 Linux 对语法的要求十分严格；同时又需要有足够的耐心，遇到的问题大多数都可以通过查阅互联网资料、PPT、Linux 自带 `manual` 或 `help` 解决。如第 20 题中用到的参数并不是日常中经常使用的参数，这些在 ppt 上都有详细地列出，最开始在网上搜索了很长时间都没有搜到。另外是要拓宽思路，同一功能可以有多种实现方式，根据实际的需求灵活采取不同的指令来实现这一能力是今后需要非常重视的，这次实验中的几个题如多种方式输出主目录绝对路径就很好地锻炼了这一点。最后就是要多操作总结经验，比如前面提到的 `rm`，再比如善用 `man`、`info` 等指令和 `--help` 参数有的时候会起到非常好的作用。