

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： 黄炯睿 专业： 信息安全 学号： 3170103455

电子邮件地址： 3170103455@zju.edu.cn

实验日期： 2019 年 5 月 14 日

一、 实验环境

计算机配置： 处理器 intel® Core™ i7-7700HQ CPU @ 2.8GHz(8 CPUs)

内 存 8G

显 卡 Nvidia GeForce GTX 1050 (6 GB)

操作系统环境： Windows 10 家庭中文版 64 位 版本： 17134.706

Linux 版本： ubuntu-17.04

二、 实验内容： (90 分)

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？

所谓软件许可证，是指“软件出版商与用户签订的，旨在指导和规范软件如何使用的合同”。软件的购买者只有使用软件的许可，软件的全部权利仍然由软件的出版商保留。“从许可证的性质和使用范围来讲，开源软件许可证和传统的商业软件许可证没有任何不同”，两者都受法律保护。但根据使用者对于软件的权利范围大小，可将软件许可证分为商业软件许可证和开源软件许可证。

商业软件许可证，对于用户使用的权利进行了较大范围的限制；未经许可时，用户不具有可以修改软件代码、再次发布等权利。商业许可证一般都是有偿获得的，具有瑕疵担保责任。

开源软件许可证，软件作者有公开源代码之义务，而用户根据不同许可证具有相应的修改软件等权利。常见的开源软件许可证有：**BSD(Berkeley Software Distribution)**许可证、**MIT(Massachusetts Institute Technology)**许可证和**GPL(General Public License)**许可证。开源许可证基本确定了用户可以复制和修改软件源代码的权利，在发布要求或再许可使用方面进行了区分。

在我国司法实务中，相较商业软件许可证，开源软件许可证涉及的争议数量上更为庞大。源代码属于知识产权的管辖范畴，而开源软件许可证同样属于格式合同，根据我国《合同法》之规定，软件许可证是基于软件作者和用户一致的意思表示、在双方平等自愿的基础上产生的民事法律行为。在实践中表现为，用户修改并发布了相关程序之行为是对于许可证同意之承诺的意思表示。一旦用户行使权力、实施了修改相关程序的行为，就要遵守软件许可证对于发布或其他的规定；反之，用户应当承担违约责任。但是在具体实施的过程中，因为在网络上达成协议本身存在不确定性，存在举证责任的负担问题。另一方面，虽然开源软件许可证规定了用户拥有修改和发布的权利，仍然会存在许可证协议的不相容性。多个许可证共同适用时，由于软件许可证是格式合同、具备要求用户对于其内容完全同意或不同意的特征，往往会形成纠纷；在向法院提请诉讼过程中，法院要判断具体某一部分属于某一许可证的管辖范围其难度相对较大。

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

```
huangjionggrui@huangjionggrui-virtual-machine:~$ ls
core 公共的 模板 视频 图片 文档 下载 音乐 桌面
```

b. \$ pwd

```
huangjionggrui@huangjionggrui-virtual-machine:~$ pwd
/home/huangjionggrui
```

c. \$ xy

#只看结果，不要问为什么没有 xy 命令

```
huangjionggrui@huangjionggrui-virtual-machine:~$ xy
xy: 未找到命令
```

d. \$ cd ..

```
huangjionggrui@huangjionggrui-virtual-machine:~$ cd ..
huangjionggrui@huangjionggrui-virtual-machine:/home$
```

e. \$ pwd

```
huangjionggrui@huangjionggrui-virtual-machine:/home$ pwd
/home
```

f. \$ cd

g. \$ pwd

```
huangjionggrui@huangjionggrui-virtual-machine:/home$ cd
huangjionggrui@huangjionggrui-virtual-machine:~$ pwd
/home/huangjionggrui
```

h. \$ cd /usr/include

```
huangjionggrui@huangjionggrui-virtual-machine:~$ cd /usr/include
huangjionggrui@huangjionggrui-virtual-machine:/usr/include$
```

i. \$ ls

```

huangjionggrui@huangjionggrui-virtual-machine:/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        tgmth.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        netlucv        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       paths.h        stdint.h        wait.h
err.h          link.h         poll.h         stdio_ext.h     wchar.h
errno.h        linux          printf.h       stdio.h         wctype.h
error.h        locale.h       protocols      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       python2.7     sudo_plugin.h   xlocale.h
fmtmsg.h       misc          python3.5m    syscall.h       xorg
huangjionggrui@huangjionggrui-virtual-machine:/usr/include$

```

j. \$ cd

```

huangjionggrui@huangjionggrui-virtual-machine:/usr/include$ cd
huangjionggrui@huangjionggrui-virtual-machine:~$

```

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）

Man ls:

```

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 ..

```

Man passwd

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	<p>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。</p> <p>密码更改</p> <p>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。</p> <p>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。</p> <p>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。</p>	

Info pwd

```

    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 neither option is given, then this implementation uses '-P' as the default unless the 'POSIXLY_CORRECT' environment variable is set.

```

Pwd --help

```

huangjionggrui@huangjionggrui-virtual-machine:~$ 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.

Exit Status:
Returns 0 unless an invalid option is given or the current directory
cannot be read.

```

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？ **huangjionggrui**
 - 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？ **huangjionggrui-virtual-machine**
 - 4) 你计算机系统的 CPU 名字是什么？ **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 **`cal 7 1752`** 和 **`cal 7 2012`**
 - 2) 1752 年有几天，为什么？ **355 天**

```

huangjionggrui@huangjionggrui-virtual-machine:~$ cal 12 1752 -j
      十二月 1752
 日 一 二 三 四 五 六
                325 326
327 328 329 330 331 332 333
334 335 336 337 338 339 340
341 342 343 344 345 346 347
348 349 350 351 352 353 354
355

```

虽然不知道为什么，但那年的九月好短？？

```

      九月
 日 一 二 三 四 五 六
                245 246 247 248 249
250 251 252 253 254 255 256
257 258 259 260 261 262 263

```

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。

已启动了 6 小时 02 分，只有一个用户使用

```

huangjionggrui@huangjionggrui-virtual-machine:~$ uptime
15:09:38 up 6:02, 1 user, load average: 0.00, 0.00, 0.00

```

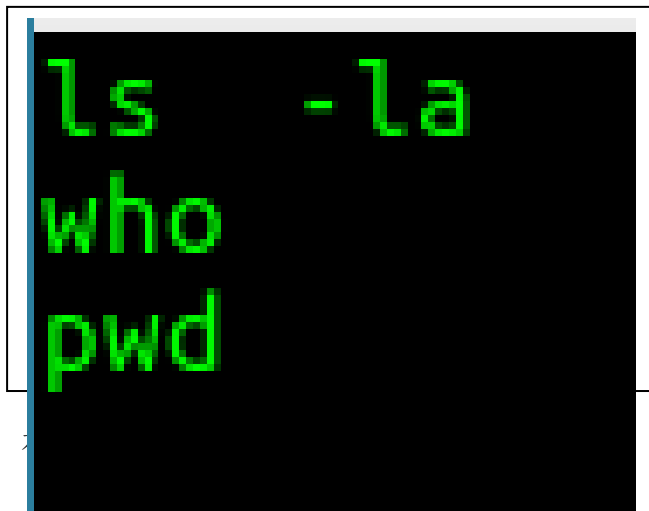
7. 通过 Linux 的 `man`、`info` 命令或互联网得到下面的 `shell` 命令、系统调用和库函数功能描述及每个命令使用例子：

命令	命令功能的简要描述	实例
<code>touch</code>	用来修改文件时间戳，或者新建一个不存在的文件。	<ul style="list-style-type: none"> ● <code>touch test.txt</code> ● <code>touch -t 201905131534.50 test.txt</code>
<code>cp</code>	<code>cp</code> 命令用来复制文件或者目录	<code>Cp log.log test5</code>
<code>mv</code>	<code>mv</code> 命令是 <code>move</code> 的缩写，可以用来移动文件或者将文件改名	<code>mv test.log test1.txt</code> <code>mv test1.txt test3</code>
<code>rm</code>	<code>Rm</code> 命令可以删除一个目录中的一个或多个文	<code>rm -i test.txt</code>

	件或目录，也可以将某个目录及其下的所有文件及子目录均删除。	<code>rm -f test.txt</code>
<code>mkdir</code>	该命令用来创建指定的名称的目录，要求创建目录的用户在当前目录中具有写权限，并且指定的目录名不能是当前目录中已有的目录。	<code>mkdir test1</code>
<code>who</code>	<code>who</code> 命令用于显示系统中有哪些使用者正在上面，显示的资料包含了使用者 ID、使用的终端机、从哪边连上来的、上线时间、呆滞时间、CPU 使用量、动作等等。	<code>who -H</code>
<code>ls</code>	通过 <code>ls</code> 命令查看 linux 文件夹包含的文件,查看文件权限(包括目录、文件夹、文件权限) 查看目录信息等等	<code>ls -l t*</code>
<code>cd</code>	<code>cd</code> 命令用来切换工作目录至 <code>dirname</code>	<code>cd ..</code>
<code>pwd</code>	<code>pwd</code> 命令来查看”当前工作目录“的完整路径。	<code>pwd</code>
<code>open</code>	<code>open()</code> 方法用于打开和创建文件	
<code>read</code>	由已打开的文件读取数据	
<code>write</code>	将数据写入已打开的文件内	
<code>close</code>	关闭文件	
<code>pipe</code>	建立管道	
<code>socket</code>	建立一个 <code>socket</code> 通信	
<code>mkfifo</code>	建立具名管道	
<code>system</code>	执行 <code>shell</code> 命令	
<code>printf</code>	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) 按

- 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?

当前有 41 个文件

```

总用量 2492
drwxr-xr-x 21 huangjiongrui huangjiongrui 4096 5月 13 23:33 .
drwxr-xr-x 3 root root 4096 12月 23 2017 ..
-rw-r--r-- 1 huangjiongrui huangjiongrui 5517 5月 13 23:29 .bash_history
-rw-r--r-- 1 huangjiongrui huangjiongrui 220 12月 23 2017 .bash_logout
-rw-r--r-- 1 huangjiongrui huangjiongrui 3771 12月 23 2017 .bashrc
drwxrwxr-x 19 huangjiongrui huangjiongrui 4096 4月 10 20:50 .cache
drwx----- 23 huangjiongrui huangjiongrui 4096 4月 17 19:47 .config
-rw-r--r-- 1 huangjiongrui huangjiongrui 2334720 4月 1 15:11 core
drwx----- 3 huangjiongrui huangjiongrui 4096 12月 23 2017 .dbus
-rw-r--r-- 1 huangjiongrui huangjiongrui 23 12月 23 2017 .dmrc
-rw-r--r-- 1 huangjiongrui huangjiongrui 582 12月 23 2017 .face
-rw-r--r-- 1 huangjiongrui huangjiongrui 17 5月 13 23:33 firscrip
-rw-r--r-- 1 huangjiongrui huangjiongrui 12288 5月 13 23:29 .firscrip.swp
drwx----- 2 huangjiongrui huangjiongrui 4096 3月 20 15:01 .gconf
drwx----- 4 huangjiongrui huangjiongrui 4096 3月 27 19:59 .gnupg
drwxr-xr-x 2 huangjiongrui huangjiongrui 4096 3月 13 19:27 .gphoto
-rw-r--r-- 1 huangjiongrui huangjiongrui 10660 5月 13 17:49 .ICEauthority
drwx----- 3 huangjiongrui huangjiongrui 4096 12月 23 2017 .local
drwx----- 5 huangjiongrui huangjiongrui 4096 4月 10 20:50 .mozilla
drwx----- 3 huangjiongrui huangjiongrui 4096 3月 1 19:34 .pki
drwx----- 2 huangjiongrui huangjiongrui 4096 12月 23 2017 .presage
-rw-r--r-- 1 huangjiongrui huangjiongrui 675 12月 23 2017 .profile
-rw-r--r-- 1 huangjiongrui huangjiongrui 241 4月 17 20:19 .recentAppLog
-rw-r--r-- 1 huangjiongrui huangjiongrui 0 3月 13 19:02 .sudo_as_admin_successful
-rw-r--r-- 1 huangjiongrui huangjiongrui 12288 2月 27 20:04 .swp
-rw-r--r-- 1 huangjiongrui huangjiongrui 12288 3月 18 22:22 .test.swo
-rw-r--r-- 1 huangjiongrui huangjiongrui 12288 2月 27 20:48 .test.swp
-rw-r--r-- 1 huangjiongrui huangjiongrui 0 5月 13 17:49 test.txt
drwx----- 5 huangjiongrui huangjiongrui 4096 4月 10 20:50 .thunderbird
-rw-r--r-- 1 huangjiongrui huangjiongrui 3 4月 10 20:54 .windowHeight
-rw-r--r-- 1 huangjiongrui huangjiongrui 74 5月 13 17:48 .Xauthority
-rw-r--r-- 1 huangjiongrui huangjiongrui 8372 5月 13 23:13 .xsession-errors
-rw-r--r-- 1 huangjiongrui huangjiongrui 11424 5月 13 16:34 .xsession-errors.old
drwxr-xr-x 2 huangjiongrui huangjiongrui 4096 12月 23 2017 公共的
drwxr-xr-x 2 huangjiongrui huangjiongrui 4096 12月 23 2017 模板
drwxr-xr-x 2 huangjiongrui huangjiongrui 4096 12月 23 2017 视频
drwxr-xr-x 2 huangjiongrui huangjiongrui 4096 12月 23 2017 图片
drwxr-xr-x 3 huangjiongrui huangjiongrui 4096 3月 27 20:18 文档
drwxr-xr-x 2 huangjiongrui huangjiongrui 4096 3月 13 19:25 下载
drwxr-xr-x 2 huangjiongrui huangjiongrui 4096 12月 23 2017 音乐
drwxr-xr-x 3 huangjiongrui huangjiongrui 4096 3月 20 15:29 桌面
huangjiongrui tty7 2019-05-13 17:48 (:0)
/home/huangjiongrui

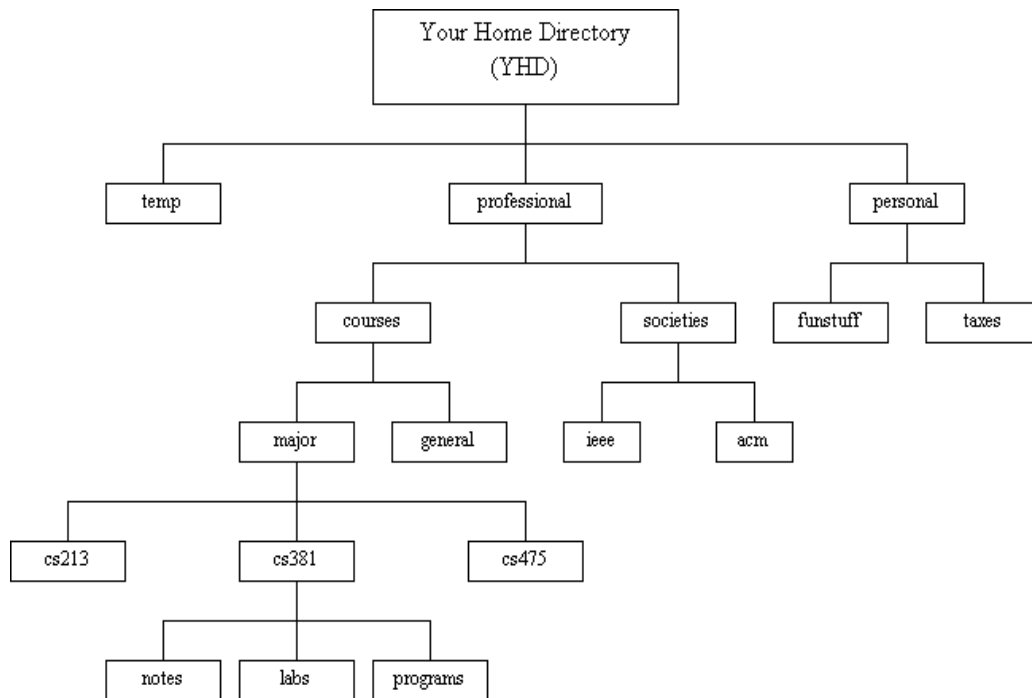
```

文件列表的最后一列是它们的名称, 第 5 列是文件的大小, 它的数字的单位为 (byte)

用户 **huangjiongrui** 在使用系统

当前工作目录为 **/home/huangjiongrui**

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)



```
huangjionggrui@huangjionggrui-virtual-machine:~$ mkdir temp
huangjionggrui@huangjionggrui-virtual-machine:~$ mkdir professional
huangjionggrui@huangjionggrui-virtual-machine:~$ mkdir personal
huangjionggrui@huangjionggrui-virtual-machine:~$ mkdir personal/funstuff
huangjionggrui@huangjionggrui-virtual-machine:~$ mkdir personal/taxes
huangjionggrui@huangjionggrui-virtual-machine:~$ cd professional/
huangjionggrui@huangjionggrui-virtual-machine:~/professional$ mkdir courses
huangjionggrui@huangjionggrui-virtual-machine:~/professional$ mkdir societies
huangjionggrui@huangjionggrui-virtual-machine:~/professional$ mkdir societies/ieee
huangjionggrui@huangjionggrui-virtual-machine:~/professional$ mkdir societies/acm
huangjionggrui@huangjionggrui-virtual-machine:~/professional$ cd courses/
huangjionggrui@huangjionggrui-virtual-machine:~/professional/courses$ mkdir major
huangjionggrui@huangjionggrui-virtual-machine:~/professional/courses$ mkdir general
huangjionggrui@huangjionggrui-virtual-machine:~/professional/courses$ cd major/
huangjionggrui@huangjionggrui-virtual-machine:~/professional/courses/major$ mkdir cs213
huangjionggrui@huangjionggrui-virtual-machine:~/professional/courses/major$ mkdir cs381
huangjionggrui@huangjionggrui-virtual-machine:~/professional/courses/major$ mkdir cs475
huangjionggrui@huangjionggrui-virtual-machine:~/professional/courses/major$ mkdir cs381/notes
huangjionggrui@huangjionggrui-virtual-machine:~/professional/courses/major$ mkdir cs381/labs
huangjionggrui@huangjionggrui-virtual-machine:~/professional/courses/major$ mkdir cs381/programs
```



```
huangjionggrui@huangjionggrui-virtual-machine:~$ 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
```

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。

```
huangjionggrui@huangjionggrui-virtual-machine:~/professional/courses/major/cs381/labs$ echo ~
/home/huangjionggrui
```

- 2) acm 目录的绝对路径是什么？

/home/huangjionggrui/professional/societies/acm

- 3) 给出 acm 目录的两个相对路径。

在 **societies** 目录下时，相对路径为 **./acm**

在 **professional** 目录下时，相对路径为 **./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。

```
huangjionggrui@huangjionggrui-virtual-machine:~/professional/courses$ cd major/cs381/labs/
huangjionggrui@huangjionggrui-virtual-machine:~/professional/courses/major/cs381/labs$ pwd
/home/huangjionggrui/professional/courses/major/cs381/labs
```

- 5) 给出获得你的主目录三个不同的命令。

```
huangjionggrui@huangjionggrui-virtual-machine:~/professional/courses/major/cs381/labs$ echo $HOME
/home/huangjionggrui
huangjionggrui@huangjionggrui-virtual-machine:~/professional/courses/major/cs381/labs$ echo ~
/home/huangjionggrui
huangjionggrui@huangjionggrui-virtual-machine:~/professional/courses/major/cs381/labs$ cd ~
huangjionggrui@huangjionggrui-virtual-machine:~$ pwd
/home/huangjionggrui
```

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？

```

huangjionggrui@huangjionggrui-virtual-machine:~$ ls -d .*
.          .dmrc          .mozilla    .test.swp
..         .face          .pki        .thunderbird
.bash_history .firscrip.swp .presage    .windowHeight
.bash_logout .gconf         .profile    .Xauthority
.bashrc      .gnupg         .recentAppLog .xsession-errors
.cache       .gphoto        .sudo_as_admin_successful .xsession-errors.old
.config      .ICEauthority  .swp
.dbus        .local         .test.swo

```

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。

```

huangjionggrui@huangjionggrui-virtual-machine:~$ stat /
 文件：/
 大小：4096          块：8          IO 块：4096   目录
设备：801h/2049d     Inode：2        硬链接：24
权限：(0755/drwxr-xr-x)  Uid：(    0/    root)  Gid：(    0/    root)
最近访问：2019-05-13 17:48:01.831622526 +0800
最近更改：2019-03-20 09:45:41.792528602 +0800
最近改动：2019-03-20 09:45:41.792528602 +0800
创建时间：-

```

```

huangjionggrui@huangjionggrui-virtual-machine:~$ stat ~
 文件：/home/huangjionggrui
 大小：4096          块：8          IO 块：4096   目录
设备：801h/2049d     Inode：677187   硬链接：24
权限：(0755/drwxr-xr-x)  Uid：( 1000/huangjionggrui)  Gid：( 1000/huangjionggrui)
最近访问：2019-05-14 13:12:28.795028065 +0800
最近更改：2019-05-14 13:10:58.335029400 +0800
最近改动：2019-05-14 13:10:58.335029400 +0800
创建时间：-

```

```

huangjionggrui@huangjionggrui-virtual-machine:~$ stat ~/temp
 文件：/home/huangjionggrui/temp
 大小：4096          块：8          IO 块：4096   目录
设备：801h/2049d     Inode：660085   硬链接：2
权限：(0755/drwxr-xr-x)  Uid：( 1000/huangjionggrui)  Gid：( 1000/huangjionggrui)
最近访问：2019-05-14 13:24:55.895017043 +0800
最近更改：2019-05-14 13:10:36.163029727 +0800
最近改动：2019-05-14 13:10:36.163029727 +0800
创建时间：-

```

```

huangjionggrui@huangjionggrui-virtual-machine:~$ stat ~/professional/
 文件：/home/huangjionggrui/professional/
 大小：4096          块：8          IO 块：4096   目录
设备：801h/2049d     Inode：678725   硬链接：4
权限：(0755/drwxr-xr-x)  Uid：( 1000/huangjionggrui)  Gid：( 1000/huangjionggrui)
最近访问：2019-05-14 13:13:10.707027447 +0800
最近更改：2019-05-14 13:13:01.275027586 +0800
最近改动：2019-05-14 13:13:01.275027586 +0800
创建时间：-
huangjionggrui@huangjionggrui-virtual-machine:~$ stat ~/personal/
 文件：/home/huangjionggrui/personal/
 大小：4096          块：8          IO 块：4096   目录
设备：801h/2049d     Inode：678726   硬链接：4
权限：(0755/drwxr-xr-x)  Uid：( 1000/huangjionggrui)  Gid：( 1000/huangjionggrui)
最近访问：2019-05-14 13:24:55.855017043 +0800
最近更改：2019-05-14 13:11:54.647028569 +0800
最近改动：2019-05-14 13:11:54.647028569 +0800
创建时间：-

```

/: 2 ~:677187

~/temp:660085 ~/professional:678725 ~/personal:678726

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。

```
huangjionggrui@huangjionggrui-virtual-machine:~/professional/courses/major/cs381/labs$ ls -dl lab1
-rw-r--r-- 1 huangjionggrui huangjionggrui 151 5月 14 13:52 lab1
```

输出的第一个字符是 ‘-’，代表普通文件类型

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。

```
huangjionggrui@huangjionggrui-virtual-machine:~/professional/courses/major/cs381/labs$ file lab1
lab1: ASCII text
```

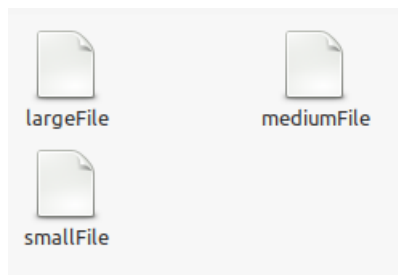
ASCII text 类型

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。

```
huangjionggrui@huangjionggrui-virtual-machine:~$ cd /usr/include
huangjionggrui@huangjionggrui-virtual-machine:/usr/include$ ls t*.h
tar.h  termio.h  termios.h  tgmath.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



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，要用什么命令？使用什么命令显示 smallFile 文件从第 6 行开始到结束全部行？给出你的会话。

Head -12 largeFile

```

huangjionggrui@huangjionggrui-virtual-machine:~/文档$ 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

```

huangjionggrui@huangjionggrui-virtual-machine:~/文档$ 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

```

huangjionggrui@huangjionggrui-virtual-machine:~/文档$ 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。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？

```
huangjionggrui@huangjionggrui-virtual-machine:~/文档$ cp smallFile dataFile
huangjionggrui@huangjionggrui-virtual-machine:~/文档$ ls -l
总用量 368
-rw-r--r-- 1 huangjionggrui huangjionggrui 986 5月 14 19:12 dataFile
drwxr-xr-x 17 huangjionggrui huangjionggrui 4096 3月 19 18:40 gnupg-2.2.14
-rw-r--r-- 1 huangjionggrui huangjionggrui 355236 5月 14 14:07 largeFile
-rw-r--r-- 1 huangjionggrui huangjionggrui 1958 5月 14 14:06 mediumFile
-rw-r--r-- 1 huangjionggrui huangjionggrui 1684 3月 27 18:55 mypublickey.asc
-rw-r--r-- 1 huangjionggrui huangjionggrui 986 5月 14 14:11 smallFile
```

它们是不同的，dataFile 文件的修改时间是这个文件的创建时间。

cp -p smallFile datafile 命令可以复制文件但是新的文件创建时间与原文件相同。

```
huangjionggrui@huangjionggrui-virtual-machine:~/文档$ stat smallFile
 文件: smallFile
 大小: 986          块: 8          IO 块: 4096   普通文件
设备: 801h/2049d    Inode: 678818      硬链接: 1
权限: (0644/-rw-r--r--)  Uid: ( 1000/huangjionggrui)   Gid: ( 1000/huangjionggrui)
最近访问: 2019-05-14 14:12:20.446975076 +0800
最近更改: 2019-05-14 14:11:56.678975427 +0800
最近改动: 2019-05-14 14:11:56.678975427 +0800
创建时间: -
```

```
huangjionggrui@huangjionggrui-virtual-machine:~/文档$ stat dataFile
 文件: dataFile
 大小: 986          块: 8          IO 块: 4096   普通文件
设备: 801h/2049d    Inode: 677376      硬链接: 1
权限: (0644/-rw-r--r--)  Uid: ( 1000/huangjionggrui)   Gid: ( 1000/huangjionggrui)
最近访问: 2019-05-14 14:12:20.446975076 +0800
最近更改: 2019-05-14 14:11:56.678975427 +0800
最近改动: 2019-05-14 19:16:55.514012860 +0800
创建时间: -
```

smallFile 的 iNode 值为 678818 dataFile 的是 677376

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？

```
huangjionggrui@huangjionggrui-virtual-machine:~/文档$ stat newDataFile
 文件: newDataFile
 大小: 986          块: 8          IO 块: 4096   普通文件
设备: 801h/2049d    Inode: 677376      硬链接: 1
权限: (0644/-rw-r--r--)  Uid: ( 1000/huangjionggrui)   Gid: ( 1000/huangjionggrui)
最近访问: 2019-05-14 14:12:20.446975076 +0800
最近更改: 2019-05-14 14:11:56.678975427 +0800
最近改动: 2019-05-14 19:22:56.098011484 +0800
创建时间: -
```

newDataFile 的 inode 值等于之前的 dataFile 为 677376。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。

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？

```
huangjionggrui@huangjionggrui-virtual-machine:/tmp$ stat newDataFile
  文件: newDataFile
  大小: 986          块: 8          IO 块: 4096   普通文件
设备: 801h/2049d    Inode: 677376   硬链接: 1
权限: (0644/-rw-r--r--)  Uid: ( 1000/huangjionggrui)  Gid: ( 1000/huangjionggrui)
最近访问: 2019-05-14 14:12:20.446975076 +0800
最近更改: 2019-05-14 14:11:56.678975427 +0800
最近改动: 2019-05-14 19:25:42.290010850 +0800
创建时间: -
```

Inode 号依然是 677376。因为移动文件不影响 inode 指向的存放文件地址。

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。

```
huangjionggrui@huangjionggrui-virtual-machine:~/文档$ wc smallFile mediumFile largeFile /tmp/newDataFile
  20   120   986 smallFile
  71   215  1958 mediumFile
5951 46427 355236 largeFile
  20   120   986 /tmp/newDataFile
6062 46882 359166 总用量
```

上图中从左到右的数字依次为行数，字数和字节数

```
huangjionggrui@huangjionggrui-virtual-machine:~/文档$ ls -l smallFile mediumFile largeFile /tmp/newDataFile
-rw-r--r-- 1 huangjionggrui huangjionggrui 355236 5月 14 14:07 largeFile
-rw-r--r-- 1 huangjionggrui huangjionggrui  1958 5月 14 14:06 mediumFile
-rw-r--r-- 1 huangjionggrui huangjionggrui   986 5月 14 14:11 smallFile
-rw-r--r-- 1 huangjionggrui huangjionggrui   986 5月 14 14:11 /tmp/newDataFile
```

上图中从左往右数第五列是该文件的字节数

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"

```
huangjionggrui@huangjionggrui-virtual-machine:~$ find ~ -name "*.c" -o -name "*.html" -o -name "*.htm"
/home/huangjionggrui/文档/gnupg-2.2.14/g10/photoid.c
/home/huangjionggrui/文档/gnupg-2.2.14/g10/encrypt.c
/home/huangjionggrui/文档/gnupg-2.2.14/g10/keygen.c
/home/huangjionggrui/文档/gnupg-2.2.14/g10/mdfilter.c
/home/huangjionggrui/文档/gnupg-2.2.14/g10/keyring.c
/home/huangjionggrui/文档/gnupg-2.2.14/g10/trust.c
/home/huangjionggrui/文档/gnupg-2.2.14/g10/test-stubs.c
/home/huangjionggrui/文档/gnupg-2.2.14/g10/compress-bz2.c
/home/huangjionggrui/文档/gnupg-2.2.14/g10/rmd160.c
/home/huangjionggrui/文档/gnupg-2.2.14/g10/misc.c
/home/huangjionggrui/文档/gnupg-2.2.14/g10/key-check.c
/home/huangjionggrui/文档/gnupg-2.2.14/g10/ecdh.c
/home/huangjionggrui/文档/gnupg-2.2.14/g10/import.c
/home/huangjionggrui/文档/gnupg-2.2.14/g10/openfile.c
```



```

/home/huangjionggrui/文档/gnupg-2.2.14/tools/gpgsplit.c
/home/huangjionggrui/文档/gnupg-2.2.14/tools/gpgtar-create.c
/home/huangjionggrui/文档/gnupg-2.2.14/tools/call-dirmngr.c
/home/huangjionggrui/文档/gnupg-2.2.14/tools/gpgconf-comp.c
/home/huangjionggrui/文档/gnupg-2.2.14/tools/send-mail.c
/home/huangjionggrui/文档/gnupg-2.2.14/tools/gpgparsemail.c
/home/huangjionggrui/文档/gnupg-2.2.14/tools/gpg-connect-agent.c
/home/huangjionggrui/文档/gnupg-2.2.14/tools/clean-sat.c
/home/huangjionggrui/文档/gnupg-2.2.14/tools/ccidmon.c
/home/huangjionggrui/文档/gnupg-2.2.14/tools/gpgconf.c
/home/huangjionggrui/文档/gnupg-2.2.14/tools/mime-maker.c
/home/huangjionggrui/文档/gnupg-2.2.14/tools/gpgtar-list.c
/home/huangjionggrui/文档/gnupg-2.2.14/tools/no-libgcrypt.c
/home/huangjionggrui/文档/gnupg-2.2.14/tools/sockprox.c
/home/huangjionggrui/文档/gnupg-2.2.14/tools/gpg-check-pattern.c
/home/huangjionggrui/文档/gnupg-2.2.14/tools/symcryptrun.c
/home/huangjionggrui/文档/gnupg-2.2.14/tools/watchgnupg.c
/home/huangjionggrui/文档/gnupg-2.2.14/tools/mime-parser.c
/home/huangjionggrui/文档/gnupg-2.2.14/tools/gpgtar.c
/home/huangjionggrui/文档/gnupg-2.2.14/tools/make-dns-cert.c
find: '/home/huangjionggrui/.cache/dconf': 权限不够
/home/huangjionggrui/a.htm
/home/huangjionggrui/a.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。

因为在 linux 虚拟机中下了很多软件，搜索的东西太多了，仅展示部分。

find ~ -cnewer 文档/smallFile

```

huangjionggrui@huangjionggrui-virtual-machine:~$ find ~ -cnewer 文档/smallFile
/home/huangjionggrui
/home/huangjionggrui/.bash_history
/home/huangjionggrui/.dbus/session-bus/4e5dd0c32c07459096789d4964324a46-0
/home/huangjionggrui/文档
/home/huangjionggrui/.recentAppLog
/home/huangjionggrui/.xsession-errors.old
/home/huangjionggrui/.xsession-errors
/home/huangjionggrui/.config/pulse/4e5dd0c32c07459096789d4964324a46-default-source
/home/huangjionggrui/.config/pulse/4e5dd0c32c07459096789d4964324a46-default-sink
/home/huangjionggrui/.config/ubuntu-kylin/indicator-china-weather/pingback.time
/home/huangjionggrui/.config/libreoffice/4
/home/huangjionggrui/.config/libreoffice/4/user
/home/huangjionggrui/.config/libreoffice/4/user/registrymodifications.xcu
/home/huangjionggrui/.config/libreoffice/4/user/uno_packages/cache/log.txt
/home/huangjionggrui/.config/libreoffice/4/user/backup
/home/huangjionggrui/.config/ukui-menu
/home/huangjionggrui/.config/ukui-menu/-usr-share-unity-icons-logout_highlight.png
/home/huangjionggrui/.config/ukui-menu/applications.list
/home/huangjionggrui/.config/ukui-menu/-usr-share-unity-icons-shutdown_highlight.png
/home/huangjionggrui/.config/ukui-menu/-usr-share-webbrowser-app-webbrowser-app.png
/home/huangjionggrui/.config/ukui-menu/-usr-share-unity-icons-restart_highlight.png
/home/huangjionggrui/.config/dconf
/home/huangjionggrui/.config/dconf/user
/home/huangjionggrui/.config/fcitx
/home/huangjionggrui/.config/fcitx/conf/fcitx-notify.config
/home/huangjionggrui/.config/fcitx/conf/fcitx-classic-ui.config
/home/huangjionggrui/.config/fcitx/profile
/home/huangjionggrui/.config/fcitx/cached_layout
/home/huangjionggrui/.config/fcitx/clipboard/history.dat
/home/huangjionggrui/.config/fcitx/dbus/4e5dd0c32c07459096789d4964324a46-0
/home/huangjionggrui/.config/peony
/home/huangjionggrui/.config/peony/accels
/home/huangjionggrui/.config/peony/desktop-metadata
/home/huangjionggrui/.cache/ubuntu_kylin_sso/sso-client.log

```

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。

```
huangjionggrui@huangjionggrui-virtual-machine:~$ find ~ -name *Linux*
/home/huangjionggrui/.mozilla/firefox/3se968yd.default/gmp/Linux_x86_64-gcc3
/home/huangjionggrui/.thunderbird/d4vaoyb4.default/gmp/Linux_x86_64-gcc3
/home/huangjionggrui/.config/libreoffice/4/user/config/javasettings_Linux_X86_64.xml
```

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	drwxr-xr-x	24	root	root	4096byte
/etc/passwd	普通	-rw-r--r--	1	root	root	2255byte
/bin/df	普通	-rwxr-xr-x	1	root	root	84776byte
~	目录	drwxr-xr-x	24	huangjionggrui	huangjionggrui	4096byte

```
huangjionggrui@huangjionggrui-virtual-machine:~$ ls -ld ~ / /etc/passwd /bin/df
drwxr-xr-x 24 root      root      4096 3月  20 09:45 /
-rwxr-xr-x  1 root      root      84776 3月  1 2017 /bin/df
-rw-r--r--  1 root      root      2255 12月 23 2017 /etc/passwd
drwxr-xr-x 24 huangjionggrui huangjionggrui 4096 5月 14 20:01 /home/huangjionggrui
```

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。

```
huangjionggrui@huangjionggrui-virtual-machine:~$ chmod u=rwx,o=rx personal professional temp
huangjionggrui@huangjionggrui-virtual-machine:~$ ls -l personal professional temp
personal:
总用量 8
drwxr-xr-x 2 huangjionggrui huangjionggrui 4096 5月 14 13:11 funstuff
drwxr-xr-x 2 huangjionggrui huangjionggrui 4096 5月 14 13:11 taxes

professional:
总用量 8
drwxr-xr-x 4 huangjionggrui huangjionggrui 4096 5月 14 13:13 courses
drwxr-xr-x 4 huangjionggrui huangjionggrui 4096 5月 14 13:13 societies

temp:
总用量 0
```



```

huangjionggrui@huangjionggrui-virtual-machine:~$ cd temp
huangjionggrui@huangjionggrui-virtual-machine:~/temp$ mkdir d1 d2 d3
huangjionggrui@huangjionggrui-virtual-machine:~/temp$ touch d1/f1
huangjionggrui@huangjionggrui-virtual-machine:~/temp$ ls -l
总用量 12
drwxr-xr-x 2 huangjionggrui huangjionggrui 4096 5月 14 20:32 d1
drwxr-xr-x 2 huangjionggrui huangjionggrui 4096 5月 14 20:32 d2
drwxr-xr-x 2 huangjionggrui huangjionggrui 4096 5月 14 20:32 d3
huangjionggrui@huangjionggrui-virtual-machine:~/temp$ ls -l d1/f1
-rw-r--r-- 1 huangjionggrui huangjionggrui 0 5月 14 20:32 d1/f1

```

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。

```

huangjionggrui@huangjionggrui-virtual-machine:~$ chmod u=x temp
huangjionggrui@huangjionggrui-virtual-machine:~$ ls -ld temp
d--xr-xr-x 5 huangjionggrui huangjionggrui 4096 5月 14 20:32 temp
huangjionggrui@huangjionggrui-virtual-machine:~$ ls -l temp
ls: 无法打开目录'temp': 权限不够
huangjionggrui@huangjionggrui-virtual-machine:~$ chmod u=rx temp
huangjionggrui@huangjionggrui-virtual-machine:~$ ls -l temp
总用量 12
drwxr-xr-x 2 huangjionggrui huangjionggrui 4096 5月 14 20:32 d1
drwxr-xr-x 2 huangjionggrui huangjionggrui 4096 5月 14 20:32 d2
drwxr-xr-x 2 huangjionggrui huangjionggrui 4096 5月 14 20:32 d3

```

可以执行 ls -ld temp 但不能 ls -l temp

成功执行 ls -l temp 的最小的权限是 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。

```

huangjionggrui@huangjionggrui-virtual-machine:~/temp$ touch hello.c
huangjionggrui@huangjionggrui-virtual-machine:~/temp$ vi hello.c
huangjionggrui@huangjionggrui-virtual-machine:~/temp$ gcc -o greeting hello.c
hello.c:2:1: warning: return type defaults to 'int' [-Wimplicit-int]
main(void)
^~~~~~

```

```
huangjionggrui@huangjionggrui-virtual-machine:~/temp$ ls -l d1/f2 hello.c greet
ing d1
-rw-r--r-- 1 huangjionggrui huangjionggrui  0 5月 14 20:50 d1/f2
-rwxr-xr-x 1 huangjionggrui huangjionggrui 8416 5月 14 22:24 greeting
-rw-r--r-- 1 huangjionggrui huangjionggrui  62 5月 14 22:23 hello.c

d1:
总用量 4
drwxr-xr-x 2 huangjionggrui huangjionggrui 4096 5月 14 20:49 d11
-rw-r--r-- 1 huangjionggrui huangjionggrui  0 5月 14 20:32 f1
-rw-r--r-- 1 huangjionggrui huangjionggrui  0 5月 14 20:50 f2
```

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--	rwxr-xr-x	rwxr-xr-x
077	rw-----	rw-r--r--	rwxr-xr-x	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。

发现更改缺省设置后，已经存在的文件权限不改变，新增的权限收到该缺省设置影响。

26. 删除~/temp 目录下的所有文件和目录。给出会话过程。

```
huangjionggrui@huangjionggrui-virtual-machine:~/temp$ rm -r ./*
```

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。

```
mkdir d1 d2 d3
cp 文档/smallFile d1/smallFile
ls -il 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。

```
huangjionggrui@huangjionggrui-virtual-machine:~/temp$ cd d2
huangjionggrui@huangjionggrui-virtual-machine:~/temp/d2$ ln ../d1/smallFile
newFile.hard
huangjionggrui@huangjionggrui-virtual-machine:~/temp/d2$ ls -il newFile.ha
rd
660085 -rw----- 2 huangjionggrui huangjionggrui 986 5月 14 23:20 newFil
e.hard
```

```
huangjionggrui@huangjionggrui-virtual-machine:~/temp$ ls -il d1/smallFile
660085 -rw----- 1 huangjionggrui huangjionggrui 986 5月 14 23:20 d1/smallFile
```

是同一个文件，因为有同一个 **inode** 值。

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。

```
huangjionggrui@huangjionggrui-virtual-machine:~/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
```

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

取消了原文件的读取权限后，它的硬链接文件也不可以被读取，可以推断出硬链接文件和原文件本质上是一个文件。

在加回读取权限后，硬链接文件也可以被读取了。

将原文件删除后，显示硬链接文件依然会有输出。

由此分析：我认为硬链接相当于是文件的另一个名字，文件本身只有一个，但通过这两个文件名都可访问到该文件。当删除的时候，仅是删除了一个文件名，文件本身没有变化，所以依然可以被访问到。

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。

```

huangjionggrui@huangjionggrui-virtual-machine:~/temp/d2$ ln -s ../d1/smallFile smallFile.soft
huangjionggrui@huangjionggrui-virtual-machine:~/temp/d2$ ls -il smallFile.soft
677380 lrwxrwxrwx 1 huangjionggrui huangjionggrui 15 5月 14 23:56 smallFile.soft -> ../d1/smallFile
huangjionggrui@huangjionggrui-virtual-machine:~/temp/d2$ ls -il ../d1/smallFile
660088 -rw----- 1 huangjionggrui huangjionggrui 986 5月 14 23:44 ../d1/smallFile

```

发现两个文件的 **inode** 号并不一样，说明不是同一个文件

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。

```

huangjionggrui@huangjionggrui-virtual-machine:~/temp/d2$ cat 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

```

```

huangjionggrui@huangjionggrui-virtual-machine:~/temp/d2$ chmod u-r ../d1/smallFile
huangjionggrui@huangjionggrui-virtual-machine:~/temp/d2$ cat smallFile.soft
cat: smallFile.soft: 权限不够

```

改变原文件的权限也会导致软链接文件权限变化，而且发现软链接的文件大小远比原文件小，推断软链接文件相当于一种指针。

增加读权限后，可以再一次的显示文件内容

```

huangjionggrui@huangjionggrui-virtual-machine:~/temp/d2$ rm ../d1/smallFile
huangjionggrui@huangjionggrui-virtual-machine:~/temp/d2$ cat smallFile.soft
cat: smallFile.soft: 没有那个文件或目录

```

我认为软链接文件就是一种指向原文件的像指针的东西，一旦原文件没有了，指针本身不能显示任何东西，也就失去了它的意义。

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。

```

huangjionggrui@huangjionggrui-virtual-machine:~/temp/d2$ ps auxw|wc -l
275

```

```
huangjionggrui@huangjionggrui-virtual-machine:~$ ps -ef|grep init
root      1      0  0 5月14 ?        00:00:07 /sbin/init splash
```

```
huangjionggrui@huangjionggrui-virtual-machine:~$ ps -ef|grep bash
huangji+  3941   3935  0 00:37 pts/0    00:00:00 bash
```

```
huangjionggrui@huangjionggrui-virtual-machine:~$ ps -l
F S      UID      PID     PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
0 S      1000     3941     3935  0  80   0  -  6078 wait   pts/0        00:00:00 bash
0 R      1000     3975     3941  0  80   0  -  8273 -      pts/0        00:00:00 ps
```

```
huangjionggrui@huangjionggrui-virtual-machine:~$ ps -ef|grep 3935
huangji+  3935   1757  0 00:37 ?        00:00:01 /usr/lib/gnome-terminal/gnome-terminal-server
```

	PID	父进程	PPID
init	1	无	0
bash	3941	gnome-terminal-server	3935
ps	3975	bash	3941

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。

```
huangjionggrui@huangjionggrui-virtual-machine:~$ (sleep 60;echo "Time for lunch");&
[1] 3986
huangjionggrui@huangjionggrui-virtual-machine:~$ ps -l
F S      UID      PID     PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
0 S      1000     3941     3935  0  80   0  -  6078 wait   pts/0        00:00:00 bash
1 S      1000     3986     3941  0  80   0  -  6078 wait   pts/0        00:00:00 bash
0 S      1000     3987     3986  0  80   0  -  2261 hrtime  pts/0        00:00:00 sleep
0 R      1000     3988     3941  0  80   0  -  8273 -      pts/0        00:00:00 ps
huangjionggrui@huangjionggrui-virtual-machine:~$ 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。

```
huangjionggrui@huangjionggrui-virtual-machine:~$ date & uname -a & who & ps
[1] 4013
[2] 4014
[3] 4015
2019年 05月 15日 星期三 00:55:35 CST
Linux huangjionggrui-virtual-machine 4.10.0-19-generic #21-Ubuntu SMP Thu Apr 6 17:04:57 UTC 2017 x86_64 x86_64 x86_64 GNU/Linux
huangjionggrui tty7      2019-05-14 18:59 (:0)
  PID TTY          TIME CMD
  3941 pts/0        00:00:00 bash
  4016 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。

```

huangjionggrui@huangjionggrui-virtual-machine:~$ date ; uname -a ; who ; ps
2019年 05月 15日 星期三 00:56:58 CST
Linux huangjionggrui-virtual-machine 4.10.0-19-generic #21-Ubuntu SMP Thu Apr 6 17:0
4:57 UTC 2017 x86_64 x86_64 x86_64 GNU/Linux
huangjionggrui tty7      2019-05-14 18:59 (:0)
  PID TTY          TIME CMD
 3941 pts/0        00:00:00 bash
 4024 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

```

huangjionggrui@huangjionggrui-virtual-machine:~$ pwd
/home/huangjionggrui
huangjionggrui@huangjionggrui-virtual-machine:~$ bash
huangjionggrui@huangjionggrui-virtual-machine:~$ cd /usr
huangjionggrui@huangjionggrui-virtual-machine:/usr$ pwd
/usr
huangjionggrui@huangjionggrui-virtual-machine:/usr$ exit
huangjionggrui@huangjionggrui-virtual-machine:~$ pwd
/home/huangjionggrui

```

① /home/huangjionggrui②/usr③/home/huangjionggrui

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）

```

huangjionggrui@huangjionggrui-virtual-machine:~$ find ~ -name foobar 2> /dev/null
/home/huangjionggrui/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。

```

huangjionggrui@huangjionggrui-virtual-machine:~$ find ~ -name foobar 1> foobar.path 2> /dev/null
huangjionggrui@huangjionggrui-virtual-machine:~$ cat foobar.path
/home/huangjionggrui/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 1> 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 <fata 1> fout

```
huangjiongrui@huangjiongrui-virtual-machine:~$ cat fout  
The time now is:2019年 05月 15日 星期三 09:26:35 CST  
The users presently logged on are:huangjiongrui tty7          2019-05-15 08:59 (: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。

```
huangjiongrui@huangjiongrui-virtual-machine:~$ ls -l | wc -l  
19  
huangjiongrui@huangjiongrui-virtual-machine:~$ ls -l | wc -m  
1234  
huangjiongrui@huangjiongrui-virtual-machine:~$ ls -l | wc -w  
164
```

有 19 行 164 个单词数 1234 个字符数

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？

```
huangjiongrui@huangjiongrui-virtual-machine:/bin$ ls -l | grep "^-" | wc -l  
139  
huangjiongrui@huangjiongrui-virtual-machine:/bin$ ls -l | grep "^d" | wc -l  
0  
huangjiongrui@huangjiongrui-virtual-machine:/bin$ ls -l | grep "^l" | wc -l  
23
```

有 139 个普通文件 1 个目录 23 个符号链接文件

三、 讨论、心得 (10 分)

本次实验主要对于之前学过的知识进行了复习回顾，让我更加熟悉了 linux 下各种命令的用法。由于在上这个课之前也用过 linux 的虚拟机，所以对于命令还算熟悉。这次的作业虽然繁琐，但一边搜索一边做，也没遇到很大的困难。

在作业过程中总共遇到了两个问题。第一个是查询进程，在网上搜到了很多种查询进程的方法，但都不太理想。有些只能查询该用户的进程，有些仅仅列出了所有的进程。最后利用了 grep 对于需要的关键字进行搜索。找到了相应的进

程。同样的通过 `grep` 搜索出所需进程的父进程分别是什么。最终解决了该问题。特别感谢 <https://www.cnblogs.com/qinduanyinghua/p/7266019.html> 的文章。

还有一个问题来自于软链接和硬链接的区别。做完了题目，我觉得我对于软链接的认识还可以，但是硬链接到底是什么样的存在我心里还是有疑惑。后来我查找相关资 <https://www.ibm.com/developerworks/cn/linux/l-cn-hardandsymb-links/>, 最终对于硬链接更深入的认识。