

# 实验报告

实验名称	实验一 Linux 常用命令（一）		
实验教室	丹青 922	实验日期	2023 年 3 月 8 日
学 号	2021223180	姓 名	廖若辰星
专业班级	计算机科学与技术 04 班		
指导教师	卢洋		

东北林业大学  
信息与计算机科学技术实验中心

## 一、 实验目的

- 1、掌握Linux下文件和目录操作命令：cd、ls、mkdir、rmdir、rm
- 2、掌握Linux下文件信息显示命令：cat、more、head、tail
- 3、掌握Linux下文件复制、删除及移动命令：cp、mv
- 4、掌握 Linux 的文件排序命令：sort

## 二、 实验环境

- (1) 计算机的硬件配置 PC 系列微机。
- (2) 计算机的软件配置 VMware 虚拟机软件及 Ubuntu 虚拟机。

### 三、 实验内容及结果

1. 使用命令切换到/etc 目录，并显示当前工作目录路径

```
chenxing@DESKTOP-UK74CBE:~$ cd /etc
chenxing@DESKTOP-UK74CBE:/etc$ pwd
/etc
chenxing@DESKTOP-UK74CBE:/etc$
```

2. 使用命令显示/home/lyj 目录下所有文件目录的详细信息，包括隐藏文件。

```
chenxing@DESKTOP-UK74CBE:/home$ cd chenxing
chenxing@DESKTOP-UK74CBE:~$ ls -a
- .a.swp      .bashrc      .profile      .viminfo      tset.txt
. .bash_history .lessht      .sudo_as_admin_successful  a
.. .bash_logout .motd_shown  .vim           test.txt
chenxing@DESKTOP-UK74CBE:~$
```

3. 使用命令创建目录/home/lyj/linux，然后删除该目录。

```
chenxing@DESKTOP-UK74CBE:~$ mkdir linux
chenxing@DESKTOP-UK74CBE:~$ ls -a
- .a.swp      .bashrc      .profile      .viminfo      test.txt
. .bash_history .lessht      .sudo_as_admin_successful  a      tset.txt
.. .bash_logout .motd_shown  .vim           linux
chenxing@DESKTOP-UK74CBE:~$
```

```
chenxing@DESKTOP-UK74CBE:~/linux$ cd ..
chenxing@DESKTOP-UK74CBE:~$ rm -i linux
rm: cannot remove 'linux': Is a directory
chenxing@DESKTOP-UK74CBE:~$ rm -r Linux
rm: cannot remove 'Linux': No such file or directory
chenxing@DESKTOP-UK74CBE:~$ rm -r linux
chenxing@DESKTOP-UK74CBE:~$ ls -a
- .a.swp      .bashrc      .profile      .viminfo      tset.txt
. .bash_history .lessht      .sudo_as_admin_successful  a
.. .bash_logout .motd_shown  .vim           test.txt
chenxing@DESKTOP-UK74CBE:~$
```

4. 使用命令 cat 用输出重定向在/home/lyj 目录下创建文件 abc，文件内容为“Hello, Linux!”，并查看该文件的内容

```
chenxing@DESKTOP-UK74CBE:~$ cat > abc
hello, Linux!
chenxing@DESKTOP-UK74CBE:~$ cat abc
hello, Linux!
chenxing@DESKTOP-UK74CBE:~$
```

5、使用命令创建目录/home/lyj/ak，然后将/home/lyj/abc文件复制到该目录下，最后将该目录及其目录下的文件一起删除。

```
chenxing@DESKTOP-UK74CBE:~$ mkdir ak
chenxing@DESKTOP-UK74CBE:~$ ls -a
- .a.swp      .bashrc      .profile      .viminfo      ak
. .bash_history .lessht      .sudo_as_admin_successful a      test.txt
.. .bash_logout .motd_shown .vim          abc      tset.txt

chenxing@DESKTOP-UK74CBE:~$ cp abc ak
chenxing@DESKTOP-UK74CBE:~$ cd ak
chenxing@DESKTOP-UK74CBE:~/ak$ ls-a
ls-a: command not found
chenxing@DESKTOP-UK74CBE:~/ak$ ls -a
. .. abc
chenxing@DESKTOP-UK74CBE:~/ak$ rm -ri ak
rm: cannot remove 'ak': No such file or directory
chenxing@DESKTOP-UK74CBE:~/ak$ cd ..
chenxing@DESKTOP-UK74CBE:~$ rm -ri ak
rm: descend into directory 'ak'? y
rm: remove regular file 'ak/abc'? y
rm: remove directory 'ak'? y
chenxing@DESKTOP-UK74CBE:~$ ls-a
ls-a: command not found
chenxing@DESKTOP-UK74CBE:~$ ls -a
- .a.swp      .bashrc      .profile      .viminfo      sk
. .bash_history .lessht      .sudo_as_admin_successful a      test.txt
.. .bash_logout .motd_shown .vim          abc      tset.txt
```

6、查看文件/etc/adduser.conf 的前 3 行内容，查看文件/etc/adduser.conf 的最后 5 行内容。

```
chenxing@DESKTOP-UK74CBE:~$ cd /etc
chenxing@DESKTOP-UK74CBE:/etc$ head -3 adduser.conf
# /etc/adduser.conf: `adduser' configuration.
# See adduser(8) and adduser.conf(5) for full documentation.

chenxing@DESKTOP-UK74CBE:/etc$ tail -5 adduser.conf
# check user and group names also against this regular expression.
#NAME_REGEX="^[a-z][-a-z0-9_]*\$"

# use extrausers by default
#USE_EXTRAUSERS=1
```

7、分屏查看文件/etc/adduser.conf 的内容。

```
chenxing@DESKTOP-UK74CBE: /etc
# /etc/adduser.conf: 'adduser' configuration.
# See adduser(8) and adduser.conf(5) for full documentation.
# The DSHELL variable specifies the default login shell on your
# system.
DSHELL=/bin/bash
# The DHOME variable specifies the directory containing users' home
--More-- (8%)
```

8、使用命令cat用输出重定向在/home/lyj目录下创建文件facebook.txt，文件内容为：

google 110 5000

baidu 100 5000

guge 50 3000

sohu 100 4500

```
chenxing@DESKTOP-UK74CBE: ~$ cat > facebook.txt
google 110 5000
baidu 100 5000
guge 50 3000
sohu 100 4500
chenxing@DESKTOP-UK74CBE: ~$ cat facebook.txt
google 110 5000
baidu 100 5000
guge 50 3000
sohu 100 4500
```

9. 第一列为公司名称，第2列为公司人数，第3列为员工平均工资。

利用sort命令完成下列排序：

- (1) 按公司字母顺序排序
- (2) 按公司人数排序
- (3) 按公司人数排序，人数相同的按照员工平均工资升序排序
- (4) 按员工工资降序排序，如工资相同，则按公司人数升序排序
- (5) 从公司英文名称的第2个字母开始进行排序。

```
chenxing@DESKTOP-UK74CBE:/etc/newt$ cd /home/chenxing
chenxing@DESKTOP-UK74CBE:~$ cat > facebook.txt
google 110 5000
baidu 100 5000
guge 50 3000
sohu 100 4500
chenxing@DESKTOP-UK74CBE:~$ cat facebook.txt
google 110 5000
baidu 100 5000
guge 50 3000
sohu 100 4500
chenxing@DESKTOP-UK74CBE:~$ sort -k 1 -t ' ' facebook.txt
baidu 100 5000
google 110 5000
guge 50 3000
sohu 100 4500
chenxing@DESKTOP-UK74CBE:~$ sort -n -k 2 -t ' ' facebook.txt
guge 50 3000
baidu 100 5000
sohu 100 4500
google 110 5000
chenxing@DESKTOP-UK74CBE:~$ sort -n -k 2 -k 3 -t ' ' facebook.txt
guge 50 3000
sohu 100 4500
baidu 100 5000
google 110 5000
chenxing@DESKTOP-UK74CBE:~$ sort -n -k 3r -k 2 -t ' ' facebook.txt
baidu 100 5000
google 110 5000
sohu 100 4500
guge 50 3000
chenxing@DESKTOP-UK74CBE:~$ sort -k 1.2 -t ' ' facebook.txt
baidu 100 5000
sohu 100 4500
google 110 5000
guge 50 3000
```

#### 四、 实验过程分析与讨论

对于一些没那么常用的命令容易忘记,比如 `head`、`more` 这类命令。`sort` 从某一行某一个位置开始这个命令也不熟悉,在再次翻看学习通记住了,当用多个 `-k` 时,仅某列使用 `r/n`,应放在数字后面,如 `-k 2r`,而不能 `-k2 -r`。

#### 五、指导教师意见

指导教师签字：卢洋

# 实验报告

实验名称	实验二 Linux 常用命令（二）		
实验教室	丹青 922	实验日期	2023 年 3 月 15 日
学 号	2021223180	姓 名	廖若辰星
专业班级	计算机科学与技术 04 班		
指导教师	卢洋		

东北林业大学  
信息与计算机科学技术实验中心



## 一、 实验目的

1. 掌握Linux下查找文件和统计文件行数、字数和字节数命令： `find` 、 `wc` ；
2. 掌握Linux下文件打包命令： `tar` ；
3. 掌握Linux下符号链接命令和文件比较命令： `ln` 、 `comm` 、 `diff` ；
4. 掌握 Linux 的文件权限管理命令： `chmod` 。

## 二、 实验环境

- (1) 计算机的硬件配置 PC 系列微机。
- (2) 计算机的软件配置 VMware 虚拟机软件及 Ubuntu 虚拟机。

### 三、 实验内容及结果

#### 1. 查找指定文件

(1) 在用户目录下新建目录 baz ， 在 baz 下新建文件 qux ， 并写如任意几行内容；

(2) 在用户目录下查找文件 qux ， 并显示该文件位置信息；

(3) 统计文件 qux 中所包含内容的行数、字数和字节数；

(4) 在用户目录下查找文件 qux ， 并删除该文件；

(5) 查看文件夹 baz 内容， 看一下是否删除了文件 qux 。

```
chenxing@DESKTOP-UK74CBE:~$ mkdir baz
chenxing@DESKTOP-UK74CBE:~$ ls
- a abc baz facebook.txt sk test.txt tset.txt
chenxing@DESKTOP-UK74CBE:~$ cd baz
chenxing@DESKTOP-UK74CBE:~/baz$ cat > qux
ni yu sady dhsujn
123 324 5432
isdidcjusd
9848493
231
78
adweien
```

```
chenxing@DESKTOP-UK74CBE:~$ find -name qux
./baz/qux
```

```
chenxing@DESKTOP-UK74CBE:~$ cd ./baz
chenxing@DESKTOP-UK74CBE:~/baz$ wc qux
7 12 65 qux
```

```
chenxing@DESKTOP-UK74CBE:~/baz$ cd ..
chenxing@DESKTOP-UK74CBE:~$ find . -name qux -exec rm {} \;
chenxing@DESKTOP-UK74CBE:~$ ls
- a abc baz facebook.txt sk test.txt tset.txt
chenxing@DESKTOP-UK74CBE:~$ cd baz
chenxing@DESKTOP-UK74CBE:~/baz$ ls
chenxing@DESKTOP-UK74CBE:~/baz$
```

#### 2. 文件打包

(1) 在用户目录下新建文件夹 path1 ， 在 path1 下新建文件 file1 和 file2 ；

- (2) 在用户目录下新建文件夹 path2 ，在 path2 下新建文件 file3 ；
- (3) 在用户目录下新建文件 file4 ；
- (4) 在用户目录下对文件夹 path1 和 file4 进行打包，生成文件 package.tar ；
- (5) 查看包 package.tar 的内容；
- (6) 向包 package.tar 里添加文件夹 path2 的内容；
- (7) 将包 package.tar 复制到用户目录下的新建文件夹 path3 中；
- (8) 进入 path3 文件夹，并还原包 package.tar 的内容。

```
chenxing@DESKTOP-UK74CBE:~$ mkdir path1
chenxing@DESKTOP-UK74CBE:~$ cd path1
chenxing@DESKTOP-UK74CBE:~/path1$ cat >file1
andeja
adna
2131242
chenxing@DESKTOP-UK74CBE:~/path1$ cat >file2
daufn
fdjid8 sfjd
ivdfhu
6757654654 6546
chenxing@DESKTOP-UK74CBE:~/path1$ cd ..
chenxing@DESKTOP-UK74CBE:~$ mkdir path2
chenxing@DESKTOP-UK74CBE:~$ cd path2
chenxing@DESKTOP-UK74CBE:~/path2$ cat >file3
32 2345 542
5 67 9
sfwe
sfsdfbd
ij
chenxing@DESKTOP-UK74CBE:~/path2$ cd ..
chenxing@DESKTOP-UK74CBE:~$ cat >file4
chenxing@DESKTOP-UK74CBE:~$ tar path1 file4
```

```

chenxing@DESKTOP-UK74CBE:~$ tar -cvf package.tar path1 file4
path1/
path1/file2
path1/file1
file4
chenxing@DESKTOP-UK74CBE:~$ ls
- a abc facebook.txt file4 package.tar path1 path2 sk test.txt tset.txt
chenxing@DESKTOP-UK74CBE:~$ tar tf package.tar
path1/
path1/file2
path1/file1
file4
chenxing@DESKTOP-UK74CBE:~$ tar rf package.tar path2
chenxing@DESKTOP-UK74CBE:~$ tar tf package.tar
path1/
path1/file2
path1/file1
file4
path2/
path2/file3

```

```

chenxing@DESKTOP-UK74CBE:~$ mkdir path3
mkdir: cannot create directory 'path3' : File exists
chenxing@DESKTOP-UK74CBE:~$ rm path3
chenxing@DESKTOP-UK74CBE:~$ mkidir path3
Command 'mkidir' not found, did you mean:
  command 'mkdir' from deb coreutils (8.32-4.1ubuntu1)
Try: sudo apt install <deb name>
chenxing@DESKTOP-UK74CBE:~$ mkdir path3
chenxing@DESKTOP-UK74CBE:~$ cp package.tar path3
chenxing@DESKTOP-UK74CBE:~$ cd pat3
-bash: cd: pat3: No such file or directory
chenxing@DESKTOP-UK74CBE:~$ cd path3
chenxing@DESKTOP-UK74CBE:~/path3$ ls
package.tar
chenxing@DESKTOP-UK74CBE:~/path3$ tar xvf package.tar
path1/
path1/file2
path1/file1
file4
path2/
path2/file3
chenxing@DESKTOP-UK74CBE:~/path3$ ls
file4 package.tar path1 path2
chenxing@DESKTOP-UK74CBE:~/path3$

```

### 3. 符号链接内容

- (1) 新建文件 foo.txt ， 内容为 123 ；
- (2) 建立 foo.txt 的硬链接文件 bar.txt ， 并比较 bar.txt 的内容和 foo.txt 是否相同， 要求用 comm 或 diff 命令；

(3) 查看 foo.txt 和 bar.txt 的 i 节点号 ( inode ) 是否相同;

(4) 修改 bar.txt 的内容为 abc , 然后通过命令判断 foo.txt 与 bar.txt 是否相同;

(5) 删除 foo.txt 文件, 然后查看 bar.txt 文件的 inode 及内容;

(6) 创建文件 bar.txt 的符号链接文件 baz.txt , 然后查看 bar.txt 和 baz.txt 的 inode 号, 并观察两者是否相同, 比较 bar.txt 和 baz.txt 的文件内容是否相同;

(7) 删除 bar.txt , 查看文件 baz.txt , 观察系统给出什么提示信息。

```
chenxing@DESKTOP-UK74CBE: ~$ cat >foo.txt
123
chenxing@DESKTOP-UK74CBE: ~$
chenxing@DESKTOP-UK74CBE: ~$ cat foo.txt
123
```

```
chenxing@DESKTOP-UK74CBE: ~$ ln foo.txt bar.txt
chenxing@DESKTOP-UK74CBE: ~$ ls -l
total 48
-rw-r--r-- 1 chenxing chenxing 100 Mar 16 21:53 -
drwxr-xr-x 3 chenxing chenxing 4096 Mar 13 11:45 a
-rw-r--r-- 1 chenxing chenxing 13 Mar 19 20:53 abc
-rw-r--r-- 2 chenxing chenxing 4 Mar 23 18:54 bar.txt
-rw-r--r-- 1 chenxing chenxing 58 Mar 19 21:09 facebook.txt
-rw-r--r-- 1 chenxing chenxing 0 Mar 23 18:29 file4
-rw-r--r-- 2 chenxing chenxing 4 Mar 23 18:54 foo.txt
-rw-r--r-- 1 chenxing chenxing 10240 Mar 23 18:46 package.tar
-rw-r--r-- 1 chenxing chenxing 13 Mar 19 20:56 sk
-rw-r--r-- 1 chenxing chenxing 63 Mar 18 08:57 test.txt
-rw-r--r-- 1 chenxing chenxing 87 Mar 18 09:32 tset.txt
chenxing@DESKTOP-UK74CBE: ~$ ls -l foo.txt bar.txt
-rw-r--r-- 2 chenxing chenxing 4 Mar 23 18:54 bar.txt
-rw-r--r-- 2 chenxing chenxing 4 Mar 23 18:54 foo.txt
```

```
chenxing@DESKTOP-UK74CBE: ~$ diff foo.txt bar.txt
chenxing@DESKTOP-UK74CBE: ~$ comm foo.txt bar.txt
123
```

```

chenxing@DESKTOP-UK74CBE:~$ ls -i foo.txt bar.txt
2122 bar.txt 2122 foo.txt

chenxing@DESKTOP-UK74CBE:~$ cat > bar.txt
abc
chenxing@DESKTOP-UK74CBE:~$ cat bar.txt
abc
chenxing@DESKTOP-UK74CBE:~$ comm foo.txt bar.txt
      abc
chenxing@DESKTOP-UK74CBE:~$ diff foo.txt bar.txt
chenxing@DESKTOP-UK74CBE:~$ rm foo.txt
chenxing@DESKTOP-UK74CBE:~$ ls -i bar.txt
2122 bar.txt
chenxing@DESKTOP-UK74CBE:~$ cat bar.txt
abc

chenxing@DESKTOP-UK74CBE:~$ ln -s bar.txt baz.txt
chenxing@DESKTOP-UK74CBE:~$ ls -i bar.txt baz.txt
2122 bar.txt 2128 baz.txt
chenxing@DESKTOP-UK74CBE:~$ diff bar.txt baz.txt
chenxing@DESKTOP-UK74CBE:~$ comm bar.txt baz.txt
      abc
chenxing@DESKTOP-UK74CBE:~$ rm bar.txt
chenxing@DESKTOP-UK74CBE:~$ cat baz.txt
cat: baz.txt: No such file or directory

```

#### 4. 权限管理

- (1) 新建文件 qux.txt ；
- (2) 为文件 qux.txt 增加执行权限（所有用户都可以执行）

```

chenxing@DESKTOP-UK74CBE:~$ cat >qux.txt
chenxing@DESKTOP-UK74CBE:~$ chmod -a +x qux.txt
chmod: invalid mode: '-a'
Try 'chmod --help' for more information.
chenxing@DESKTOP-UK74CBE:~$ chmod a+x qux.txt
chenxing@DESKTOP-UK74CBE:~$ ls -l qux.txt
-rwxr-xr-x 1 chenxing chenxing 0 Mar 23 19:35 qux.txt

```

#### 四、 实验过程分析与讨论

对命令掌握不是很熟悉，需要在执行对应命令前复习，需要多看多记相关的命令语句，各种命令适用于不同的环境和问题下，需要多使用才能对这些命令更加熟练。

#### 五、 指导教师意见

指导教师签字：卢洋

# 实验报告

实验名称	实验三 vim 编辑器及 gcc 编译器的使用		
实验教室	丹青 922	实验日期	2023 年 3 月 22 日
学 号	2021223180	姓 名	廖若辰星
专业班级	计算机科学与技术 04 班		
指导教师	卢洋		

东北林业大学  
信息与计算机科学技术实验中心



## 一、 实验目的

掌握 vim 编辑器及 gcc 编译器的使用方法

## 二、 实验环境

(1) 计算机的硬件配置 PC 系列微机。

(2) 计算机的软件配置 VMware 虚拟机软件及 Ubuntu 虚拟机。

## 三、 实验内容及结果

1. vim 编辑器和 gcc 编译器的简单使用:

(1) 在用户目录下新建一个目录，命名为 workspace1 ；

(2) 进入目录 workspace1 ；

(3) 在 workspace1 下用 vim 编辑器新建一个 c 语言程序文件，文件名为 test.c ， 内容为：

```
#include <stdio.h>

int main( )
{
    printf("hello world!\n");
    return 0;
```

```
}
```

(4) 保存 `test.c` 的内容，并退出；

(5) 编译 `test.c` 文件，生成可执行文件 `test`，并执行，查看执行结果。

```
chenxing@DESKTOP-UK74CBE:~$ rm -r workspace1
chenxing@DESKTOP-UK74CBE:~$ mkdir workspace1
chenxing@DESKTOP-UK74CBE:~$ cd workspace1
chenxing@DESKTOP-UK74CBE:~/workspace1$ touch test.c
chenxing@DESKTOP-UK74CBE:~/workspace1$ vi test.c
chenxing@DESKTOP-UK74CBE:~/workspace1$ gcc test.c -o test
chenxing@DESKTOP-UK74CBE:~/workspace1$ ./test
hello world!
```

2. vim 编辑器的详细使用：

(1) 在用户目录下创建一个名为 `workspace2` 的目录；

(2) 进入 `workspace2` 目录；

(3) 使用以下命令：

```
cat /etc/gai.conf > ./gai.conf
```

将文件 `/etc/gai.conf` 的内容复制到当前目录下的新建文件 `gai.conf` 中；

(4) 使用 `vim` 编辑当前目录下的 `gai.conf`；

(5) 将光标移到第 18 行；

(6) 复制该行内容；

(7) 将光标移到最后一行行首；

(8) 粘贴复制行的内容；

(9) 撤销第 8 步的动作；

(10) 存盘但不退出；

- (11) 将光标移到首行;
- (12) 插入模式下输入 "Hello, this is vim world!" ;
- (13) 删除字符串 "this" ;
- (14) 强制退出 vim , 不存盘

```
chenxing@DESKTOP-UK74CBE:~$ mkdir workspace2
chenxing@DESKTOP-UK74CBE:~$ cd workspace2
chenxing@DESKTOP-UK74CBE:~/workspace2$ cp /etc/gai.conf ./gai.conf
chenxing@DESKTOP-UK74CBE:~/workspace2$ vi gai.conf
chenxing@DESKTOP-UK74CBE:~/workspace2$ cat /etc/gai.conf > ./gai.conf
chenxing@DESKTOP-UK74CBE:~/workspace2$ vi gai.conf
chenxing@DESKTOP-UK74CBE:~/workspace2$ vi gai.conf
```

```
#scopev4 ::ffff:169.254.0.0/112 2
#scopev4 ::ffff:127.0.0.0/104 2
#scopev4 ::ffff:0.0.0.0/96 14
# Add another rule to the RFC 3484 label table. See section 2.1 in
```

```
#scopev4 ::ffff:169.254.0.0/112 2
#scopev4 ::ffff:127.0.0.0/104 2
#scopev4 ::ffff:0.0.0.0/96 14
```

```
1 line less; before #2 29 seconds ago
```

```
#scopev4 ::ffff:127.0.0.0/104 2
#scopev4 ::ffff:0.0.0.0/96 14
```

```
"gai.conf" 65L, 2584B written
```

```
#label fc00::/7 6
#label 2001:0::/32 7
:w
```

```
hello,this is vim world!
# Configuration for getaddrinfo
#
```

```
hello,is vim world!
# Configuration for ge
```

```
#label fec0::/10 5
#label fc00::/7 6
:q!
```

#### 四、 实验过程分析与讨论

vim 提供了 3 种模式用于不同的使用，其中命令行模式的命令最多，但很多是易于记忆的，编辑模式个人认为比较常用的，底行模式最常用的就是退出方式，在本次实验中我对 vim 的使用不断熟悉，也渐渐理解了什么是 vim。

#### 五、 指导教师意见

指导教师签字：卢洋

# 实验报告

实验名称	实验四 用户和用户组管理		
实验教室	丹青 922	实验日期	2023 年 3 月 29 日
学 号	2021223180	姓 名	廖若辰星
专业班级	计算机科学与技术 04 班		
指导教师	卢洋		

东北林业大学  
信息与计算机科学技术实验中心

## 一、 实验目的

1. 掌握用户管理命令，包括命令 `useradd` 、 `usermod` 、 `userdel` 、 `newusers` ；
2. 掌握用户组管理命令，包括命令 `groupadd` 、 `groupdel` 、 `gpasswd` ；
3. 掌握用户和用户组维护命令，包括命令 `passwd` 、 `su` 、 `sudo` 。

## 二、 实验环境

- (1) 计算机的硬件配置 PC 系列微机。
- (2) 计算机的软件配置 VMware 虚拟机软件及 Ubuntu 虚拟机。

### 三、 实验内容及结果

1. 创建一个名为 foo ，描述信息为 bar ，登录 shell 为 /bin/sh ，家目录为 /home/foo 的用户，并设置登陆口令为 123456 ；

```
chenxing@DESKTOP-UK74CBE:~$ su sudo
su: user sudo does not exist or the user entry does not contain all the required fields
chenxing@DESKTOP-UK74CBE:~$ sudo su
root@DESKTOP-UK74CBE:/home/chenxing# useradd -c bar -s /bin/sh -d /home/foo foo
root@DESKTOP-UK74CBE:/home/chenxing# passwd foo
New password:
Retype new password:
passwd: password updated successfully
root@DESKTOP-UK74CBE:/home/chenxing#
```

2. 使用命令从 root 用户切换到用户 foo ，修改 foo 的 UID 为 2000 ，其 shell 类型为 /bin/csh ；

```
root@DESKTOP-UK74CBE:/home/chenxing# su foo
$ usermod -u 2000 -s /bin/csh foo
usermod: user foo is currently used by process 133
```

```
root@DESKTOP-UK74CBE:/home/chenxing# usermod -u 2000 -s /bin/csh foo
```

3. 从用户 foo 切换到 root ；

```
chenxing@DESKTOP-UK74CBE:~$ sudo passwd root
[sudo] password for chenxing:
New password:
Retype new password:
passwd: password updated successfully
chenxing@DESKTOP-UK74CBE:~$ su root
Password:
root@DESKTOP-UK74CBE:/home/chenxing#
```

4. 删除 foo 用户，并在删除该用户的同时一并删除其家目录；

```
root@DESKTOP-UK74CBE:/home/chenxing# userdel -r foo
userdel: foo mail spool (/var/mail/foo) not found
userdel: foo home directory (/home/foo) not found
```

5. 使用命令 newusers 批量创建用户，并使用命令 chpasswd 为这些批量创建的用户设置密码（密码也需要批量设置），查看 /etc/passwd 文件检查用户是否创建成功；

```

root@DESKTOP-UK74CBE:/home/chenxing# touch userfile.txt
root@DESKTOP-UK74CBE:/home/chenxing# touch userpwdfile.txt
root@DESKTOP-UK74CBE:/home/chenxing# cat > userfile.txt
win00:x:520:520::/home/win00:/sbin/nologin
win01:x:521:521::/home/win01:/sbin/nologin
^Z
[1]+  Stopped                  cat > userfile.txt
root@DESKTOP-UK74CBE:/home/chenxing# cat > userpwdfile.txt
win00:123456
win01:654321
^Z
[2]+  Stopped                  cat > userpwdfile.txt
root@DESKTOP-UK74CBE:/home/chenxing# newusers userfile.txt
root@DESKTOP-UK74CBE:/home/chenxing# chpasswd < userpwdfile.txt

```

```

root@DESKTOP-UK74CBE:/home/chenxing# tail -1 /etc/passwd
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:102:105::/nonexistent:/usr/sbin/nologin
systemd-timesync:x:103:106:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
syslog:x:104:111::/home/syslog:/usr/sbin/nologin
_apt:x:105:65534::/nonexistent:/usr/sbin/nologin
uidd:x:106:112::/run/uidd:/usr/sbin/nologin
tcpdump:x:107:113::/nonexistent:/usr/sbin/nologin
chenxing:x:1000:1000::,/home/chenxing:/bin/bash
win00:x:520:520::/home/win00:/sbin/nologin
win01:x:521:521::/home/win01:/sbin/nologin

```

6. 创建用户组 group1 ，并在创建时设置其 GID 为 3000 ；

```

root@DESKTOP-UK74CBE:/home/chenxing# groupadd -g 3000 group1

```

7. 在用户组 group1 中添加两个之前批量创建的用户；

```

root@DESKTOP-UK74CBE:/home/chenxing# usermod -g group1 win00
root@DESKTOP-UK74CBE:/home/chenxing# usermod -g group1 win01

```

8. 切换到 group1 组中的任一用户，在该用户下使用 sudo 命令查看 /etc/shadow 文件，检查上述操作是否可以执行；若不能执行，修改 sudoers 文件使得该用户可以查看文件 /etc/shadow 的内容

```

chenxing@DESKTOP-UK74CBE:~$ su win00
Password:
This account is currently not available.
chenxing@DESKTOP-UK74CBE:~$ su root
Password:
root@DESKTOP-UK74CBE:/home/chenxing# usermod -s /bin/bash win00
root@DESKTOP-UK74CBE:/home/chenxing# su win00
win00@DESKTOP-UK74CBE:/home/chenxing$ sudo tail -1 /etc/shadow
[sudo] password for win00:
win00 is not in the sudoers file. This incident will be reported.
win00@DESKTOP-UK74CBE:/home/chenxing$ su root
Password:
root@DESKTOP-UK74CBE:/home/chenxing# visudo
root@DESKTOP-UK74CBE:/home/chenxing# su win00
win00@DESKTOP-UK74CBE:/home/chenxing$ sudo tail -1 /etc/shadow
[sudo] password for win00:
systemd-resolve:*:19398:0:99999:7:::
messagebus:*:19398:0:99999:7:::
systemd-timesync:*:19398:0:99999:7:::
syslog:*:19398:0:99999:7:::
_apt:*:19398:0:99999:7:::
uidd:*:19398:0:99999:7:::
tcpdump:*:19398:0:99999:7:::
chenxing:$y$j9T$TLxgWBjNbi17B2HaJl1b1$LfQwfgVswvmhRO/GxkG08wGau/7XgVbjdqrVgkzmh3:19418:0:99999:7:::
win00:$y$j9T$WRFehhKEY3EVOKJ8g5L3s/$HtwcYhIJiVKHOMdEslitjZDVR8CV745167E7rriabA1:19498:0:99999:7:::
win01:$y$j9T$SogoOrr4Jmi8ni7uEfbRMD/$bWuSuWpwZvdDKbXGWWrUThwCns1AWrsqk8EwYqR8nG0:19498:0:99999:7:::
win00@DESKTOP-UK74CBE:/home/chenxing$

```



#### 四、 实验过程分析与讨论

第 5 个问题一开始不知道 `newusers` 和命令 `chpasswd` 的使用方法，通过查询 `csdn` 发现需要先建立两个文本文件 `userfile.txt` 和 `userpwdfile.txt` 分别存放用户和密码，最后使用这两个命令完成批量创建用户。

在创建用户时 `shell` 使用的是 `/sbin /nologin`，导致在完成第 8 个问题时使用 `su` 其中一个用户，反馈是 `This account is currently not available`，通过搜索发现需要将 `shell` 改成 `/bin/bash` 才能使用。

最后使用 `sudo tail -l /etc/shadow` 无法查询是因为修改了用户名，没有对应修改 `sudoers` 文件或者 `group` 文件，这是可以使用 `visudo` 修改 `sudoers` 文件，也可以直接用 `su -` 用 `root` 调用查询。

#### 五、指导教师意见

指导教师签字：卢洋

# 实验报告

实验名称	实验五 Shell 程序的创建及条件判断语句		
实验教室	丹青 922	实验日期	2023 年 4 月 5 日
学 号	2021223180	姓 名	廖若辰星
专业班级	计算机科学与技术 04 班		
指导教师	卢洋		

东北林业大学  
信息与计算机科学技术实验中心

## 一、 实验目的

1. 掌握Shell程序的创建过程及Shell程序的执行方法；
2. 掌握Shell变量的定义方法，及用户定义变量、参数位置等；
3. 掌握变量表达式，包括字符串比较、数字比较、逻辑测试、文件测试；
4. 掌握条件判断语句，如 if 语句、 case 语句。

## 二、 实验环境

- (1) 计算机的硬件配置 PC 系列微机。
- (2) 计算机的软件配置 VMware 虚拟机软件及 Ubuntu 虚拟机。

## 三、 实验内容及结果

1. 定义变量 foo 的值为 200 ，并将其显示在屏幕上（终端上执行）；

```
chenxing@DESKTOP-UK74CBE: ~$ foo=200
chenxing@DESKTOP-UK74CBE: ~$ echo foo
foo
```

2. 定义变量 bar 的值为 100 ，并使用 test 命令比较其值是否大于 150 ，并显示 test 命令的退出码（终端上执行）；

```
chenxing@DESKTOP-UK74CBE: ~$ tt=150
chenxing@DESKTOP-UK74CBE: ~$ test $foo -gt $tt
chenxing@DESKTOP-UK74CBE: ~$ echo $?
0
```

3. 创建一个Shell程序，其功能为显示计算机主机名(hostname)和系统时间(date)；

```

chenxing@DESKTOP-UK74CBE: ~$ mkdir shell_mulu
chenxing@DESKTOP-UK74CBE: ~$ cd shell_mulu
chenxing@DESKTOP-UK74CBE: ~/shell_mulu$ cat > test_1_shell.sh
#!/bin/bash
homename
date
^Z
[1]+  Stopped                  cat > test_1_shell.sh
chenxing@DESKTOP-UK74CBE: ~/shell_mulu$ vi test_1_shell.sh

chenxing@DESKTOP-UK74CBE: ~/shell_mulu$ bash test_1_shell.sh
test_1_shell.sh: line 2: homename: command not found
Sun May 21 22:11:27 CST 2023

```

4. 创建一个Shell程序，要求可以处理一个输入参数，判断该输入参数是否为水仙花数；

所谓水仙花数是指一个 3 位数，该数字每位数字的 3 次幂之和等于其本身，例如：

$$153 == 1^3 + 3^3 + 5^3$$

根据上述定义 153 是水仙花数。编写程序时要求首先进行输入参数个数判断，判断是否有输入参数存在：如果没有则给出提示信息；否则给出该数是否是水仙花数。要求对 153 、 124 和 370 进行测试判断。

```

#!/bin/bash
num=$1
n=$num
sum=0
while [ $n -gt 0 ]
do
    m=$((n%10))
    sum=$((sum+m*m*m))
    n=$((n/10))
done
if [ $sum -eq $num ]
then
    echo "$num是水仙花数"
else
    echo "$num不是水仙花数"
fi

```

```
chenxing@DESKTOP-UK74CBE:~/shell_mulu$ vim test_1_shell.sh
chenxing@DESKTOP-UK74CBE:~/shell_mulu$ bash test_1_shell.sh 153
153是水仙花数
chenxing@DESKTOP-UK74CBE:~/shell_mulu$ bash test_1_shell.sh 124
124不是水仙花数
chenxing@DESKTOP-UK74CBE:~/shell_mulu$ bash test_1_shell.sh 370
370是水仙花数
```

5. 创建一个Shell程序，输入 3 个参数，计算 3 个输入变量的和并输出；

```
chenxing@DESKTOP-UK74CBE:~/shell_mulu$ vim test_1_shell.sh
chenxing@DESKTOP-UK74CBE:~/shell_mulu$ bash test_1_shell.sh
The sum of the three numbers is:0
chenxing@DESKTOP-UK74CBE:~/shell_mulu$ bash test_1_shell.sh 1 2 3
The sum of the three numbers is:6
```

```
#!/bin/bash
num1=$1
num2=$2
num3=$3
sum=$((num1+num2+num3))
echo "The sum of the three numbers is:$sum"
```

6. 创建一个Shell程序，输入学生成绩，给出该成绩对应的等级：

90 分以上为 A ， 80-90 为 B ， 70-80为 C ， 60-70 为 D ， 小于 60 分为 E 。要求使用

if

elif

else

fi

实现。

```
chenxing@DESKTOP-UK74CBE:~/shell_mulu$ vim test_1_shell.sh
chenxing@DESKTOP-UK74CBE:~/shell_mulu$ bash test_1_shell.sh
请输入成绩: 77
C
chenxing@DESKTOP-UK74CBE:~/shell_mulu$ bash test_1_shell.sh
请输入成绩: 56
E
```

```
#!/bin/bash
read -p '请输入成绩: ' score
if [ $score -ge 90 ]
then
    echo "A"
elif [ $score -ge 80 ]
then
    echo "B"
elif [ $score -ge 70 ]
then
    echo "C"
elif [ $score -ge 60 ]
then
    echo "D"
else
    echo "E"
fi
```

#### 四、 实验过程分析与讨论

我在输入判断语句的时候，经常遗忘[]中的空格，导致代码出错，又因为错误很小，所以经常需要找很久，浪费时间。据此，我学习到了敲代码一定要注意细节，在学习的时候就要注意这些用法规范，而不要等到多次实践出错才发现这一要点。

#### 五、指导教师意见

指导教师签字：卢洋

# 实验报告

实验名称	实验六 Shell 循环控制语句		
实验教室	丹青 922	实验日期	2023 年 4 月 12 日
学 号	2021223180	姓 名	廖若辰星
专业班级	计算机科学与技术 04 班		
指导教师	卢洋		

东北林业大学  
信息与计算机科学技术实验中心

## 一、 实验目的

1. 熟练掌握Shell循环语句： `for` 、 `while` 、 `until` ；
2. 熟练掌握 Shell 循环控制语句： `break` 、 `continue` 。

## 二、 实验环境

- (1) 计算机的硬件配置 PC 系列微机。
- (2) 计算机的软件配置 VMware 虚拟机软件及 Ubuntu 虚拟机。

## 三、 实验内容及结果

1. 编写一个Shell脚本，利用 `for` 循环把当前目录下的所有 `*.c` 文件复制到指定的目录中（如 `~/workspace` ）；  
可以事先在当前目录下建立若干 `*.c` 文件用于测试。

```
target_file=~/workspace/  
for file in *.c;do  
    cp "$file" "$target_file"  
done
```

```
chenxing@DESKTOP-UK74CBE:~$ mkdir workspace  
chenxing@DESKTOP-UK74CBE:~$ vim test_2_shell.sh  
chenxing@DESKTOP-UK74CBE:~$ bash test_2_shell.sh  
chenxing@DESKTOP-UK74CBE:~$ cd workspace  
chenxing@DESKTOP-UK74CBE:~/workspace$ ls  
test.c
```

2. 编写Shell脚本，利用 `while` 循环求前 10 个偶数之和，并输出结果；



```
chenxing@DESKTOP-UK74CBE: ~/workspace$ vim test_2_shell.sh
chenxing@DESKTOP-UK74CBE: ~/workspace$ bash test_2_shell.sh
110
```

```
n=1
sum=0
while ((n<=20))
do
    if ((n%2 == 0))
    then
        sum=$((sum+n))
    fi
    n=$((n+1))
done
echo "$sum"
```

3. 编写Shell脚本，利用 `until` 循环求 1 到 10 的平方和，并输出结果；

```
chenxing@DESKTOP-UK74CBE: ~/workspace$ vim test_2_shell.sh
chenxing@DESKTOP-UK74CBE: ~/workspace$ bash test_2_shell.sh
385
```

```
n=1
sum=0
until ((n>10))
do
    s=$((n*n))
    sum=$((sum+s))
    n=$((n+1))
done
echo "$sum"
```

4. 运行下列程序，并观察程序的运行结果。将程序中的 `---` 分别替换为 `break`、`break 2`、`continue`、`continue 2`，并观察四种情况下的实验结果。

```
#!/bin/bash
```

```
for i in a b c d; do
```

```
    echo -n $i
```

```
    for j in 1 2 3 4 5 6 7 8 9 10; do
```

```
        if [[ $j -eq 5 ]]; then
```

fi

echo -n \$j

done

echo ''

done

```
#!/bin/bash
for i in a b c d; do
    echo -n $i
    for j in 1 2 3 4 5 6 7 8 9 10;do
        if [[ $j -eq 5 ]];then
            break
        fi
        echo -n $j
    done
    echo ''
done
```

```
chenxing@DESKTOP-UK74CBE: ~/workspace$ vim test_2_shell.sh
chenxing@DESKTOP-UK74CBE: ~/workspace$ bash test_2_shell.sh
a1234
b1234
c1234
d1234
```

```
#!/bin/bash
for i in a b c d; do
    echo -n $i
    for j in 1 2 3 4 5 6 7 8 9 10;do
        if [[ $j -eq 5 ]];then
            break 2
        fi
        echo -n $j
    done
    echo ''
done
```

```
chenxing@DESKTOP-UK74CBE: ~/workspace$ vim test_2_shell.sh
chenxing@DESKTOP-UK74CBE: ~/workspace$ bash test_2_shell.sh
a1234chenxing@DESKTOP-UK74CBE: ~/workspace$
```

```
#!/bin/bash
for i in a b c d; do
    echo -n $i
    for j in 1 2 3 4 5 6 7 8 9 10;do
        if [[ $j -eq 5 ]];then
            break 2
        fi
        echo -n $j
    done
    echo ''
done
```

```
chenxing@DESKTOP-UK74CBE: ~/workspace$ bash test_2_shell.sh
a1234678910
b1234678910
c1234678910
d1234678910

#!/bin/bash
for i in a b c d; do
    echo -n $i
    for j in 1 2 3 4 5 6 7 8 9 10;do
        if [[ $j -eq 5 ]];then
            continue 2
        fi
        echo -n $j
    done
    echo ', '
done
```

```
chenxing@DESKTOP-UK74CBE: ~/workspace$ vim test_2_shell.sh
chenxing@DESKTOP-UK74CBE: ~/workspace$ bash test_2_shell.sh
a1234b1234c1234d1234chenxing@DESKTOP-UK74CBE: ~/workspace$
```

#### 四、 实验过程分析与讨论

在创建文件或者文件夹的时候经常容易手误，又因为 Linux 不像其他编译器会及时提醒，导致经常在后续使用时发现不能使用该文件/文件夹而纠错很久，最后发现还是一个很小的失误。从这件事上，学习到了一定要细心，不论是在敲代码还是做其他事情。

#### 五、指导教师意见

指导教师签字：卢洋

# 实验报告

实验名称	实验七 Shell 函数		
实验教室	丹青 922	实验日期	2023 年 4 月 19 日
学 号	2021223180	姓 名	廖若辰星
专业班级	计算机科学与技术 04 班		
指导教师	卢洋		

东北林业大学  
信息与计算机科学技术实验中心

## 一、 实验目的

1. 掌握Shell函数的定义方法；
2. 掌握Shell函数的参数传递、调用和返回值；
3. 掌握Shell函数的递归调用方法；
4. 理解 Shell 函数的嵌套。

## 二、 实验环境

- (1) 计算机的硬件配置 PC 系列微机。
- (2) 计算机的软件配置 VMware 虚拟机软件及 Ubuntu 虚拟机。

## 三、 实验内容及结果

1. 编写Shell脚本，实现一个函数，对两个数的和进行求解，并输出结果；

```
function addition() {  
    sum=$(( $1 + $2 ))  
    echo "$sum"  
}  
addition 7 9
```

```
chenxing@DESKTOP-UK74CBE:~/workspace$ vim test_2_shell.sh  
chenxing@DESKTOP-UK74CBE:~/workspace$ bash test_2_shell.sh  
16
```

2. 编写Shell脚本，在脚本中定义一个递归函数，实现  $n$  的阶乘的求解；

```
chenxing@DESKTOP-UK74CBE:~/workspace$ vim test_2_shell.sh  
chenxing@DESKTOP-UK74CBE:~/workspace$ bash test_2_shell.sh  
input:5  
120
```

```
#!/bin/bash
function fun() {
    result=1
    for((i=1;i<=$1;i++))
    do
        result=$((result*i))
    done
    echo $result
}
read -p 'input:' n
if (( $? == 0 ))
then
    fun $n
else
    echo "wrong"
fi
```

3. 一个Shell脚本的内容如下所示:

```
#!/bin/bash

function first() {

    function second() {

        function third() {

            echo "-3- here is in the third func."

        }

        echo "-2- here is in the second func."

        third

    }

    echo "-1- here is in the first func."

    second

}

echo "starting..."

first
```

试运行该程序，并观察程序运行结果，理解函数嵌套的含义。

```
chenxing@DESKTOP-UK74CBE:~/shell_mulu$ cat> test_1_shell.sh
#!/bin/bash
function first() {
function wecond() {
gunction third() {
echo "-3- here is in the third func."
}
echo "-2- here is in the second func."
third
}
echo "-1- here is in the first func."
second
}
echo "starting...}
first
chenxing@DESKTOP-UK74CBE:~/shell_mulu$ vim test_1_shell.sh
chenxing@DESKTOP-UK74CBE:~/shell_mulu$ bash test_1_shell.sh
test_1_shell.sh: line 4: syntax error near unexpected token `('
test_1_shell.sh: line 4: `gunction third() {'
chenxing@DESKTOP-UK74CBE:~/shell_mulu$ vim test_1_shell.sh
chenxing@DESKTOP-UK74CBE:~/shell_mulu$ bash test_1_shell.sh
starting...
-1- here is in the first func.
-2- here is in the second func.
-3- here is in the third func.
```

答：Linux中的函数嵌套与C语言中的本质相同，在函数内部再定义或调用多个函数。

#### 四、 实验过程分析与讨论

在本次实验中，我重点学习了 shell 脚本中的函数的建立与调用，其中 `function` 是 Shell 中的关键字，专门用来定义函数。shell 中的函数与 C 语言的相似，可以 `return` 返回值，给予参数等，不同的是：Shell 也不限制定义和调用的顺序，可以将定义放在调用的前面，也将定义放在调用的后面。

#### 五、指导教师意见

指导教师签字：卢洋



# 实验报告

实验名称	实验八 sed 和 awk		
实验教室	丹青 922	实验日期	2023 年 4 月 26 日
学 号	2021223180	姓 名	廖若辰星
专业班级	计算机科学与技术 04 班		
指导教师	卢洋		

东北林业大学  
信息与计算机科学技术实验中心

## 一、 实验目的

1. 掌握 sed 基本编辑命令的使用方法；
2. 掌握 sed 与Shell变量的交互方法；
3. 掌握 awk 命令的使用方法；
4. 掌握 awk 与 Shell 变量的交互方法。

## 二、 实验环境

- (1) 计算机的硬件配置 PC 系列微机。
- (2) 计算机的软件配置 VMware 虚拟机软件及 Ubuntu 虚拟机。

## 三、 实验内容及结果

1. 文件 quote.txt 的内容如下所示：

The honeysuckle band played all night long for only \$90.

It was an evening of splendid music and company.

Too bad the disco floor fell through at 23:10.

The local nurse Miss P. Neave was in attendance.

试使用 sed 命令实现如下功能：

- (1) 删除 \$ 符号；

```
chenxing@DESKTOP-UK74CBE: ~$ sed 's/\$/g' quote.txt
The honeysuckle band played all night long for only 90.
It was an evening of splendid music and company.
Too bad the disco floor fell through at 23:10.
The local nurse Miss P. Neave was in attendance.
```

- (2) 显示包含 music 文字的行内容及行号；

```
chenxing@DESKTOP-UK74CBE:~$ cat quote.txt | sed -n '/music/p'  
It was an evening of splendid music and company.
```

(3) 在第 4 行后面追加内容: "hello world!";

```
chenxing@DESKTOP-UK74CBE:~$ cat quote.txt | sed '4a hello world!'  
The honeysuckle band played all night long for only $90.  
It was an evening of splendid music and company.  
Too bad the disco floor fell through at 23:10.  
The local nurse Miss P.Neave was in attendance.  
hello world!
```

(4) 将文本 "The" 替换为 "Quod";

```
chenxing@DESKTOP-UK74CBE:~$ cat quote.txt | sed 's/The/Quod/g'  
Quod honeysuckle band played all night long for only $90.  
It was an evening of splendid music and company.  
Too bad the disco floor fell through at 23:10.  
Quod local nurse Miss P.Neave was in attendance.
```

(5) 将第 3 行内容修改为: "This is the third line.";

```
chenxing@DESKTOP-UK74CBE:~$ cat quote.txt | sed '3c This is the third line.'  
The honeysuckle band played all night long for only $90.  
This is the third line.  
The local nurse Miss P.Neave was in attendance.
```

(6) 删除第 2 行内容;

```
chenxing@DESKTOP-UK74CBE:~$ cat quote.txt | sed '2d'  
The honeysuckle band played all night long for only $90.  
Too bad the disco floor fell through at 23:10.  
The local nurse Miss P.Neave was in attendance.
```

(7) 设置Shell变量 var 的值为 evening, 用 sed 命令查找  
匹配 var 变量值的行。

```
chenxing@DESKTOP-UK74CBE:~$ var=evening  
chenxing@DESKTOP-UK74CBE:~$ cat quote.txt | sed -n '/$var/p'
```

2. 文件 numbers.txt 的内容如下所示:

one : two : three

four : five : six

注: 每个冒号前后都有空格。

试使用 awk 命令实现如下功能: 分别以 空格 和 冒号 做分隔

符，显示第 2 列的内容，观察两者的区别；

```
chenxing@DESKTOP-UK74CBE:~$ cat <<EOF >number.txt
> one : two : three
> four : five : six
> EOF
chenxing@DESKTOP-UK74CBE:~$ cat number.txt |awk ' {FS=":"} {print $2}'
:
five
chenxing@DESKTOP-UK74CBE:~$ cat number.txt |awk ' {FS=" "} {print $2}'
:
:
```

3. 已知文件 `foo.txt` 中存储的都是数字，且每行都包含 3 个数字，数字之前以空格作为分隔符。试找出

`foo.txt` 中的所有偶数进行打印，并输出偶数的个数。

要求：判断每行的 3 个数字是否为偶数时用循环结果，即要求程序里包含循环和分支结构。

例如： `foo.txt` 内容为：

2 4 3

15 46 79

则输出为：

even:

2

4

46

numbers:

3

```
chenxing@DESKTOP-UK74CBE:~$ cat >foo.txt
2 4 3
15 46 79
```

```
chenxing@DESKTOP-UK74CBE:~$ awk ' {for(i=1;i<=NF;i++) if($i%2==0) {print $i;count++;}} END{print "
The total number of even numbers is: "count}' foo.txt
2
4
46
The total number of even numbers is: 3
```

4. 脚本的内容如下所示:

```
#!/bin/bash
```

```
read -p "enter search pattern: " pattern
```

```
awk "/$pattern/" '{ nmatches++; print } END { print nmatches,
"found." }' info.txt
```

试运行该脚本，并理解该脚本实现的功能

```
chenxing@DESKTOP-UK74CBE:~/shell_mulu$ vim test_1_shell.sh
chenxing@DESKTOP-UK74CBE:~/shell_mulu$ bash test_1_shell.sh
enter search pattern: abc
awk: fatal: cannot open file `info.txt' for reading: No such file or directory
chenxing@DESKTOP-UK74CBE:~/shell_mulu$ cat >info.txt
1 2 3
123
abc
3 found.
chenxing@DESKTOP-UK74CBE:~/shell_mulu$ bash test_1_shell.sh
enter search pattern: abc
abc
1 found.
```

功能：匹配字符串，根据输入的字符串在指定文件文件 info.txt 中查找，输出匹配到的字符串出现的位置以及出现的次数。

#### 四、 实验过程分析与讨论

在本次实验中，我重点练习了 sed 和 awk 的使用方法，其中需要注意的是，当使用 sed 删除指定字符时，当字符是普通字符，如字母等，就可以直接使用 ‘s/字符//’，而当字符是特殊字符时，如\$，就需要使用 ‘\’ ,如 ‘s\\$/’。

#### 五、 指导教师意见

指导教师签字：卢洋