

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: _____ 专业: 信息安全 学号: _____

电子邮件地址: _____

实验日期: 2020 年 05 月 30 日

一、 实验环境

计算机配置:

处理器: Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz (8 CPUs), ~1.8GHz

内存: 8192MB RAM

显卡名称: Intel(R) UHD Graphics 620

芯片类型: Intel(R) UHD Graphics Family

操作系统:

Windows 10 家庭中文版 64-bit (10.0, Build 18363)

Linux 版本:

Ubuntu 18.04.4 LTS

二、 实验内容和结果及分析（90 分）

//问题回答及分析；上机实验输入输出显示的结果截图；放在这里

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

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

国际上常用的软件许可证包括：GPL、BSD、LGPL、MIT 许可证。

GPL 许可证:

GPL 许可证是自由软件联盟 GNU 的开放源代码软件许可证的一种，是开放源代码软件领域最富盛名的一种许可证，同时，GPL 也是开放源代码软件领域对被许可人

权利限制最严的许可证，它集中体现了纯粹的 Copyleft 精神。

GPL 首先是承认软件的版权——软件有原始的版权所有者，然后纳入自由软件的版权许可约束。在承认版权的前提下，通过软件的版权许可来实现自由权利的要求。GPL 许可证最重要的特点在于它要求根据 GPL 许可证发布的软件（如：最有影响力的 Linux）修改、翻译的演绎作品，甚至只要其任何一部分代码是以 GPL 发布的，那么全部程序也必须受 GPL 许可证的约束，即继续遵守 GPL 许可证的规定，有些人将其称之为 GPL 的“病毒效应”。

BSD 许可证：

相对于其他开源软件许可证，特别是 GPL 许可证，BSD (Berkly Software Distribution) 许可证可能对被许可人来说是最“宽容”的，虽然 BSD 许可证具备开源软件许可证普遍的要求，但 BSD 许可证只要求被许可者附上该许可证的原文以及所有开发者的版权资料。通俗地说，BSD 许可证看重的是“名”，在“利”方面，BSD 许可证给予被许可者充分使用（包含商业使用）源代码的权利。

BSD 许可证完全不会产生像 GPL 许可证那样的“病毒效应”，为开源软件，特别是作为学校、公共科研机构科研成果的开源软件转化为产品等充分利用敞开了大门。

LGPL 许可证：

LGPL 是 GPL 的一个为主要为类库使用设计的开源协议。和 GPL 要求任何使用/修改/衍生之 GPL 类库的软件必须采用 GPL 协议不同。LGPL 允许商业软件通过类库引用(link)方式使用 LGPL 类库而不需要开源商业软件的代码。这使得采用 LGPL 协议的开源代码可以被商业软件作为类库引用并发布和销售。

但是如果修改 LGPL 协议的代码或者衍生，则所有修改的代码，涉及修改部分的额外代码和衍生的代码都必须采用 LGPL 协议。因此 LGPL 协议的开源代码很适合作为第三方类库被商业软件引用，但不适合希望以 LGPL 协议代码为基础，通过修改和衍生的方式做二次开发的商业软件采用。GPL/LGPL 都保障原作者的知识产权，避免有人利用开源代码复制并开发类似的产品。

MIT 许可证：

MIT 是和 BSD 一样宽泛的许可协议，作者只想保留版权，而无任何其他限制。也就是说，你必须在你的发行版里包含原许可协议的声明，无论你是以二进制发布的还是以源代码发布的。

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

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

j. \$ cd

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

  --author
      print the author of each file

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

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
    passwd 命令用来更改用户账户的密码。普通用户通常只更改其自己账户的密码，而超级用户可以更改任何账户的密码。passwd 也更改账户或相关的密码有效期。

    密码更改
    如果有旧密码，首先提示用户输入旧密码。加密这个密码然后和存储的密码进行比较。用户只有一次机会输入正确密码。允许超级用户略过这个步骤，以便更改忘记了密码。

    输入了密码之后，会检查密码年龄信息，以查看此时是否允许用户更改密码。如果不允许，passwd 拒绝更改密码，然后退出。

    提示用户输入两次新密码。第二次和第一次进行比较，并且需要相同才能更改密码。

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

info pwd

```
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-----
Unknown command (M-a)
```

who --help

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

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

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？

xx

- 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？

xx-virtual-machine

- 4) 你计算机系统的 CPU 名字是什么？

x86_64

```
xx@xx-virtual-machine:~$ info pwd
xx@xx-virtual-machine:~$ whoami
xx
xx@xx-virtual-machine:~$ uname
Linux
xx@xx-virtual-machine:~$ uname -n
xx-virtual-machine
xx@xx-virtual-machine:~$ uname -p
x86_64
```

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

- 1) 给出你显示以上年份年历的命令

cal 4


```
xx@xx-virtual-machine:~$ cal 4
4
      一月          二月          三月
日 一 二 三 四 五 六 日 一 二 三 四 五 六 日 一 二 三 四 五 六
    1  2  3  4  5          1  2          1
  6  7  8  9 10 11 12    3  4  5  6  7  8  9    2  3  4  5  6  7  8
13 14 15 16 17 18 19    10 11 12 13 14 15 16    9 10 11 12 13 14 15
20 21 22 23 24 25 26    17 18 19 20 21 22 23    16 17 18 19 20 21 22
27 28 29 30 31          24 25 26 27 28 29          23 24 25 26 27 28 29
                                     30 31

      四月          五月          六月
日 一 二 三 四 五 六 日 一 二 三 四 五 六 日 一 二 三 四 五 六
    1  2  3  4  5          1  2  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

      七月          八月          九月
日 一 二 三 四 五 六 日 一 二 三 四 五 六 日 一 二 三 四 五 六
    1  2  3  4  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

      十月          十一月          十二月
日 一 二 三 四 五 六 日 一 二 三 四 五 六 日 一 二 三 四 五 六
    1  2  3  4          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
```

cal 1752

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

      四月          五月          六月
日 一 二 三 四 五 六 日 一 二 三 四 五 六 日 一 二 三 四 五 六
    1  2  3  4          1  2          1  2  3  4  5  6
  5  6  7  8  9 10 11    3  4  5  6  7  8  9    7  8  9 10 11 12 13
12 13 14 15 16 17 18    10 11 12 13 14 15 16    14 15 16 17 18 19 20
19 20 21 22 23 24 25    17 18 19 20 21 22 23    21 22 23 24 25 26 27
26 27 28 29 30          24 25 26 27 28 29 30    28 29 30
                          31

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

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

cal 7 2012

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

2) 1752 年有几天，为什么？

```
xx@xx-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
```

从上图中可以看出 355 天，除二月有 29 天外，由 1) 可知九月有 19 天，少了 11 天。因为 1752 年 9 月大英帝国极其所属美洲殖民地的恺撒历法被格里高利教皇历法所取代。由于恺撒历法比格里高利历法迟 11 天，因此 9 月 2 日当天改历法后，次日须为 9 月 14 日。

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。

```
xx@xx-virtual-machine:~$ uptime
00:46:01 up 7:25, 1 user, load average: 0.00, 0.00, 0.00
```

系统已经启动了 7 小时 25 分钟，当前系统中有一个登录用户。

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
touch	修改文件或目录的时间属性，若文件不存在则会新建一个文件	touch 1.txt
cp	复制文件或者目录	cp log.log 1
mv	移动文件或目录或者为文件或目录改名	mv 1.c 1.txt
rm	删除文件或目录	rm -i 1.txt
mkdir	创建指定的名称的目录	mkdir 1
who	显示系统中有哪些使用者正在上面	who -count
ls	查看文件夹包含的文件、文件权限、目录信息等	ls -l *t
cd	切换工作目录	cd ~
pwd	查看当前工作目录的完整路径	pwd
open	打开和创建文件	
read	由已经打开的文件读取数据	

write	将数据写入已经打开的文件
close	关闭文件
pipe	建立管道
socket	建立一个 socket 通信
mkfifo	建立具名管道
system	执行 shell 命令
printf	格式化输出数据

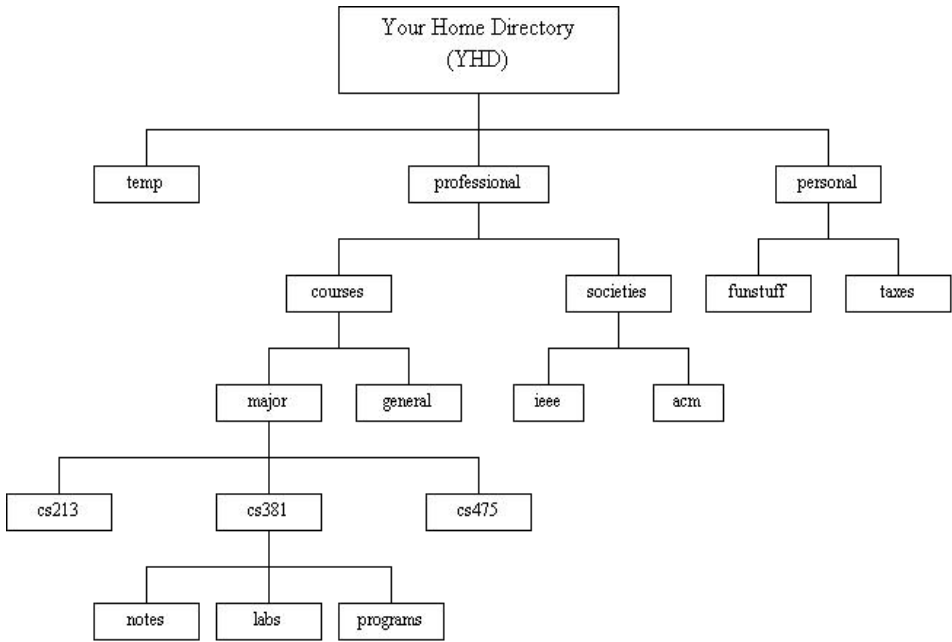

```

xx@xx-virtual-machine:~$ vi firscrip
xx@xx-virtual-machine:~$ bash ./firscrip
总用量 140
drwxr-xr-x 18 xx xx 4096 5月 31 01:18 .
drwxr-xr-x 3 root root 4096 3月 3 21:31 ..
-rw-r--r-- 1 xx xx 988 5月 31 01:13 .bash_history
-rw-r--r-- 1 xx xx 220 3月 3 21:31 .bash_logout
-rw-r--r-- 1 xx xx 3771 3月 3 21:31 .bashrc
drwx----- 19 xx xx 4096 4月 17 15:42 .cache
drwx----- 16 xx xx 4096 4月 10 18:35 .config
drwx----- 3 root root 4096 3月 12 12:52 .dbus
-rw-r--r-- 1 xx xx 8980 3月 3 21:31 examples.desktop
-rw-r--r-- 1 xx xx 16 5月 31 01:18 firscrip
-rw-r--r-- 1 xx xx 12288 5月 31 01:15 .firscrip.swo
-rw-r--r-- 1 xx xx 12288 5月 31 01:13 .firscrip.swp
drwx----- 3 xx xx 4096 3月 3 21:39 .gnupg
-rw-r--r-- 1 xx xx 5116 5月 31 01:10 .ICEauthority
drwxrwxr-x 3 xx xx 4096 3月 17 19:52 .kingsoft
drwx----- 3 xx xx 4096 3月 3 21:53 .local
drwx----- 5 xx xx 4096 3月 3 21:59 .mozilla
-rw-r--r-- 1 xx xx 807 3月 3 21:31 .profile
-rw-r--r-- 1 xx xx 0 3月 3 22:45 .sudo_as_admin_successful
drwx----- 6 xx xx 4096 4月 17 15:42 .thunderbird
-rw-rw-r-- 1 xx xx 131 3月 12 12:47 .xinputrc
drwxr-xr-x 2 xx xx 4096 3月 3 21:53 公共的
drwxr-xr-x 2 xx xx 4096 3月 17 19:42 模板
drwxr-xr-x 2 xx xx 4096 3月 3 21:53 视频
drwxr-xr-x 2 xx xx 4096 3月 4 00:15 图片
drwxr-xr-x 2 xx xx 4096 3月 3 21:53 文档
drwxr-xr-x 2 xx xx 4096 3月 3 21:53 下载
drwxr-xr-x 2 xx xx 4096 3月 3 21:53 音乐
drwxr-xr-x 2 xx xx 4096 5月 30 17:11 桌面
xx      :0      2020-05-31 01:10 (:0)
/home/xx

```

当前的工作目录中有 29 个文件，它们的名称如最后一列所示，大小如第 5 列所示。用户 xx 在使用计算机系统，当前工作目录为/home/xx

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）



命令输入:

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

结果输出:

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

27 directories
```

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

- 1) 你的主目录的绝对路径是什么? 给出获得该绝对路径的命令及命令输出。

绝对路径: `/home/xx` 命令: `echo ~`

输出：

```
xx@xx-virtual-machine:~$ cd professional/courses
xx@xx-virtual-machine:~/professional/courses$ echo ~
/home/xx
```

2) **acm** 目录的绝对路径是什么？

/home/xx/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。

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

5) 给出获得你的主目录三个不同的命令。

1. **echo \$HOME**

2. **echo ~**

3. **cd ~**

pwd

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

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？

```
xx@xx-virtual-machine:~$ ls -d .*
.          .bashrc   .fircscrip.swo .ICEauthority .profile
..         .cache    .fircscrip.swp .kingsoft     .sudo_as_admin_successful
.bash_history .config   .gnupg         .local        .thunderbird
.bash_logout .dbus     .gvfs         .mozilla     .xinputrc
```

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。


```

xx@xx-virtual-machine:~$ stat /
  文件: /
  大小: 4096          块: 8          IO 块: 4096   目录
设备: 801h/2049d      Inode: 2          硬链接: 25
权限: (0755/drwxr-xr-x)  Uid: (  0/   root)  Gid: (  0/   root)
最近访问: 2020-05-30 17:10:17.488987040 +0800
最近更改: 2020-03-04 00:01:32.334391285 +0800
最近改动: 2020-03-04 00:01:32.334391285 +0800
创建时间: -
xx@xx-virtual-machine:~$ stat ~
  文件: /home/xx
  大小: 4096          块: 8          IO 块: 4096   目录
设备: 801h/2049d      Inode: 798432     硬链接: 23
权限: (0755/drwxr-xr-x)  Uid: ( 1000/   xx)  Gid: ( 1000/   xx)
最近访问: 2020-05-31 13:34:07.033467867 +0800
最近更改: 2020-05-31 13:21:49.545441350 +0800
最近改动: 2020-05-31 13:21:49.545441350 +0800
创建时间: -
xx@xx-virtual-machine:~$ stat ~/temp
  文件: /home/xx/temp
  大小: 4096          块: 8          IO 块: 4096   目录
设备: 801h/2049d      Inode: 800222     硬链接: 2
权限: (0755/drwxr-xr-x)  Uid: ( 1000/   xx)  Gid: ( 1000/   xx)
最近访问: 2020-05-31 02:47:16.031282712 +0800
最近更改: 2020-05-31 01:31:49.846785316 +0800
最近改动: 2020-05-31 01:31:49.846785316 +0800
创建时间: -
xx@xx-virtual-machine:~$ stat ~/professional
  文件: /home/xx/professional
  大小: 4096          块: 8          IO 块: 4096   目录
设备: 801h/2049d      Inode: 803272     硬链接: 4
权限: (0755/drwxr-xr-x)  Uid: ( 1000/   xx)  Gid: ( 1000/   xx)
最近访问: 2020-05-31 02:47:16.031282712 +0800
最近更改: 2020-05-31 01:34:33.030784947 +0800
最近改动: 2020-05-31 01:34:33.030784947 +0800
创建时间: -
xx@xx-virtual-machine:~$ stat ~/personal
  文件: /home/xx/personal
  大小: 4096          块: 8          IO 块: 4096   目录
设备: 801h/2049d      Inode: 803273     硬链接: 4
权限: (0755/drwxr-xr-x)  Uid: ( 1000/   xx)  Gid: ( 1000/   xx)
最近访问: 2020-05-31 02:47:16.031282712 +0800
最近更改: 2020-05-31 01:33:59.842785022 +0800
最近改动: 2020-05-31 01:33:59.842785022 +0800
创建时间: -

```

1、主目录（/home/xx）、~/temp、~/professional、和~/personal 的 inode 号分别为 2、798432、800222、803272、803273

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。

-代表普通文件类型

```

xx@xx-virtual-machine:~$ cd professional/courses/major/cs381/labs
xx@xx-virtual-machine:~/professional/courses/major/cs381/labs$ touch lab1
xx@xx-virtual-machine:~/professional/courses/major/cs381/labs$ vi lab1
xx@xx-virtual-machine:~/professional/courses/major/cs381/labs$ ls -dl lab1
-rw-r--r-- 1 xx xx 152 5月 31 13:46 lab1

```

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

ASCII text

```

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

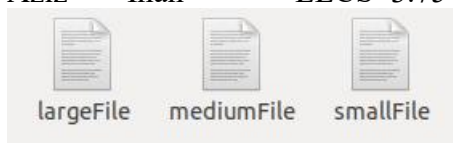
```

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

```
xx@xx-virtual-machine:~$ cd /usr/include
xx@xx-virtual-machine:/usr/include$ ls t*.h
tar.h  termio.h  termios.h  tgmash.h  thread_db.h  time.h  ttyent.h
```

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

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



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


```
xx@xx-virtual-machine:~$ head -12 largeFile
BASH(1)                                     General Commands Manual                                     BASH(1)

NAME
    bash - GNU Bourne-Again Shell

SYNOPSIS
    bash [options] [command_string | file]

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

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

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

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

然后再把文件 `newDataFile` 移动到 `/tmp` 目录下，文件 `/tmp/newDataFile` 的 `inode` 号是多少？比较结果如何，为什么？

给出完成上述工作的会话过程。

```

xx@xx-virtual-machine:~$ cp smallFile dataFile
xx@xx-virtual-machine:~$ ls -l
总用量 412
-rw-r--r-- 1 xx xx 963 5月 31 14:01 dataFile
-rw-r--r-- 1 xx xx 8980 3月 3 21:31 examples.desktop
-rw-r--r-- 1 xx xx 16 5月 31 01:18 firscrip
-rw-r--r-- 1 xx xx 341953 5月 31 13:53 largeFile
-rw-r--r-- 1 xx xx 1992 5月 31 13:52 mediumFile
drwxr-xr-x 4 xx xx 4096 5月 31 01:33 personal
drwxr-xr-x 4 xx xx 4096 5月 31 01:34 professional
-rw-r--r-- 1 xx xx 963 5月 31 13:55 smallFile
drwxr-xr-x 2 xx xx 4096 5月 31 01:31 temp
drwxr-xr-x 3 xx xx 4096 5月 31 02:45 tree-1.8.0
drwxr-xr-x 2 xx xx 4096 3月 3 21:53 公共的
drwxr-xr-x 2 xx xx 4096 3月 17 19:42 模板
drwxr-xr-x 2 xx xx 4096 3月 3 21:53 视频
drwxr-xr-x 2 xx xx 4096 3月 4 00:15 图片
drwxr-xr-x 2 xx xx 4096 3月 3 21:53 文档
drwxr-xr-x 2 xx xx 4096 3月 3 21:53 下载
drwxr-xr-x 2 xx xx 4096 3月 3 21:53 音乐
drwxr-xr-x 2 xx xx 4096 5月 30 17:11 桌面
xx@xx-virtual-machine:~$ cp -p smallFile datafile
xx@xx-virtual-machine:~$ ls -l
总用量 416
-rw-r--r-- 1 xx xx 963 5月 31 13:55 datafile
-rw-r--r-- 1 xx xx 963 5月 31 14:01 dataFile
-rw-r--r-- 1 xx xx 8980 3月 3 21:31 examples.desktop
-rw-r--r-- 1 xx xx 16 5月 31 01:18 firscrip
-rw-r--r-- 1 xx xx 341953 5月 31 13:53 largeFile
-rw-r--r-- 1 xx xx 1992 5月 31 13:52 mediumFile
drwxr-xr-x 4 xx xx 4096 5月 31 01:33 personal
drwxr-xr-x 4 xx xx 4096 5月 31 01:34 professional
-rw-r--r-- 1 xx xx 963 5月 31 13:55 smallFile
drwxr-xr-x 2 xx xx 4096 5月 31 01:31 temp
drwxr-xr-x 3 xx xx 4096 5月 31 02:45 tree-1.8.0
drwxr-xr-x 2 xx xx 4096 3月 3 21:53 公共的
drwxr-xr-x 2 xx xx 4096 3月 17 19:42 模板
drwxr-xr-x 2 xx xx 4096 3月 3 21:53 视频
drwxr-xr-x 2 xx xx 4096 3月 4 00:15 图片
drwxr-xr-x 2 xx xx 4096 3月 3 21:53 文档
drwxr-xr-x 2 xx xx 4096 3月 3 21:53 下载
drwxr-xr-x 2 xx xx 4096 3月 3 21:53 音乐
drwxr-xr-x 2 xx xx 4096 5月 30 17:11 桌面

```

这两个文件的修改时间不同的，dataFile 文件的修改时间应该是这个文件的创建时间。

cp -p smallFile datafile 命令可以复制文件使得新的文件修改时间与原文件相同。

```

xx@xx-virtual-machine:~$ stat smallFile
 文件: smallFile
 大小: 963          块: 8          IO 块: 4096   普通文件
设备: 801h/2049d    Inode: 802406    硬链接: 1
权限: (0644/-rw-r--r--)  Uid: ( 1000/   xx)   Gid: ( 1000/   xx)
最近访问: 2020-05-31 13:56:43.881516655 +0800
最近更改: 2020-05-31 13:55:50.185514724 +0800
最近改动: 2020-05-31 13:55:50.185514724 +0800
创建时间: -
xx@xx-virtual-machine:~$ stat dataFile
 文件: dataFile
 大小: 963          块: 8          IO 块: 4096   普通文件
设备: 801h/2049d    Inode: 800272    硬链接: 1
权限: (0644/-rw-r--r--)  Uid: ( 1000/   xx)   Gid: ( 1000/   xx)
最近访问: 2020-05-31 14:01:11.249526269 +0800
最近更改: 2020-05-31 14:01:11.253526269 +0800
最近改动: 2020-05-31 14:01:11.253526269 +0800
创建时间: -

```

smallFile 和 dataFile 这两个文件的 inode 号分别为 802406 和 800272。

```
xx@xx-virtual-machine:~$ mv dataFile newDataFile
xx@xx-virtual-machine:~$ ls -li newDataFile
800272 newDataFile
```

把文件名 dataFile 改成 newDataFile，文件 newDataFile 的 inode 号为 800272，与 dataFile 文件的 inode 号相同，因为重命名文件不影响 inode 指向的文件地址。

```
xx@xx-virtual-machine:~$ mv newDataFile /tmp
xx@xx-virtual-machine:~$ cd /tmp
xx@xx-virtual-machine:/tmp$ ls -li newDataFile
800272 newDataFile
```

再把文件 newDataFile 移动到/tmp 目录下，文件/tmp/newDataFile 的 inode 号仍为 800272，与 dataFile 文件的 inode 号相同，因为重命名文件不影响 inode 指向的文件地址。

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。

```
$ wc smallFile mediumFile largeFile /tmp/newDataFile
```

第 1、2、3 列分别为行数，字数和字节数

```
xx@xx-virtual-machine:~$ wc smallFile mediumFile largeFile /tmp/newDataFile
 20   120   963 smallFile
 71   216  1992 mediumFile
5129 46300 341953 largeFile
 20   120   963 /tmp/newDataFile
5240 46756 345871 总用量
```

```
$ ls -li smallFile mediumFile largeFile /tmp/newDataFile
```

第 5 列为字节数

```
xx@xx-virtual-machine:~$ ls -li smallFile mediumFile largeFile /tmp/newDataFile
-rw-r--r-- 1 xx xx 341953 5月 31 13:53 largeFile
-rw-r--r-- 1 xx xx 1992 5月 31 13:52 mediumFile
-rw-r--r-- 1 xx xx 963 5月 31 13:55 smallFile
-rw-r--r-- 1 xx xx 963 5月 31 14:01 /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。

```
sudo find ~ -name "*.c" -o -name "*.html" -o -name "*.htm"
```

```
xx@xx-virtual-machine:~$ sudo find ~ -name "*.c" -o -name "*.html" -o -name "*.htm"
/home/xx/test.c
/home/xx/test.html
/home/xx/test2.htm
/home/xx/tree-1.8.0/unix.c
/home/xx/tree-1.8.0/html.c
/home/xx/tree-1.8.0/file.c
/home/xx/tree-1.8.0/tree.c
/home/xx/tree-1.8.0/xml.c
/home/xx/tree-1.8.0/json.c
/home/xx/tree-1.8.0/color.c
/home/xx/tree-1.8.0/hash.c
/home/xx/tree-1.8.0/strverscmp.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。

\$ sudo find ~ -cnewer ~/smallFile

下图为部分截图

```
xx@xx-virtual-machine: /usr/include$ sudo find ~ -cnewer ~/smallFile
/home/xx
/home/xx/.mozilla/firefox
/home/xx/.mozilla/firefox/installs.ini
/home/xx/.mozilla/firefox/profiles.ini
/home/xx/.mozilla/firefox/6ied6i5d.default-release-1590907261793
/home/xx/.mozilla/firefox/6ied6i5d.default-release-1590907261793/sessionstore.jsonlz4
/home/xx/.mozilla/firefox/6ied6i5d.default-release-1590907261793/sessionstore-backups
/home/xx/.mozilla/firefox/6ied6i5d.default-release-1590907261793/sessionstore-backups/upgrade.jsonlz4-20200117190643
/home/xx/.mozilla/firefox/6ied6i5d.default-release-1590907261793/sessionstore-backups/previous.jsonlz4
/home/xx/.mozilla/firefox/6ied6i5d.default-release-1590907261793/cert_override.txt
/home/xx/.mozilla/firefox/6ied6i5d.default-release-1590907261793/xulstore.json
/home/xx/.mozilla/firefox/6ied6i5d.default-release-1590907261793/webappsstore.sqlite
/home/xx/.mozilla/firefox/6ied6i5d.default-release-1590907261793/weave
/home/xx/.mozilla/firefox/6ied6i5d.default-release-1590907261793/weave/bookmarks.sqlite
/home/xx/.mozilla/firefox/6ied6i5d.default-release-1590907261793/weave/changes
/home/xx/.mozilla/firefox/6ied6i5d.default-release-1590907261793/weave/changes/history.json
/home/xx/.mozilla/firefox/6ied6i5d.default-release-1590907261793/weave/changes/forms.json
/home/xx/.mozilla/firefox/6ied6i5d.default-release-1590907261793/weave/changes/addons.json
/home/xx/.mozilla/firefox/6ied6i5d.default-release-1590907261793/weave/changes/passwords.json
/home/xx/.mozilla/firefox/6ied6i5d.default-release-1590907261793/weave/changes/clients.json
/home/xx/.mozilla/firefox/6ied6i5d.default-release-1590907261793/weave/toFetch
/home/xx/.mozilla/firefox/6ied6i5d.default-release-1590907261793/weave/toFetch/history.json
/home/xx/.mozilla/firefox/6ied6i5d.default-release-1590907261793/weave/toFetch/extension-storage.json
/home/xx/.mozilla/firefox/6ied6i5d.default-release-1590907261793/weave/toFetch/forms.json
/home/xx/.mozilla/firefox/6ied6i5d.default-release-1590907261793/weave/toFetch/bookmarks.json
/home/xx/.mozilla/firefox/6ied6i5d.default-release-1590907261793/weave/toFetch/tabs.json
/home/xx/.mozilla/firefox/6ied6i5d.default-release-1590907261793/weave/toFetch/addons.json
/home/xx/.mozilla/firefox/6ied6i5d.default-release-1590907261793/weave/toFetch/passwords.json
/home/xx/.mozilla/firefox/6ied6i5d.default-release-1590907261793/weave/toFetch/clients.json
/home/xx/.mozilla/firefox/6ied6i5d.default-release-1590907261793/weave/toFetch/prefs.json
/home/xx/.mozilla/firefox/6ied6i5d.default-release-1590907261793/weave/addonsreconciler.json
/home/xx/.mozilla/firefox/6ied6i5d.default-release-1590907261793/weave/logs
/home/xx/.mozilla/firefox/6ied6i5d.default-release-1590907261793/weave/logs/error-sync-1590907903534.txt
/home/xx/.mozilla/firefox/6ied6i5d.default-release-1590907261793/weave/logs/error-sync-1590907995581.txt
/home/xx/.mozilla/firefox/6ied6i5d.default-release-1590907261793/weave/logs/error-sync-1590907462307.txt
/home/xx/.mozilla/firefox/6ied6i5d.default-release-1590907261793/weave/logs/error-sync-1590907550219.txt
```

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。

\$ sudo find ~ -name *Linux*

```
xx@xx-virtual-machine: ~$ sudo find ~ -name *Linux*
/home/xx/.config/libreoffice/4/user/pack/config/javasettings_Linux_X86_64.pack
/home/xx/.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	25	root	root	4096
/etc/passwd	普通	-rw-r--r--	1	root	root	2395
/bin/df	普通	-rwxr-xr-x	1	root	root	84776
~	目录	drwxr-xr-x	24	xx	xx	4096

```
xx@xx-virtual-machine: ~$ ls -ld / /etc/passwd /bin/df ~
drwxr-xr-x 25 root root 4096 3月 4 00:01 /
-rwxr-xr-x 1 root root 84776 1月 18 2018 /bin/df
-rw-r--r-- 1 root root 2395 3月 3 21:31 /etc/passwd
drwxr-xr-x 24 xx xx 4096 5月 31 15:18 /home/xx
```

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。

```
xx@xx-virtual-machine:~$ chmod u=rwx,o=rx personal professional temp
xx@xx-virtual-machine:~$ ls -l
总用量 412
-rw-r--r-- 1 xx xx 963 5月 31 13:55 datafile
-rw-r--r-- 1 xx xx 8980 3月 3 21:31 examples.desktop
-rw-r--r-- 1 xx xx 16 5月 31 01:18 firstrip
-rw-r--r-- 1 xx xx 341953 5月 31 13:53 largeFile
-rw-r--r-- 1 xx xx 1992 5月 31 13:52 mediumFile
drwxr-xr-x 4 xx xx 4096 5月 31 01:33 personal
drwxr-xr-x 4 xx xx 4096 5月 31 01:34 professional
-rw-r--r-- 1 xx xx 963 5月 31 13:55 smallFile
drwxr-xr-x 2 xx xx 4096 5月 31 01:31 temp
-rw-r--r-- 1 xx xx 0 5月 31 15:18 test2.htm
-rw-r--r-- 1 xx xx 0 5月 31 14:22 test.c
-rw-r--r-- 1 xx xx 0 5月 31 15:17 test.html
drwxr-xr-x 3 xx xx 4096 5月 31 02:45 tree-1.8.0
drwxr-xr-x 2 xx xx 4096 3月 3 21:53 公共的
drwxr-xr-x 2 xx xx 4096 3月 17 19:42 模板
drwxr-xr-x 2 xx xx 4096 3月 3 21:53 视频
drwxr-xr-x 2 xx xx 4096 3月 4 00:15 图片
drwxr-xr-x 2 xx xx 4096 3月 3 21:53 文档
drwxr-xr-x 2 xx xx 4096 3月 3 21:53 下载
drwxr-xr-x 2 xx xx 4096 3月 3 21:53 音乐
drwxr-xr-x 2 xx xx 4096 5月 31 15:09 桌面
xx@xx-virtual-machine:~$ cd temp
xx@xx-virtual-machine:~/temp$ mkdir d1 d2 d3
xx@xx-virtual-machine:~/temp$ touch d1/f1
xx@xx-virtual-machine:~/temp$ ls -l
总用量 12
drwxr-xr-x 2 xx xx 4096 5月 31 15:36 d1
drwxr-xr-x 2 xx xx 4096 5月 31 15:36 d2
drwxr-xr-x 2 xx xx 4096 5月 31 15:36 d3
xx@xx-virtual-machine:~/temp$ ls -l d1/f1
-rw-r--r-- 1 xx xx 0 5月 31 15:36 d1/f1
```

d1,d2,d3 所有者：读|写|执行，组和其他：读|执行

f1 所有者：读|写，组合其他：读

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。

```
xx@xx-virtual-machine:~$ chmod u=x temp
xx@xx-virtual-machine:~$ ls -ld temp
d--xr--r-- 5 xx xx 4096 5月 31 15:36 temp
xx@xx-virtual-machine:~$ ls -l temp
ls: 无法打开目录'temp': 权限不够
xx@xx-virtual-machine:~$ chmod u=rx temp
xx@xx-virtual-machine:~$ ls -l temp
总用量 12
drwxr-xr-x 2 xx xx 4096 5月 31 15:36 d1
drwxr-xr-x 2 xx xx 4096 5月 31 15:36 d2
drwxr-xr-x 2 xx xx 4096 5月 31 15:36 d3
```

可以执行 **ls -ld temp**，但 **ls -l temp** 权限不够。成功执行 **ls -l temp** 的最小权限是 **rx**，因为 **r** 和 **x** 分别代表可以读出目录的内容和检索这个目录。

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

```
#include <stdio.h>

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

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

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

umask 值	文件权限			
	f2	hello.c	greeting	d11 / d21
022	-rw-rw-r--	-rw-r--r--	-rwxr-xr-x	drwxrwxr-x
077	-rw-----	-rw-r--r--	-rwxr-xr-x	drwx-----

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。

```
xx@xx-virtual-machine:~$ umask
0022
xx@xx-virtual-machine:~$ mkdir ~/temp/d1/d11
xx@xx-virtual-machine:~$ touch ~/temp/d1/f2
xx@xx-virtual-machine:~$ cd ~/temp
xx@xx-virtual-machine:~/temp$ touch hello.c
xx@xx-virtual-machine:~/temp$ vi hello.c
xx@xx-virtual-machine:~/temp$ gcc -o greeting hello.c
hello.c:2:1: warning: return type defaults to 'int' [-Wimplicit-int]
main(void)
^
xx@xx-virtual-machine:~/temp$ ls -l d1/f2 hello.c greeting d1
-rw-rw-r-- 1 xx xx 0 5月 31 16:06 d1/f2
-rwxr-xr-x 1 xx xx 8304 5月 31 16:10 greeting
-rw-r--r-- 1 xx xx 62 5月 31 16:10 hello.c

d1:
总用量 4
drwxrwxr-x 2 xx xx 4096 5月 31 16:05 d11
-rw-rw-r-- 1 xx xx 0 5月 31 15:36 f1
-rw-rw-r-- 1 xx xx 0 5月 31 16:06 f2
```

```
xx@xx-virtual-machine:~$ umask 077
xx@xx-virtual-machine:~$ mkdir ~/temp/d2/d21
xx@xx-virtual-machine:~$ touch ~/temp/d2/f2
xx@xx-virtual-machine:~$ cd temp
xx@xx-virtual-machine:~/temp$ ls -l d2/f2 hello.c greeting d2
-rw----- 1 xx xx 0 5月 31 16:17 d2/f2
-rwxr-xr-x 1 xx xx 8304 5月 31 16:10 greeting
-rw-r--r-- 1 xx xx 62 5月 31 16:10 hello.c

d2:
总用量 4
drwx----- 2 xx xx 4096 5月 31 16:17 d21
-rw----- 1 xx xx 0 5月 31 16:17 f2
```

26. 删除~/temp 目录下的所有文件和目录。给出会话过程。

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

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

```
xx@xx-virtual-machine:~/temp$ mkdir d1 d2 d3
xx@xx-virtual-machine:~/temp$ cp ~/smallFile ./d1
xx@xx-virtual-machine:~/temp$ ls -il d1/smallFile
800387 -rw----- 1 xx xx 963 5月 31 16:25 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。

```
xx@xx-virtual-machine:~/temp$ cd d2
xx@xx-virtual-machine:~/temp/d2$ ln ../d1/smallFile newFile.hard
xx@xx-virtual-machine:~/temp/d2$ ls -il newFile.hard
800387 -rw----- 3 xx xx 963 5月 31 16:25 newFile.hard
xx@xx-virtual-machine:~/temp/d2$ ls -il ~/temp/d1/smallFile
800387 -rw----- 3 xx xx 963 5月 31 16:25 /home/xx/temp/d1/smallFile
```

是同一个文件，不是因为链接数一样而是因为有同一个 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。

```
xx@xx-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
xx@xx-virtual-machine:~/temp/d2$ chmod u-r ../d1/smallFile
xx@xx-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，可以推断出硬链接是一个指向文件索引节点的指针，它的改变能够引起指向文件的同步改变。

加回读取权限后，硬链接文件可以被读取。

将原文件删除后，显示硬链接文件依然会有输出。

说明硬链接相当于一个灾备系统，数据存放在两处，与复制不同的是两处之间存在同步机制，一处数据的改变会实时同步到另一处，另外一处数据如果被删除了，不会影响到另一处的数据。

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。

```
xx@xx-virtual-machine:~/temp/d2$ ln -s ../d1/smallFile smallFile.soft
xx@xx-virtual-machine:~/temp/d2$ ls -il smallFile.soft
798836 lrwxrwxrwx 1 xx xx 15 5月 31 16:51 smallFile.soft -> ../d1/smallFile
xx@xx-virtual-machine:~/temp/d2$ ls -il ../d1/smallFile
800387 -rw-r--r-- 3 xx xx 963 5月 31 16:25 ../d1/smallFile
```

两个文件的 inode 号并不一样，说明不是同一个文件。可以通过两个文件的大小不同判断，但是本质上的条件还是 inode 号。

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。

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

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


原本能够正常显示文件。

取消读权限后，系统提示权限不够根据以上能够推断出软链接是一个快捷方式，它指向了源文件，因为源文件的权限被修改，所以不能够使用快捷方式访问它。增加读权限后又能够显示。

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

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。

```
xx@xx-virtual-machine:~$ ps -l
F S  UID      PID     PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
0 S   1000    13808   13798  0  80   0  -  7425 wait  pts/0        00:00:00 bash
0 R   1000    13930   13808  0  80   0  -  9006 -      pts/0        00:00:00 ps

xx@xx-virtual-machine:~$ ps auxw|wc -l
283
xx@xx-virtual-machine:~$ ps -ef|grep bash
xx      13808   13798  0 16:04 pts/0    00:00:00 bash
xx      13937   13808  0 17:09 pts/0    00:00:00 grep --color=auto bash
xx@xx-virtual-machine:~$ ps -ef|grep 13798
xx      13798     859  0 16:04 ?          00:00:08 /usr/lib/gnome-terminal/gnome-terminal-server
xx      13808   13798  0 16:04 pts/0    00:00:00 bash
xx@xx-virtual-machine:~$ ps -ef|grep init
13935 pts/0    S+   0:00  \_ grep --color=auto init CLUTTER_IM_MODULE=xim LS_COLORS=rs=0:di=01
```

有 283 个进程。

	PID	父进程	PPID
init	13935	无	0
bash	13808	gnome-terminal-server	13798
ps	13930	bash	13808

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。

```
xx@xx-virtual-machine:~$ (sleep 60; echo "Time for lunch!"); &
[1] 13978
xx@xx-virtual-machine:~$ ps -l
F S  UID      PID     PPID  C PRI  NI ADDR SZ WCHAN  TTY          TIME CMD
0 S   1000    13808   13798  0  80   0  -  7425 wait  pts/0        00:00:00 bash
1 S   1000    13978   13808  0  80   0  -  7425 wait  pts/0        00:00:00 bash
0 S   1000    13979   13978  0  80   0  -  3643 hrtim  pts/0        00:00:00 sleep
0 R   1000    13980   13808  0  80   0  -  9006 -      pts/0        00:00:00 ps
xx@xx-virtual-machine:~$ Time for lunch!
[1]+  已完成                  ( sleep 60; echo "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。

```
xx@xx-virtual-machine:~$ date & uname -a & who & ps
[1] 13990
[2] 13991
2020年 05月 31日 星期日 17:46:11 CST
Linux xx-virtual-machine 5.3.0-28-generic #30~18.04.1-Ubuntu SMP Fri Jan 17 06:14:09 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
[3] 13992
xx      :0      2020-05-31 13:21 (:0)
  PID TTY          TIME CMD
 13808 pts/0        00:00:00 bash
 13993 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。

```
xx@xx-virtual-machine:~$ date && uname -a && who && ps
2020年 05月 31日 星期日 17:47:35 CST
Linux xx-virtual-machine 5.3.0-28-generic #30~18.04.1-Ubuntu SMP Fri Jan 17 06:14:09 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
xx      :0      2020-05-31 13:21 (:0)
  PID TTY          TIME CMD
 13808 pts/0        00:00:00 bash
 13997 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

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

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）

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

```
xx@xx-virtual-machine:~$ find ~ -name foobar 1> foobar.path 2> /dev/null
xx@xx-virtual-machine:~$ cat foobar.path
/home/xx/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



```
xx@xx-virtual-machine:~$ exec <fdata> fout
```



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。

```
xx@xx-virtual-machine:~$ ls -l |wc -clw
26      227     1259
```

26 行，227 个单词，1259 个字符。

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？

```
xx@xx-virtual-machine:/bin$ find -type f|wc -l
139
xx@xx-virtual-machine:/bin$ find -type d|wc -l
1
xx@xx-virtual-machine:/bin$ find -type l|wc -l
23
```

139 个普通文件，1 个目录文件，23 个符号链接文件。

三、 讨论、心得（必填）（10 分）

本次实验遇到的主要问题有 tree 命令安装时提示域名解析错误，上网查找后得知是因为网络问题下载失败。一开始想要手动安装，发现缺少 gcc 的编译环境，而 gcc 仍然无法通过命令行安装。通过换源用国内比较快的阿里云镜像，最终成功解决了这一问题，也顺带终于把 gcc 的环境安装成功，这也暴露了自己安装系统却疏于配置和使用的问题，以后要更加适应日常使用 Linux 系统和在 Linux 环境下编程。

```
xx@xx-virtual-machine:~$ sudo apt install tree
[sudo] xx 的密码:
正在读取软件包列表... 完成
正在分析软件包的依赖关系树
正在读取状态信息... 完成
下列软件包是自动安装的并且现在不需要了:
 fonts-liberation2 fonts-opensymbol gstreamer1.0-gtk3 libabw-0.1-1 libboost-date-time1.65.1
 libboost-filesystem1.65.1 libboost-iostreams1.65.1 libboost-locale1.65.1 libcdr-0.1-1
 libclucene-contribs1v5 libclucene-core1v5 libcmis-0.5-5v5 libcolamd2 libe-book-0.1-1 libeot0
 libepubgen-0.1-1 libetonyek-0.1-1 libexttextcat-2.0-0 libexttextcat-data libfreehand-0.1-1
 libgpgmepp6 liblangtag-common liblangtag1 libmhash2 libmispub-0.1-1 libmwaw-0.3-3
 libmythes-1.2-0 libneon27-gnutls libodfgen-0.1-1 liborcus-0.13-0 libpagemaker-0.0-0
 libraptor2-0 librasqal3 librdf0 librevenge-0.0-0 libsuitesparseconfig5 libvisio-0.1-1
 libwps-0.10-10 libwpg-0.3-3 libwps-0.4-4 libxmlsec1-nss libyajl2 lp-solve uno-libs3 ure
使用'sudo apt autoremove'来卸载它(它们)。
下列【新】软件包将被安装:
 tree
升级了 0 个软件包，新安装了 1 个软件包，要卸载 0 个软件包，有 0 个软件包未被升级。
需要下载 40.7 kB 的归档。
解压缩后会消耗 105 kB 的额外空间。
错误:1 http://cn.archive.ubuntu.com/ubuntu bionic/universe amd64 tree amd64 1.7.0-5
 无法解析域名“cn.archive.ubuntu.com”
E: 无法下载 http://cn.archive.ubuntu.com/ubuntu/pool/universe/t/tree/tree_1.7.0-5_amd64.deb 无法
解析域名“cn.archive.ubuntu.com”
E: 有几个软件包无法下载，要不运行 apt-get update 或者加上 --fix-missing 的选项再试试?
xx@xx-virtual-machine:~$ apt-get update
正在读取软件包列表... 完成
E: 无法打开锁文件 /var/lib/apt/lists/lock - open (13: 权限不够)
E: 无法对目录 /var/lib/apt/lists/ 加锁
W: 用 unlink 删除文件 /var/cache/apt/pkgcache.bin 出错 - RemoveCaches (13: 权限不够)
W: 用 unlink 删除文件 /var/cache/apt/srcpkgcache.bin 出错 - RemoveCaches (13: 权限不够)
xx@xx-virtual-machine:~$ sudo snap install tree # version 1.8.0+pkg-3fd6
error: cannot install "tree": Post https://api.snapcraft.io/v2/snaps/refresh: dial tcp: lookup
api.snapcraft.io: no such host
```

通过上网查找也对 tree 这一有用的命令有了更多了解，一些参数附在后面。

实验期间还遇到过虚拟机的整个网络崩溃，后来通过查找多种资料更改更新设置和 DNS 配置终于能直接通过主机的代理上网。虚拟机确实在电脑上运行的稳定性略差，从一开始的不习惯觉得配置麻烦，到最后也对 vim 编辑器和这种命令行交互修改配置的方式有了更深的了解，感受到了 Linux 系统的魅力。

Linux 命令行的各种快捷键与 Windows 界面的非常不同，比如 Ctrl+D 结束输入，Ctrl+Z 挂起一个进程，也需要进一步的记忆和熟练使用。而且一个空格键也会造成命令无法识别，在学习时要注意准确性。而且一种操作也可以通过不同的命令实现，要注意灵活运用。

此外还通过自己的操作复习了对各种权限的控制,对上课的时候不是非常明白的软硬链接的区别和实质也有了进一步的了解和认识。

附 tree 命令行参数:

- a 显示所有文件和目录。
- A 使用 ANSI 绘图字符显示树状图而非以 ASCII 字符组合。
- C 在文件和目录清单加上色彩,便于区分各种类型。
- d 显示目录名称而非内容。
- D 列出文件或目录的更改时间。
- f 在每个文件或目录之前,显示完整的相对路径名称。
- F 在执行文件,目录,Socket,符号连接,管道名称名称,各自加上 "*","/","=", "@","|"号。
- g 列出文件或目录的所属群组名称,没有对应的名称时,则显示群组识别码。
- i 不以阶梯状列出文件或目录名称。
- I 不显示符合范本样式的文件或目录名称。
- l 如遇到性质为符号连接的目录,直接列出该连接所指向的原始目录。
- n 不在文件和目录清单加上色彩。
- N 直接列出文件和目录名称,包括控制字符。
- p 列出权限标示。
- P 只显示符合范本样式的文件或目录名称。
- q 用 "?"号取代控制字符,列出文件和目录名称。
- s 列出文件或目录大小。
- t 用文件和目录的更改时间排序。
- u 列出文件或目录的拥有者名称,没有对应的名称时,则显示用户识别码。
- x 将范围局限在现行的文件系统中,若指定目录下的某些子目录,其存放于另一个文件系统上,则将该子目录予以排除在寻找范围外。