

shell 命令

吴梦埏 计算机科学与技术学院
3180105091 995862798@qq.com

日期：2020 年 5 月 31 日

1 实验环境

操作系统：Microsoft Windows 10 家庭中文版

处理器：Intel(R) Core(TM) i7-8750H CPU @ 2.20GHz 2.21GHz

内存：16.0GB

虚拟机：VMware

Linux 版本：Ubuntu 19.04

2 实验内容和结果及分析

2.1 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？

2.1.1 软件许可证

软件许可证（Software License）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。

2.1.2 许可证分类

常用的软件许可证包括：GPL、BSD 许可证、私权软件许可证等等。根据许可证使用时间可以做如下分类。

- 终身许可证

是一旦与软件开发商达成协议，签订合同后可终身无限制的使用该软件。此类许可证多见于个人用户领域。

- 年度许可证

指的是客户与软件开发商签订协议，按年付费来使用该软件。此类软件许可证多见于商业软件领域。相比终身许可证，年度许可证不太像是购买软件，而更像是租赁软件使用，不过却更为灵活。

2.1.3 举例分析

- GPL 许可证:

GPL 许可证的核心含义是, 允许任何人观看、修改, 并散播程序软件里的原始程序码, 条件是如果要发布修改后的版本需要连源代码一起公布。

- BSD 许可证:

BSD 授权许可证 (Free BSD Copyright Information) 具有多种授权许可证。用户可以对软件任意处理, 只需要在软件中注明其来源。用户具有更大的自由度来处置软件。BSD 授权许可证没有实现" 通透性" 自由, 也就是其不保证软件源代码开放的连续性。

- GPL 与 BSD 的区别:

GPL 许可证希望代码是自由的, 凡是使用了 GPL 的代码的项目都必须开放。也就是说 GPL 代码只能在 GPL 项目中使用, GPL 要求 GPL 代码开放的继承性, 即确保所有 GPL 代码的派生工作的开放性。而 BSD 许可证则相对宽泛的多, 允许自由使用 BSD 代码, 使用者有权对代码所做的修改进行保密, 不要求 BSD 代码派生工作的开放性。

2.2 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

```
ydream@ydream:~$ ls
Desktop  Downloads      IoTSystem  Pictures  Templates
Documents examples.desktop Music       Public    Videos
ydream@ydream:~$ pwd
/home/ydream
ydream@ydream:~$ xy
xy: command not found
ydream@ydream:~$ cd ..
ydream@ydream:/home$ pwd
/home
ydream@ydream:/home$ cd
ydream@ydream:~$ pwd
/home/ydream
ydream@ydream:~$ cd /usr/include
ydream@ydream:/usr/include$ ls
aio.h      expat_external.h  link.h      nl_types.h      semaphore.h    threads.h
aliases.h  expat.h           linux       nss.h            setjmp.h      time.h
alloca.h   fcntl.h           locale.h    obstack.h        sgTTY.h        ttyent.h
argp.h     features.h        malloc.h    openvpn          shadow.h       uchar.h
argz.h     fenv.h            math.h      paths.h          signal.h       ucontext.h
ar.h       fmtmsg.h          mcheck.h    poll.h           sound          ulimit.h
arpa       fnmatch.h         memory.h    printf.h         spawn.h        unistd.h
asm-generic fstab.h           misc        proc_service.h   stab.h         utime.h
assert.h   fts.h             mntent.h    pthread.h        stdc-predef.h  utmp.h
byteswap.h ftw.h             monetary.h  ptty.h           stdint.h       utmpx.h
c++        gccv.h            mqqueue.h   pwd.h            stdio_ext.h    values.h
complex.h  getopt.h          mtd         python2.7        stdio.h        video
cpio.h     glob.h            net         python3.7m       stdlib.h       wait.h
crypt.h    gnumake.h         netash      rdma             string.h       wchar.h
ctype.h    gnu-versions.h   netatalk    re_comp.h        strings.h      wctype.h
dirent.h   grp.h            netax25     regex.h          stropts.h     wordexp.h
dlfcn.h    gshadow.h        netdb.h     regext.h         sudo_plugin.h  X11
drm         iconv.h          neteconet   reglib           syscall.h      x86_64-linux-gnu
elf.h      ifaddrs.h        netinet     resolv.h         sysaux.h       xen
endian.h   inttypes.h       netipx      rpc              syslog.h       xorg
envz.h     langinfo.h       netiuvc     rpcsvc           tar.h          termio.h
err.h      lastlog.h        netpacket   sched.h          termios.h     tgmth.h
errno.h    libgen.h         netrose     scsi             thread_db.h
error.h    libintl.h        nfs         search.h
execinfo.h limits.h
```

分析:

1. \$ ls

ls(list) 显示当前目录中的文件名字, 不加参数的话它只显示除隐藏文件以外的所有文件和

目录的名字

2. \$ pwd

pwd(print working directory) 显示当前所在工作目录的全路径

3. \$ xy

没有改命令

4. \$ cd..

cd(change directory) 切换当前工作目录至 dirName(目录参数), 参数为.. 表示返回上一级目录

5. \$ pwd

上一步返回了上一层目录, 因此这次 pwd 显示的结果是上一次的上一级目录

6. \$ cd

cd 后无参数, 切换至使用者的 Home 目录

7. \$ pwd

可以验证上一结果

8. \$ cd /usr/include

进入 usr/include 目录

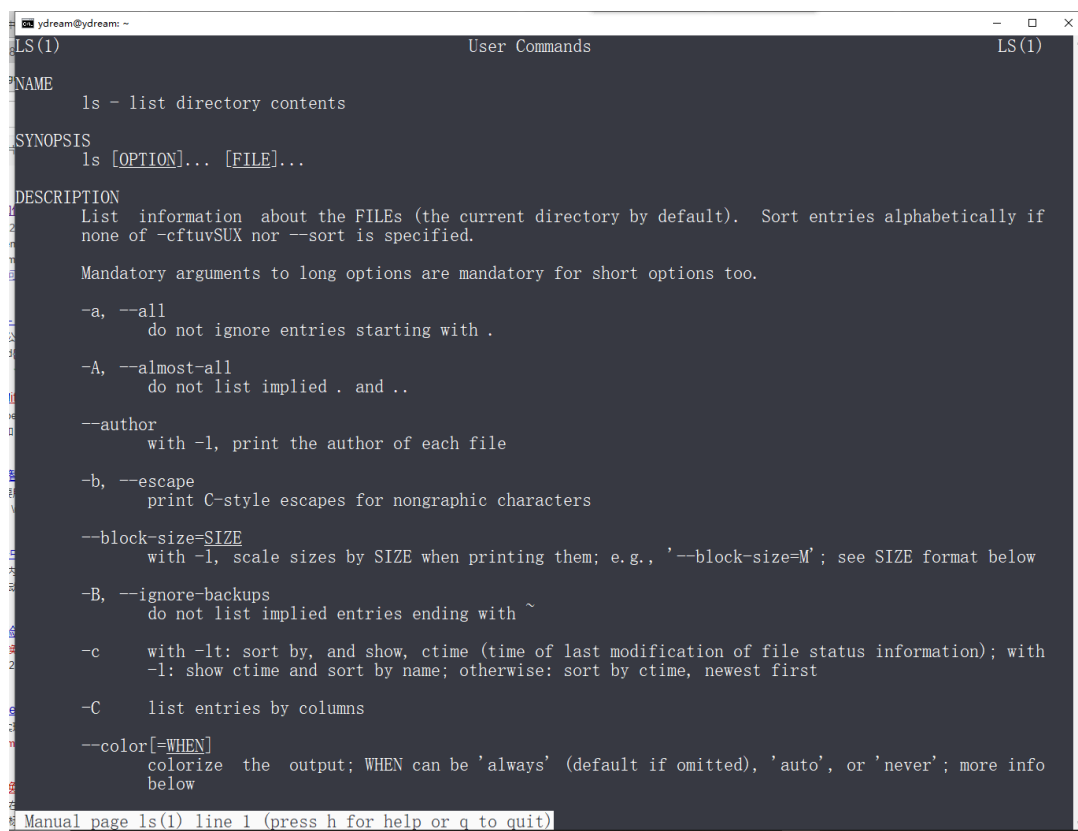
9. \$ ls

显示/usr/include 目录下文件和目录名称

10. \$ cd

切换至 Home

2.3 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。



```
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 specified.
  Mandatory arguments to long options are mandatory for short options too.
  -a, --all
    do not ignore entries starting with .
  -A, --almost-all
    do not list implied . and ..
  --author
    with -l, print the author of each file
  -b, --escape
    print C-style escapes for nongraphic characters
  --block-size=SIZE
    with -l, scale sizes by SIZE when printing them; e.g., '--block-size=M'; see SIZE format below
  -B, --ignore-backups
    do not list implied entries ending with ~
  -c
    with -lt: sort by, and show, ctime (time of last modification of file status information); with -l: show ctime and sort by name; otherwise: sort by ctime, newest first
  -C
    list entries by columns
  --color[=WHEN]
    colorize the output; WHEN can be 'always' (default if omitted), 'auto', or 'never'; more info below
Manual page ls(1) line 1 (press h for help or q to quit)
```

图 1: man ls

```
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，以便更改忘记了的密码。

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

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

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

    • 小写字母
    • 数字 0 到 9
    • 标点符号

    必须要留意不能包含系统默认的擦除和杀死字符。passwd 会拒绝复杂度不满足要求的密码。

    关于用户密码的提示
    密码的安全性依赖于加密算法的强度和密钥空间的大小。旧的 UNIX 系统加密算法基于 NBS DES 算法。推荐使用更新的算法(查看 ENCRYPT_METHOD)。密钥空间的大小依赖于选择的密码的随机性。

    由于粗心或处理选择密码，会危及密码的安全。由于这个原因，您不应该选择出现在词典中或者必须要写下来才能记住的密码。密码也不应该是一个名字、许可证号、生日或者街道号。所有这些可以用于猜测来损害系统安全。

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

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

图 2: man passwd

```
Next: stty invocation, Up: Working context

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

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

    pwd [OPTION]...

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

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

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

    If '-L' and '-P' are both given, the last one takes precedence. If
    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 -----
Welcome to Info version 6.5. Type H for help, h for tutorial.
```

图 3: info pwd

```

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

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

GNU coreutils online help: <https://www.gnu.org/software/coreutils/>
Report translation bugs to <https://translationproject.org/team/>
Full documentation at: <https://www.gnu.org/software/coreutils/who>
or available locally via: info '(coreutils) who invocation'

```

图 4: who --help

分析：man 得到的内容比用 help 更多更详细，而且 man 没有内建与外部命令的区分，因为 man 工具是显示系统手册页中的内容，也就是一本电子版的字典，这些内容大多数都是对命令的解释信息，还有一些相关的描述。info 得到的信息比 man 还要多，info 来自自由软件基金会的 GNU 项目，是 GNU 的超文本帮助系统，能够更完整的显示出 GNU 信息。

2.4 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)

```

ydream@ydream: $ whoami
ydream
ydream@ydream: $ uname
Linux
ydream@ydream: $ uname -n
ydream
ydream@ydream: $ uname -p
x86_64

```

1. 您的用户名是什么
ydream
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？
ydream
4. 你计算机系统的 CPU 名字是什么？
x86_64

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

2.5.1 显示以上年份

指令如下

```
1 cal -y 4
2 cal -y 1752
3 cal 7 2012
```

```
ydream@ydream:~$ 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
   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
```

图 5: 显示 4 年日历

```
ydream@ydream:~$ cal -y 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   30 31
 26 27 28 29 30 31     23 24 25 26 27 28 29

      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
```

图 6: 显示 1752 年日历

```
ydream@ydream:~$ 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
```

图 7: 显示 2012 年 7 月日历

2.5.2 1752 年的天数

指令如下

```
1 cal 12 1752 -j
```

用-j 命令得到 1752 年最后一个月份的具体日期在当年的天数排序, 最后一天的天数排序就是 1752 年的总天数

如图, 1752 年共有 355 天


```

ydream@ydream:~$ 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

```

2.6 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。

```

ydream@ydream:~$ uptime
11:23:44 up 20 min, 2 users, load average: 0.18, 0.14, 0.22

```

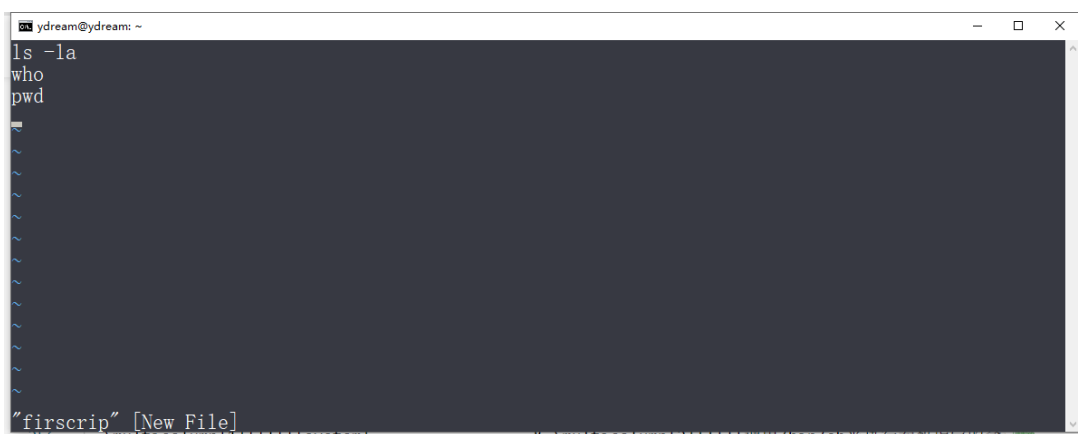
根据结果可知，当前系统已运行 20 分钟，有 2 个用户。这是因为我在虚拟机中登录了一个用户，另外在 windows 系统下用 cmd 利用 ssh 协议远程操控虚拟机。

2.7 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

见附

2.8 vi 编辑器的使用

1. 在 shell 提示符下，输入 vi firscrip 并按 <Enter> 键。vi 的界面将出现在显示屏上；
2. 按 <A> 键，输入 ls -la，并按 <Enter> 键；
3. 输入 who，并按 <Enter> 键；
4. 输入 pwd，再按 <Enter> 键。这时屏幕将如下图所示：



5. 在按 <Esc> 键；
6. 再输入:wq，并按 <Enter> 键；
7. 在 shell 提示符下，输入 bash ./firscrip 并按 <Enter> 键；

```

ydream@ydream:~$ ydream@ydream:~$ vi firscrip
ydream@ydream:~$ ydream@ydream:~$ bash ./firscrip
total 140
drwxr-xr-x 21 ydream ydream 4096 5月 31 15:51 .
drwxr-xr-x 3 root root 4096 5月 19 10:52 ..
-rw----- 1 ydream ydream 5673 5月 23 23:44 .bash_history
-rw-r--r-- 1 ydream ydream 220 5月 19 10:52 .bash_logout
-rw-r--r-- 1 ydream ydream 3929 5月 22 11:23 .bashrc
drwx----- 21 ydream ydream 4096 5月 22 17:48 .cache
drwx----- 20 ydream ydream 4096 5月 22 17:03 .config
drwx----- 3 root root 4096 5月 22 10:35 .dbus
drwxr-xr-x 2 ydream ydream 4096 5月 22 17:39 Desktop
drwxr-xr-x 2 ydream ydream 4096 5月 20 19:38 Documents
drwxr-xr-x 3 ydream ydream 4096 5月 22 18:31 Downloads
-rw-r--r-- 1 ydream ydream 8980 5月 19 10:52 examples.desktop
-rw-rw-r-- 1 ydream ydream 16 5月 31 15:51 firscrip
-rw-r--r-- 1 ydream ydream 26 5月 23 23:27 .gitconfig
drwx----- 3 ydream ydream 4096 5月 20 19:38 .gnupg
-rw----- 1 ydream ydream 2548 5月 31 11:04 .ICEauthority
drwxr-xr-x 3 ydream ydream 4096 5月 22 17:24 IoTSystem
drwx----- 3 ydream ydream 4096 5月 20 19:38 .local
drwx----- 5 ydream ydream 4096 5月 21 20:34 .mozilla
drwxr-xr-x 2 ydream ydream 4096 5月 20 19:38 Music
-rw-r--r-- 1 ydream ydream 57 5月 22 11:07 .pam_environment
drwxr-xr-x 2 ydream ydream 4096 5月 20 19:38 Pictures
drwx----- 3 ydream ydream 4096 5月 21 20:40 .pki
-rw-r--r-- 1 ydream ydream 807 5月 19 10:52 .profile
drwxr-xr-x 2 ydream ydream 4096 5月 20 19:38 Public
-rw----- 1 ydream ydream 12 5月 22 10:19 .python_history
drwx----- 2 ydream ydream 4096 5月 22 17:25 .ssh
-rw-r--r-- 1 ydream ydream 0 5月 21 20:32 .sudo_as_admin_successful
drwxr-xr-x 2 ydream ydream 4096 5月 20 19:38 Templates
drwxrwxr-x 3 ydream ydream 4096 5月 22 12:06 .tooling
drwxr-xr-x 2 ydream ydream 4096 5月 20 19:38 Videos
drwxrwxr-x 3 ydream ydream 4096 5月 22 10:09 .vscode
-rw----- 1 ydream ydream 52 5月 30 11:07 .xauthority
ydream :0 2020-05-31 11:03 (:0)
ydream pts/1 2020-05-31 11:06 (192.168.247.1)
/home/ydream

```

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？

共有 140 个文件，有两个不同的用户使用该计算机，当前工作目录是/home/ydream

2.9 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）

```

1  cd
2  mkdir temp professional personal
3  cd professional/
4  mkdir courses societies
5  cd ~/personal/
6  mkdir funstuff taxes
7  cd ~/professional/courses
8  mkdir major general
9  cd ~/professional/societies/
10 mkdir ieee acm
11 cd ~/professional/courses/major
12 mkdir cs213 cs381 cs475

```

```
13 cd ~/professional/courses/major/cs381
14 mkdir notes labs programs
```

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

2.10 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。

绝对路径为：/home/ydream

```
ydream@ydream:~$ cd
ydream@ydream:~$ pwd
/home/ydream
```

2. acm 目录的绝对路径是什么？

/home/ydream/professional/societies/acm

3. 给出 acm 目录的两个相对路径。

../societies/acm

../../professional/societies/acm

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

```
ydream@ydream:~$ cd professional/courses
ydream@ydream:~/professional/courses$ cd major/cs381/labs
ydream@ydream:~/professional/courses/major/cs381/labs$ pwd
/home/ydream/professional/courses/major/cs381/labs
```

5. 给出获得你的主目录三个不同的命令。

```
1 echo $HOME
2 cd;pwd
3 egrep ^$(whoami) /etc/passwd|awk -F: '{print $6}'
```

```
ydream@ydream:~/professional$ echo $HOME
/home/ydream
ydream@ydream:~/professional$ cd
ydream@ydream:~$ pwd
/home/ydream
ydream@ydream:~$ egrep ^$(whoami) /etc/passwd|awk -F: '{print $6}'
/home/ydream
```

2.11 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？

```
ydream@ydream:~$ ls -d .*
.          .bashrc    .gitconfig .mozilla   .python_history .vscode
..         .cache     .gnupg     .pam_environment .ssh            .Xauthority
.bash_history .config    .ICEauthority .pki       .sudo_as_admin_successful
.bash_logout .dbus      .local     .profile   .tooling
```

2.12 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。

```
1 cd /
2 ls -li -a
3 cd
4 ls -li -a
```

```
ydream@ydream:~$ cd /
ydream@ydream:/$ ls -li -a
2 .          1048577 etc          16 lib64        1 proc          12 swapfile
2 ..         393217  home         17 libx32       262146 root          1 sys
13 bin        19 initrd.img  11 lost+found  2 run           393218 tmp
131073 boot    20 initrd.img.old 655361 media      18 sbin         786435 usr
267733 cdrom   14 lib        786433 mnt          917506 snap      1179649 var
2 dev        15 lib32      917505 opt          131074 srv          21 vmlinuz
```

```
ydream@ydream:/$ cd
ydream@ydream:~$ ls -li -a
410006 .          393369 firscrip    554890 professional
393217 ..         430830 .gitconfig      410007 .profile
397230 .bash_history 393363 .gnupg          395730 Public
410009 .bash_logout 393358 .ICEauthority   400441 .python_history
402991 .bashrc      394243 IoTSystem      554772 .ssh
393365 .cache       393366 .local          393269 .sudo_as_admin_successful
393298 .config      395567 .mozilla        554766 temp
402927 .dbus        395732 Music          395729 Templates
395724 Desktop   396023 .pam_environment 425965 .tooling
395731 Documents  554940 personal       395734 Videos
395728 Downloads 395733 Pictures     401125 .vscode
410008 examples.desktop 398465 .pki           397241 .Xauthority
```

- / 2
- /home/ydream 410006
- /home/ydream/temp 554766
- /home/ydream/professional 554890
- /home/ydream/personal 554940

2.13 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。

```

ydream@ydream:~/professional/courses/major/cs381/labs$ ls -l
total 4
-rw-rw-r-- 1 ydream ydream 149 5月 31 18:28 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。

```

ydream@ydream:~/professional/courses/major/cs381/labs$ cat lab1
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

```

2.14 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。

```

ydream@ydream:~/professional/courses/major/cs381/labs$ cd /usr/include
ydream@ydream:/usr/include$ ls -l t*.h
-rw-r--r-- 1 root root 3785 4月 8 2019 tar.h
-rw-r--r-- 1 root root 214 4月 8 2019 termio.h
-rw-r--r-- 1 root root 3598 4月 8 2019 terminios.h
-rw-r--r-- 1 root root 31489 4月 8 2019 tgmath.h
-rw-r--r-- 1 root root 16023 4月 8 2019 thread_db.h
-rw-r--r-- 1 root root 6660 4月 8 2019 threads.h
-rw-r--r-- 1 root root 10360 4月 8 2019 time.h
-rw-r--r-- 1 root root 2494 4月 8 2019 ttyent.h

```

2.15 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

2.16 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。

```

1 more +12 largeFile
2 tail -n 5 smallFile
3 tail -n +6 smallFile

```

结果如下

```

ydream@ydream:~$ more +12 largeFile
DESCRIPTION
    Bash is an sh-compatible command language interpreter that executes commands read from
    the standard input or from a file. Bash also incorporates useful features from the
    Korn and C shells (ksh and csh).

    Bash is intended to be a conformant implementation of the Shell and Utilities portion
    of the IEEE POSIX specification (IEEE Standard 1003.1). Bash can be configured to be
    POSIX-conformant by default.

OPTIONS
    All of the single-character shell options documented in the description of the set
    builtin command, including -o, can be used as options when the shell is invoked. In
    addition, bash interprets the following options when it is invoked:

    -c      If the -c option is present, then commands are read from the first non-option
            argument command_string. If there are arguments after the command_string,
            the first argument is assigned to $0 and any remaining arguments are assigned
            to the positional parameters. The assignment to $0 sets the name of the
            shell, which is used in warning and error messages.
    -i      If the -i option is present, the shell is interactive.
    -l      Make bash act as if it had been invoked as a login shell (see INVOCATION be-
            low).
    -r      If the -r option is present, the shell becomes restricted (see RESTRICTED
            SHELL below).
    -s      If the -s option is present, or if no arguments remain after option process-
            ing, then commands are read from the standard input. This option allows the
            positional parameters to be set when invoking an interactive shell or when
            reading input through a pipe.
    -v      Print shell input lines as they are read.
    -x      Print commands and their arguments as they are executed.
    -D      A list of all double-quoted strings preceded by $ is printed on the standard
            output. These are the strings that are subject to language translation when
            the current locale is not C or POSIX. This implies the -n option; no com-
            mands will be executed.
    [+>]0 [shopt_option]
            shopt_option is one of the shell options accepted by the shopt builtin (see
            SHELL BUILTIN COMMANDS below). If shopt_option is present, -0 sets the value
            of that option; +0 unsets it. If shopt option is not supplied, the names and

```

```

ydream@ydream:~$ tail -n 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

```

```

ydream@ydream:~$ 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

```

2.17 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？

拷贝及查看修改时间

```

ydream@ydream:~$ cp smallFile dataFile
ydream@ydream:~$ ls -l dataFile
-rw-rw-r-- 1 ydream ydream 974 5月 31 18:46 dataFile
ydream@ydream:~$ ls -l smallFile
-rw-rw-r-- 1 ydream ydream 974 5月 31 18:36 smallFile

```

保持修改时间不变

```

ydream@ydream:~$ cp -a smallFile dataFile2
ydream@ydream:~$ ls -l smallFile
-rw-rw-r-- 1 ydream ydream 974 5月 31 18:36 smallFile
ydream@ydream:~$ ls -l dataFile2
-rw-rw-r-- 1 ydream ydream 974 5月 31 18:36 dataFile2

```

inode

```

ydream@ydream:~$ ls -i -a smallFile dataFile
395872 dataFile 393370 smallFile

```

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

```

ydream@ydream:~$ mv dataFile newDataFile
ydream@ydream:~$ ls -i -a newDataFile
395872 newDataFile

```

inode 相同，因为 mv 只改变了文件名。

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？

```

ydream@ydream:~$ mv newDataFile /tmp
ydream@ydream:~$ ls -i /tmp/newdataFile
395872 /tmp/newdataFile

```

inode 相同，因为 mv 只是把文件移动到了其他的路径，文件本身没有发生变化。

2.18 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

```

ydream@ydream:~$ wc smallFile mediumFile largeFile /tmp/newdataFile
 20   120   974 smallFile
 71   216  1996 mediumFile
5319 48540 357292 largeFile
 20   120   974 /tmp/newdataFile
5430 48996 361236 total

```


- 2.19 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。

```
ydream@ydream:~$ touch 1.c 2.c
ydream@ydream:~$ find -name "*.c"
./2.c
./1.c
```

- 2.20 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

```
ydream@ydream:~$ find -newer smallFile
.
./local/share/gvfs-metadata
./local/share/gvfs-metadata/root-7337c594.log
./local/share/gvfs-metadata/root
./local/share/gnome-shell
./local/share/gnome-shell/application_state
./local/share/tracker/data/tracker-store.journal
./2.c
./1.c
./mediumFile
./largeFile
find: './.config/enchant': Permission denied
./config/qv2ray/Qv2ray.conf
find: './.cache/gvfs': Permission denied
./cache/tracker/meta.db-wal
./cache/tracker/meta.db-shm
./cache/tracker/meta.db
find: './.cache/doc': Permission denied
find: './.cache/dconf': Permission denied
find: './.dbus': Permission denied
```

- 2.21 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。

```
ydream@ydream:~$ grep "Linux" ./ -Rn -l
./local/share/app-info/xmls/extensions-web.xml
./local/share/tracker/data/tracker-store.journal
```

- 2.22 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。

```
ydream@ydream:~$ ls -ld /etc/passwd /bin/df / ~
drwxr-xr-x 20 root root 4096 5月 22 10:08 /
-rwxr-xr-x 1 root root 92968 1月 14 2019 /bin/df
-rw-r--r-- 1 root root 2681 5月 22 16:42 /etc/passwd
drwxr-xr-x 23 ydream ydream 4096 5月 31 19:01 /home/ydream
```

文件	文件类型	存取权限	链接数	所有者	组	文件大小
/	d	drwxr-xr-x	20	root	root	4096
/bin/df	-	-rwxr-xr-x	1	root	root	92968
/etc/passwd	-	-rw-r--r--	1	root	root	2681
/home/ydream	d	drwxr-xr-x	23	ydream	ydream	4096

2.23 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。

```
ydream@ydream:~$ chmod u=rwx *
ydream@ydream:~$ chmod o-w *
ydream@ydream:~$ cd temp/
ydream@ydream:~/temp$ mkdir d1 d2 d3
ydream@ydream:~/temp$ cd d1
ydream@ydream:~/temp/d1$ touch f1
ydream@ydream:~/temp/d1$
```

2.24 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。

```
ydream@ydream:~$ chmod 111 temp
ydream@ydream:~$ ls -ld temp/
d--x--x--x 5 ydream ydream 4096 5月 31 19:16 temp/
ydream@ydream:~$ ls -l temp/
ls: cannot open directory 'temp/': Permission denied
ydream@ydream:~$ chmod 511 temp/
ydream@ydream:~$ ls -l temp/
total 12
drwxrwxr-x 2 ydream ydream 4096 5月 31 19:16 d1
drwxrwxr-x 2 ydream ydream 4096 5月 31 19:16 d2
drwxrwxr-x 2 ydream ydream 4096 5月 31 19:16 d3
```

2.25 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:

```
ydream@ydream:~$ umask
0002
ydream@ydream:~$ cd temp/d1
ydream@ydream:~/temp/d1$ mkdir d11
ydream@ydream:~/temp/d1$ touch f2
ydream@ydream:~/temp/d1$ vi hello.c
ydream@ydream:~/temp/d1$ ydream@ydream:~/temp/d1$ gcc -o greeting hello.c
hello.c:2:1: warning: return type defaults to 'int' [-Wimplicit-int]
main(void)
^~~~~
```

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	644	644	755	755/-
077	600	-	-	-/700

2.26 删除 /temp 目录下的所有文件和目录。给出会话过程。

```
ydream@ydream:~$ rm -rf temp
```

2.27 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。

```
ydream@ydream:~$ cp smallFile ~/temp/d1
ydream@ydream:~$ ls -l smallFile
-rwxrw-r-- 1 ydream ydream 974 5月 31 18:36 smallFile
ydream@ydream:~$ ls -li smallFile
393370 smallFile
ydream@ydream:~$ wc smallFile
20 120 974 smallFile
```

2.28 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。

```
ydream@ydream:~/temp/d2$ ln ../d1/smallFile newFile.hard
ydream@ydream:~/temp/d2$ ls -li
total 4
665438 -rwxrw-r-- 2 ydream ydream 974 5月 31 19:29 newFile.hard
ydream@ydream:~/temp/d2$ cd ../d1
ydream@ydream:~/temp/d1$ ls -li
total 4
665438 -rwxrw-r-- 2 ydream ydream 974 5月 31 19:29 smallFile
```

根据 inode 来判断，两者 inode 是相同的，因此是用一个文件的两个名字

2.29 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。

使用硬链接文件 smallFile.hard 显示 smallFile 文件的内容：

```

ydream@ydream:~/temp/d1$ more smallFile ../d2/newFile.hard
:::~::~:
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.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
:::~::~:
../d2/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

```

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？

```

ydream@ydream:~/temp/d1$ chmod u-r smallFile
ydream@ydream:~/temp/d1$ more smallFile ../d2/newFile.hard
more: cannot open smallFile: Permission denied
more: cannot open ../d2/newFile.hard: Permission denied

```

SmallFile 操作会影响到 newFile.hard

对 smallFile 文件增加读权限，再一次显示文件内容，发生了什么？

```
ydreame@ydream:~/temp/d1$ chmod u+r smallFile
ydream@ydream:~/temp/d1$ more smallFile ../d2/newFile.hard
::::::::::::
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.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
::::::::::::
../d2/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
```

文件又可以显示了

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

```

ydream@ydream:~/temp/d1$ rm smallFile
ydream@ydream:~/temp/d1$ more ../d2/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	

文件 `.hard` 可以显示，硬连接 == 拷贝，所以源文件被删除后，还有几个备份。硬链接能跨不同的文件系统，记录的是目标的 `node`。

2.30 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。

```
ydream@ydream:~/temp/d1$ ln -s ~/temp/d1/smallFile ~/temp/d2/smallFile.soft
ydream@ydream:~/temp/d1$ ls -il smallFile ../d2/smallFile.soft
665442 lrwxrwxrwx 1 ydream ydream 30 5月 31 20:02 ../d2/smallFile.soft -> /home/ydream/temp/d1/smallFile
665440 -rwxrw-r-- 1 ydream ydream 974 5月 31 19:29 smallFile
```

不能根据文件大小判断，应当根据 inode 来判断，可以看到两者 inode 不同。

2.31 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。

使用软链接文件 smallFile.soft 显示 smallFile 文件的内容：

```
ydream@ydream:~/temp/d1$ more ../d2/smallFile.soft
FirstName LastName Major GPA Email Phone
John Doe ECE 3.54 doe@jd.home.org 111.222.3333
James Davis ECE 3.71 davis@jd.work.org 111.222.1111
Al Davis CS 2.63 davis@a.lakers.org 111.222.2222
Ahmad Rashid MBA 3.04 ahmad@mba.org 111.222.4444
Sam Chu ECE 3.68 chu@sam.ab.com 111.222.5555
Arun Roy SS 3.86 roy@ss.arts.edu 111.222.8888
Rick Marsh CS 2.34 marsh@a.b.org 111.222.6666
James Adam CS 2.77 jadam@a.b.org 111.222.7777
Art Pohm ECE 4.00 pohm@ap.a.org 111.222.9999
John Clark ECE 2.68 clark@xyz.ab.com 111.111.5555
Nabeel Ali EE 3.56 ali@ee.eng.edu 111.111.8888
Tom Nelson ECE 3.81 nelson@tn.abc.org 111.111.6666
Pat King SS 3.77 king@pk.xyz.org 111.111.7777
Jake Zulu CS 3.00 zulu@jz.sa.org 111.111.9999
John Lee EE 3.64 jlee@j.lee.com 111.111.2222
Sunil Raj ECE 3.86 raj@sr.cs.edu 111.111.3333
Charles Right EECS 3.31 right@cr.abc.edu 111.111.4444
Diane Rover ECE 3.87 rover@dr.xyz.edu 111.111.5555
Aziz Inan EECS 3.75 ainan@ai.abc.edu 111.111.1111
```

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？

再对 smallFile 文件增加读权限，再一次显示文件内容，发生了什么？

```

ydream@ydream:~/temp/d1$ chmod u-r smallFile
ydream@ydream:~/temp/d1$ more ../d2/smallFile.soft
more: cannot open ../d2/smallFile.soft: Permission denied
ydream@ydream:~/temp/d1$ chmod u+r smallFile
ydream@ydream:~/temp/d1$ more ../d2/smallFile.soft

```

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

SmallFile 操作会影响到 smallFile.soft

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。

```

ydream@ydream:~/temp/d1$ rm smallFile
ydream@ydream:~/temp/d1$ more ../d2/smallFile.soft
more: stat of ../d2/smallFile.soft failed: No such file or directory

```

软连接 == 指向源文件的 path，所以源文件被删除后，虽然有路径但是没有源文件了。软链接能跨不同的文件系统，软连接记录的是目标的 path

2.32 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。

```

ydream@ydream:~$ ps -ef | wc -l
298
ydream@ydream:~$ ps -ef | grep init bash ps
grep: bash: No such file or directory
grep: ps: No such file or directory
ydream@ydream:~$ ps -ef | grep init
root      1      0  0 11:02 ?        00:00:10 /sbin/init splash
ydream    7972  4277  0 21:09 pts/1    00:00:00 grep --color=auto init
ydream@ydream:~$ ps -ef | wc -l
297
ydream@ydream:~$ ps -ef | grep init
root      1      0  0 11:02 ?        00:00:10 /sbin/init splash
ydream    7976  4277  0 21:09 pts/1    00:00:00 grep --color=auto init
ydream@ydream:~$ ps -ef | grep ps
root      999      1  0 11:03 ?        00:00:00 /usr/sbin/cups-browsed
root     1025      1  0 11:03 ?        00:00:00 /usr/sbin/cupsd -l
lp       1027    1025  0 11:03 ?        00:00:00 /usr/lib/cups/notifier/dbus dbus://
whoopsie 1138      1  0 11:03 ?        00:00:00 /usr/bin/whoopsie -f
kernoops 1142      1  0 11:03 ?        00:00:00 /usr/sbin/kerneloops --test
kernoops 1145      1  0 11:03 ?        00:00:00 /usr/sbin/kerneloops
ydream    2653   2303  0 11:04 tty2    00:00:00 /usr/lib/tracker/tracker-miner-apps
ydream    7977  4277  0 21:10 pts/1    00:00:00 ps -ef
ydream    7978  4277  0 21:10 pts/1    00:00:00 grep --color=auto ps
ydream@ydream:~$ ps -ef | grep bash
ydream    4277  4276  0 11:06 pts/1    00:00:00 -bash
ydream    7980  4277  0 21:10 pts/1    00:00:00 grep --color=auto bash

```

2.33 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。

```

ydream@ydream:~$ cat > test <& stop
> Time for lunch!
> stop
ydream@ydream:~$ sleep 10s;cat test
Time for lunch!

```

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

```

ydream@ydream:~$ date&uname -a&who&ps
[1] 7984
[2] 7985
[3] 7986
Linux ydream 5.0.0-13-generic #14-Ubuntu SMP Mon Apr 15 14:59:14 UTC 2019 x86_64 x86_64 x86_64 GNU
U/Linux
2020年 05月 31日 星期日 21:13:19 CST
ydream :0          2020-05-31 11:03 (:0)
ydream pts/1      2020-05-31 11:06 (192.168.247.1)
  PID TTY          TIME CMD
  4277 pts/1    00:00:00 bash
  7987 pts/1    00:00:00 ps
[1] Done          date
[2]- Done          uname -a
[3]+ Done          who

```

- 2.35 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。

```
ydream@ydream:~$ date && uname -a && who && ps
2020年 05月 31日 星期日 21:14:51 CST
Linux ydream 5.0.0-13-generic #14-Ubuntu SMP Mon Apr 15 14:59:14 UTC 2019 x86_64 x86_64 x86_64 GNU/Linux
ydream  :0          2020-05-31 11:03 (:0)
ydream  pts/1      2020-05-31 11:06 (192.168.247.1)
  PID TTY          TIME CMD
 4277 pts/1      00:00:00 bash
 7991 pts/1      00:00:00 ps
```

- 2.36 在 shell 下执行下面的命令。3 个 **pwd** 命令的运行结果是什么？

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

- 2.37 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）

```
ydream@ydream:~$ mkdir foobar
ydream@ydream:~$ find foobar 2>/dev/null
foobar
```

- 2.38 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。

```
ydream@ydream:~$ find foobar >foobar.path 2>/dev/null
ydream@ydream:~$ cat foobar.path
foobar
```

- 2.39 **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。

```
ydream@ydream:~$ cat student.records 1>output.data 2>error.log
```


- 2.40 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?

```
ydream@ydream:~$ vi fdata
ydream@ydream:~$ ydream@ydream:~$ bash < ./fdata > ./fout
-bash: ./fdata: No such file or directory
ydream@ydream:~$ bash < ./fdata > ./fout
ydream@ydream:~$ vi ./fout
```

```
The time now is:2020年 05月 31日 星期日 21:26:24 CST
The users presently logged on are:ydream :0      2020-05-31 11:03 (:0)
ydream pts/1      2020-05-31 11:06 (192.168.247.1)
```

- 2.41 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。

```
ydream@ydream:~$ ydream@ydream:~$ ls -l |wc -w
236
ydream@ydream:~$ ls -l |wc -l
27
ydream@ydream:~$ ls -l |wc -m
1458
```

- 2.42 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?

```
ydream@ydream:/bin$ find /bin -type f|wc -l
0
ydream@ydream:/bin$ find /bin -type d|wc -l
0
ydream@ydream:/bin$ find /bin -type l|wc -l
1
```

3 讨论、心得

实验过程中, 了解了 **Linux** 的基本语法和使用方法, 掌握了一定的基础命令使用方法, 也满足了自己在这一方面的好奇心。在真正经过实际操作之后, 对于各种操作有了更深的理解。

实验中对于权限的设置不是很清晰, 关于文件权限的题目经常搞错, 只得全部重新来。查找了很多相关资料, 对权限的授予有了更加深刻的理解。

设置软链接的实验, 由于当前工作目录在 **/home/ydream/temp/d1** 设置软链接的时候用的地址是相对地址, 因此软连接失效。在查阅过资料后, 突然明白了这个问题。

Linux 是一种十分具有魅力的操作系统, 我现在的使用还不够得心应手。实践出真知, 只有真正逼迫自己使用该系统, 才能够真正学会使用它。希望自己能够获得更大的提升

4 附

命令	命令功能的简要描述	实例
touch	用于把已存在文件的时间标签更新为系统当前的时间（默认方式），它们的数据将原封不动地保留下来 用来创建新的空文件	a 改变档案的读取时间记录。
		m 改变档案的修改时间记录。
		c 假如目的档案不存在，不会建立新的档案。与 --no-create 的效果一样。
		f 不使用，是为了与其他 unix 系统的相容性而保留。
		r 使用参考档的时间记录，与 --file 的效果一样。
		d 设定时间与日期，可以使用各种不同的格式。
		t 设定档案的时间记录，格式与 date 指令相同。
		--no-create 不会建立新档案。
cp	复制文件或目录	-a：此选项通常在复制目录时使用，它保留链接、文件属性，并复制目录下的所有内容。其作用等于dpR参数组合。
		-d：复制时保留链接。这里所说的链接相当于Windows系统中的快捷方式。
		-f：覆盖已经存在的目标文件而不给出提示。
		-i：与-f选项相反，在覆盖目标文件之前给出提示，要求用户确认是否覆盖，回答"y"时目标文件将被覆盖。
		-p：除复制文件的内容外，还把修改时间和访问权限也复制到新文件中。
		-r：若给出的源文件是一个目录文件，此时将复制该目录下所有的子目录和文件。
		-l：不复制文件，只是生成链接文件。
mv	移动文件或者将文件改名，经常用来备份文件或者目录。	-b：若需覆盖文件，则覆盖前先行备份。
		-f：force 强制的意思，如果目标文件已经存在，不会询问而直接覆盖；
		-i：若目标文件 (destination) 已经存在时，就会询问是否覆盖！
		-u：若目标文件已经存在，且 source 比较新，才会更新(update)
		-t：--target-directory=DIRECTORY move all SOURCE arguments into DIRECTORY，即指定mv的目标目录，该选项适用于移动多个源文件到一个目录的情况，此时目标目录在前，源文件在后。
rm	删除文件或目录	-d --directory 删除可能仍有数据的目录 (只限超级用户)
		-f --force 略过不存在的文件，不显示任何信息，强制删除

命令	命令功能的简要描述	实例
		-i --interactive 进行任何删除操作前必须先确认
		-r/R --recursive 同时删除该目录下的所有目录层
mkdir	创建目录，要求创建目录的用户在当前目录中具有写权限，并且指定的目录名不能是当前目录中已有的目录	-m --mode=模式，设定权限
		-p --parents 递归创建目录
who	显示关于当前在本地系统上的所有用户的信息	-H或--heading：显示各栏位的标题信息列
		-i或-u或--idle：显示闲置时间，若该用户在前一分钟之内有进行任何动作，将标示成"."号，如果该用户已超过24小时没有任何动作，则标示出"old"字符串
		-m：此参数的效果和指定"am i"字符串相同
		-q或--count：只显示登入系统的帐号名称和总人数
		-s：此参数将忽略不予处理，仅负责解决who指令其他版本的兼容性问题
		-w或-T或--mesg或--message或--writable：显示用户的信息状态栏
ls	显示当前目录中的文件名字	ls -a：显示当前目录中的所有文件，包含隐藏文件
		ls -l：显示文件及其详细信息
cd	切换当前工作目录至 dirName(目录参数)	cd：进入用户主目录
		cd ~：进入用户主目录
		cd -：返回进入此目录之前所在目录
		cd ..：返回上一级目录
		cd ../../：返回上两级目录
		cd !\$：把上个命令的参数作为cd 参数使用
		cd /：进入根目录
		cd .：当前目录
pwd	显示当前所在工作目录的全路径	pwd [--help][--version]
		参数说明: --help 在线帮助。 --version 显示版本信息
open	open用于打开一个文件，通过设置不同的flag，可以让进程只读，只写，可读/可写等操作	
read	用于从标准输入读取数值。read 内部命令被用来从标准输入读取单行数据。这个命令可以用来读取键盘输入，当使用重定向的时候，可以读取文件中的一行数据	
write	write函数向打开的设备或文件中写数据	
close	关闭一个已经打开的文件	

命令	命令功能的简要描述	实例
pipe	把两个进程之间的标准输入和标准输出连接起来，从而提供一种让多个进程间通信的方法	
socket	从客户端发送一条消息后，服务端接收这条消息，并在服务端显示	
mkfifo	创建一个命名管道	
system	调用/bin/sh来执行参数指定的命令	
printf	格式化输出	