

实验1 Linux shell基本命令

实验目的

1. 初步了解Linux的命令格式；
2. 学会如何得到帮助信息；
3. 练习几个最常用的命令；
4. 练习用vi编辑器编辑文本文件；
5. 学习掌握Linux文件类型概念
6. 学习如何创建一个Linux目录的层次结构
7. 学习掌握有关绝对路径和相对路径概念，掌握主目录(home directory)、工作目录（当前目录）概念
8. 学习如何有效浏览Linux目录层次，有关文件内容类型和隐含文件
9. 学习有关文件属性，如何确定文件的大小
10. 学习如何显示文本文件的内容
11. 学习如何复制、追加、移动和删除文件，如何合并文件
12. 学习Linux的文件访问权限，用户的类型和文件访问权限的类型
13. 学习如何设置和改变一个文件的访问权限
14. 学习如何在文件或目录的创建时设置缺省访问权限
15. 学习理解硬链接、符号链接
16. 学习了解Linux进程的属性
17. 学习理解Linux的前台进程、后台进程及守护进程
18. 学习理解Linux命令的顺序执行和并发执行
19. 学习使用Linux的I/O重定向操作和管道操作
20. 学习使用data、cal、pwd、w、who、uname、cat、nl、head、tail、ls、cp、wc、mkdir、cd、mv、find、grep等命令
21. 学习使用id、chmod、chgrp、chown、umask、ln、ps、kill、fg、bg、job、top、pstree、crontab、at、|、<、>等命令

实验提示

略去

实验内容

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？重点研究Linux的软件许可存在哪些法律风险和应用风
险？（10分）

软件许可证：

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。通常作为安装过程的一部分嵌入到软件本身中。

常用的软件许可证：

软件可以按照版权的许可证，大致分成商业软件、开源软件（自由软件）和共享软件三大类：

- 商业许可证：保护商业利益
 - 商业软件的许可证一般明确许可方的版权归属、法定权利并比较完整地保证了软件开发者的权益。
 - 商业软件一般提供瑕疵担保，违约责任。
 - 商业软件许可证针对不同的环境和被许可人（用户）提供各种不同的文本。
 - 越来越多的商业软件许可证中包含了如安装、培训、运行支持、排错性维护和版本升级等技术服务内容。明确约定由谁负责提供服务，是否收费，即技术服务费用是否含在许可费用中，或者需要另行收费等。
 - 便于开发软件的企业生存，有效地保护了开发者的权益，最大程度保护了版权；但也向软件开发引入了更多商业的元素，可能会阻碍软件的普及以及开发的进步
 - 通过软件的使用许可费，商业软件企业得以收回成本、获得利益，并继续开发新产品，或不断提升软件的性能和品质，形成良性循环
- 开源许可证：源代码公开
 - 经OSIA批准
 - 源代码公开，但依然保留一部分版权
 - 降低了软件开发的成本
 - 盈利较为困难，开发者难以仅凭开发软件生存；可以让所有开发者一起完善代码，开发更多功能
- 共享许可证：先使用后付费

而常用的软件许可证，主要是对许可证的延续性、源代码的商业性、专利的保护性等方面有不同的要求，目前比较流行的几种有：

- GPL
 - 承认软件的版权
 - 可复制软件
 - 可修改软件
 - 可发布软件，但软件修改后的衍生软件仍然需要适用GPL
 - 可收取费用，但商业软件不能用GPL协议的代码。这是因为如果源代码采用了GPL协议，那么再发布产品的所有源代码也应该公开（所谓GPL的“传染性”），这种要求对于商业软件显然是不合理的
 - 衍生的协议有LGPL协议
- MIT
 - 宽松，除了保留版权外几乎没有限制，也因此大受欢迎
 - 可以使用、复制和修改软件
 - 必须附有MIT协议，商业软件也可以使用，因为其宽松，再发布产品只需要声明版权即可
- BSD
 - 十分宽松，对包含软件的重新分发施加的限制目前也是最小的
 - 如果再发布的产品中包含源代码，则在源代码中必须带有原来代码中的BSD协议
 - 不可以用开源代码的作者/机构名字和原来产品的名字做市场推广
 - BSD与其他许可证不同，不是必须要分发源代码，这可能使得发布的代码不具有连续性；但对商业应用来说是一个好的选择，也只需要在产品中声明版权即可
- Apache
 - 要求保留版权声明和免责声明
 - 鼓励代码共享和尊重原作者的著作权，同样允许代码修改，再发布
 - 对商业应用友好的许可
- Creative Commons
 - 不开放源代码授权，通常用于设计项目
 - 不可用于商业目的

- 必须以相同方式共享
- 禁止修改
- 这个许可证较为严格，也因此有许多修改版本；需要注意的是，该许可证可以要求商业使用者担责

Linux的软件许可存在的法律风险和应用风险：

一般来说，开源许可证存在的法律风险是比商业许可证的高的。对于大部分开源软件许可协议来说，其存在的法律风险主要分为**侵害以及许可限制**。前者主要是指将受保护的代码添加到了开源代码中，侵害了他人的权利；后者主要是指开源协议本身存在的限制条件所造成的关系侵害。

此处主要讨论Linux和大多数GNU软件采用的*GPL*。GPL的主要内容是：允许复制软件、修改软件、发布软件，但不允许修改后和衍生的代码做为闭源的商业软件发布和销售；要求只要一个软件中使用了GPL协议的产品，那么该软件也必须采用GPL协议，此为“传染性”。

其中存在的法律风险和应用风险：

- GPL的法律性质存在争议
 - 国外关于GPL的探讨主要集中在GPL是否具有强制力，以及GPL到底是合同还是许可证两个方面
 - 在我国现行法律下，GPL是合同关系。而我国目前尚无法律直接调整开源软件或自由软件许可行为，开源软件的风险防控问题尤为重要和复杂。
 - 最高人民法院对“独立程序”的认定方式对GPL开源软件风控问题有一定启示意义，即商业软件公司在使用包含GPL开源协议文件时，即使将其与商业软件产品放置在不同空间，也需考量二者在展示方式、所用技术、功能分工等方面关联程度，以进一步规避风险。目前我国法律也未对相关法律问题表示出明显的倾向或立场。
 - 在我国，相关问题目前还是个开放问题，相关立法并不成熟
- GPL的法律主体存在的风险
 - 对于许可方来说，法律的地域性和GPL未明确所适用的实体法和程序法，导致权利人所主张的权利是否受到法律保护或权利受侵害后是否能够及时得到救济在许可当时处于不确定状态
 - 对于许可方来说，对方存在违反GPL的行为举证难度较大
 - 对于被许可方来说，需要尤其注意GPL的“传染性”，要将相关的源代码开源并注意独立作品的认定
 - 对于被许可方来说，需要了解违反GPL而需承担的法律责任；违反GPL存在违约责任和侵权责任竞合问题

2. 通过对HarmonyOS或OpenHarmony与当前操作系统（Windows、Linux、i_OS、Android）的比较分析，如何构建鸿蒙的技术和生态，请你支招？（10分）

鸿蒙与其他几个操作系统的最主要区别是内核的区别：鸿蒙采用微内核，而其他几个操作系统采用宏内核或混合内核。微内核给鸿蒙带来的特点是：跨设备、低延时、更安全、全场景。

技术：

- 要搭建对开发者友好的开发工具，降低开发者的适应成本与开发成本，如继续完善统一IDE，开发强大的调试工具，IDE要适配各个操作系统，甚至可以开发将其他操作系统下的程序移植到鸿蒙的移植工具
- 出台相关政策吸引、激励开发者参与鸿蒙的开发，如给予开发者一定的经济鼓励，维护好开发者社区
- 考虑到微内核的功能较弱，很多功能需要由用户态完整，需要准备好相关的开发文档与必要的API，以积极的态度协助开发者解决开发中的问题
- 可以学习安卓的思想，将一部分代码开源，**便于开发者更好地开发**；但是也要借鉴iOS的设计思想，保证系统具有一定的封闭性以**确保安全性**，尤其是鸿蒙将会在很多IoT设备上运行，需要非常

注意安全问题

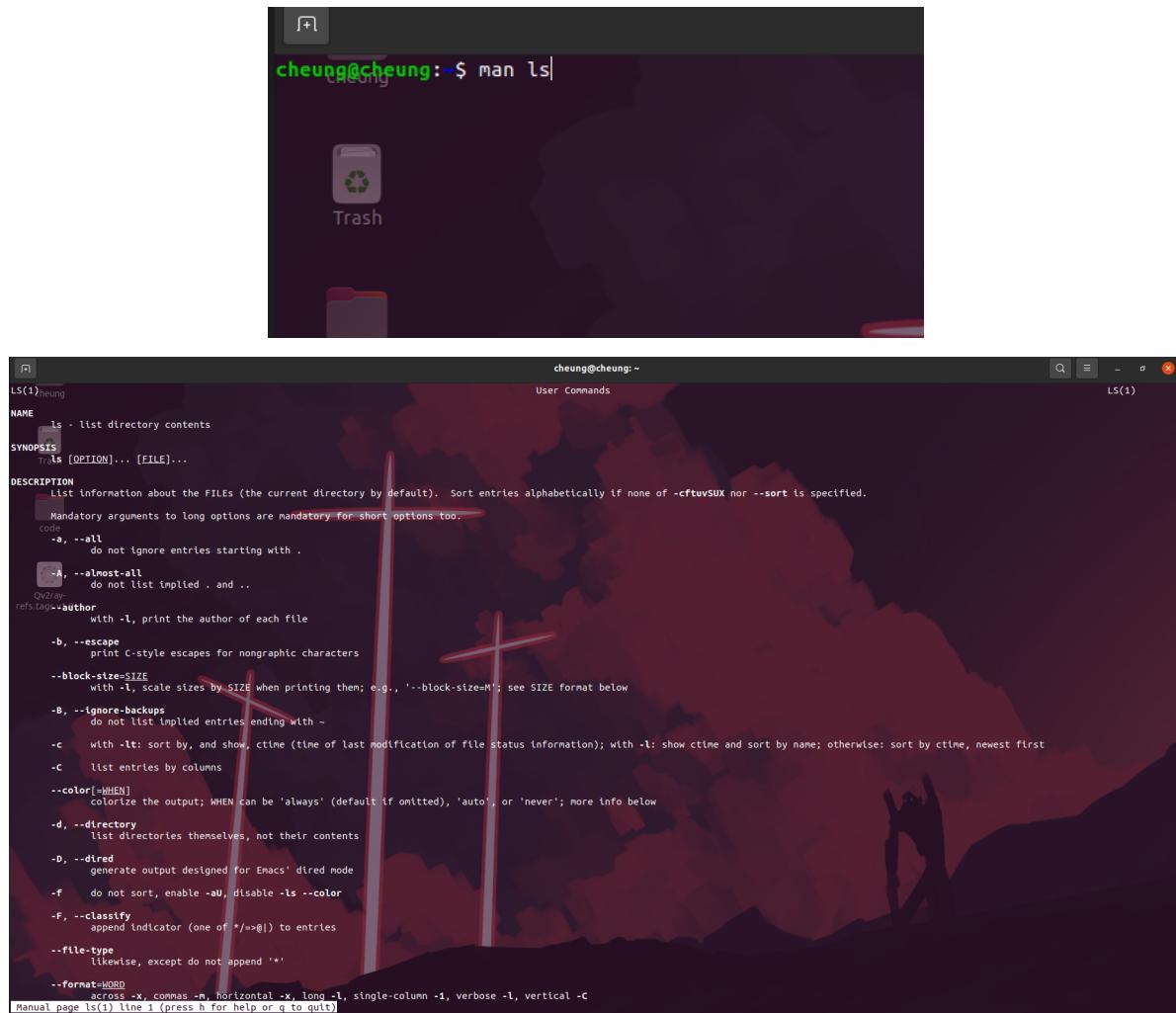
- 继续学习谷歌GMS+AOSP+Android的模式，完善HMS+OpenHarmony+HarmonyOS的模式，扩大技术版图

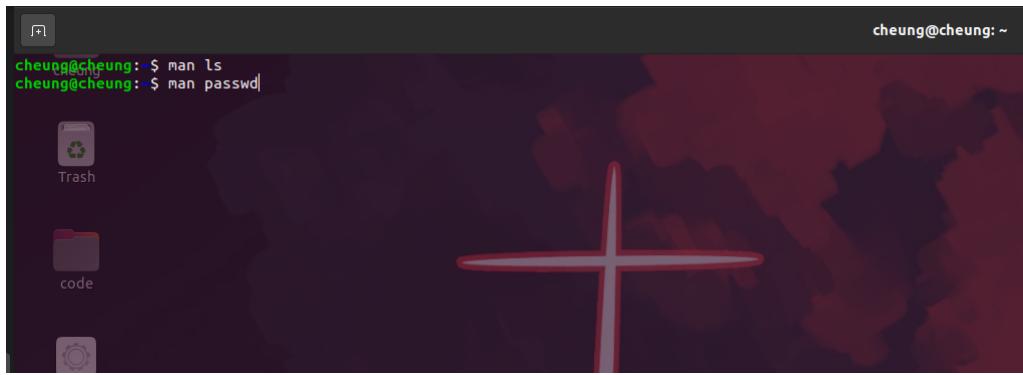
生态：

- 要突出鸿蒙跨终端协作的特点，华为自身还要大力开发各种智能终端或者说IoT设备，扩大鸿蒙的版图，使得鸿蒙支持的设备尽可能多；也可以结合自身的发展特点，利用已有业务，将鸿蒙用于电信设备的管理
- 拉拢更多的智能终端设备企业成为盟友，为盟友设计适合的适配方案，共同扩大鸿蒙的版图，使得鸿蒙支持的设备尽可能多
- 系统的UI设计以及操作逻辑上可以借鉴Windows和iOS等系统，降低用户的适应成本
- 借鉴苹果的思路，先对现有的架构（x86、arm等）与体系（Android、iOS、Windows等）做好适配优化，再逐步完善发展鸿蒙的生态

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。

截图如下：





```
cheung@cheung:~$ man ls
cheung@cheung:~$ man passwd|
```

```
cheung@cheung:~$ man passwd(1)
PASSWD(1)                               User Commands                               PASSWD(1)

NAME
    passwd - change user password

SYNOPSIS
    passwd [options] [LOGIN]

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

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

        After the password has been entered, password aging information is checked to see if the user is permitted to change the password at this time. If not, passwd refuses to change the password and exits.

        QVZayg
    The user is then prompted twice for a replacement password. The second entry is compared against the first and both are required to match in order for the password to be changed.

    Then, the password is tested for complexity. As a general guideline, passwords should consist of 6 to 8 characters including one or more characters from each of the following sets:
        • lower case alphabets
        • digits 0 thru 9
        • punctuation marks

    Care must be taken not to include the system default erase or kill characters. passwd will reject any password which is not suitably complex.

    Hints for user passwords
        The security of a password depends upon the strength of the encryption algorithm and the size of the key space. The legacy UNIX System encryption method is based on the NBS DES algorithm. More recent methods are now recommended (see ENCRYPT_METHOD). The size of the key space depends upon the randomness of the password which is selected.

        Compromises in password security normally result from careless password selection or handling. For this reason, you should not select a password which appears in a dictionary or which must be written down. The password should also not be a proper name, your license number, birth date, or street address. Any of these may be used as guesses to violate system security.

        You can find advice on how to choose a strong password on http://en.wikipedia.org/wik/Password_strength

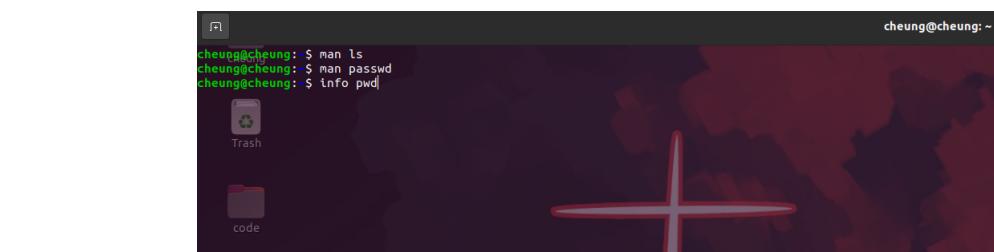
OPTIONS
    The options which apply to the passwd command are:
        -a, --all
            This option can be used only with -s and causes show status for all users.

        -d, --delete
            Delete a user's password (make it empty). This is a quick way to disable a password for an account. It will set the named account passwordless.

        -e, --expire
            Immediately expire an account's password. This in effect can force a user to change their password at the user's next login.

        -h, --help
            Display help message and exit.

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



```
cheung@cheung:~$ man ls
cheung@cheung:~$ man passwd
cheung@cheung:~$ info pwd|
```

```
cheung@cheung:~$
```

```
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 'code'
  --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
    components. Otherwise, fall back to default '-P' handling.
    QVZayg
  -P 'tags.v1.9...'
  --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.
    Due to shell aliases and built-in 'pwd' functions, using an unadorned
    'pwd' interactively or in a script may get you different functionality
    than that described here. Invoke it via 'env' (i.e., 'env pwd ...') to
    avoid interference from the shell.
    An exit status of zero indicates success, and a nonzero value
    indicates failure.

-----Info: (coreutils)pwd Invocation, 37 lines --All-----
```

```
Welcome to Info version 6.7. Type H for help, h for tutorial.
```

```
cheung@cheung:~$ man ls
cheung@cheung:~$ man passwd
cheung@cheung:~$ info pwd
cheung@cheung:~$ man passwd
cheung@cheung:~$ info pwd
cheung@cheung:~$ 
cheung@cheung:~$ who --help
cheung@cheung:~$ who --help
Usage: who [OPTION]... [FILE | ARG1 ARG2]
Print information about users who are currently logged in.

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

If FILE is not specified, use /var/run/utmp. /var/log/wtmp as FILE is common.
If ARG1 ARG2 given, -m presumed: 'am i' or 'mon likes' are usual.

GNU coreutils online help: <https://www.gnu.org/software/coreutils/>
Full documentation at: <https://www.gnu.org/software/coreutils/who>
or available locally via: info '(coreutils) who invocation'
cheung@cheung:~$
```

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？

cheung

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？

cheung

4) 你计算机系统的CPU名字是什么？

x86_64

```

cheung@cheung: ~
cheung@cheung: ~ whoami
cheung
cheung@cheung: ~ uname
Linux
cheung@cheung: ~ uname -n
cheung
cheung@cheung: ~ uname -p
x86_64
cheung@cheung: ~ uname --help
Usage: uname [OPTION]...
Print certain system information. With no OPTION, same as -s.

-a, --all          print all information, in the following order,
                   except omit -p and -l if unknown:
-s, --kernel-name print the kernel name
-n, --nodename    print the network node hostname
-r, --kernel-release print the kernel release
-v, --kernel-version print the kernel version
-mQV4--machine   print the machine hardware name
-pprocessor       print the processor type (non-portable)
-l, --hardware-platform print the hardware platform (non-portable)
-o, --operating-system print the operating system
--help            display this help and exit
--version         output version information and exit

GNU coreutils online help: <https://www.gnu.org/software/coreutils/>
Full documentation at: <https://www.gnu.org/software/coreutils/uname>
or available locally via: info '(coreutils) uname invocation'

cheung@cheung: ~ uname -s
Linux
cheung@cheung: ~ uname -m
x86_64
cheung@cheung: ~ uname -i
x86_64
cheung@cheung: ~ uname -o
GNU/Linux
cheung@cheung: ~ |

```

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

1) 给出你显示以上年份年历的命令

```

cal -y 4
cal -y 1752
cal -m 7 2012

```

2) 1752年有几天，为什么？

355天。见第四张图，通过命令 `cal -j` 显示每一天是该年的第几天。通过第二张图也可以看到，1752年的9月少了11天。这是因为即使每四年加入一次闰日，也无法完全消除纪年法与公转周期的差。而这个差一直累积到1752年，英国议会才决定抹除11天以消除这个差。

```

cheung@cheung: ~
cheung@cheung: ~ cal -help
Usage: cal [general options] [-jy] [[month] year]
        cal [general options] [-j] [-m month] [year]
        ncal -C [general options] [-jy] [[month] year]
        ncal -c [general options] [-jy] [-m month] [year]
        ncal [general options] [-bhjpwySM] [-H yyyy-mm-dd] [-s country_code] [[month] year]
        Tcal [general options] [-bhjeoSMy] [year]
General options: [-31] [-A months] [-B months] [-d yyyy-mm]
cheung@cheung: ~ cal -y 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
code 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
QV2r4y-
refstags: 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 31 25 26 27 28 29 30 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 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 2           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

```

```

cheung@cheung: $ cal -y 1752
Qv2ray- 1752
refs/tags/v1--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 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21
      5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 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           5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
      5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27
      12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 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           5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
      5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27
      12 13 14 15 16 17 18 19 20 21 22 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

```

```

cheung@cheung: $ cal -m 7 2012
July 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

```

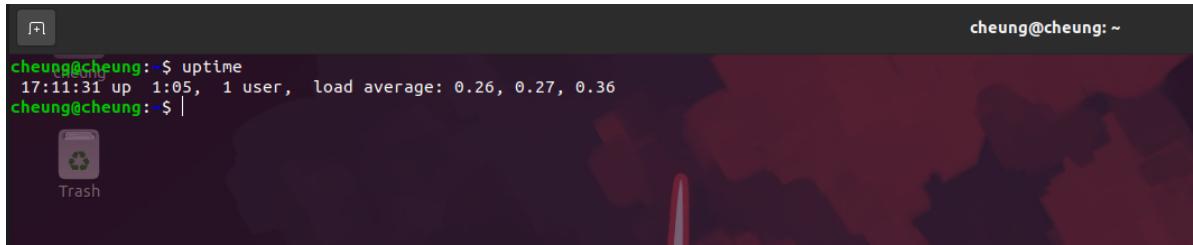
```

cheung@cheung: $ cal -j 12 1752
December 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

```

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。

由截图可以看出，只有一个登录用户，系统启动了1:05



```

cheung@cheung: ~
cheung@cheung: $ uptime
17:11:31 up 1:05, 1 user,  load average: 0.26, 0.27, 0.36
cheung@cheung: $

```

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：

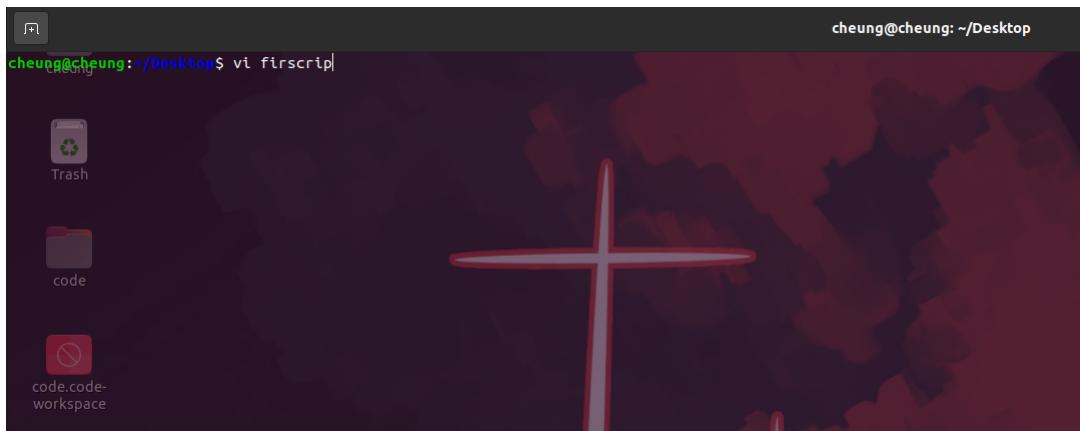
- 实验中发现cd命令不能通过man查阅，因此我用cd --help查阅。
- 由于字数较多，我只将每个命令的描述的第一句话放在表格中。

command	description	sample
touch	Update the access and modification times of each FILE to the current time.	<code>touch test</code>
cp	Copy SOURCE to DEST, or multiple SOURCE(s) to DIRECTORY.	<code>cp ./test ~/Desktop</code>
mv	Rename SOURCE to DEST, or move SOURCE(s) to DIRECTORY.	<code>mv ./test ~/Desktop</code>
rm	This manual page documents the GNU version of rm. rm removes each specified file. By default, it does not remove directories.	<code>rm ./test</code>
mkdir	Create the DIRECTORY(ies), if they do not already exist.	<code>mkdir testd</code>
who	Print information about users who are currently logged in.	<code>who -a</code>
ls	List information about the FILES (the current directory by default). Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.	<code>ls -al</code>
cd	Change the shell working directory. (--help)	<code>cd ~</code>
pwd	Print the full filename of the current working directory.	<code>pwd</code>
open	The open() system call opens the file specified by pathname. If the specified file does not exist, it may optionally (if O_CREAT is specified in flags) be created by open().	
read	read() attempts to read up to count bytes from file descriptor fd into the buffer starting at buf.	<code>read input</code>
write	The write utility allows you to communicate with other users, by copying lines from your terminal to theirs.	<code>write testuser</code>
close	close() closes a file descriptor, so that it no longer refers to any file and may be reused. Any record locks (see fcntl(2)) held on the file it was associated with, and owned by the process, are removed (regardless of the file descriptor that was used to obtain the lock).	
pipe	pipe() creates a pipe, a unidirectional data channel that can be used for interprocess communication. The array pipefd is used to return two file descriptors referring to the ends of the pipe. pipefd[0] refers to the read end of the pipe. pipefd[1] refers to the write end of the pipe. Data written to the write end of the pipe is buffered by the kernel until it is read from the read end of the pipe. For further details, see pipe(7).	
socket	socket() creates an endpoint for communication and returns a file descriptor that refers to that endpoint. The file descriptor returned by a successful call will be the lowest-numbered file descriptor not currently open for the process.	

command	description	sample
mkfifo	Create named pipes (FIFOs) with the given NAMES.	mkfifo pipetest
system	The system() library function uses fork(2) to create a child process that executes the shell command specified in command using execl(3)	
printf	Print ARGUMENT(s) according to FORMAT, or execute according to OPTION	printf "%d" 2

8. vi编辑器的使用：

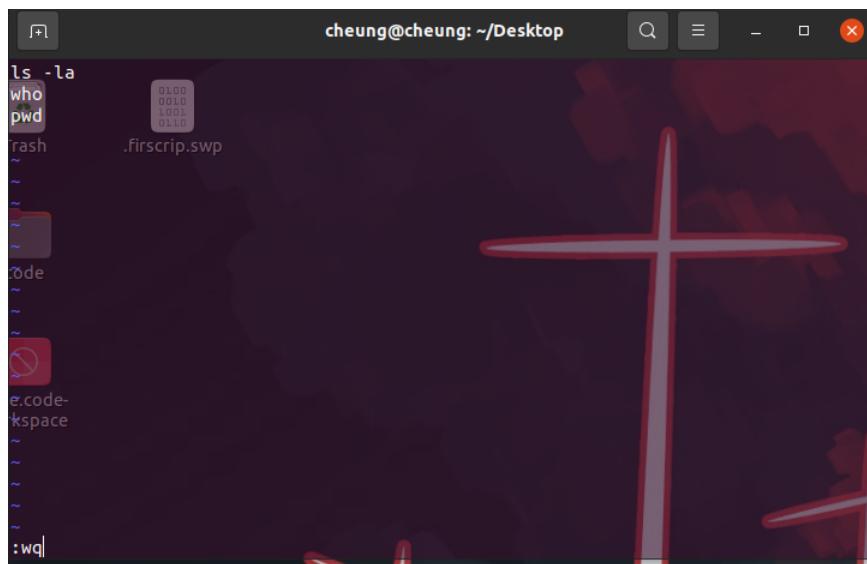
1) 在shell提示符下，输入vi firscript并按<Enter>键。vi的界面将出现在显示屏上；



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。



5) 在按<Esc>键；

6) 再输入:wq，并按<Enter>键；

```
ls -la
who
pwd
.firscript.swp
~
~
~
code
~
~
e.code-
workspace
~
~
~
~
:wq|
```

7) 在shell提示符下，输入**bash ./firscript**并按<Enter>键；

```
cheung@cheung:~/Desktop$ vi firscript
cheung@cheung:~/Desktop$ bash ./firscript
total 20
drwxr-xr-x 3 cheung cheung 4096 7月 15 14:45 .
drwxr-xr-x 21 cheung cheung 4096 7月 11 23:59 ..
drwxrwxr-x 5 cheung cheung 4096 7月 14 18:48 code
-rw-rw-r-- 1 cheung cheung 87 7月 14 18:47 code.code-workspace
-rw-rw-r-- 1 cheung cheung 16 7月 15 14:45 firscript
cheung :0          2021-07-15 14:09 (:0)
/home/cheung/Desktop
cheung@cheung:~/Desktop$ |
code
```

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？

从结果可以看出，当前工作目录有两个文件和一个文件夹。

文件夹名为code，占用4096B；文件code.code-workspace占用87B；文件firscript占用16B。

没有其他人在使用计算机系统，只有我在使用的cheung。

当前的工作目录为 /home/cheung/Desktop。

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。

如截图所示，第一个截图为会话，第二个为结果验证。

```
cheung@cheung:~$ pwd
/home/cheung
cheung@cheung:~$ mkdir temp
cheung@cheung:~$ mkdir -p personal/{funstuff,taxes}
cheung@cheung:~$ mkdir -p professional/societies/{ieee,acm}
cheung@cheung:~$ mkdir -p professional/courses/{major,general}
cheung@cheung:~$ cd professional/courses/major/
cheung@cheung:~/professional/courses/major$ mkdir cs{213,381,475}
cheung@cheung:~/professional/courses/major$ mkdir cs381/{notes,labs,programs}
cheung@cheung:~/professional/courses/major$ cd
```

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。

```
$ echo $HOME  
/home/cheung/
```

2) acm目录的绝对路径是什么?

```
$ cd ../../societies/acm  
$ pwd  
/home/cheung/professional/societies/acm
```

3) 给出acm目录的两个相对路径。

```
~/professional/courses: ..../societies/acm  
~/professional: ./societies/acm
```

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

```
$ cd major/cs381/labs/  
$ pwd
```

给出获得你的主目录三个不同的命令。

```
$ echo $HOME  
$ cd ~; pwd  
$ cat /etc/passwd | grep "cheung"
```

```

cheung@cheung:~$ cd professional/courses/
cheung@cheung:~/professional/courses$ echo $HOME
/home/cheung
cheung@cheung:~/professional/courses$ cd ..;/societies/acm/
cheung@cheung:~/professional/societies/acm$ pwd
/home/cheung/professional/societies/acm
cheung@cheung:~/professional/societies/acm$ cd ~/professional/courses/
cheung@cheung:~/professional/courses$ cd major/cs381/labs/
cheung@cheung:~/professional/courses/major/cs381/labs$ pwd
/home/cheung/professional/courses/major/cs381/labs
cheung@cheung:~/professional/courses/major/cs381/labs$ echo $HOME
/home/cheung
cheung@cheung:~/professional/courses/major/cs381/labs$ cd ~;pwd
/home/cheung
cheung@cheung:~$ cat /etc/passwd | grep "cheung"
cheung:x:1000:1000:cheung,,,:/home/cheung:/bin/bash
cheung@cheung:~$ |

```

code.code-workspace

最后一条命令输出中的 `/home/cheung` 即为文件中保存的主目录路径

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？

```

cheung@cheung:~$ cd
cheung@cheung:~$ ls -dal .
drwxr-xr-x 24 cheung cheung 4096 7月 15 14:56 .
drwxr-xr-x 3 root root 4096 7月 10 05:52 ..
-rw----- 1 cheung cheung 9375 7月 15 14:23 .bash_history
-rw-r--r-- 1 cheung cheung 220 7月 16 05:52 .bash_logout
-rw-r--r-- 1 cheung cheung 3771 7月 16 05:52 .bashrc
drwxr-xr-x 18 cheung cheung 4096 7月 15 14:16 .cache
drwxr-xr-x 20 cheung cheung 4096 7月 15 15:23 .config
drwxrwxr-x 3 cheung cheung 4096 7月 11 20:15 .deepinwine
-rw-rw-r-- 1 cheung cheung 48 7月 11 23:59 .gitconfig
drwxr-xr-x 3 cheung cheung 4096 7月 14 17:42 .gnugc
drwxr-xr-x 3 cheung cheung 4096 7月 10 14:05 .local
drwxr-xr-x 5 cheung cheung 4096 7月 10 14:06 .mozilla
drwxr-xr-x 3 cheung cheung 4096 7月 11 11:40 .pki
-rw-r--r-- 1 cheung cheung 887 7月 10 05:52 .profile
drwxr-xr-x 2 cheung cheung 4096 7月 12 00:05 .ssh
-rw-r--r-- 1 cheung cheung 0 7月 9 14:10 .sudo_as_admin_successful
drwxrwxr-x 3 cheung cheung 4096 7月 11 11:39 .vscode
-rw-rw-r-- 1 cheung cheung 211 7月 12 00:15 .wget-hsts
-rw-rw-r-- 1 cheung cheung 131 7月 9 15:53 .xinputrc
cheung@cheung:~$ ls -al
total 132
drwxr-xr-x 24 cheung cheung 4096 7月 15 14:56 .
drwxr-xr-x 3 root root 4096 7月 10 05:52 ..
-rw----- 1 cheung cheung 9375 7月 15 14:23 .bash_history
-rw-r--r-- 1 cheung cheung 220 7月 16 05:52 .bash_logout
-rw-r--r-- 1 cheung cheung 3771 7月 16 05:52 .bashrc
drwxr-xr-x 18 cheung cheung 4096 7月 15 14:16 .cache
drwxr-xr-x 20 cheung cheung 4096 7月 15 15:23 .config
drwxrwxr-x 3 cheung cheung 4096 7月 11 20:15 .deepinwine
drwxr-xr-x 3 cheung cheung 4096 7月 15 14:45 .desktop
drwxr-xr-x 3 cheung cheung 4096 7月 11 20:15 .documents
drwxr-xr-x 2 cheung cheung 4096 7月 15 14:13 .downloads
-rw-rw-r-- 1 cheung cheung 48 7月 11 23:59 .gitconfig
drwxrwxr-x 3 cheung cheung 4096 7月 11 20:38 .gitdeepin
drwxr-xr-x 3 cheung cheung 4096 7月 14 17:42 .gnugc
drwxr-xr-x 3 cheung cheung 4096 7月 10 14:05 .local
drwxr-xr-x 5 cheung cheung 4096 7月 10 14:06 .mozilla
drwxr-xr-x 2 cheung cheung 4096 7月 10 14:05 .music
drwxrwxr-x 4 cheung cheung 4096 7月 15 14:55 .personal
drwxr-xr-x 2 cheung cheung 4096 7月 10 14:05 .Pictures
drwxr-xr-x 3 cheung cheung 4096 7月 11 11:40 .pki
drwxrwxr-x 4 cheung cheung 4096 7月 15 14:57 .professional
-rw-r--r-- 1 cheung cheung 887 7月 10 05:52 .profile
drwxr-xr-x 2 cheung cheung 4096 7月 10 14:05 .public
drwxr-xr-x 3 cheung cheung 4096 7月 9 14:44 .snap
drwxr-xr-x 2 cheung cheung 4096 7月 12 00:05 .ssh
-rw-r--r-- 1 cheung cheung 0 7月 9 14:10 .sudo_as_admin_successful
drwxrwxr-x 2 cheung cheung 4096 7月 15 14:55 .tmp
drwxr-xr-x 2 cheung cheung 4096 7月 10 14:05 .templates
drwxr-xr-x 2 cheung cheung 4096 7月 10 14:05 .Videos
drwxrwxr-x 3 cheung cheung 4096 7月 11 11:39 .vscode
-rw-rw-r-- 1 cheung cheung 211 7月 12 00:15 .wget-hsts
-rw-rw-r-- 1 cheung cheung 131 7月 9 15:53 .xinputrc
cheung@cheung:~$ 

```

由截图可见，除去目录本身与父目录，隐藏文件（不包括隐藏目录）有8个，分别是 `.bash_history`, `.bash_logout`, `.bashrc`, `.gitconfig`, `.profile`, `.sudo_as_admin_successful`, `.wget-hsts` 和 `.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。

```
cheung@cheung: ~
```

```
cheung@cheung: $ stat / | grep "Inode"
Device: 807h/2055d      Inode: 2          Links: 20
cheung@cheung: $ stat ~ | grep "Inode"
Device: 807h/2055d      Inode: 2763600    Links: 24
cheung@cheung: $ stat ~/temp | grep "Inode"
Device: 807h/2055d      Inode: 2893501    Links: 2
cheung@cheung: $ stat ~/professional | grep "Inode"
Device: 807h/2055d      Inode: 2893505    Links: 4
cheung@cheung: $ stat ~/personal | grep "Inode"
Device: 807h/2055d      Inode: 2893502    Links: 4
cheung@cheung: $ |
```

code

(No)

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命回答这个问题，给出会话过程。

regular file

```
cheung@cheung: ~/professional/courses/major/cs381/labs
```

```
cheung@cheung: $ cd ./professional/courses/major/cs381/labs/
cheung@cheung: ~/professional/courses/major/cs381/labs$ gedit lab1
```

Trash

code

code.code-workspace

Firscript

```
cheung@cheung: ~/professional/courses/major/cs381/labs
```

```
cheung@cheung: $ cd ./professional/courses/major/cs381/labs/
cheung@cheung: ~/professional/courses/major/cs381/labs$ gedit lab1
cheung@cheung: ~/professional/courses/major/cs381/labs$ file lab1
lab1: ASCII text
cheung@cheung: ~/professional/courses/major/cs381/labs$ stat lab1
  File: 'lab1'
  Size: 152          Blocks: 8          IO Block: 4096   regular file
Device: 807h/2055d      Inode: 2893518      Links: 1
Access: (0664/-rw-rw-r--)
Access: ( 1000/ cheung)  Uid: ( 1000/ cheung)
Access: 2021-07-15 15:46:12.199467317 +0800
Modify: 2021-07-15 15:45:29.512390364 +0800
Change: 2021-07-15 15:45:29.512390364 +0800
 Birthdate:
cheung@cheung: ~/professional/courses/major/cs381/labs$ ls -l
total 4
-rw-rw-r-- 1 cheung cheung 152 7月 15 15:45 lab1
cheung@cheung: ~/professional/courses/major/cs381/labs$ |
```

code-code-workspace

2) lab1文件内容的类型，用Linux命回答这个问题，给出会话过程。

ASCII text

```

cheung@cheung: ~/professional/courses/major/cs381/labs
cheung@cheung: $ cd ./professional/courses/major/cs381/labs/
cheung@cheung:~/professional/courses/major/cs381/labs$ gedit lab1
cheung@cheung:~/professional/courses/major/cs381/labs$ file lab1
lab1: ASCII text
cheung@cheung:~/professional/courses/major/cs381/labs$ stat lab1
  File: lab1
  Size: 152          Blocks: 8          IO Block: 4096   regular file
Device: 807h/2055d  Inode: 2893518      Links: 1
Access: (0664/-rw-rw-r--)  Uid: ( 1000/  cheung)   Gid: ( 1000/  cheung)
Access: 2021-07-15 15:46:12.199467317 +0800
Modify: 2021-07-15 15:45:29.512390364 +0800
Change: 2021-07-15 15:45:29.512390364 +0800
 Birth:de
cheung@cheung:~/professional/courses/major/cs381/labs$ ls -l
total 4
-rw-rw-r-- 1 cheung cheung 152 7月 15 15:45 lab1
cheung@cheung:~/professional/courses/major/cs381/labs$ |
code.code-
workspace

```

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。

```

cheung@cheung: ~$ cd /usr/include/
cheung@cheung: /usr/include$ ls -l t*.h
-rw-r--r-- 1 root root 3786 12月 16 2020 tar.h
-rw-r--r-- 1 root root 3471 2月 26 2020 termcap.h
-rw-r--r-- 1 root root 9096 2月 26 2020 term_entry.h
-rw-r--r-- 1 root root 41369 2月 26 2020 term.h
-rw-r--r-- 1 root root 214 12月 16 2020 termio.h
-rw-r--r-- 1 root root 3599 12月 16 2020 termios.h
-rw-r--r-- 1 root root 37419 12月 16 2020 tgmath.h
-rw-r--r-- 1 root root 16024 12月 16 2020 thread_db.h
-rw-r--r-- 1 root root 6661 12月 16 2020 threads.h
-rw-r--r-- 1 root root 14830 2月 26 2020 tic.h
-rw-r--r-- 1 root root 10276 12月 16 2020 time.h
-rw-r--r-- 1 root root 2494 12月 16 2020 ttyent.h
cheung@cheung: /usr/include$ |
code.code-
workspace

```

15. 创建几个大小不等的文本文件，供本实验和后面几个实验用：用man cat > mediumFile 命令创建中等大小的文件；用man bash >largeFile 命令创建一个大文件；再创建一个名字为smallFile关于学生数据的小文件

```

cheung@cheung: ~$ cd ~/temp/
cheung@cheung: ~/temp$ man cat>mediumFile
cheung@cheung: ~/temp$ man bash>largeFile
cheung@cheung: ~/temp$ gedit smallFile

```

Open	Save
1 FirstName LastName Major GPA Email Phone	
2 John Doe ECE 3.54 doe@jd.home.org 111.222.3333	
3 James Davis ECE 3.71 davis@jd.work.org 111.222.1111	
4 Al Davis CS 2.63 davis@lakers.org 111.222.2222	
5 Ahmad Rashid MBA 3.04 ahmad@mba.org 111.222.4444	
6 Sam Chu ECE 3.68 chugsam.ab.com 111.222.5555	
7 Arun Roy SS 3.86 roy@ss.arts.edu 111.222.8888	
8 Rick Marsh CS 2.34 marsh@ab.org 111.222.6666	
9 James Adam CS 2.77 jadam@ab.org 111.222.7777	
10 Art Pohm ECE 4.00 pohm@ap.a.org 111.222.9999	
11 John Clark ECE 2.68 clark@xyz.ab.com 111.111.5555	
12 Nabeel Ali EE 3.56 ali@ee.eng.edu 111.111.8888	
13 Tom Nelson ECE 3.81 nelson@tn.abc.org 111.111.6666	
14 Pat King SS 3.77 king@pk.xyz.org 111.111.7777	
15 Jake Zulu CS 3.00 zulu@jz.sa.org 111.111.9999	
16 John Lee EE 3.64 jlee@j.lee.com 111.111.2222	
17 Sunil Raj ECE 3.86 raj@sr.cs.edu 111.111.3333	
18 Charles Right EECS 3.31 right@cr.abc.edu 111.111.4444	
19 Diane Rover ECE 3.87 rover@dr.xyz.edu 111.111.5555	
20 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。

```

cheung@cheung:~/temp$ head -n 12 largeFile
Bash(1)                                General Commands Manual                                BASH(1)
NAME
    bash - GNU Bourne-Again SHell
SYNOPSIS
    bash [options] [command_string | file]
COPYRIGHT
    Bash is Copyright (C) 1989-2018 by the Free Software Foundation, Inc.

DESCRIPTION
cheung@cheung:~/temp$ tail -n 6 smallFile
John      Lee      ECE   3.64  jlee@j.lee.com      111.111.2222
Sunil     Raj      ECE   3.86  rajgsr.cs.edu      111.111.3333
Charles   Right   EECS  3.31  rightgcr.abc.edu  111.111.4444
Diane    Rover    ECE   3.87  rover@dr.xyz.edu  111.111.5555
Aziz     Inan    EECS  3.75  atnan@at.abc.edu  111.111.1111
cheung@cheung:~/temp$ tail -n 6 smallFile
Sam       Chu      ECE   3.68  chugsan.ab.com    111.222.5555
Arun     Roy      SS    3.86  roysgs.arts.edu   111.222.8888
Rick     Marsh   CS    3.66  marshba.arts.org  111.222.6666
James   Adam    ECE   2.77  jadams@cs.abc.org  111.222.7777
Art      Pohn    ECE   4.00  pohn@ap.s.org    111.222.9999
John     Clark   ECE   2.68  clarkgxyz.ab.com  111.111.5555
Nabeel   Alt     EE   3.156  nabeel@eng.edu   111.111.8888
Tom      Nelson ECE   3.81  allgee@tn.abc.org  111.111.6666
Pat      King    SS    3.69  kingpat@cs.abc.org  111.111.7777
Jesse   Zulu   CS    3.69  jzeluzulu@cs.abc.org  111.111.9999
John     Lee      ECE   3.64  jlee@j.lee.com      111.111.2222
Sunil     Raj      ECE   3.86  rajgsr.cs.edu      111.111.3333
Charles   Right   EECS  3.31  rightgcr.abc.edu  111.111.4444
Diane    Rover    ECE   3.87  rover@dr.xyz.edu  111.111.5555
Aziz     Inan    EECS  3.75  atnan@at.abc.edu  111.111.1111
cheung@cheung:~/temp$ |

```

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？

问题一：

用 touch -m 命令即可。inode号分别为2893525和2893524

```

cheung@cheung:~/temp$ cp smallFile dataFile
cheung@cheung:~/temp$ ls -l
total 336
-rw-rw-r-- 1 cheung cheung 1004 7月 15 16:27 dataFile
-rw-rw-r-- 1 cheung cheung 327764 7月 15 16:13 largeFile
-rw-rw-r-- 1 cheung cheung 2179 7月 15 16:13 mediumFile
-rw-rw-r-- 1 cheung cheung 1004 7月 15 16:24 smallFile
cheung@cheung:~/temp$ stat dataFile
  File: dataFile
  Size: 1004          Blocks: 8          IO Block: 4096   regular file
Device: 807h/2055d  Inode: 2893525        Links: 1
Access: (0664/-rw-rw-r--)  Uid: ( 1000/  cheung)  Gid: ( 1000/  cheung)
Access: 2021-07-15 16:27:54.080535748 +0800
Modify: 2021-07-15 16:27:54.080535748 +0800
Change: 2021-07-15 16:27:54.080535748 +0800
 Birth: -
cheung@cheung:~/temp$ stat smallFile
  File: smallFile
  Size: 1004          Blocks: 8          IO Block: 4096   regular file
Device: 807h/2055d  Inode: 2893524        Links: 1
Access: (0664/-rw-rw-r--)  Uid: ( 1000/  cheung)  Gid: ( 1000/  cheung)
Access: 2021-07-15 16:24:40.923841294 +0800
Modify: 2021-07-15 16:24:33.259972555 +0800
Change: 2021-07-15 16:24:33.315971595 +0800
 Birth: -
cheung@cheung:~/temp$ |

```

```

cheung@cheung:~/temp$ touch -m -d"2021-07-15 16:24:33.259972555 +0800" dataFile
cheung@cheung:~/temp$ stat dataFile
  File: dataFile
  Size: 1004          Blocks: 8          IO Block: 4096   regular file
Device: 807h/2055d  Inode: 2893525        Links: 1
Access: (0664/-rw-rw-r--)  Uid: ( 1000/  cheung)  Gid: ( 1000/  cheung)
Access: 2021-07-15 16:27:54.080535748 +0800
Modify: 2021-07-15 16:24:33.259972555 +0800
Change: 2021-07-15 16:24:33.1902198925 +0800
 Birth: -
cheung@cheung:~/temp$ |

```

问题二：

newDataFile的inode为2893525，与dataFile的相同。这是因为移动（重命名）时，文件本身没有发生变化，则inode不变，只是指向该inode的文件名变化了。

```
cheung@cheung:~/temp$ mv dataFile newDataFile
cheung@cheung:~/temp$ ls
'2021-07-15 16:24:33' '2021-07-15 16:24:33.259972555 +0800' largeFile mediumFile newDataFile smallFile
cheung@cheung:~/temp$ stat newDataFile
  File: newDataFile
  Size: 1004          Blocks: 8          IO Block: 4096   regular file
Device: 807h/2055d      Inode: 2893525      Links: 1
Access: (0664/-rw-rw-r--)  Uid: ( 1000/ cheung)  Gid: ( 1000/ cheung)
Access: 2021-07-15 16:27:54.080535748 +0800
Modify: 2021-07-15 16:24:33.259972555 +0800
Change: 2021-07-15 16:35:18.196240813 +0800
 Birth: -
cheung@cheung:~/temp$ |
```

问题三：

文件/tmp/newDataFile的inode号是2893525，与newDataFile的相同。这是因为移动时，文件本身没有发生变化，则inode不变，只是指向该inode的文件名变化了。

此外，我还额外做了一步拷贝的实验。发现拷贝出来的新文件的inode号不同了，系统为其新建了一个inode以标识一个新文件。

```
cheung@cheung:~/temp$ mv newDataFile /tmp/
cheung@cheung:~/temp$ cd /tmp/
cheung@cheung:/tmp$ stat newDataFile
  File: newDataFile
  Size: 1004          Blocks: 8          IO Block: 4096   regular file
Device: 807h/2055d      Inode: 2893525      Links: 1
Access: (0664/-rw-rw-r--)  Uid: ( 1000/ cheung)  Gid: ( 1000/ cheung)
Access: 2021-07-15 16:27:54.080535748 +0800
Modify: 2021-07-15 16:24:33.259972555 +0800
Change: 2021-07-15 16:37:16.645070458 +0800
 Birth: -
cheung@cheung:/tmp$ cp newDataFile ~/Desktop/
cheung@cheung:/tmp$ stat ~/Desktop/newDataFile
  File: /home/cheung/Desktop/newDataFile
  Size: 1004          Blocks: 8          IO Block: 4096   regular file
Device: 807h/2055d      Inode: 2778390      Links: 1
Access: (0664/-rw-rw-r--)  Uid: ( 1000/ cheung)  Gid: ( 1000/ cheung)
Access: 2021-07-15 16:39:01.289127573 +0800
Modify: 2021-07-15 16:39:01.289127573 +0800
Change: 2021-07-15 16:39:01.289127573 +0800
 Birth: -
cheung@cheung:/tmp$ |
```

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。

用ls -l 或者 wc

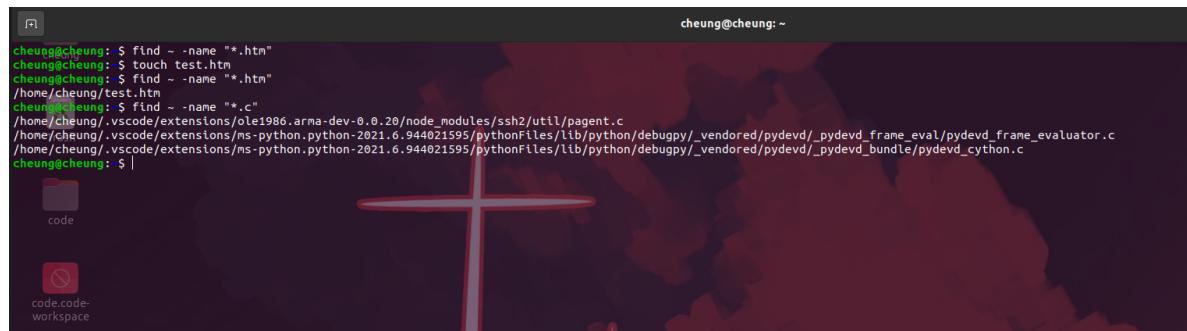
```
cheung@cheung:/tmp$ cd ~/temp/
cheung@cheung:~/temp$ wc smallFile mediumFile largeFile /tmp/newDataFile
 20    120    1004 smallFile
 69    215    2179 mediumFile
3443  48116  327764 largeFile
 20    120    1004 /tmp/newDataFile
3552 48571 331951 total
cheung@cheung:~/temp$ ls -l
total 332
-rw-rw-r-- 1 cheung cheung 327764 7月 15 16:13 largeFile
-rw-rw-r-- 1 cheung cheung  2179 7月 15 16:13 mediumFile
-rw-rw-r-- 1 cheung cheung   1004 7月 15 16:24 smallFile
```

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。

结果较多，只放一部分

```
cheung@cheung: ~
cheung@cheung: $ find ~ -name "*.html"
/home/cheung/.config/google-chrome/Default/Extensions/fmflejelblhoaflnjajjjjkkgbefpn/2.10.4_0/index.html
/home/cheung/.config/google-chrome/Default/Extensions/fmflejelblhoaflnjajjjjkkgbefpn/2.10.4_0/options.html
/home/cheung/.config/google-chrome/Default/Extensions/lmpkjlgappg/kkflleppmihakdmaab/0.9.8.74_0/background.html
/home/cheung/.config/google-chrome/Default/Extensions/lmpkjlgappg/kkflleppmihakdmaab/0.9.8.74_0/options.html
/home/cheung/.config/google-chrome/Default/Extensions/rrbanmkthbegcgfonabangnfafrbdikn/0.4.2_0/popup.html
/home/cheung/.config/google-chrome/Default/Extensions/dhdgffkkekbnkfjojejmpbldmpofkfo/4.13_0/ask.html
/home/cheung/.config/google-chrome/Default/Extensions/dhdgffkkekbnkfjojejmpbldmpofkfo/4.13_0/background.html
/home/cheung/.config/google-chrome/Default/Extensions/dhdgffkkekbnkfjojejmpbldmpofkfo/4.13_0/action.html
/home/cheung/.config/google-chrome/Default/Extensions/dhdgffkkekbnkfjojejmpbldmpofkfo/4.13_0/userscript.html
/home/cheung/.config/google-chrome/Default/Extensions/dhdgffkkekbnkfjojejmpbldmpofkfo/4.13_0/options.html
/home/cheung/.config/google-chrome/Default/Extensions/aapocclcgogkrnckokdprmnonfgoek/0.10_0/main.html
/home/cheung/.config/google-chrome/Default/Extensions/nclcbchpeplkfjdhnoepgdnmjkkci/2.0.6_0/about.html
/home/cheung/.config/google-chrome/Default/Extensions/nclcbchpeplkfjdhnoepgdnmjkkci/2.0.6_0/setting.html
/home/cheung/.config/google-chrome/Default/Extensions/nclcbchpeplkfjdhnoepgdnmjkkci/2.0.6_0/select_line.html
/home/cheung/.config/google-chrome/Default/Extensions/nclcbchpeplkfjdhnoepgdnmjkkci/2.0.6_0/buyvip.html
/home/cheung/.config/google-chrome/Default/Extensions/nclcbchpeplkfjdhnoepgdnmjkkci/2.0.6_0/login.html
/home/cheung/.config/google-chrome/Default/Extensions/nclcbchpeplkfjdhnoepgdnmjkkci/2.0.6_0/welcome.html
/home/cheung/.config/google-chrome/Default/Extensions/nclcbchpeplkfjdhnoepgdnmjkkci/2.0.6_0/helper/open_tracker.html
/home/cheung/.config/google-chrome/Default/Extensions/nclcbchpeplkfjdhnoepgdnmjkkci/2.0.6_0/helper/systemdomain.html
/home/cheung/.config/google-chrome/Default/Extensions/nclcbchpeplkfjdhnoepgdnmjkkci/2.0.6_0/helper/only_vip_site.html
/home/cheung/.config/google-chrome/Default/Extensions/nclcbchpeplkfjdhnoepgdnmjkkci/2.0.6_0/helper/proxydomain.html
/home/cheung/.config/google-chrome/Default/Extensions/nclcbchpeplkfjdhnoepgdnmjkkci/2.0.6_0/helper/fc_tips.html
/home/cheung/.config/google-chrome/Default/Extensions/nclcbchpeplkfjdhnoepgdnmjkkci/2.0.6_0/helper/payment.html
/home/cheung/.config/google-chrome/Default/Extensions/nclcbchpeplkfjdhnoepgdnmjkkci/2.0.6_0/helper/bypassdomain.html
/home/cheung/.config/google-chrome/Default/Extensions/nclcbchpeplkfjdhnoepgdnmjkkci/2.0.6_0/page.html
/home/cheung/.config/google-chrome/Default/Extensions/nclcbchpeplkfjdhnoepgdnmjkkci/2.0.6_0/main.html
/home/cheung/.config/google-chrome/Default/Extensions/cfhdojbkjhnlbpkdaibdcddlllfdbb/11_0/day1.html
/home/cheung/.config/google-chrome/Default/Extensions/cfhdojbkjhnlbpkdaibdcddlllfdbb/11_0/composer.html
/home/cheung/.config/google-chrome/Default/Extensions/cfhdojbkjhnlbpkdaibdcddlllfdbb/11_0/devtools.html
/home/cheung/.config/google-chrome/Default/Extensions/cfhdojbkjhnlbpkdaibdcddlllfdbb/3.11_0/first-run.html
/home/cheung/.config/google-chrome/Default/Extensions/cfhdojbkjhnlbpkdaibdcddlllfdbb/3.11_0/issue-reporter.html
/home/cheung/.config/google-chrome/Default/Extensions/cfhdojbkjhnlbpkdaibdcddlllfdbb/3.11_0/problem.html
/home/cheung/.config/google-chrome/Default/Extensions/cfhdojbkjhnlbpkdaibdcddlllfdbb/3.11_0/options.html
/home/cheung/.config/google-chrome/Default/Extensions/cfhdojbkjhnlbpkdaibdcddlllfdbb/3.11_0/updates.html
/home/cheung/.config/google-chrome/Default/Extensions/cfhdojbkjhnlbpkdaibdcddlllfdbb/3.11_0/popup-dummy.html
/home/cheung/.config/google-chrome/Default/Extensions/cfhdojbkjhnlbpkdaibdcddlllfdbb/3.11_0/desktop-options.html
/home/cheung/.config/google-chrome/Default/Extensions/cfhdojbkjhnlbpkdaibdcddlllfdbb/3.11_0/proxy.html
/home/cheung/.config/google-chrome/Default/Extensions/cfhdojbkjhnlbpkdaibdcddlllfdbb/3.11_0/popup.html
/home/cheung/.config/google-chrome/Default/Extensions/cfhdojbkjhnlbpkdaibdcddlllfdbb/3.11_0/devtools-panel.html
/home/cheung/.config/google-chrome/Default/Extensions/akimgiweeqognljlfchpkpkfbmeapkh/2.1.1_0/index.html
/home/cheung/.config/google-chrome/Default/Extensions/akimgiweeqognljlfchpkpkfbmeapkh/2.1.1_0/options.html
/home/cheung/.config/google-chrome/Default/Extensions/ikhdkkncnognlghjlkmcimnlkkteamad/4.0_0/main.html
/home/cheung/.config/google-chrome/Default/Extensions/ikhdkkncnognlghjlkmcimnlkkteamad/4.0_0/settings.html
/home/cheung/.config/google-chrome/Default/Extensions/ikhdkkncnognlghjlkmcimnlkkteamad/4.0_0/pdf-viewer/web/viewer.html
/home/cheung/.config/google-chrome/Default/Extensions/ihblhcdfebdnlfelmnfcidctb1gcoo/2.0_0/options.html
/home/cheung/.config/google-chrome/Default/Extensions/ihblhcdfebdnlfelmnfcidctb1gcoo/2.0_0/popup.html
/home/cheung/.config/google-chrome/Default/Extensions/eladnpbcfbnmkbkopoojfekhknhdteeh/4.9.34.0/uil/stylesheet-editor/index.html
/home/cheung/.config/google-chrome/Default/Extensions/eladnpbcfbnmkbkopoojfekhknhdteeh/4.9.34.0/uil/devtools/index.html
/home/cheung/.config/google-chrome/Default/Extensions/eladnpbcfbnmkbkopoojfekhknhdteeh/4.9.34.0/uil/popup/index.html
/home/cheung/.config/google-chrome/Default/Extensions/felcaaldbndcclmgdncncolepbglejp/1.2_0/main.html
/home/cheung/.config/google-chrome/Default/Extensions/nbcbgbppomkndfbplepljnakkboejgh/1.0.3_0/reader-container.html
/home/cheung/.config/google-chrome/Default/Extensions/nbcbgbppomkndfbplepljnakkboejgh/1.0.3_0/newtab.html
/home/cheung/.config/google-chrome/Default/Extensions/mhcabhnmonkkndfhniemimakhnorinh/1.0.3_0/background.html
```

```
cheung@cheung: ~
cheung@cheung: $ find ~ -name "*.ht*"
cheung@cheung: $ touch test.htm
cheung@cheung: $ find ~ -name "*.ht*"
/home/cheung/test.htm
cheung@cheung: $ find ~ -name "*.*c"
/home/cheung/.vscode/extensions/o1e1986.arma-dev-0.0.20/node_modules/ssh2/util/pagent.c
/home/cheung/.vscode/extensions/ms-python.python-2021.6.944021595/pythonFiles/lib/python/debugpy/_vendored/pydevd/_pydevd_frame_eval/_pydevd_frame_evaluator.c
/home/cheung/.vscode/extensions/ms-python.python-2021.6.944021595/pythonFiles/lib/python/debugpy/_vendored/pydevd/_pydevd_bundle/pydevd_cython.c
cheung@cheung: $ |
```



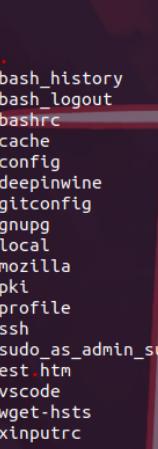
20. 给出一条命令，在主目录下显示所有文件内容中包含字符串“Linux”的文件名。

结果较多，只放一部分

```
cheung@cheung: ~
cheung@cheung: $ grep -rl "Linux" >Q20
cheung@cheung: $ head -n 30 Q20
.local/share/xorg/Xorg.0.log.old
.local/share/xorg/Xorg.0.log
.local/share/Trash/files/doc/readme.md
.local/share/Trash/files/deepin-wine-for-ubuntu/deepin-libwine-dev_2.18-12_i386/usr/include/deepin-wine/windows/objidl.idl
.local/share/Trash/files/deepin-wine-for-ubuntu/udis6_1.72-2_i386/usr/include/libudis6/types.h
.local/share/Trash/files/deepin-wine-for-ubuntu/deepin-libwine_2.18-12_i386/usr/lib/i386-linux-gnu/deepin-wine/winaspi.dll16.so
.local/share/Trash/files/deepin-wine-for-ubuntu/deepin-libwine_2.18-12_i386/usr/lib/i386-linux-gnu/deepin-wine/dinput.dll.so
.local/share/Trash/files/deepin-wine-for-ubuntu/deepin-libwine_2.18-12_i386/usr/lib/i386-linux-gnu/deepin-wine/wnaspi32.dll.so
.mozilla/firefox/xz1gdihy.default-release/assetsstore-backups/previous.jsonlz4
.mozilla/firefox/xz1gdihy.default-release/compatibility.int
.mozilla/firefox/xz1gdihy.default-release/webappsstore.sqlite
.mozilla/firefox/xz1gdihy.default-release/places.sqlite
.mozilla/firefox/xz1gdihy.default-release/datareporting/archived/2021-07/1625980202655_449b2684-fa4b-4e7b-b811-acfbe704ecc0.main.jsonlz4
.mozilla/firefox/xz1gdihy.default-release/datareporting/archived/2021-07/1626008738648_21c88985-484c-499a-85d8-bb1648054e44.main.jsonlz4
.mozilla/firefox/xz1gdihy.default-release/datareporting/archived/2021-07/162581713529_fb9a8c27-31df-4e87-88bb-140f1fe06802.main.jsonlz4
.mozilla/firefox/xz1gdihy.default-release/datareporting/archived/2021-07/162581474682_b754-42cd-96bf-c9ac9df0f189a.main.jsonlz4
.mozilla/firefox/xz1gdihy.default-release/datareporting/archived/2021-07/1626010304182_2ab0558f-f32d-4bf6-bc88-a00668690059.main.jsonlz4
.mozilla/firefox/xz1gdihy.default-release/datareporting/archived/2021-07/162620914219_ae59d486_3a4c-45ed-908d-7d50db3d024.main.jsonlz4
.mozilla/firefox/xz1gdihy.default-release/datareporting/archived/2021-07/1626248530052_6ce6b92a-7f79-4576-b61a-1fac3afb4a2d.main.jsonlz4
.mozilla/firefox/xz1gdihy.default-release/datareporting/archived/2021-07/1626249612266_7378cd6b-527a-4d72-9446-54402edb8d1.main.jsonlz4
.mozilla/firefox/xz1gdihy.default-release/datareporting/archived/2021-07/1626010753319_3500e364-92fa-4ff0-b79c-94c9f2004cd9.main.jsonlz4
.mozilla/firefox/xz1gdihy.default-release/datareporting/archived/2021-07/1626248552757_5f12ab55-457d-490a-972e-870f0bbec108.main.jsonlz4
.mozilla/firefox/xz1gdihy.default-release/datareporting/archived/2021-07/1626248501131_a5337cbc-cdc5-4966-a087-cc724f559bc9.main.jsonlz4
.mozilla/firefox/xz1gdihy.default-release/datareporting/archived/2021-07/16262485440664_557ded34-4981-4b06-87d6-d7f31044a083.main.jsonlz4
.mozilla/firefox/xz1gdihy.default-release/datareporting/archived/2021-07/162625523150_0ceda779-57c2-461a-ba05-411ac488059a.main.jsonlz4
.mozilla/firefox/xz1gdihy.default-release/datareporting/archived/2021-07/162624861527_12d2ba25-1d5f-48be-abec-76040dfa9e35.main.jsonlz4
.mozilla/firefox/xz1gdihy.default-release/datareporting/archived/2021-07/1625817634174_137bdb5e-zece-4702-b5ed-edd2584a4c0.main.jsonlz4
.mozilla/firefox/xz1gdihy.default-release/datareporting/archived/2021-07/1626019335727_3294a3af-d0cd-4380-b243-7d332246cf4d.main.jsonlz4
cheung@cheung: $ |
```

21. 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	大小
/	directory	rwxr-xr-x	20	root	root	4096B
/etc/passwd	file	rwxr-xr-x	1	root	root	10B
/bin/df	file	rwxr-xr-x	1	root	root	5B
~	directory	rwxr-xr-x	24	cheung	cheung	4096B



```

cheung@cheung:~$ ls -al /etc/passwd /bin/df
-rw-r--r-- 1 root root 93128 9月  5 2019 /bin/df
-rw-r--r-- 1 root root 2741 7月 10 14:04 /etc/passwd
cheung@cheung:~$ ls -al / | grep "\."
drwxr-xr-x 20 root root 4096 7月 11 20:05 .
drwxr-xr-x 20 root root 4096 7月 11 20:05 ..
cheung@cheung:~$ ls -al . | grep "\."
drwxr-xr-x 24 cheung cheung 4096 7月 15 17:26 .
drwxr-xr-x 3 root  root 4096 7月 10 05:52 ..
-rw----- 1 cheung cheung 13191 7月 15 17:25 .bash_history
-rw-r--r-- 1 cheung cheung 220 7月 10 05:52 .bash_logout
-rw-r--r-- 1 cheung cheung 3771 7月 10 05:52 .bashrc
drwxrwxr-x 18 cheung cheung 4096 7月 15 14:16 .cache
drwxr-xr-x 21 cheung cheung 4096 7月 15 17:26 .config
drwxrwxr-x 3 cheung cheung 4096 7月 11 20:15 .deepinwine
-rw-rw-r-- 1 cheung cheung 48 7月 11 23:59 .gitconfig
drwxr-xr-x 3 cheung cheung 4096 7月 14 17:42 .gnupg
drwxr-xr-x 3 cheung cheung 4096 7月 10 14:05 .local
drwxrwxr-x 5 cheung cheung 4096 7月 10 14:06 .mozilla
drwxrwxr-x 3 cheung cheung 4096 7月 11 11:40 .pki
-rw-r--r-- 1 cheung cheung 807 7月 10 05:52 .profile
drwxr----- 2 cheung cheung 4096 7月 12 00:05 .ssh
-rw-r--r-- 1 cheung cheung 0 7月 9 14:10 .sudo_as_admin_successful
-rw-rw-r-- 1 cheung cheung 0 7月 15 17:02 test.htm
drwxrwxr-x 3 cheung cheung 4096 7月 11 11:39 .vscode
-rw-rw-r-- 1 cheung cheung 211 7月 12 00:15 .wget-hsts
-rw-rw-r-- 1 cheung cheung 131 7月 9 15:53 .xinputrc
cheung@cheung:~|

```

22. 在你的主目录中的temp、professional和个人三个子目录，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。

```
cheung@cheung: ~$ chmod 755 temp
cheung@cheung: ~$ chmod 755 professional
cheung@cheung: ~$ chmod 755 personal
cheung@cheung: ~$ ls -lt | grep "personal\|temp\|professional"
drwxr-xr-x 2 cheung cheung 4096 7月 15 17:25 temp
drwxr-xr-x 4 cheung cheung 4096 7月 15 14:57 professional
drwxr-xr-x 4 cheung cheung 4096 7月 15 14:55 personal
cheung@cheung: ~$ cd temp/
cheung@cheung: ~/temp$ mkdir d{1,2,3}
cheung@cheung: ~/temp$ cd d1
cheung@cheung: ~/temp/d1$ touch f1
cheung@cheung: ~/temp/d1$ ls -l
total 0
cheung@cheung: ~/temp/d1$ cd ..
cheung@cheung: ~/temp$ ls -l
total 344
drwxrwxr-x 2 cheung cheung 4096 7月 15 19:33 d1
drwxrwxr-x 2 cheung cheung 4096 7月 15 19:33 d2
drwxrwxr-x 2 cheung cheung 4096 7月 15 19:33 d3
-rw-rw-r-- 1 cheung cheung 327764 7月 15 16:13 largeFile
-rw-rw-r-- 1 cheung cheung 2179 7月 15 16:13 mediumFile
-rw-rw-r-- 1 cheung cheung 1004 7月 15 16:24 smallFile
cheung@cheung: ~/temp$ |
  Firscrip

```

23. 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。

由截图可见，仅有执行权限时，可以执行 `ls -ld` 但不能执行 `ls -l`。

仅有读权限时，能读出目录下有哪些文件及其文件类型，但是读不出其他具体信息。

有读和执行权限时，可以完整执行 ls -l。

```
cheung@cheung:~$ id
uid=1000(cheung) gid=1000(cheung) groups=1000(cheung),4(adm),24(cdrom),27(sudo),30(dlp),46(plugdev),120(lpadmin),131(lxd),132(sambashare)
cheung@cheung:~$ ls -l | grep "temp"
drwxr-xr-x 5 cheung cheung 4096 7月 15 19:33 temp
cheung@cheung:~$ chmod u=x temp
cheung@cheung:~$ ls -l | grep "temp"
d--xrwxr-x 5 cheung cheung 4096 7月 15 19:33 temp
cheung@cheung:~$ ls -ld temp
d--xrwxr-x 5 cheung cheung 4096 7月 15 19:33 temp
cheung@cheung:~$ ls -l temp
ls: cannot open directory 'temp': Permission denied
cheung@cheung:~$ chmod u=r temp
cheung@cheung:~$ ls -l temp
ls: cannot access 'temp/smallFile': Permission denied
ls: cannot access 'temp/d1': Permission denied
ls: cannot access 'temp/largeFile': Permission denied
ls: cannot access 'temp/d2': Permission denied
ls: cannot access 'temp/d3': Permission denied
ls: cannot access 'temp/mediumFile': Permission denied
total 344
d????????? ? ? ? ?          ? d1
d????????? ? ? ? ?          ? d2
d????????? ? ? ? ?          ? d3
-????????? ? ? ? ?          ? largeFile
-????????? ? ? ? ?          ? mediumFile
-????????? ? ? ? ?          ? smallFile
cheung@cheung:~$ chmod u=rx temp
cheung@cheung:~$ ls -l temp
total 344
drwxrwxr-x 2 cheung cheung 4096 7月 15 19:33 d1
drwxrwxr-x 2 cheung cheung 4096 7月 15 19:33 d2
drwxrwxr-x 2 cheung cheung 4096 7月 15 19:33 d3
-rw-rw-r-- 1 cheung cheung 327764 7月 15 16:13 largeFile
-rw-rw-r-- 1 cheung cheung 2179 7月 15 16:13 mediumFile
-rw-rw-r-- 1 cheung cheung 1004 7月 15 16:24 smallFile
cheung@cheung:~$ |
```

24. 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。

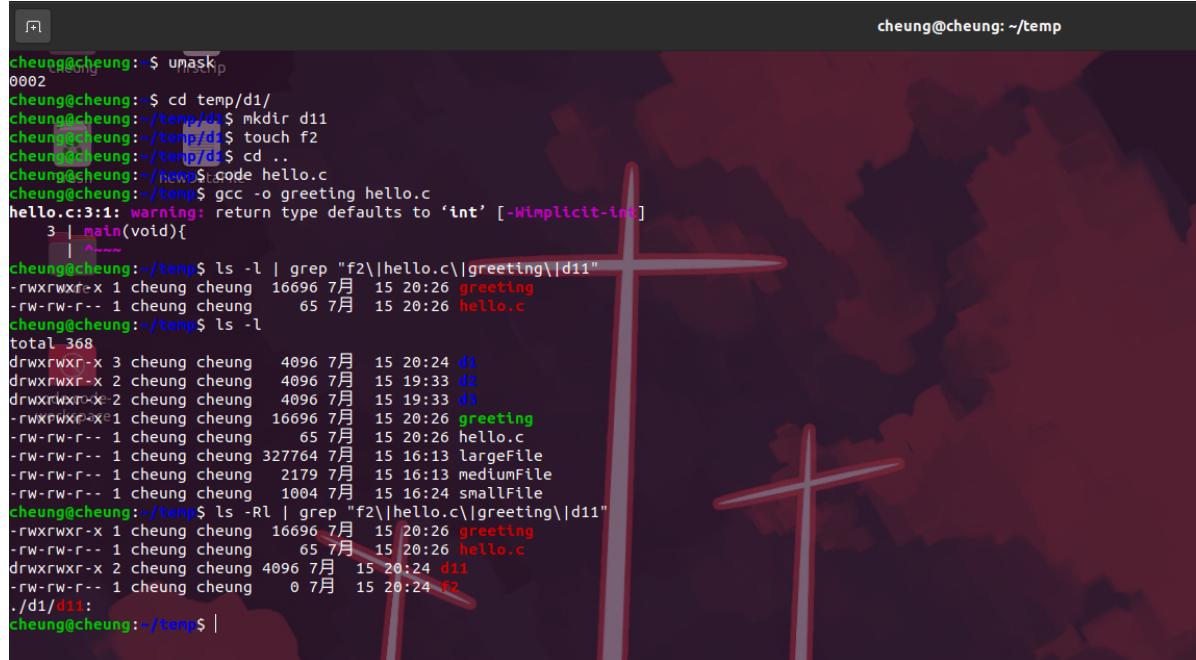
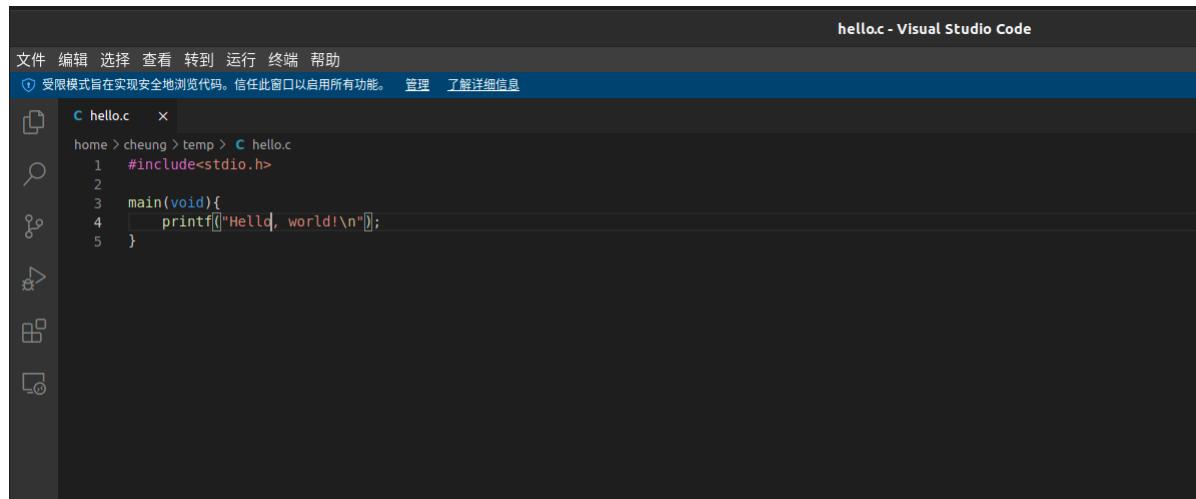
再运行命令 `gcc -o greeting hello.c`，生成了可执行文件 `greeting`。长列表显示 `f2`、`hello.c`、`greeting` 和 `d1` 文件访问权限。

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
0002	664	664	775	775
0077	600	664	700	700

结论：掩码位为1，则生成的权限位必为0；掩码位为0，则生成的权限位为默认的权限位。

0002：



```
cheung@cheung:~/temp$ umask 0002
cheung@cheung:~/temp$ cd temp/d1/
cheung@cheung:~/temp/d1$ mkdir d11
cheung@cheung:~/temp/d1$ touch f2
cheung@cheung:~/temp/d1$ cd ..
cheung@cheung:~/temp$ code hello.c
cheung@cheung:~/temp$ gcc -o greeting hello.c
hello.c:3:1: warning: return type defaults to 'int' [-Wimplicit-int]
  3 | main(void){
  | ~~~~
cheung@cheung:~/temp$ ls -l | grep "f2|hello.c|greeting|d11"
-rwxrwxr-x 1 cheung cheung 16696 7月 15 20:26 greeting
-rw-rw-r-- 1 cheung cheung 65 7月 15 20:26 hello.c
cheung@cheung:~/temp$ ls -l
total 368
drwxrwxr-x 3 cheung cheung 4096 7月 15 20:24 d1
drwxrwxr-x 2 cheung cheung 4096 7月 15 19:33 d2
drwxrwxr-x 2 cheung cheung 4096 7月 15 19:33 d3
-rwxrwxr-x 1 cheung cheung 16696 7月 15 20:26 greeting
-rw-rw-r-- 1 cheung cheung 65 7月 15 20:26 hello.c
-rw-rw-r-- 1 cheung cheung 327764 7月 15 16:13 largeFile
-rw-rw-r-- 1 cheung cheung 2179 7月 15 16:13 mediumFile
-rw-rw-r-- 1 cheung cheung 1004 7月 15 16:24 smallFile
cheung@cheung:~/temp$ ls -Rl | grep "f2|hello.c|greeting|d11"
-rwxrwxr-x 1 cheung cheung 16696 7月 15 20:26 greeting
-rw-rw-r-- 1 cheung cheung 65 7月 15 20:26 hello.c
drwxrwxr-x 2 cheung cheung 4096 7月 15 20:24 d11
-rw-rw-r-- 1 cheung cheung 0 7月 15 20:24 f2
./d1/d11:
cheung@cheung:~/temp$ |
```

0077：重新打开了code.c并保存，然后又重新编译生成了greeting。我发现编辑文件不会改变文件原有的权限。

```
cheung@cheung:~/temp$ umask 077
cheung@cheung:~/temp$ umask
0077
cheung@cheung:~/temp$ cd d2
cheung@cheung:~/temp/d2$ mkdir d21
cheung@cheung:~/temp/d2$ touch f2
cheung@cheung:~/temp/d2$ cd ..
cheung@cheung:~/temp$ ls -Rl | grep "f2\|hello.c\|greeting\|d21"
-rwxrwxr-x 1 cheung cheung 16696 7月 15 20:26 greeting
-rw-rw-r-- 1 cheung cheung 65 7月 15 20:33 hello.c
-rw-rw-r-- 1 cheung cheung 0 7月 15 20:24 f2
drwx----- 2 cheung cheung 4096 7月 15 20:34 d21
-rw-rw-r-- 1 cheung cheung 0 7月 15 20:35 f2
./d2/d21:
cheung@cheung:~/temp$ code hello.c
cheung@cheung:~/temp$ ls -Rl | grep "f2\|hello.c\|greeting\|d21"
-rwxrwxr-x 1 cheung cheung 16696 7月 15 20:26 greeting
-rw-rw-r-- 1 cheung cheung 66 7月 15 20:35 hello.c
-rw-rw-r-- 1 cheung cheung 0 7月 15 20:24 f2
drwx----- 2 cheung cheung 4096 7月 15 20:34 d21
-rw-rw-r-- 1 cheung cheung 0 7月 15 20:35 f2
./d2/d21:
cheung@cheung:~/temp$ gcc -o greeting hello.c
hello.c:3:1: warning: return type defaults to 'int' [-Wimplicit-int]
  3 | main(void){
  | ^~~~~
cheung@cheung:~/temp$ ls -Rl | grep "f2\|hello.c\|greeting\|d21"
-rwx----- 1 cheung cheung 16696 7月 15 20:36 greeting
-rw-rw-r-- 1 cheung cheung 66 7月 15 20:35 hello.c
-rw-rw-r-- 1 cheung cheung 0 7月 15 20:24 f2
drwx----- 2 cheung cheung 4096 7月 15 20:34 d21
-rw-rw-r-- 1 cheung cheung 0 7月 15 20:35 f2
./d2/d21:
cheung@cheung:~/temp$ |
```

25. 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。

在077掩码下进行：

```
cheung@cheung:~/temp$ umask 002
cheung@cheung:~/temp$ umask 077
cheung@cheung:~/temp$ cp smallFile
cp: missing destination file operand after 'smallFile'
Try 'cp --help' for more information.
cheung@cheung:~/temp$ cp smallFile d1
cheung@cheung:~/temp$ ls -l
total 368
drwxrwxr-x 3 cheung cheung 4096 7月 15 20:44 d1
drwxrwxr-x 3 cheung cheung 4096 7月 15 20:35 d2
drwxrwxr-x 2 cheung cheung 4096 7月 15 19:33 d3
-rwx----- 1 cheung cheung 16696 7月 15 20:36 greeting
-rw-rw-r-- 1 cheung cheung 66 7月 15 20:35 hello.c
-rw-rw-r-- 1 cheung cheung 327764 7月 15 16:13 largeFile
-rw-rw-r-- 1 cheung cheung 2179 7月 15 16:13 mediumFile
-rw-rw-r-- 1 cheung cheung 1004 7月 15 16:24 smallFile
cheung@cheung:~/temp$ cd d1
cheung@cheung:~/temp/d1$ ls -l
total 8
drwxrwxr-x 2 cheung cheung 4096 7月 15 20:24 d1
-rw-rw-r-- 1 cheung cheung 0 7月 15 19:33 f1
-rw-rw-r-- 1 cheung cheung 0 7月 15 20:24 f2
-rw-rw-r-- 1 cheung cheung 1004 7月 15 20:44 smallFile
cheung@cheung:~/temp/d1$ ls -li | grep "small*"
2893444 -rw-rw-r-- 1 cheung cheung 1004 7月 15 20:44 smallFile
cheung@cheung:~/temp/d1$ |
```

在002掩码下进行：

```

cheung@cheung:~/temp$ umask 002
cheung@cheung:~/temp$ cp -f smallFile d1
cheung@cheung:~/temp$ ls -li | grep "small*"
2893524 -rw-rw-r-- 1 cheung cheung 1004 7月 15 16:24 smallFile
cheung@cheung:~/temp$ cd d1
cheung@cheung:~/temp/d1$ ls -li | grep "small*"
2893444 -rw-rw-r-- 1 cheung cheung 1004 7月 15 20:50 smallFile
cheung@cheung:~/temp/d1$ rm smallFile
cheung@cheung:~/temp/d1$ ls -li | grep "small*"
cheung@cheung:~/temp/d1$ cd ..
cheung@cheung:~/temp$ cp smallFile d1
cheung@cheung:~/temp$ cd d1
cheung@cheung:~/temp/d1$ ls -li | grep "small*"
2893444 -rw-rw-r-- 1 cheung cheung 1004 7月 15 20:51 smallFile
cheung@cheung:~/temp/d1$ |

```

code.code-workspace

26. 在~/temp目录下，把当前目录改变成d2。创建一个名字为smallFile.hard硬链接到d1目录下的smallFile文件。长列表格式显示smallFile.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。

两个文件的inode号相同且其余所有信息一致。

```

cheung@cheung:~/temp/d1$ cd ..
cheung@cheung:~/temp$ cd d2
cheung@cheung:~/temp/d2$ ln .../smallFile smallFile.hard
cheung@cheung:~/temp/d2$ ls -li smallFile.hard
2893524 -rw-rw-r-- 2 cheung cheung 1004 7月 15 16:24 smallFile.hard
cheung@cheung:~/temp/d2$ ls -li .../smallFile
2893524 -rw-rw-r-- 1 cheung cheung 1004 7月 15 16:24 .../smallFile
cheung@cheung:~/temp/d2$ cat smallFile.hard
FirstName LastName Major GPA Email Phone
John Doe ECE 3.54 doe@jd.home.org 111.222.3333
James Davis ECE 3.71 davis@jd.work.org 111.222.1111
Al Davis CS 2.63 davis@a.lakers.org 111.222.2222
Ahmad Rashid MBA 3.04 ahmad@mba.org 111.222.4444
Sam Chu ECE 3.68 chu@sam.arts.edu 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 de.code- Clark ECE 2.68 clark@xyz.ab.com 111.111.5555
Nabeel kspace 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@s.r.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
cheung@cheung:~/temp/d2$ cat .../smallFile
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.arts.edu 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@s.r.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
cheung@cheung:~/temp/d2$ |

```

27. 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。

取消读权限后，不能通过硬链接显示文件内容。系统提示没有权限，此时查看硬链接的权限可以看到，硬链接的读权限也被取消，这是因为硬连接只是原文件的别名，所有信息与属性均与原文件一致。增加读权限后，又可以正常显示文件内容。此时查看硬链接的权限可以看到，硬链接又有读权限了。做备份时，可以看到备份文件的权限均与原文件一致，但inode号不同。然后删除原文件，无法通过

原文件显示文件内容；查看硬链接的权限可以看到硬连接的硬链接数变为1；依然可以通过硬链接显示文件内容。这是因为该文件的硬链接数不为0，该文件的inode与文件本身并未被删除。只是指向该inode的原文件被删除了，少了一个访问该inode的入口，而通过硬链接访问该inode的入口还在。

```
cheung@cheung:~/temp/d2$ cat smallFile.hard
FirstName LastName Major GPA Email Phone
John Doe ECE 3.54 doe@jd.home.org 111.222.3333
James Davis ECE 3.71 davis@jd.work.org 111.222.1111
Al Davis CS 2.63 davis@a.lakers.org 111.222.2222
Ahmad Rashid MBA 3.04 ahmad@mba.org 111.222.4444
Sam Chu DataFile 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@s.r.cs.edu 111.111.3333
Charles Right EECS 3.31 right@cr.abc.edu 111.111.4444
Diane Rover ECE 3.87 rover@d.r.xyz.edu 111.111.5555
Aziz Inan EECS 3.75 ainan@ai.abc.edu 111.111.1111
cheung@cheung:~/temp/d2$ chmod u-r ./smallFile.hard
cheung@cheung:~/temp/d2$ ls -li smallFile.hard
2893524 --w-rw-r-- 2 cheung cheung 1004 7月 15 16:24 smallFile.hard
cheung@cheung:~/temp/d2$ cat smallFile.hard
cat: smallFile.hard: Permission denied
cheung@cheung:~/temp/d2$ chmod u+r ./smallFile.hard
cheung@cheung:~/temp/d2$ ls -li smallFile.hard
2893524 -rw-rw-r-- 2 cheung cheung 1004 7月 15 16:24 smallFile.hard
cheung@cheung:~/temp/d2$ cat smallFile.hard
FirstName LastName Major GPA Email Phone
John Doe ECE 3.54 doe@jd.home.org 111.222.3333
James Davis ECE 3.71 davis@jd.work.org 111.222.1111
Al Davis CS 2.63 davis@a.lakers.org 111.222.2222
Ahmad Rashid MBA 3.04 ahmad@mba.org 111.222.4444
Sam Chu DataFile 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@s.r.cs.edu 111.111.3333
Charles Right EECS 3.31 right@cr.abc.edu 111.111.4444
Diane Rover ECE 3.87 rover@d.r.xyz.edu 111.111.5555
Aziz Inan EECS 3.75 ainan@ai.abc.edu 111.111.1111
cheung@cheung:~/temp/d2$ |
```

```

cheung@cheung: ~/temp/d2$ 
cat: smallFile.hard: Permission denied
cheung@cheung:~/temp/d2$ chmod u+r ../smallFile
cheung@cheung:~/temp/d2$ ls -li smallFile.hard
2893524 -rw-rw-r-- 2 cheung cheung 1004 7月 15 16:24 smallFile.hard
cheung@cheung:~/temp/d2$ cat smallFile.hard
FirstName LastName Major GPA Email Phone
John Doe ECE 3.54 doe@jd.home.org 111.222.3333
James Davis ECE 3.71 davis@jd.work.org 111.222.1111
AL Davis CS 2.63 davis@lakers.org 111.222.2222
Ahmad Rashid MBA 3.04 ahmad@mba.org 111.222.4444
Sam Chu ECE 3.68 chug@ab.com 111.222.5555
Arun Roy SS 3.86 roy@ss.arts.edu 111.222.8888
Rick Marsh CS 2.34 marsh@ab.org 111.222.6666
James Adam CS 2.77 jadam@ab.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
Patodeode King SS 3.77 king@pk.xyz.org 111.111.7777
Jake@rkspace 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@s.r.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
cheung@cheung:~/temp/d2$ cp ../smallFile .../smallFile.copy
cheung@cheung:~/temp/d2$ ls -li .../smallFile.copy
2893452 -rw-rw-r-- 1 cheung cheung 1004 7月 15 20:59 .../smallFile.copy
cheung@cheung:~/temp/d2$ rm .../smallFile
cheung@cheung:~/temp/d2$ ls -li .../smallFile
ls: cannot access '.../smallFile': No such file or directory
cheung@cheung:~/temp/d2$ ls -li smallFile.hard
2893524 -rw-rw-r-- 1 cheung cheung 1004 7月 15 16:24 smallFile.hard
cheung@cheung:~/temp/d2$ cat smallFile.hard
FirstName LastName Major GPA Email Phone
John Doe ECE 3.54 doe@jd.home.org 111.222.3333
James Davis ECE 3.71 davis@jd.work.org 111.222.1111
AL Davis CS 2.63 davis@lakers.org 111.222.2222
Ahmad Rashid MBA 3.04 ahmad@mba.org 111.222.4444
Sam Chu ECE 3.68 chug@ab.com 111.222.5555
Arun Roy SS 3.86 roy@ss.arts.edu 111.222.8888
Rick Marsh CS 2.34 marsh@ab.org 111.222.6666
James Adam CS 2.77 jadam@ab.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@s.r.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
cheung@cheung:~/temp/d2$ |

```

28. 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。

两个文件的inode号不同。查看软连接的信息时，可以看到最后有箭头指向原文件。通过软连接也可以直接读出文件内容，同时可以看到**软连接的权限与原文件的权限不同**。实验中我还发现，建立软链接时，似乎只能输入绝对路径，如果输入相对路径，系统会提示文件不存在。

```

cheung@cheung:~/temp$ ln -s ~/temp/d1/smallFile d2/smallFile.soft
cheung@cheung:~/temp$ ls -li d2/smallFile.soft
2893456 lrwxrwxrwx 1 cheung cheung 30 7月 15 21:23 d2/smallFile.soft -> /home/cheung/temp/d1/smallFile
cheung@cheung:~/temp$ ls -li di/smallFile
2893444 -rw-rw-r-- 1 cheung cheung 1004 7月 15 20:51 d1/smallFile
cheung@cheung:~/temp$ cat di/smallFile
FirstName LastName Major GPA Email Phone
John Doe ECE 3.54 doe@jd.home.org 111.222.3333
James Davis ECE 3.71 davids@jd.work.org 111.222.1111
AL Davis CS 2.63 davisa@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 royo@ss.arts.edu 111.222.8888
Rick Marsh CS 2.34 marsha@b.org 111.222.6666
James Adam CS 2.77 jadam@a.b.org 111.222.7777
Art Pohn ECE 4.00 pohn@ap.a.org 111.222.9999
John Clark ECE 2.68 clarkxyz@ab.com 111.111.5555
Nabeel Ali EE 3.56 ali@ee.eng.edu 111.111.8888
Tomode.code- Nelson ECE 3.81 nelson@tn.abc.org 111.111.6666
Patworkspace 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
Suntl 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@i.abc.edu 111.111.1111
cheung@cheung:~/temp$ cat d2/smallFile.soft
FirstName LastName Major GPA Email Phone
John Doe ECE 3.54 doe@jd.home.org 111.222.3333
James Davis ECE 3.71 davisa@lakers.org 111.222.2222
AL Davis CS 2.63 davisa@lakers.org 111.222.4444
Ahmad Rashid MBA 3.04 ahmad@mba.org 111.222.5555
Sam Chu ECE 3.68 chu@sam.ab.com 111.222.8888
Arun Roy SS 3.86 royo@ss.arts.edu 111.222.6666
Rick Marsh CS 2.34 marsha@b.org 111.222.7777
James Adam CS 2.77 jadam@a.b.org 111.222.9999
Art Pohn ECE 4.00 pohn@ap.a.org 111.111.5555
John Clark ECE 2.68 clarkxyz@ab.com 111.111.8888
Nabeel Ali EE 3.56 ali@ee.eng.edu 111.111.6666
Tom Nelson ECE 3.81 nelson@tn.abc.org 111.111.7777
Pat King SS 3.77 king@pk.xyz.org 111.111.9999
Jake Zulu CS 3.00 zulu@jz.sa.org 111.111.2222
John Lee EE 3.64 jlee@j.lee.com 111.111.3333
Suntl Raj ECE 3.86 raj@sr.cs.edu 111.111.4444
Charles Right EECS 3.31 right@cr.abc.edu 111.111.5555
Diane Rover ECE 3.87 rover@dr.xyz.edu 111.111.1111
Aziz Inan EECS 3.75 ainan@i.abc.edu 111.111.1111
cheung@cheung:~/temp$ 

```

29. 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？最后做一个smallFile文件的备份，并删除smallFile文件，用smallFile.soft显示smallFile文件内容，又发生了什么？请你解释一下练习过程中的现象。

取消读权限后，不能通过软链接显示文件内容。系统提示没有权限，此时查看软链接的权限可以看到，软链接的权限依然为777，但是因为原文件不可以被读，因此无法显示内容。推断在通过软连接访问文件时，系统需要进行两次权限检查；不能通过软链接绕开原文件的权限检查。

增加读权限后，又可以正常显示文件内容。

做备份时，可以看到备份文件的权限均与原文件一致，但inode号不同。然后删除原文件，无法通过软链接显示文件内容。此时通过ls显示软链接的信息，可以看到箭头前后的文件名都变成了红字，代表文件不存在或失效。这是因为原文件的硬链接数归零，原文件及其inode被删除，软链接虽然还在，但是其指向的文件已经不存在了。

```
cheung@cheung:~/temp$ cat d2/smallFile.soft
FirstName LastName Major GPA Email Phone
John Doe ECE 3.54 doe@jd.home.org 111.222.3333
James Davis ECE 3.71 davis@jd.work.org 111.222.1111
Al Davis CS 2.63 davisa@lakers.org 111.222.2222
Ahmad Rashid MBA 3.04 ahmad@mba.org 111.222.4444
Sam Chu ChuDataFile 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@ab.org 111.222.6666
James Adam CS 2.77 jadam@ab.org 111.222.7777
Art Pohn ECE 4.00 pohn@ap.a.org 111.222.9999
John Clark ECE 2.68 clark@xyz.ab.com 111.111.5555
Nabeel Ali EE 3.56 ali@ee.eng.edu 111.111.8888
Tom Nelson ECE 3.81 nelson@tn.abc.org 111.111.6666
Pat King SS 3.77 king@pk.xyz.org 111.111.7777
Jake Zulu CS 3.00 zulu@jz.sa.org 111.111.9999
John Lee EE 3.64 jlee@j.lee.com 111.111.2222
Sunil Raj ECE 3.86 raj@sr.cs.edu 111.111.3333
Charles Right EECS 3.31 right@cr.abc.edu 111.111.4444
Diane Rover ECE 3.87 rover@dr.xyz.edu 111.111.5555
Aziz Inan EECS 3.75 ainan@ai.abc.edu 111.111.1111
cheung@cheung:~/temp$ chmod u-r d1/smallfile
cheung@cheung:~/temp$ cat d2/smallFile.soft
cat: d2/smallFile.soft: Permission denied
cheung@cheung:~/temp$ ls -li d2/smallFile.soft
2893456 lrwxrwxrwx 1 cheung cheung 30 7月 15 21:23 d2/smallFile.soft -> /home/cheung/temp/d1/smallFile
cheung@cheung:~/temp$ ls -li d1/smallFile
2893444 --w-rw-r-- 1 cheung cheung 1004 7月 15 20:51 d1/smallFile
cheung@cheung:~/temp$ chmod u+r d1/smallFile
cheung@cheung:~/temp$ cat d2/smallFile.soft
FirstName LastName Major GPA Email Phone
John Doe ECE 3.54 doe@jd.home.org 111.222.3333
James Davis ECE 3.71 davis@jd.work.org 111.222.1111
Al Davis CS 2.63 davisa@lakers.org 111.222.2222
Ahmad Rashid MBA 3.04 ahmad@mba.org 111.222.4444
Sam Chu ChuDataFile 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@ab.org 111.222.6666
James Adam CS 2.77 jadam@ab.org 111.222.7777
Art Pohn ECE 4.00 pohn@ap.a.org 111.222.9999
John Clark ECE 2.68 clark@xyz.ab.com 111.111.5555
Nabeel Ali EE 3.56 ali@ee.eng.edu 111.111.8888
Tom Nelson ECE 3.81 nelson@tn.abc.org 111.111.6666
Pat King SS 3.77 king@pk.xyz.org 111.111.7777
Jake Zulu CS 3.00 zulu@jz.sa.org 111.111.9999
John Lee EE 3.64 jlee@j.lee.com 111.111.2222
Sunil Raj ECE 3.86 raj@sr.cs.edu 111.111.3333
Charles Right EECS 3.31 right@cr.abc.edu 111.111.4444
Diane Rover ECE 3.87 rover@dr.xyz.edu 111.111.5555
Aziz Inan EECS 3.75 ainan@ai.abc.edu 111.111.1111
cheung@cheung:~/temp$ |
```

```
cheung@cheung:~/temp$ cat d2/smallFile.soft
FirstName LastName Major GPA Email Phone
John Doe ECE 3.54 doe@jd.home.org 111.222.3333
James Davis ECE 3.71 davis@jd.work.org 111.222.1111
Al Davis CS 2.63 davisa@lakers.org 111.222.2222
Ahmad Rashid MBA 3.04 ahmad@mba.org 111.222.4444
Sam Chu ChuDataFile 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@ab.org 111.222.6666
James Adam CS 2.77 jadam@ab.org 111.222.7777
Art Pohn ECE 4.00 pohn@ap.a.org 111.222.9999
John Clark ECE 2.68 clark@xyz.ab.com 111.111.5555
Nabeel Ali EE 3.56 ali@ee.eng.edu 111.111.8888
Tom Nelson ECE 3.81 nelson@tn.abc.org 111.111.6666
Pat King SS 3.77 king@pk.xyz.org 111.111.7777
Jake Zulu CS 3.00 zulu@jz.sa.org 111.111.9999
John Lee EE 3.64 jlee@j.lee.com 111.111.2222
Sunil Raj ECE 3.86 raj@sr.cs.edu 111.111.3333
Charles Right EECS 3.31 right@cr.abc.edu 111.111.4444
Diane Rover ECE 3.87 rover@dr.xyz.edu 111.111.5555
Aziz Inan EECS 3.75 ainan@ai.abc.edu 111.111.1111
cheung@cheung:~/temp$ chmod u-r d1/smallfile
cheung@cheung:~/temp$ cat d2/smallFile.soft
cat: d2/smallFile.soft: Permission denied
cheung@cheung:~/temp$ ls -li d2/smallFile.soft
2893456 lrwxrwxrwx 1 cheung cheung 30 7月 15 21:23 d2/smallFile.soft -> /home/cheung/temp/d1/smallFile
cheung@cheung:~/temp$ ls -li d1/smallFile
2893444 --w-rw-r-- 1 cheung cheung 1004 7月 15 20:51 d1/smallFile
cheung@cheung:~/temp$ chmod u+r d1/smallFile
cheung@cheung:~/temp$ cat d2/smallFile.soft
FirstName LastName Major GPA Email Phone
John Doe ECE 3.54 doe@jd.home.org 111.222.3333
James Davis ECE 3.71 davis@jd.work.org 111.222.1111
Al Davis CS 2.63 davisa@lakers.org 111.222.2222
Ahmad Rashid MBA 3.04 ahmad@mba.org 111.222.4444
Sam Chu ChuDataFile 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@ab.org 111.222.6666
James Adam CS 2.77 jadam@ab.org 111.222.7777
Art Pohn ECE 4.00 pohn@ap.a.org 111.222.9999
John Clark ECE 2.68 clark@xyz.ab.com 111.111.5555
Nabeel Ali EE 3.56 ali@ee.eng.edu 111.111.8888
Tom Nelson ECE 3.81 nelson@tn.abc.org 111.111.6666
Pat King SS 3.77 king@pk.xyz.org 111.111.7777
Jake Zulu CS 3.00 zulu@jz.sa.org 111.111.9999
John Lee EE 3.64 jlee@j.lee.com 111.111.2222
Sunil Raj ECE 3.86 raj@sr.cs.edu 111.111.3333
Charles Right EECS 3.31 right@cr.abc.edu 111.111.4444
Diane Rover ECE 3.87 rover@dr.xyz.edu 111.111.5555
Aziz Inan EECS 3.75 ainan@ai.abc.edu 111.111.1111
cheung@cheung:~/temp$ cp d1/smallfile d1/smallFile.copy
cheung@cheung:~/temp$ ls -li d1/small*
2893444 -rw-rw-r-- 1 cheung cheung 1004 7月 15 20:51 d1/smallFile
2893457 -rw-rw-r-- 1 cheung cheung 1004 7月 15 21:26 d1/smallFile.copy
cheung@cheung:~/temp$ rm d1/smallFile
cheung@cheung:~/temp$ cat d2/smallFile.soft
cat: d2/smallFile.soft: No such file or directory
cheung@cheung:~/temp$ ls -li d2/smallFile.soft
2893456 lrwxrwxrwx 1 cheung cheung 30 7月 15 21:23 d2/smallFile.soft -> /home/cheung/temp/d1/smallFile
cheung@cheung:~/temp$ |
```

30. 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。

可以看出，有279个进程在运行（包括了ps）。经过查询资料，我认为我的系统已经用systemd取代了init，因此下面的叙述全部用systemd而不是init。

systemd、bash、ps的PID分别为1、28449、28897。

systemd的PPID为0，代表其为最顶层的进程，不存在父进程。bash的PPID为28438，gnome-terminal-，翻译过来是桌面的虚拟终端。ps的PPID为28449，bash。

```
cheung@cheung:~$ ps -Al | wc -l
279
cheung@cheung:~$ ps -Al | grep "CMD\|systemd\|bash\|ps"
F S  UID   PID  PPID C PRI  NI ADDR SZ WCHAN TTY          TIME CMD
4 S  0     1    0  80  - 41925 - ?          00:00:03 systemd
4 S  0     306   1  0  79 -21189 - ?          00:00:03 systemd-journal
4 S Tras:0 354newDataFil1  0  80  0 - 6113 - ?          00:00:01 systemd-udevd
4 S  101   795   1  0  80  0 - 6074 - ?          00:00:02 systemd-resolve
4 S  102   796   1  0  80  0 - 22565 - ?          00:00:00 systemd-timesyncd
4 S  0     818   1  0  80  0 - 9301 - ?          00:00:01 cupsd
4 S  0     858   1  0  80  0 - 4193 - ?          00:00:02 systemd-logind
4 S  0     905   1  0  80  0 - 44600 - ?          00:00:00 cups-browsed
4 S c120  1142   1  0  80  0 - 81806 - ?          00:00:00 whoopsie
1 S  116   1145   1  0  80  0 - 2816 - ?          00:00:01 kerneloops
1 S  116   1147   1  0  80  0 - 2816 - ?          00:00:01 kerneloops
4 S  1000  1561   1  0  80  0 - 4808 ep_pol ?          00:00:01 systemd
0 S  1000  28449  28438  0  80  0 - 4879 do_wai pts/0  00:00:00 bash
4 R  1000  28897  28449  0  80  0 - 5013 - pts/0  00:00:00 ps
cheung@cheung:~$ ps -Al | grep "28449\|28438"
0 Swo 1000ce 28438  1561  1  80  - 242625 poll_s ?          00:00:10 gnome-terminal-
0 S  1000  28449  28438  0  80  0 - 4879 do_wai pts/0  00:00:00 bash
4 R  1000  28902  28449  0  80  0 - 5013 - pts/0  00:00:00 ps
0 S  1000  28903  28449  0  80  0 - 4419 pipe_r pts/0  00:00:00 grep
cheung@cheung:~$ |
```

31. 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。

```
cheung@cheung:~$ sleep 1m; echo 'Time for Lunch!';
Time for Lunch!
cheung@cheung:~|
```

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

cheung@cheung:~\$ date&uname -a&who&ps&

[1] 29474 pts/0 2021-07-15 22:01:00 CST

[2] 29475 pts/0 2021-07-15 22:01:00 bash

[3] 29476 pts/0 2021-07-15 22:01:00 Recent

[4] 29477 pts/0 2021-07-15 22:01:00 ps

cheung@cheung:~\$ Linux cheung 5.8.0-59-generic #66~20.04.1-Ubuntu SMP Thu Jun 17 11:14:10 UTC 2021 x86_64 x86_64 x86_64 GNU/Linux

PID TTY TIME CMD

28449 pts/0 00:00:00 bash

29476 pts/0 00:00:00 bash

29477 pts/0 00:00:00 ps

cheung :0 2021-07-15 22:01:00 (:0)

[1] Done date

[2] Done uname -a

[3] Done who

[4]+ Done ps

cheung@cheung:~\$ |

code Pictures

wor Videos

33. 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。

cheung@cheung:~\$ date && uname -a && who && ps

2021年 07月 15日 星期四 22:08:48 CST

Linux cheung 5.8.0-59-generic #66~20.04.1-Ubuntu SMP Thu Jun 17 11:14:10 UTC 2021 x86_64 x86_64 x86_64 GNU/Linux

cheung :0 2021-07-15 22:08:48 (:0)

PID TTY TIME CMD

28449 pts/0 00:00:00 bash

30329 pts/0 00:00:00 ps

cheung@cheung:~\$ |

code

34. 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
```

cheung@cheung:~\$ pwd

/home/cheung

cheung@cheung:~\$ bash

cheung@cheung:~\$ cd /usr

cheung@cheung:/usr\$ pwd

/usr

cheung@cheung:/usr\$ exit

cheung@cheung:~\$ pwd

/home/cheung

cheung@cheung:~\$ |

code

35. 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)

```
cheung@cheung:~$ touch foobar
cheung@cheung:~$ find ~ -name "foobar" 2>/dev/null
/home/cheung/foobar
cheung@cheung:~$ |
```

36. 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。

```
cheung@cheung:~$ find ~ -name "foobar" >foobar.path 2>/dev/null
cheung@cheung:~$ cat foobar.path
/home/cheung/foobar
cheung@cheung:~$ |
```

37. 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。

```
cheung@cheung:~$ cd temp/
cheung@cheung:~/temp$ cat student.records 1>output.data 2>error.log
cheung@cheung:~/temp$ |
```

38. 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
```

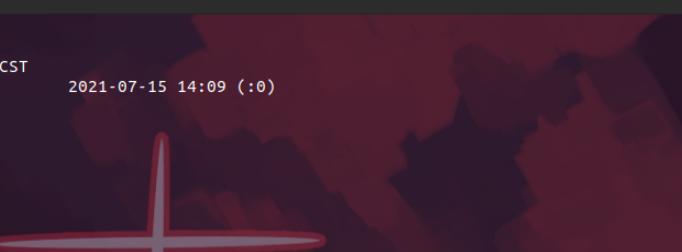
提示：使用exec命令

使用 bash 而不是 exec：



```
cheung@cheung:~/temp$ gedit fdata
cheung@cheung:~/temp$ cat fdata
echo -n "The time now is: "
date
echo -n "The users presently logged on are: "
who
cheung@cheung:~/temp$ bash <fdata 1>fout
cheung@cheung:~/temp$ cat fout
The time now is: 2021年 07月 15日 星期四 22:31:12 CST
The users presently logged on are: cheung :0          2021-07-15 14:09 (:0)
cheung@cheung:~/temp$ exec < fdata 1>fout|
```

使用 `exec`，由于当前进程被替换成 `exec` 的子进程，因此在 `exec` 执行完毕后当前 shell 会退出。再次打开终端，可以查看到刚才执行的结果。再次进入终端时，可以发现按动上下箭头可以看到 `fdata` 中的命令；而这一点是 `bash` 命令不能做到的。



```
cheung@cheung:~$ cd temp/
cheung@cheung:~/temp$ cat fout
The time now is: 2021年 07月 15日 星期四 22:31:55 CST
The users presently logged on are: cheung :0          2021-07-15 14:09 (:0)
cheung@cheung:~/temp$ |
```

39. 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。



```
cheung@cheung:~$ ls -l | wc
      18    155    967
cheung@cheung:~$ ls -l | wc -c
967
cheung@cheung:~$ ls -l | wc -w
155
cheung@cheung:~$ ls -l | wc -l
18
cheung@cheung:~$ |
```

40. 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？

1206 个普通文件，0 个目录文件与 349 个符号链接文件。我用了两个方法来得到答案，其中统计目录文件时，`find` 会把当前目录也统计进去，因此其结果为 1 而不是 0。

此统计没有统计隐藏文件。

```
cheung@cheung: ~$ cd bin
cheung@cheung:/bin$ find . -type f | wc -l
1206
cheung@cheung:/bin$ ls -l | grep "^-" | wc -l
1206
cheung@cheung:/bin$ find . -type d | wc -l
1 Trash
cheung@cheung:/bin$ ls -l | grep "^d" | wc -l
0
cheung@cheung:/bin$ find . -type d
.
cheung@cheung:/bin$ find . -type l | wc -l
349 code
cheung@cheung:/bin$ ls -l | grep "^l" | wc -l
349
cheung@cheung:/bin$ |
code.code-
workspace
```

讨论心得

实验中遇到的一个印象深刻的问题是查看进程时，利用 grep 命令无法搜索到init。在利用 man 命令查询后，我又在互联网搜索了相关资料，确认了我输入的命令没有问题。然后继续搜索相关资料，发现了 systemd 是一个比 init 更为先进的初始化系统，是系统启动的第一个进程，从某个 Linux 版本开始 systemd 就取代了 init。因此，我的 ubuntu20 系统中第一个进程时 systemd 而不是 init。

第二个印象深刻的问题就是软链接的问题。一开始进行相关的实验，为了方便我在建立软链接时输入的路径都是相对路径。虽然链接文件可以建立，但是我却无法通过它打开源文件。用编辑器打开该链接文件，发现它是空文件。这时我在图形化界面打开链接文件所在目录，发现该文件的icon上有红色的叉，点击后提示我原文件不存在。建立软链接命令本身不难掌握，但是看过的资料都没有强调过要输入绝对路径。几番尝试之后，我发现只有用绝对路径才能建立软链接。目前还不知道这是否与系统有关。

本次实验量很大，但是也让我掌握了很多命令的使用，也让我对正则表达式、管道等工具有了更深的了解。实验中会遇到很多问题，有许多都是可以通过指定选项解决的，需要熟练掌握 `man` 的使用以及互联网。