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 4 月 10 日
学　　号	2021213125	姓　　名	程啟勇
专业班级	计算机科学与技术 01 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

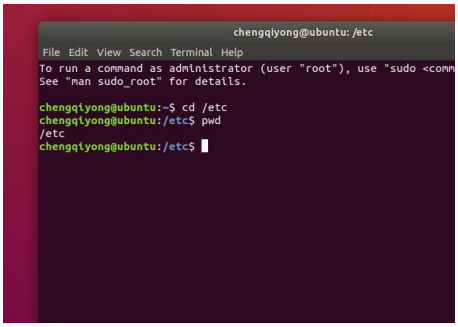
二、 实验环境

（1）计算机的硬件配置 PC 系列微机。

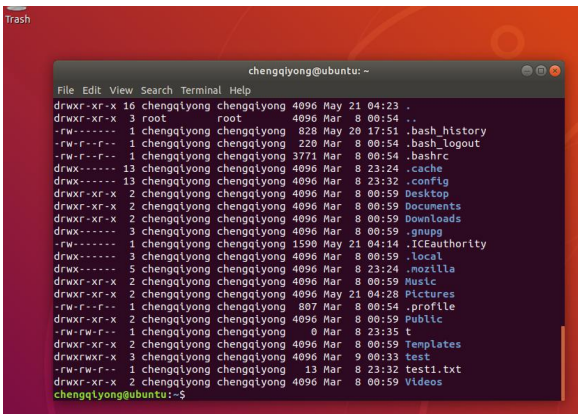
（2）计算机的软件配置 VMware 虚拟机软件及 Ubuntu 虚拟机。

三、 实验内容及结果

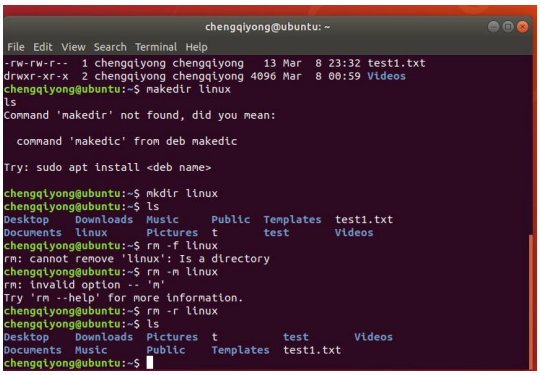
1. 使用命令切换到/etc 目录，并显示当前工作目录路径



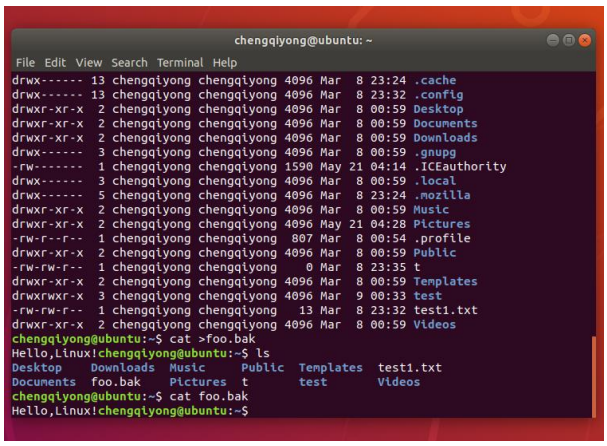
2、使用命令显示/home/lyj 目录下所有文件目录的详细信息，包括隐藏文件。

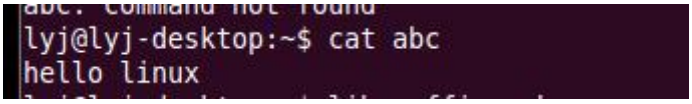


3、使用命令创建目录/home/lyj/linux，然后删除该目录。

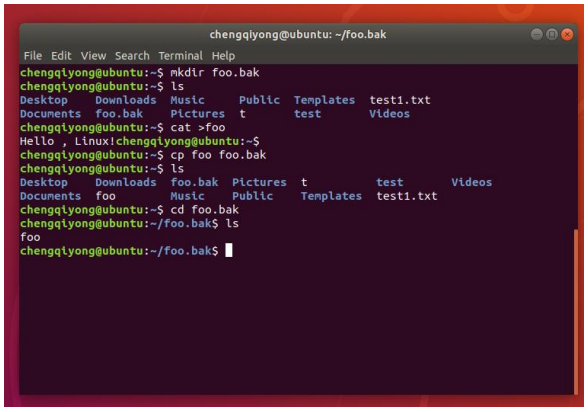


4、使用命令 cat 用输出重定向在/home/lyj 目录下创建文件 abc，文件内容为“Hello, Linux!”，并查看该文件的内容

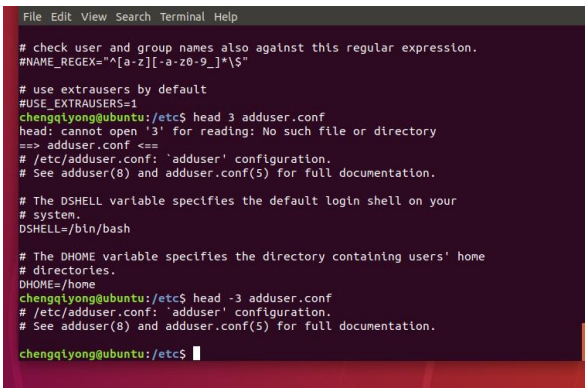




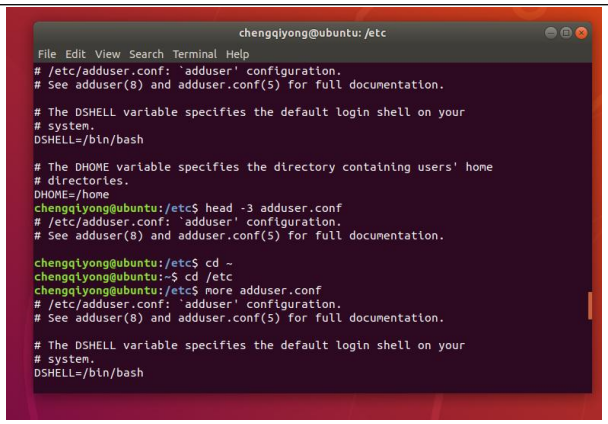
5、使用命令创建目录/home/lyj/ak，然后将/home/lyj/abc文件复制到该目录下，最后将该目录及其目录下的文件一起删除。



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。



7、分屏查看文件/etc/adduser.conf 的内容。



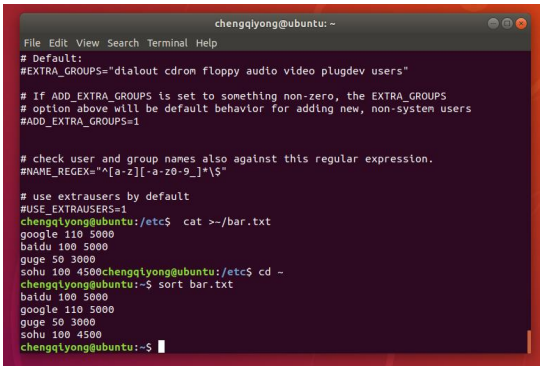
8、使用命令cat用输出重定向在/home/lyj目录下创建文件bar.txt，文件内容为：

google 110 5000

baidu 100 5000

guge 50 3000

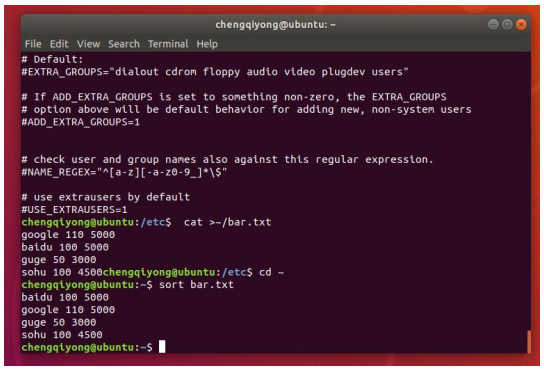
sohu 100 4500



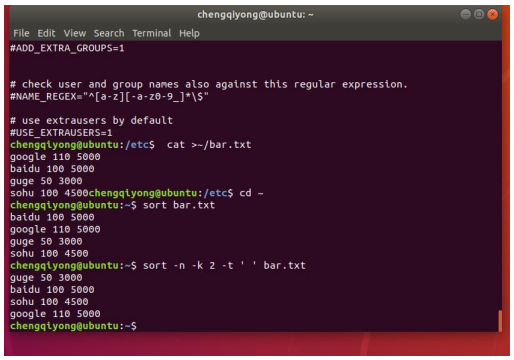
9．第一列为公司名称，第2列为公司人数，第3列为员工平均工资。

利用sort命令完成下列排序：

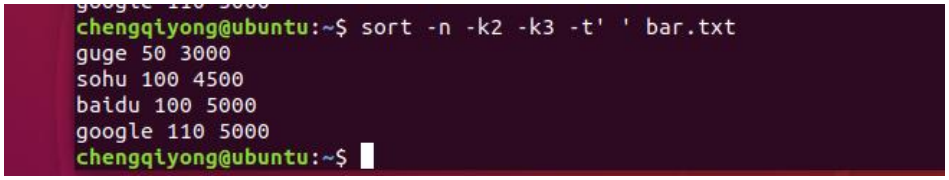
（1）按公司字母顺序排序



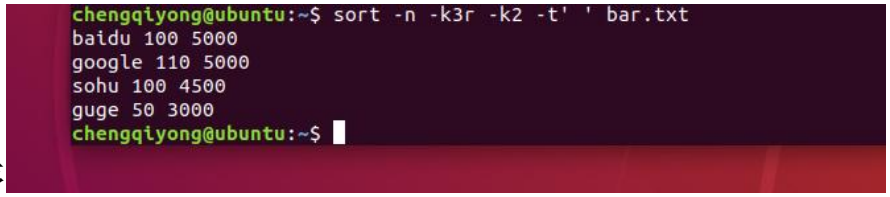
（2）按公司人数排序



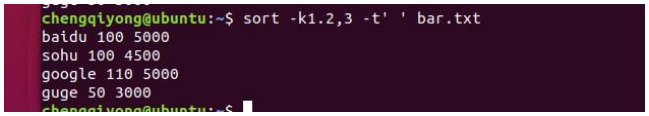
（3）按公司人数排序，人数相同的按照员工平均工资升序排序



（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。



四、 实验过程分析与讨论

遇到的困难在最后 soty 排序的部分，对于多重要求和非第一行的排序命令还是不太熟悉，在查询 CSDN 之后学会了相关命令。

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 4 月 20 日
学　　号	2021213125	姓　　名	程啟勇
专业班级	计算机科学与技术 01 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

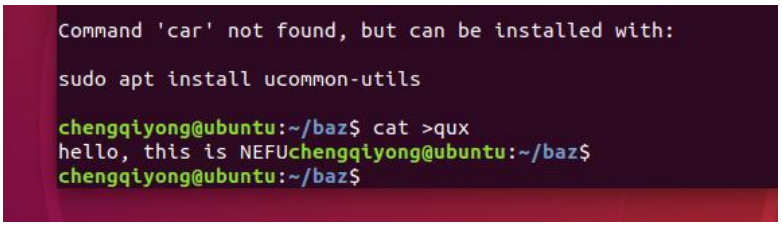
（1）计算机的硬件配置 PC 系列微机。

（2）计算机的软件配置 VMware 虚拟机软件及 Ubuntu 虚拟机。

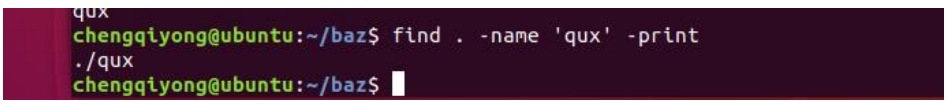
三、 实验内容及结果

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；

```
chengqiyong@ubuntu:~/baz$ wc qux
 0  4 19 qux
chengqiyong@ubuntu:~/baz$ cat >>qux
nini
haohaochengqiyong@ubuntu:~/baz$ cat qux
hello, this is NEFUnini
haohaochengqiyong@ubuntu:~/baz$ wc qux
 1  5 30 qux
chengqiyong@ubuntu:~/baz$
```

(4)在用户目录下查找文件 `qux` ，并删除该文件；

```
 1  5 30 qux
chengqiyong@ubuntu:~/baz$ cd ~
chengqiyong@ubuntu:~$ find . -name 'qux' -print
./baz/qux
chengqiyong@ubuntu:~$ rm -f ~/baz/qux
chengqiyong@ubuntu:~$ cd baz
chengqiyong@ubuntu:~/baz$ ls
chengqiyong@ubuntu:~/baz$
```

(5)查看文件夹 `baz` 内容，看一下是否删除了文件 `qux` 。

```
chengqiyong@ubuntu:~$ cd baz
chengqiyong@ubuntu:~/baz$ ls
chengqiyong@ubuntu:~/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` ；

```
chengqiyong@ubuntu:~$ mkdir path1
chengqiyong@ubuntu:~$ touch path1/file1
chengqiyong@ubuntu:~$ ls path1
file1
chengqiyong@ubuntu:~$ touch path1/file2
chengqiyong@ubuntu:~$ ls
bar.txt  Desktop    Downloads  foo.bak  path1     Public  Templat
baz      Documents  foo        Music    Pictures  t       test
chengqiyong@ubuntu:~$ ls path1
file1  file2
chengqiyong@ubuntu:~$
```

(2)在用户目录下新建文件夹 `path2` ，在 `path2` 下新建文件 `file3` ；

```
chengqiyong@ubuntu:~$ mkdir path2
chengqiyong@ubuntu:~$ cd path2
chengqiyong@ubuntu:~/path2$ touch file3
```

(3)在用户目录下新建文件 `file4` ；

```
chengqiyong@ubuntu:~/path2$ cd ..
chengqiyong@ubuntu:~$ touch file4
chengqiyong@ubuntu:~$
```

(4)在用户目录下对文件夹 `path1` 和 `file4` 进行打包，生成文件 `package.tar` ；

```
chengqiyong@ubuntu:~$ tar -cvf package.tar file4 path1
file4
path1/
path1/file2
path1/file1
```

(5)查看包 `package.tar` 的内容；

```
chengqiyong@ubuntu:~$ tar -tf package.tar
file4
path1/
path1/file2
path1/file1
```

(6)向包 `package.tar` 里添加文件夹 `path2` 的内容；

```
chengqiyong@ubuntu:~$ tar -rf package.tar path2
chengqiyong@ubuntu:~$ tar -tf package.tar
file4
path1/
path1/file2
path1/file1
path2/
path2/file3
```

(7)将包 `package.tar` 复制到用户目录下的新建文件夹 `path3` 中；

```
chengqiyong@ubuntu:~$ mkdir path3
chengqiyong@ubuntu:~$ cp package.tar path3
chengqiyong@ubuntu:~$ ls path3
package.tar
chengqiyong@ubuntu:~$
```

(8)进入 `path3` 文件夹，并还原包 `package.tar` 的内容。

```
package.tar
chengqiyong@ubuntu:~$ cd path3
chengqiyong@ubuntu:~/path3$ tar -xvf package.tar
file4
path1/
path1/file2
path1/file1
path2/
path2/file3
chengqiyong@ubuntu:~/path3$ ls
file4  package.tar  path1  path2
chengqiyong@ubuntu:~/path3$
```

3. 符号链接内容

(1)新建文件 `foo.txt`，内容为 `123`；

```
chengqiyong@ubuntu:~$ cat > foo.txt
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;

```
chengqiyong@ubuntu:~$ ln foo.txt bar.txt
chengqiyong@ubuntu:~$ diff foo.txt bar.txt
chengqiyong@ubuntu:~$ cat bar.txt
123
```

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

```
123
chengqiyong@ubuntu:~$ ls -i bar.txt
274223 bar.txt
chengqiyong@ubuntu:~$ ls -i foo.txt
274223 foo.txt
chengqiyong@ubuntu:~$
```

(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;

```
chengqiyong@ubuntu:~$ cat >bar.txt
abc
chengqiyong@ubuntu:~$ diff foo.txt bar.txt
chengqiyong@ubuntu:~$ comm foo.txt bar.txt
                abc
chengqiyong@ubuntu:~$ cat foo.txt
abc
```

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

```
abc
chengqiyong@ubuntu:~$ rm -f foo.txt
chengqiyong@ubuntu:~$ ls -i bar.txt
274223 bar.txt
chengqiyong@ubuntu:~$ cat bar.txt
abc
```

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

```
chengqiyong@ubuntu:~$ ls -i bar.txt
274223 bar.txt
chengqiyong@ubuntu:~$ ls -i link
273564 link
chengqiyong@ubuntu:~$ cat link
abc
chengqiyong@ubuntu:~$ cat bar.txt
abc
```

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

```
abc
chengqiyong@ubuntu:~$ rm -f bar.txt
chengqiyong@ubuntu:~$ cat link.txt
cat: link.txt: No such file or directory
chengqiyong@ubuntu:~$
```

4. 权限管理

(1)新建文件 `qux.txt`；

```
chengqiyong@ubuntu:~$ touch qux.txt
```

(2)为文件 `qux.txt` 增加执行权限(所有用户都可以执行)。

```
chengqiyong@ubuntu:~$ chmod a+x qux.txt
chengqiyong@ubuntu:~$ ls -a qux.txt
qux.txt
chengqiyong@ubuntu:~$ ls -l qux.txt
-rwxrwxr-x 1 chengqiyong chengqiyong 0 May 21 06:02 qux.txt
```

四、 实验过程分析与讨论

困难在文件打包的部分，对于打包，还原包的命令还是不太熟悉，在查询 CSDN，查询相关资料之后学会了相关命令。

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 4 月 28 日
学　　号	2021213125	姓　　名	程啟勇
专业班级	计算机科学与技术 01 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 ；

```
chengqiyong@ubuntu:~$ mkdir workspace1
```

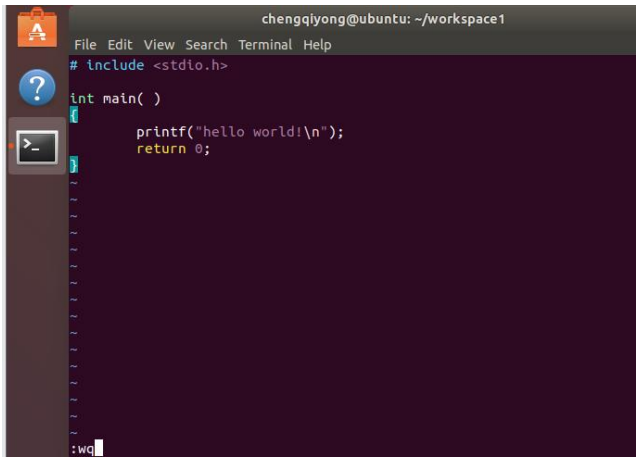
（2)进入目录 workspace1 ；

```
chengqiyong@ubuntu:~$ cd 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。

2. vim 编辑器的详细使用：

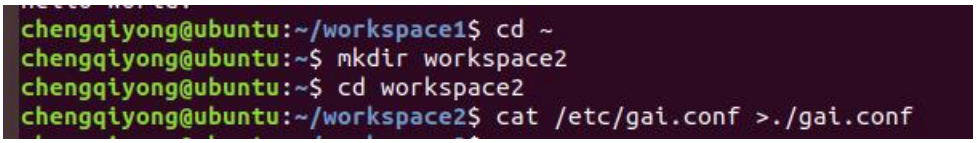
(1)在用户目录下创建一个名为 workspace2 的目录；

(2)进入 workspace2 目录；

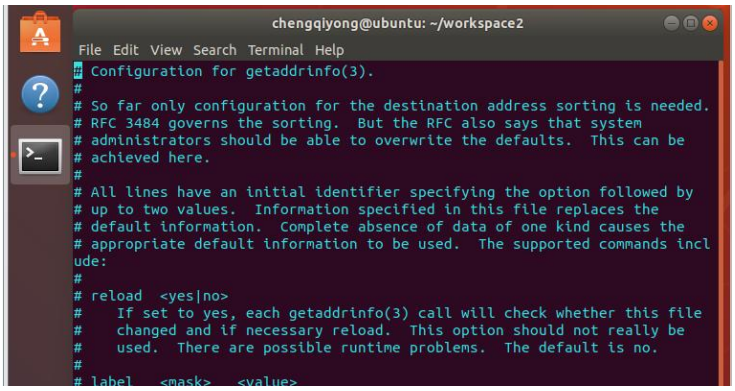
(3)使用以下命令:

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

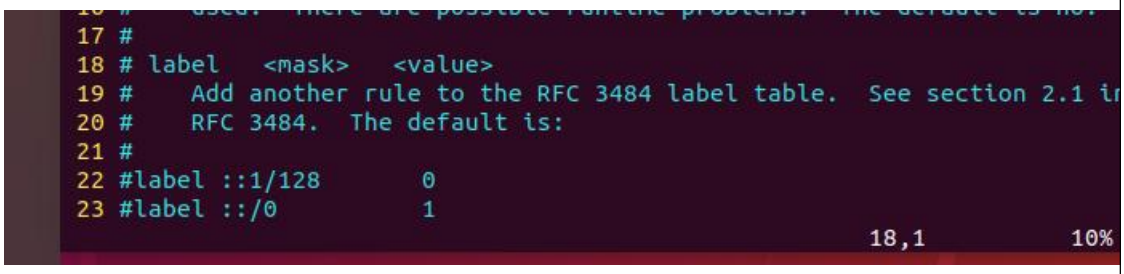
将文件 /etc/gai.conf 的内容复制到当前目录下的新建文件 gai.conf 中;



(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;

```
18 # label   <mask>   <value>
19 #    Add another rule to the RFC 3484 label table.  See section 2.1 in
20 #    RFC 3484.  The default is:
21 #
22 #label ::1/128       0
23 #label ::/0          1
24 #label 2002::/16     2
25 #label ::/96         3
26 #label ::ffff:0:0/96 4
27 #label fec0::/10     5
28 #label fc00::/7      6
29 #label 2001:0::/32   7
30 #
                                        y         18,1          22%
```

(7)将光标移到最后一行行首；

```
61 #    The defaults are equivalent to:
62 #
63 #scopev4 ::ffff:169.254.0.0/112  2
64 #scopev4 ::ffff:127.0.0.0/104    2
65 #scopev4 ::ffff:0.0.0.0/96       14
~
                                                      65,1          B
```

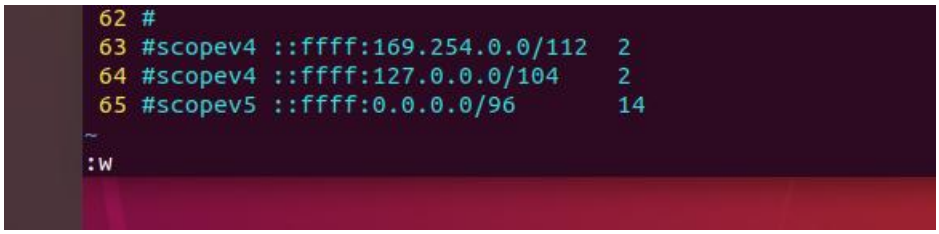
(8)粘贴复制行的内容；

```
64 #scopev4 ::ffff:127.0.0.0/104    2
65 #scopev4 ::ffff:0.0.0.0/96       14
66 # label   <mask>   <value>
                                                      66,1          B
```

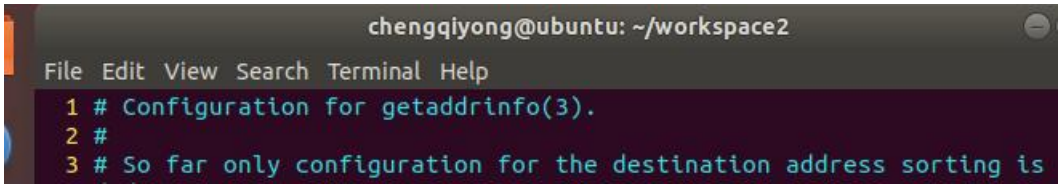
(9)撤销第 8 步的动作；

```
60 #    used.  Changing these defaults should hardly ever be n
61 #    The defaults are equivalent to:
62 #
63 #scopev4 ::ffff:169.254.0.0/112  2
64 #scopev4 ::ffff:127.0.0.0/104    2
65 #scopev4 ::ffff:0.0.0.0/96       14
~
1 line less; before #2  67 seconds ago                65,1
```

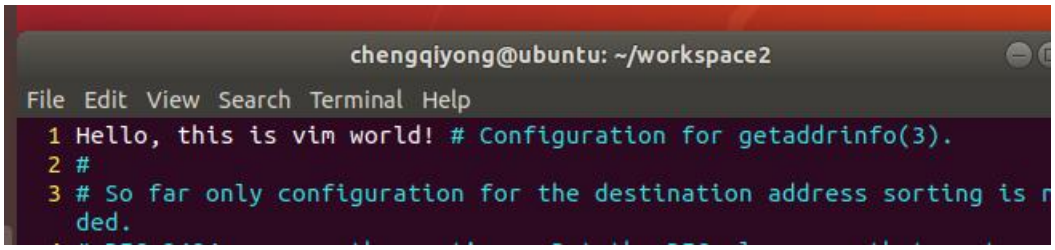
(10)存盘但不退出；



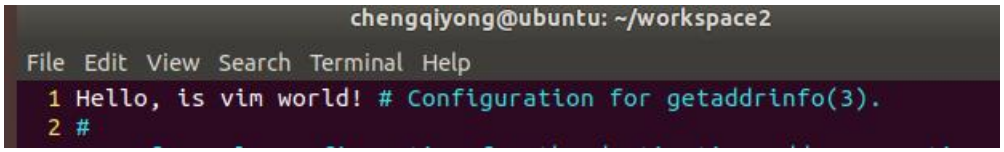
(11)将光标移到首行;



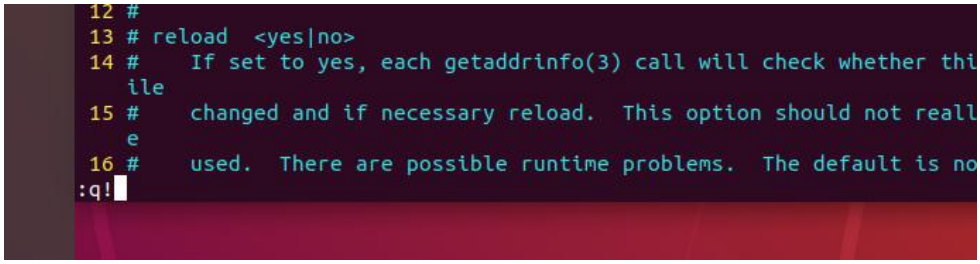
(12)插入模式下输入 "Hello, this is vim world!" ;



(13)删除字符串 "this" ;



(14)强制退出 vim ，不存盘。



四、 实验过程分析与讨论

vim 各个模式的切换，每个模式都有其特定的指令，初步操作起来很不习惯，但随着多加练习，相信会慢慢熟悉 vim 编辑器。

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 5 月 5 日
学　　号	2021213125	姓　　名	程啟勇
专业班级	计算机科学与技术 01 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`；

```
root@ubuntu:/home/chengqiyong# useradd foo -m -c "bar" -s /bin/sh -d /home/foo -
p 123456
root@ubuntu:/home/chengqiyong#
```

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@ubuntu:/home/chengqiyong# su foo_2

root@ubuntu:/home/chengqiyong# usermod -u 2000 -s /bin/csh foo_2
root@ubuntu:/home/chengqiyong#
```

```
root@ubuntu:/home/chengqiyong# su foo_2
$ su root
Password:
root@ubuntu:/home/chengqiyong#
```

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@ubuntu:/home/chengqiyong# userdel -r foo_2
```

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；

```
newusers: error detected, changes ignored
root@ubuntu:/home/chengqiyong# vim users
root@ubuntu:/home/chengqiyong# vim passwds
root@ubuntu:/home/chengqiyong# newusers users
root@ubuntu:/home/chengqiyong# chpasswd paswds
root@ubuntu:/home/chengqiyong# chpasswd < paswds
bash: paswds: No such file or directory
root@ubuntu:/home/chengqiyong# chpasswd < passwds
root@ubuntu:/home/chengqiyong# cat /etc/passwd | tail -3
zhangsan:x:522:522:zhangsan:/home/zhangsan:/bin/bash
lisi:x:523:523:lisi:/home/lisi:/bin/bash
wangwu:x:524:524:wangwu:/home/wangwu:/bin/bash
root@ubuntu:/home/chengqiyong#
```

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

```
wangwu:x:524:524:wangwu:/home/wangwu:/bin/bash
root@ubuntu:/home/chengqiyong# groupadd group1 -g 3000
root@ubuntu:/home/chengqiyong#
```

7. 在用户组 `group1` 中添加两个之前批量创建的用户；

```
root@ubuntu:/home/chengqiyong# groupadd group1 -g 3000
root@ubuntu:/home/chengqiyong# gpasswd -a zhangsan group1
Adding user zhangsan to group group1
root@ubuntu:/home/chengqiyong# gpasswd -a lisi group1
Adding user lisi to group group1
root@ubuntu:/home/chengqiyong#
```

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。

```
root@ubuntu:/home/chengqiyong# su zhangsan
zhangsan@ubuntu:/home/chengqiyong$ sudo cat /etc/shadow
[sudo] password for zhangsan:
zhangsan is not in the sudoers file.  This incident will be reported.
zhangsan@ubuntu:/home/chengqiyong$
```

```
# User privilege specification
root    ALL=(ALL:ALL) ALL
zhangsan ALL=(ALL:ALL) ALL
# Members of the admin group may gain root privileges
%admin ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "#include" directives:
```

```
File  Edit  View  Search  Terminal  Help
dnsmasq:*:18885:0:99999:7:::
rtkit:*:18885:0:99999:7:::
cups-pk-helper:*:18885:0:99999:7:::
speech-dispatcher:!:18885:0:99999:7:::
whoopsie:*:18885:0:99999:7:::
kernoops:*:18885:0:99999:7:::
saned:*:18885:0:99999:7:::
avahi:*:18885:0:99999:7:::
colord:*:18885:0:99999:7:::
hplip:*:18885:0:99999:7:::
geoclue:*:18885:0:99999:7:::
pulse:*:18885:0:99999:7:::
gnome-initial-setup:*:18885:0:99999:7:::
gdm:*:18885:0:99999:7:::
chengqiyong:$5$oJwEE93MCkY7GRjY$CS9QV7NLg7B0utjVZH6PmfdWNOZL2XjGRf8z04sSUpA:1942
4:0:99999:7:::
foo_1:!:19499:0:99999:7:::
zhangsan:$6$rdqSdzmG$whO.i8ZJLg4hARBZXbvuZjJjUDqPXgWmfuoB/fJEH7j8/fwwOFXRx67imaH
f2L.sh1tKT2tn/PvOy3KKUm6ne0:19499:0:99999:7:::
lisi:$6$oxuyljyr$1sg5i1o8liQHRl3GL671eHG/pxi35lGkiO4AH7DLWjtF.VmzKjP7TkVfZA7YHMm
LAXmpNlY78qGULt3BXRxR3/:19499:0:99999:7:::
wangwu:$6$YCOSfF2U$djvklV/rhuPgg6.BfBkH4gvkLcMTqMHojsbQsSQOmHTnvQ53rEyAiyTR7DJwE
NugJ6ZIAQhqc3VODOTGZX0/2.:19499:0:99999:7:::
zhangsan@ubuntu:/home/chengqiyong$
```

四、 实验过程分析与讨论

本次实验过程中对于 newusesr、groupadd 、 groupdel 、gpasswd 运用还是不熟练，相关选项还需要多加练习。

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	20023 年 5 月 10 日
学　　号	2021213125	姓　　名	程啟勇
专业班级	计算机科学与技术 01 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。

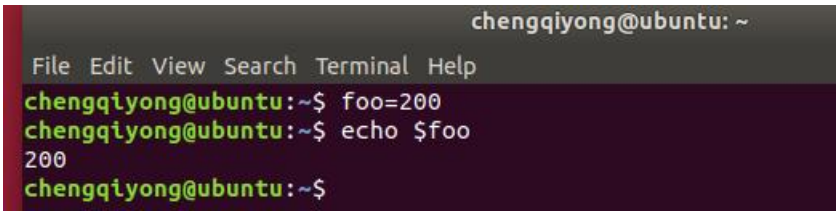
二、 实验环境

（1）计算机的硬件配置 PC 系列微机。

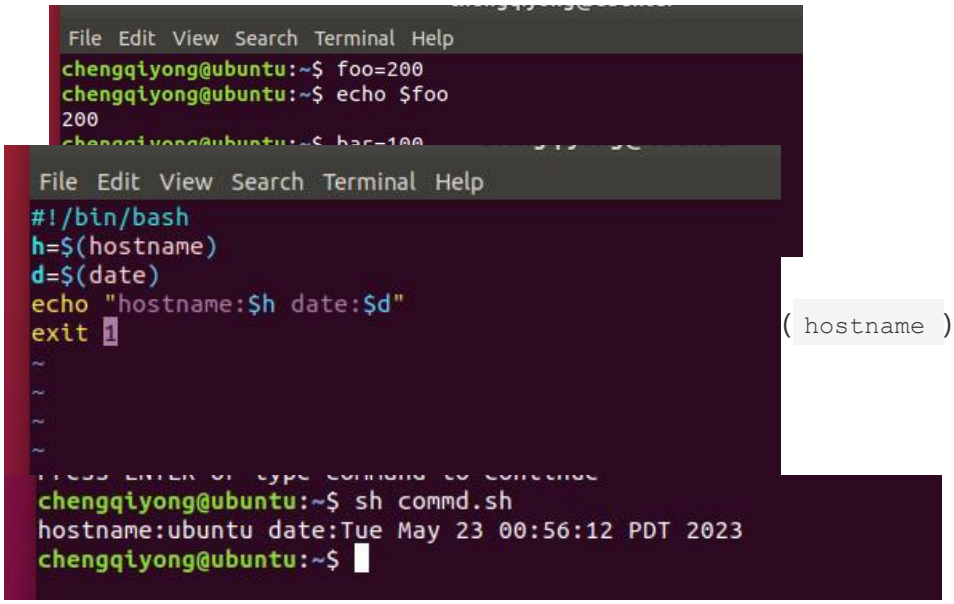
（2）计算机的软件配置 VMware 虚拟机软件及 Ubuntu 虚拟机。

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)；



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)；

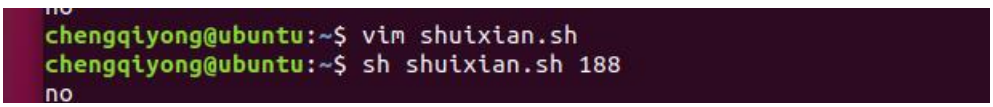


(`hostname`)和系统时间

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 。

```
File  Edit  View  Search  Terminal  Help
#!/bin/bash
num=$#
if [ $num -lt 1 ];then
        echo "error!"
else
        x=$1
        if [ $x -gt 999 -o $x -lt 100 ]; then
                echo "error"
        else
                a=$(($x/100))
                b=$(($x/10%10))
                c=$(($x%10))
                a=$(($a*$a*$a))
                b=$(($b*$b*$b))
                c=$(($c*$c*$c))
                if [ $x = $(($a+$b+$c)) ];then
                        echo "yes"
                else
                        echo "no"
                fi
        fi
fi
~
"shuixian.sh" 22L, 310C                                    16,10-24
```

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；

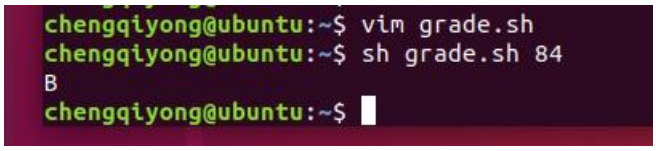
```
55+66+77
chengqiyong@ubuntu:~$ vim adder.sh
chengqiyong@ubuntu:~$ sh adder.sh
number1:55
number2:66
number3:77
198
chengqiyong@ubuntu:~$
```

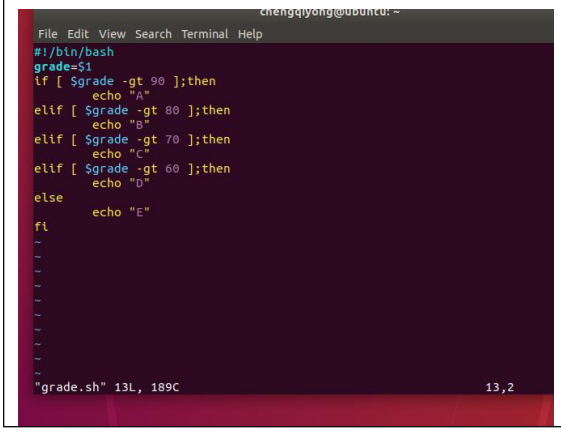
```
File  Edit  View  Search  Terminal  Help
#!/bin/bash
read -p "number1:" n1
read -p "number2:" n2
read -p "number3:" n3
x=$(($n1+$n2+$n3))
echo "$x"
~
~
~
~
~
~
~
~
~
~
~
```

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`， 为`C`， `60-70` 为 `D`，小于 `60` 分为 `E`。要求使用

```
if
elif
else
fi
```

实现。





四、 实验过程分析与讨论

通过本次实验我掌握了 bash 程序的设计方法，学习运用了 if 等条件语句，在编程细节上还需更加注意，如$符号不能缺少。

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 5 月 18 日
学　　号	2021213125	姓　　名	程啟勇
专业班级	计算机科学与技术 01 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

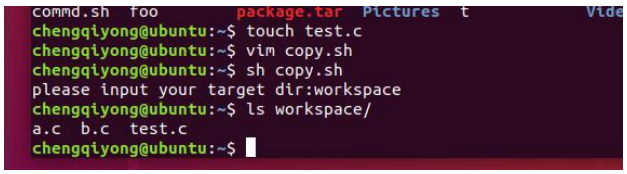
（1）计算机的硬件配置 PC 系列微机。

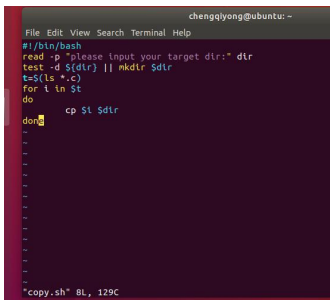
（2）计算机的软件配置 VMware 虚拟机软件及 Ubuntu 虚拟机。

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。





```
chengqiyong@ubuntu:~$ vim addBy2.sh
chengqiyong@ubuntu:~$ sh addBy2.sh
sum= 90
chengqiyong@ubuntu:~$
```

chengqiyong@ubuntu: ~

File Edit View Search Terminal Help

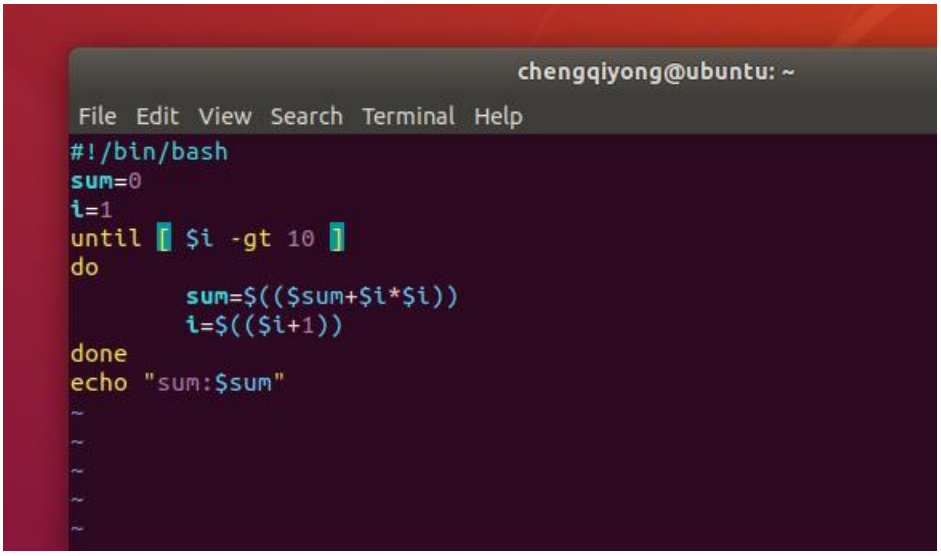
```
#!/bin/bash
n=1
sum=0
a=0
while [ "$n" -le "10" ]
do
        if [ "$(($a%2))" -eq "0" ];then
                sum=$(($sum+$a))
                n=$(($n+1))
        fi
        a=$(($a+1))
done
echo "sum= $sum"
~
~
```

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

chengqiyong@ubuntu: ~

File Edit View Search Terminal Help

```
chengqiyong@ubuntu:~$ vim add2x.sh
chengqiyong@ubuntu:~$ sh add2x.sh
sum:385
chengqiyong@ubuntu:~$
```

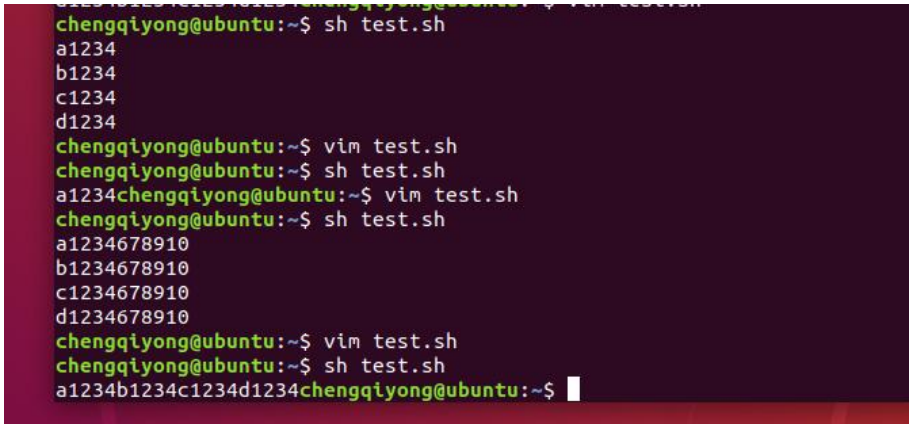


4. 运行下列程序，并观察程序的运行结果。将程序中的

`---` 分别替换为 `break` 、 `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

            ---

        f i

        echo –n $j

    done

    echo ''

done
```



四、 实验过程分析与讨论

编写 shell 程序时要特别注意特殊符号的使用，如$,fi 等，以及小括号的配合使用，空格的使用也很重要，否则容易出现错误。

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 5 月 20 日
学　　号	2021213125	姓　　名	程啟勇
专业班级	计算机科学与技术 01 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

```
99
chengqiyong@ubuntu:~$ vim he.sh
chengqiyong@ubuntu:~$ sh he.sh
input1:44
input2:55
99
```

并输出结果；

```
chengqiyong@
File  Edit  View  Search  Terminal  Help
#!/bin/bash
f1(){
        return $(($1+$2))
}
read -p "input1:" n1
read -p "input2:" n2
f1 $n1 $n2
sum=$?
echo "$sum"
~
```

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；

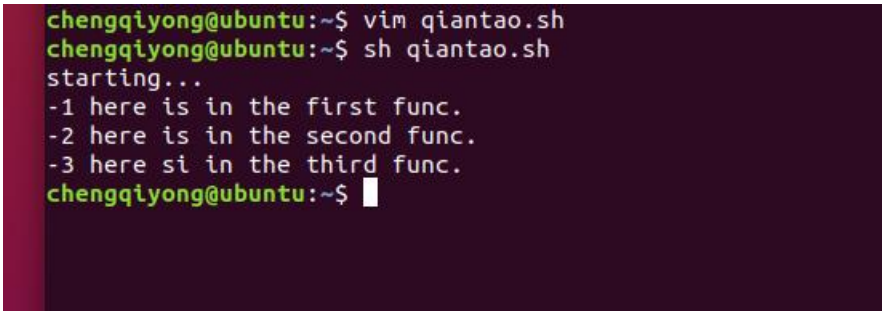
```
digui.sh: 8: digui.sh: Syntax error: "fi" unexpected (expecting "then")
chengqiyong@ubuntu:~$ vim digui.sh
chengqiyong@ubuntu:~$ sh digui.sh
3
6
chengqiyong@ubuntu:~$
```

```
File  Edit  View  Search  Terminal  Help
#!/bin/bash
fun1(){
        if [ $1 -eq 1 ]; then
                return 1;
        else
                fun1 $(($1-1))
                return $(($?*$1))
        fi
}
read n
fun1 $n
end=$?
echo "$end"
~
~
```

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。



四、 实验过程分析与讨论

对函数的运用出现了困难，比如 shell 函数参数的传递，函数值的调用，需要多加练习。

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 5 月 23 日
学　　号	2021213125	姓　　名	程啟勇
专业班级	计算机科学与技术 01 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。

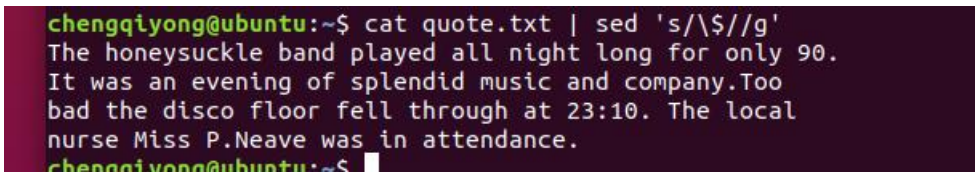
三、 实验内容及结果

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
```

试使用 `sed` 命令实现如下功能:

(1)删除 `$` 符号;



(2)显示包含 `music` 文字的行内容及行号;

```
bad the disco floor fell through at 23:10. The local
chengqiyong@ubuntu:~$ cat quote.txt | sed -n '/music/p'
It was an evening of splendid music and company.Too
chengqiyong@ubuntu:~$
```

(3)在第 `4` 行后面追加内容： `"hello world!"` ；

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

(5) 将文本 `"The"` 替换为 `"Quod"` ；

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

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

```
chengqiyong@ubuntu:~$ cat quote.txt | sed '3c This is the third line'
The honeysuckle band played all night long for only $90.
It was an evening of splendid music and company.Too
This is the third line
nurse Miss P.Neave was in attendance.
chengqiyong@ubuntu:~$
```

(6)删除第 `2` 行内容；

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

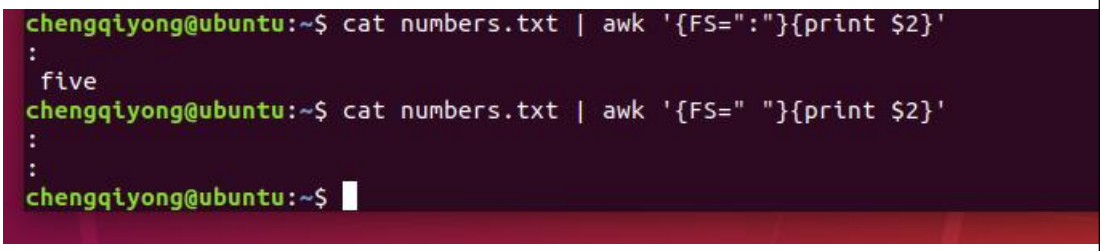
```
chengqiyong@ubuntu:~$ cat quote.txt | sed -n "/$var/p"
It was an evening of splendid music and company.Too
chengqiyong@ubuntu:~$
```

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；



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
```

```
chengqiyong@ubuntu:~$ cat foo.txt | awk '{for(i=1;i<NF;i++) if($i%2==0){print $i
;sum++}} END{print 'sum=' sum}'
2
4
46
3
chengqiyong@ubuntu:~$
```

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。

```
chengqiyong@ubuntu:~$ vim pp.sh
chengqiyong@ubuntu:~$ sh pp.sh
enter search pattern:a
The honeysuckle band played all night long for only $90.
It was an evening of splendid music and company.Too
bad the disco floor fell through at 23:10. The local
nurse Miss P.Neave was in attendance.
4 found.
chengqiyong@ubuntu:~$
```

实现的功能：
在文本中匹配模式串，输出模式串所在行，以及匹配成功的次数。

四、 实验过程分析与讨论

指令 sed, awk 的功能十分强大，sed 的替换、删除、新增等功能，awk 的强大分割功能，语言简洁高效，需要牢牢掌握！

五、指导教师意见

指导教师签字：卢洋