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	2021223280	姓　　名	刘嘉琦
专业班级	计算机科学与技术 06 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

```
the@LAPTOP-CGNJ5JQJ:~$ cd /etc
the@LAPTOP-CGNJ5JQJ:/etc$ pwd
/etc
the@LAPTOP-CGNJ5JQJ:/etc$
```

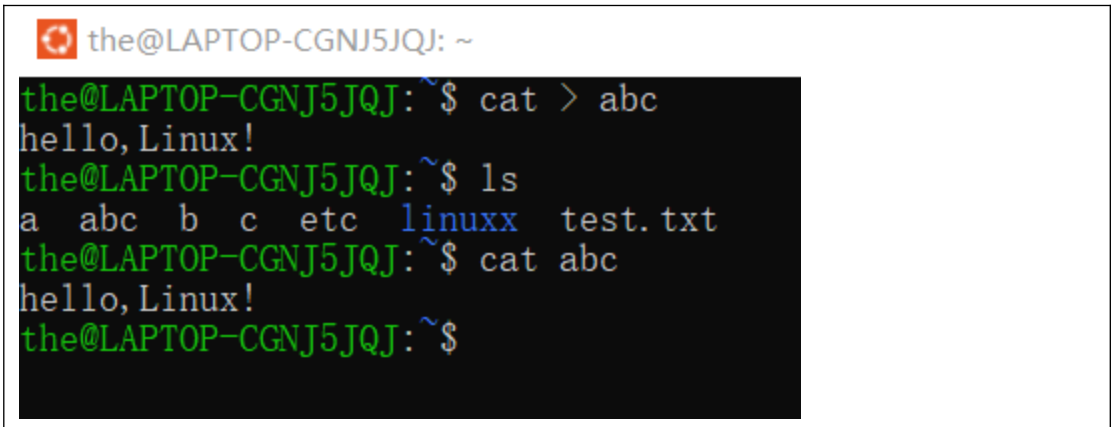
2、使用命令显示/home/lyj 目录下所有文件目录的详细信息，包括隐藏文件。

```
the@LAPTOP-CGNJ5JQJ:/home$ cd the
the@LAPTOP-CGNJ5JQJ:~$ ls -a
.   .bash_history  .bashrc   .motd_shown  .sudo_as_admin_successful  b  etc
..  .bash_logout   .lesshst  .profile     a                          c  test.
the@LAPTOP-CGNJ5JQJ:~$
```

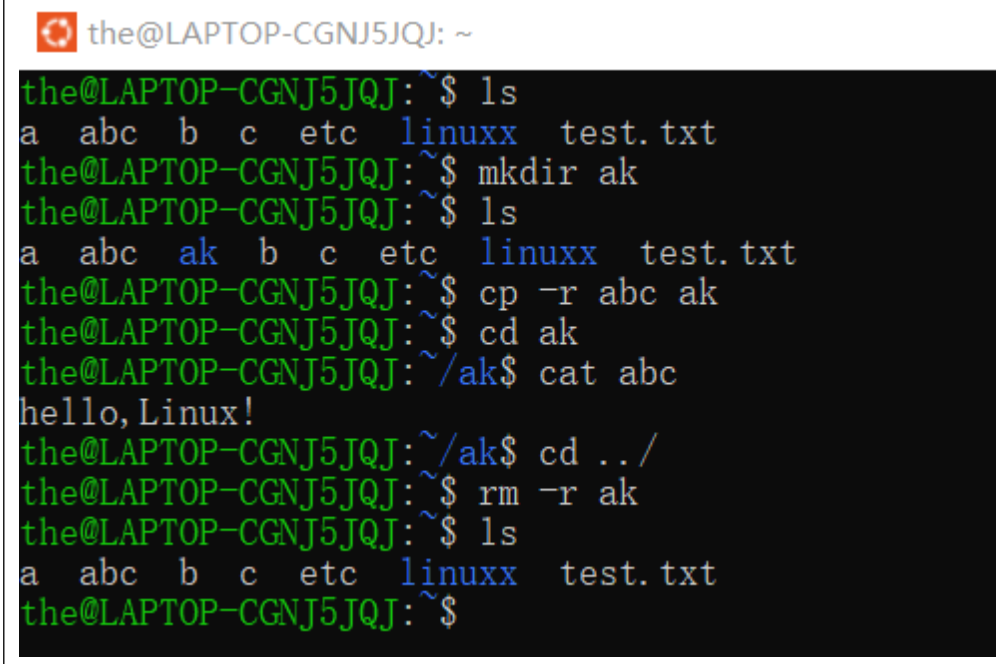
3、使用命令创建目录/home/lyj/linux，然后删除该目录。


```
the@LAPTOP-CGNJ5JQJ:~$ mkdir linux
the@LAPTOP-CGNJ5JQJ:~$ cd linux
the@LAPTOP-CGNJ5JQJ:~/linux$ cd ../
the@LAPTOP-CGNJ5JQJ:~$ rmdir linux
the@LAPTOP-CGNJ5JQJ:~$ cd linux
-bash: cd: linux: No such file or directory
the@LAPTOP-CGNJ5JQJ:~$
```

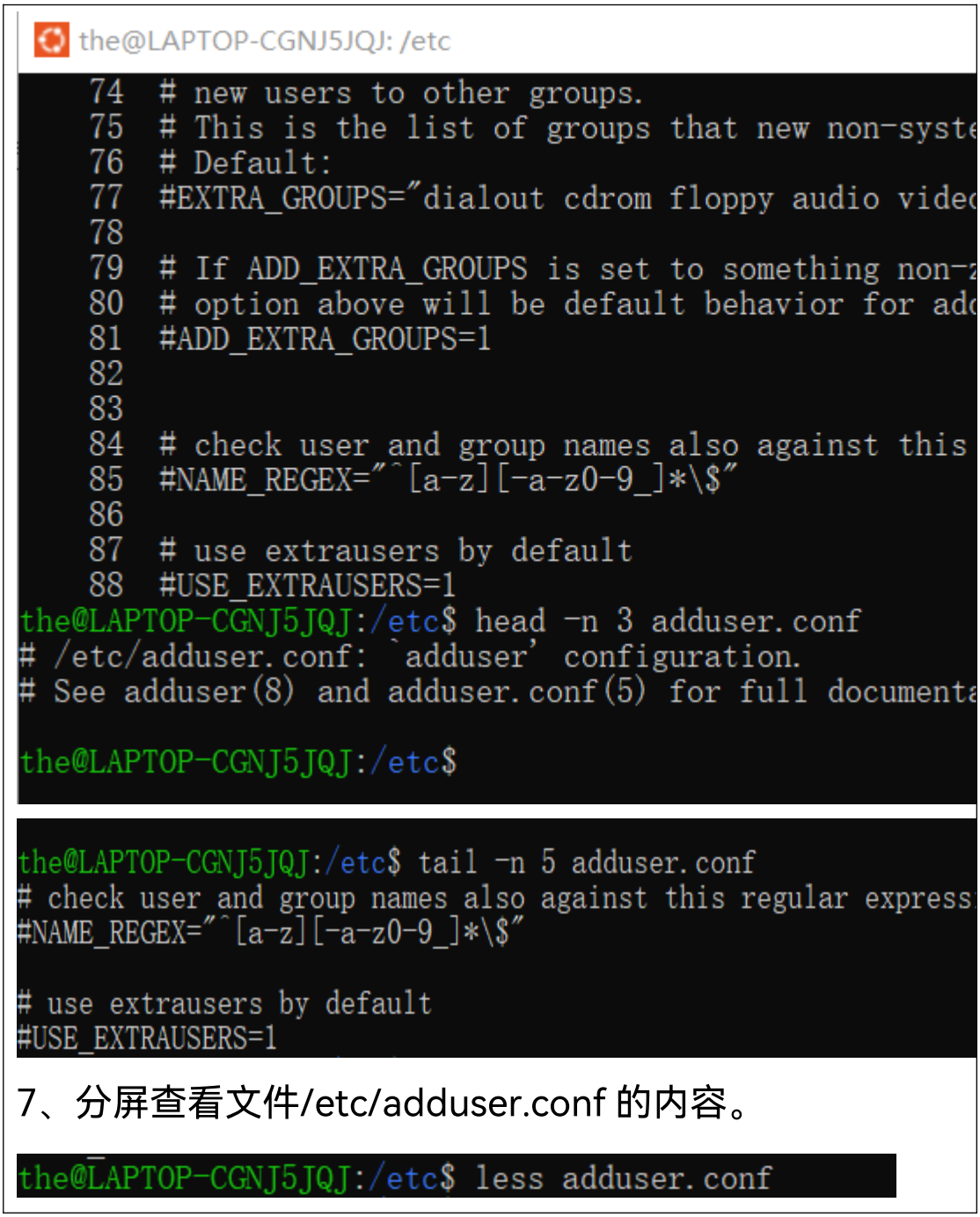
4、使用命令 cat 用输出重定向在/home/lyj 目录下创建文件 abc，文件内容为“Hello, Linux!”，并查看该文件的内容



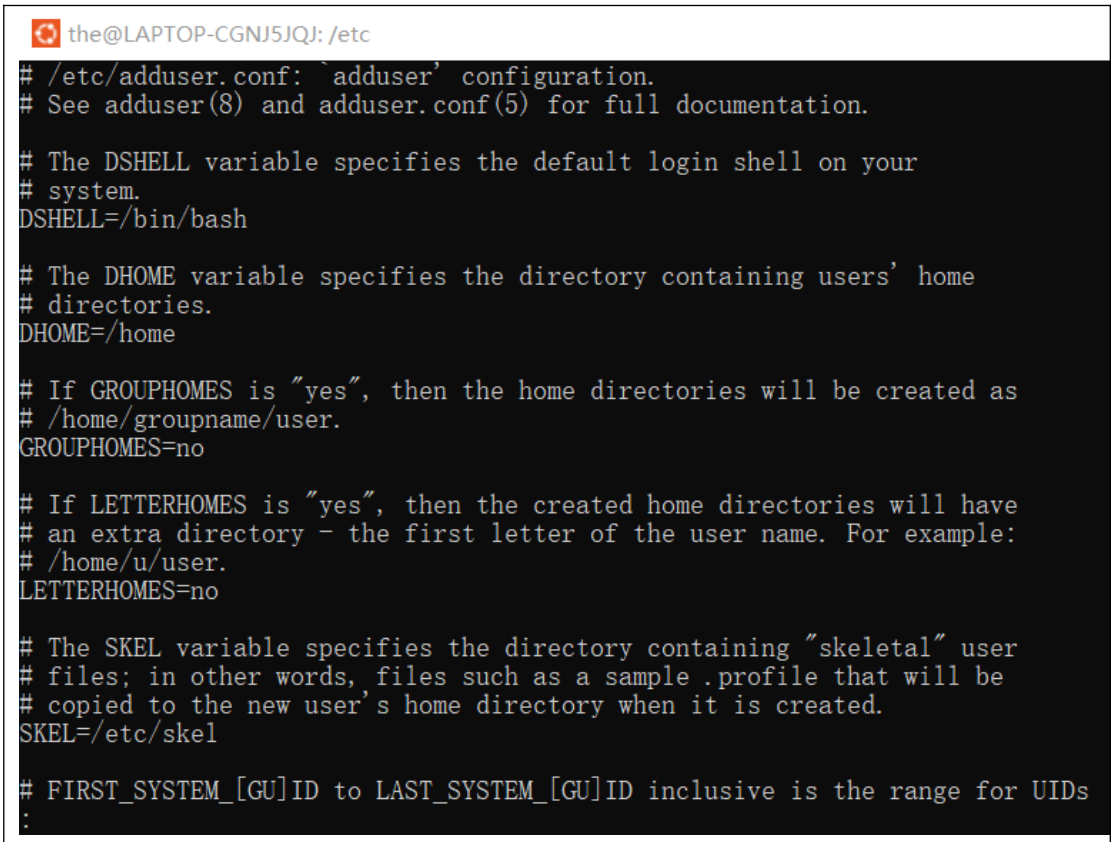
5、使用命令创建目录/home/lyj/ak，然后将/home/lyj/abc文件复制到该目录下，最后将该目录及其目录下的文件一起删除。



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facebook.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

```
the@LAPTOP-CGNJ5JQJ:~$ sort facebook.txt -t " " -k 1
baidu 100 5000
google 110 5000
guge 50 3000
sohu 100 4500
```

（2）按公司人数排序

```
the@LAPTOP-CGNJ5JQJ:~$ sort facebook.txt -t " " -k 2 -r
guge 50 3000
baidu 100 5000
sohu 100 4500
google 110 5000
the@LAPTOP-CGNJ5JQJ:~$
```

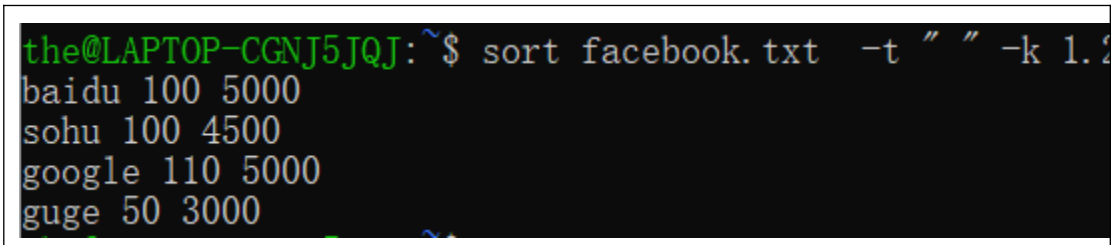
（3）按公司人数排序，人数相同的按照员工平均工资升序排序

```
the@LAPTOP-CGNJ5JQJ:~$ sort facebook.txt -n  -t " " -k 2 -k
guge 50 3000
sohu 100 4500
baidu 100 5000
google 110 5000
```

（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

```
the@LAPTOP-CGNJ5JQJ:~$ sort facebook.txt -n  -t " " -k3r  -
baidu 100 5000
google 110 5000
sohu 100 4500
guge 50 3000
```

（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

遇到的困难在最后一道题排序的部分，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	2021223280	姓　　名	刘嘉琦
专业班级	计算机科学与技术 06 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

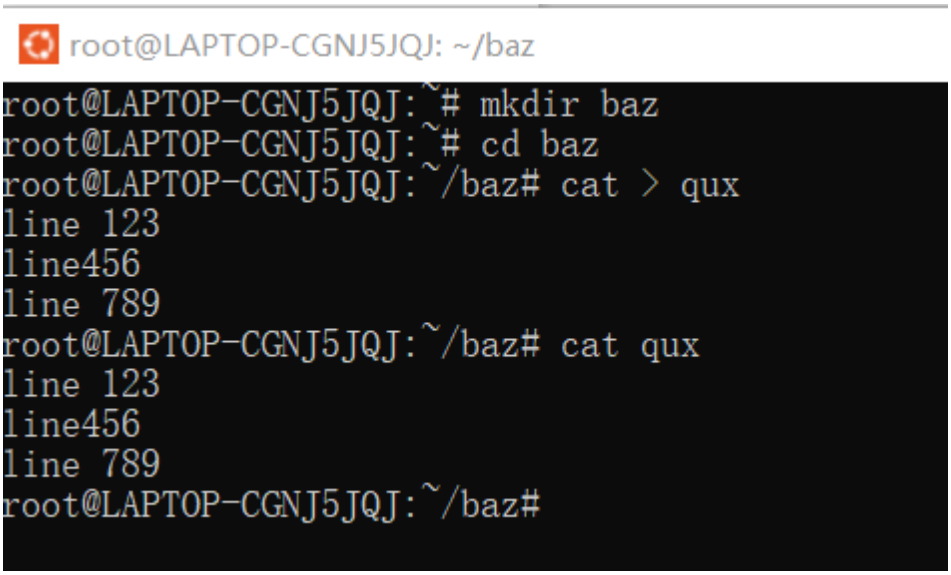
（1）计算机的硬件配置 PC 系列微机。

（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;

```
root@LAPTOP-CGNJ5JQJ:~# find  -name qux -print
./baz/qux
root@LAPTOP-CGNJ5JQJ:~#
```

（3）统计文件 qux 中所包含内容的行数、字数和字节数;

```
root@LAPTOP-CGNJ5JQJ:~# cd baz
root@LAPTOP-CGNJ5JQJ:~/baz# wc qux
 3  5 26 qux
root@LAPTOP-CGNJ5JQJ:~/baz#
```

（4）在用户目录下查找文件 qux ，并删除该文件;

```
root@LAPTOP-CGNJ5JQJ:~# find  -name qux -delete
root@LAPTOP-CGNJ5JQJ:~# find  -name qux
root@LAPTOP-CGNJ5JQJ:~#
```

（5）查看文件夹 baz 内容，看一下是否删除了文件 qux 。

```
root@LAPTOP-CGNJ5JQJ:~# cd baz
root@LAPTOP-CGNJ5JQJ:~/baz# ls -a
.  ..
root@LAPTOP-CGNJ5JQJ:~/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 ;

```
root@LAPTOP-CGNJ5JQJ:~# mkdir path1
root@LAPTOP-CGNJ5JQJ:~# cd path1
root@LAPTOP-CGNJ5JQJ:~/path1# touch file1
root@LAPTOP-CGNJ5JQJ:~/path1# touch file2
root@LAPTOP-CGNJ5JQJ:~/path1# ls
file1  file2
root@LAPTOP-CGNJ5JQJ:~/path1#
```

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

```
root@LAPTOP-CGNJ5JQJ:~# mkdir path2
root@LAPTOP-CGNJ5JQJ:~# cd path2
root@LAPTOP-CGNJ5JQJ:~/path2# touch file3
root@LAPTOP-CGNJ5JQJ:~/path2# cd ../
root@LAPTOP-CGNJ5JQJ:~# ls
baz  linux  locate  ml  path1  path2
root@LAPTOP-CGNJ5JQJ:~# cd path2
root@LAPTOP-CGNJ5JQJ:~/path2# ls
file3
root@LAPTOP-CGNJ5JQJ:~/path2#
```

（3）在用户目录下新建文件 file4 ；

```
root@LAPTOP-CGNJ5JQJ:~/path2# cd ../
root@LAPTOP-CGNJ5JQJ:~# mkdir file4
root@LAPTOP-CGNJ5JQJ:~# ls
baz  file4  linux  locate  ml  path1  path2
root@LAPTOP-CGNJ5JQJ:~#
```

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

```
root@LAPTOP-CGNJ5JQJ:~# tar -cvf package.tar path1 file4
path1/
path1/file1
path1/file2
file4/
```

（5）查看包 package.tar 的内容；

```
root@LAPTOP-CGNJ5JQJ:~# tar -tvf package.tar
drwxr-xr-x root/root         0 2023-05-23 22:38 path1/
-rw-r--r-- root/root         0 2023-05-23 22:38 path1/file1
-rw-r--r-- root/root         0 2023-05-23 22:38 path1/file2
drwxr-xr-x root/root         0 2023-05-23 22:43 file4/
```

（6）向包 package.tar 里添加文件夹 path2 的内容；

```
root@LAPTOP-CGNJ5JQJ:~# tar -rvf package.tar path2
path2/
path2/file3
```

```
root@LAPTOP-CGNJ5JQJ:~# tar -tvf package.tar
drwxr-xr-x root/root         0 2023-05-23 22:38 path1/
-rw-r--r-- root/root         0 2023-05-23 22:38 path1/file1
-rw-r--r-- root/root         0 2023-05-23 22:38 path1/file2
drwxr-xr-x root/root         0 2023-05-23 22:43 file4/
drwxr-xr-x root/root         0 2023-05-23 22:42 path2/
-rw-r--r-- root/root         0 2023-05-23 22:42 path2/file3
```

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

```
root@LAPTOP-CGNJ5JQJ:~# cp package.tar path3
root@LAPTOP-CGNJ5JQJ:~# cd path3
```

（8）进入 path3 文件夹，并还原包 package.tar 的内容。

```
root@LAPTOP-CGNJ5JQJ:~/path3# tar -xvf package.tar
path1/
path1/file1
path1/file2
file4/
path2/
path2/file3
```

3. 符号链接内容

（1）新建文件 foo.txt ，内容为 123 ；

```
root@LAPTOP-CGNJ5JQJ:~# touch foo.txt
root@LAPTOP-CGNJ5JQJ:~# echo 123 > foo.txt
root@LAPTOP-CGNJ5JQJ:~# cat foo
cat: foo: No such file or directory
root@LAPTOP-CGNJ5JQJ:~# cat foo.txt
123
root@LAPTOP-CGNJ5JQJ:~#
```

（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；

```
root@LAPTOP-CGNJ5JQJ:~# ln foo.txt bar.txt
root@LAPTOP-CGNJ5JQJ:~# comm foo.txt bar.txt
                123
root@LAPTOP-CGNJ5JQJ:~# diff foo.txt bar.txt
root@LAPTOP-CGNJ5JQJ:~#
```

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

```
root@LAPTOP-CGNJ5JQJ:~# ls -i foo.txt bar.txt
29429 bar.txt  29429 foo.txt
root@LAPTOP-CGNJ5JQJ:~#
```

（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;

```
root@LAPTOP-CGNJ5JQJ:~# echo abc > bar.txt
root@LAPTOP-CGNJ5JQJ:~# comm foo.txt bar.txt
                abc
root@LAPTOP-CGNJ5JQJ:~#
```

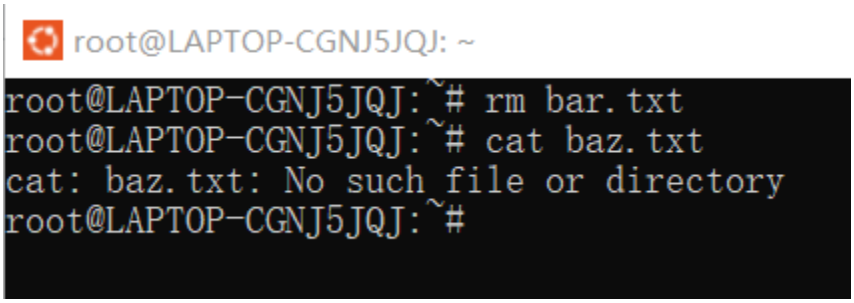
（5）删除 foo.txt 文件，然后查看 bar.txt 文件的 inode 及内容;

```
root@LAPTOP-CGNJ5JQJ:~# rm -rf foo.txt
root@LAPTOP-CGNJ5JQJ:~# cat bar.txt
abc
root@LAPTOP-CGNJ5JQJ:~# ls -i bar.txt
29429 bar.txt
root@LAPTOP-CGNJ5JQJ:~#
```

（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;

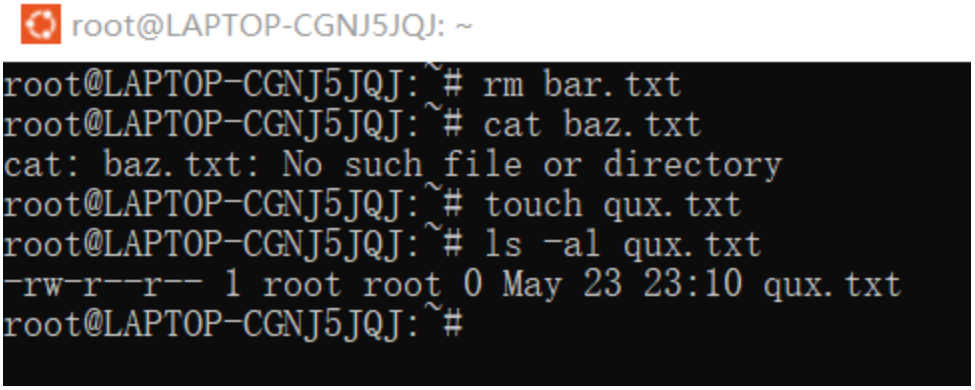
```
root@LAPTOP-CGNJ5JQJ:~# ln -s bar.txt baz.txt
root@LAPTOP-CGNJ5JQJ:~# comm bar.txt baz.txt
                abc
root@LAPTOP-CGNJ5JQJ:~# diff bar.txt baz.txt
root@LAPTOP-CGNJ5JQJ:~# ls -i bar.txt baz.txt
29429 bar.txt  29430 baz.txt
root@LAPTOP-CGNJ5JQJ:~#
```

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

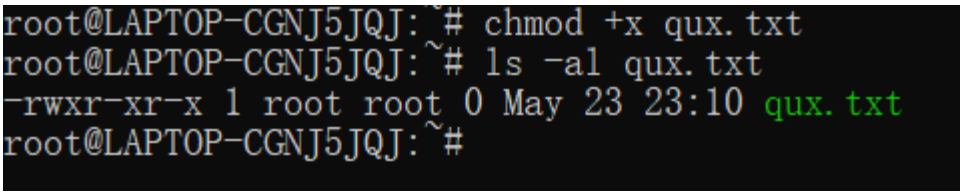


4. 权限管理

（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）



四、实验过程分析与讨论

注意区分压缩命令的区别：

打包压缩是：-zcvf

打包是：-cvf

解包是：-xvf

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	2021223280	姓　　名	刘嘉琦
专业班级	计算机科学与技术 06 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。

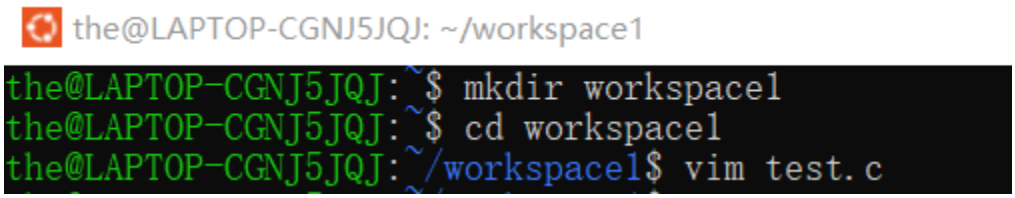
三、实验内容及结果

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 。



```
the@LAPTOP-CGNJ5JQJ: ~/workspace1
#inlcude <stdio.h>

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

（4）保存 test.c 的内容，并退出；

```
~
~
:wq
```

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

```
the@LAPTOP-CGNJ5JQJ:~/workspace1$ gcc test.c -o test
the@LAPTOP-CGNJ5JQJ:~/workspace1$ ./test
hello world!
the@LAPTOP-CGNJ5JQJ:~/workspace1$
```

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；

```
the@LAPTOP-CGNJ5JQJ:~/workspace1$ cd ../
the@LAPTOP-CGNJ5JQJ:~$ mkdir workspace2
the@LAPTOP-CGNJ5JQJ:~$ cd workspace2
the@LAPTOP-CGNJ5JQJ:~/workspace2$
```

（3）使用以下命令：

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；

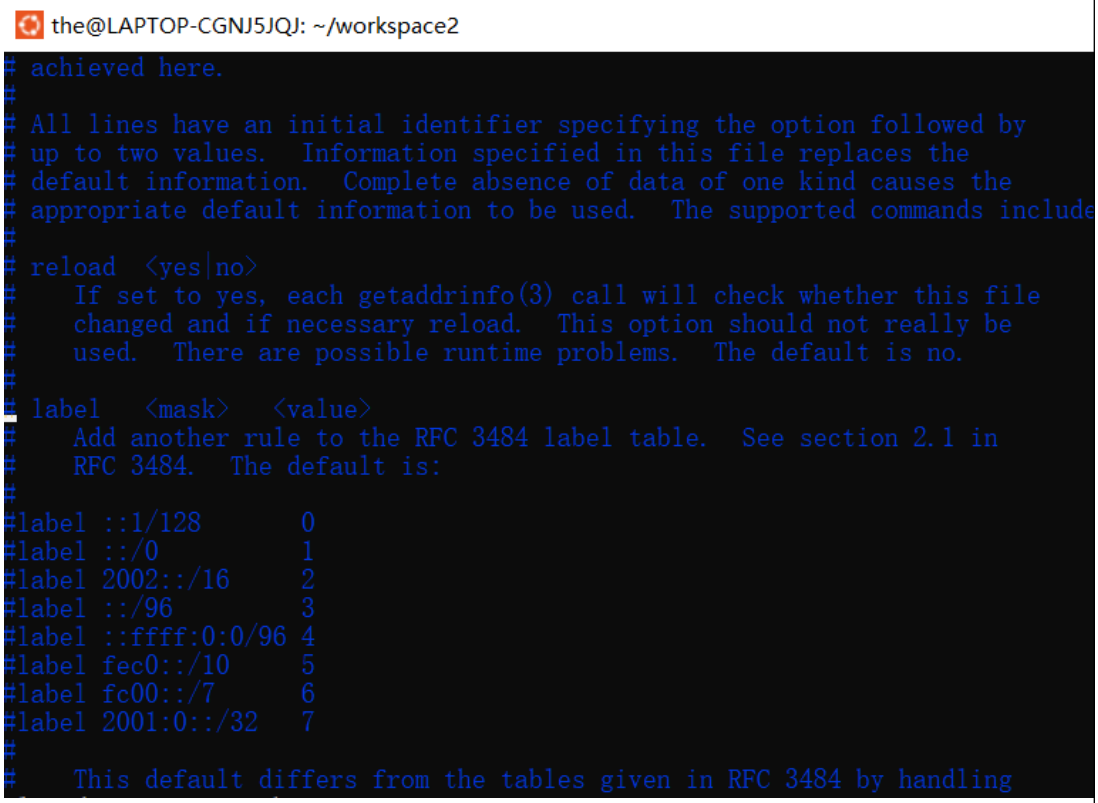
```
the@LAPTOP-CGNJ5JQJ:~/workspace2$ ls
the@LAPTOP-CGNJ5JQJ:~/workspace2$ cat /etc/gai.conf > ./gai.
the@LAPTOP-CGNJ5JQJ:~/workspace2$ ls
gai.conf
the@LAPTOP-CGNJ5JQJ:~/workspace2$
```

（4）使用 vim 编辑当前目录下的 gai.conf ；

```
the@LAPTOP-CGNJ5JQJ:~/workspace2$ vim gai.conf
```

（5）将光标移到第 18 行;

在命令模式下按18G



（6）复制该行内容;

命令模式下按yy

（7）将光标移到最后一行行首;

末行G

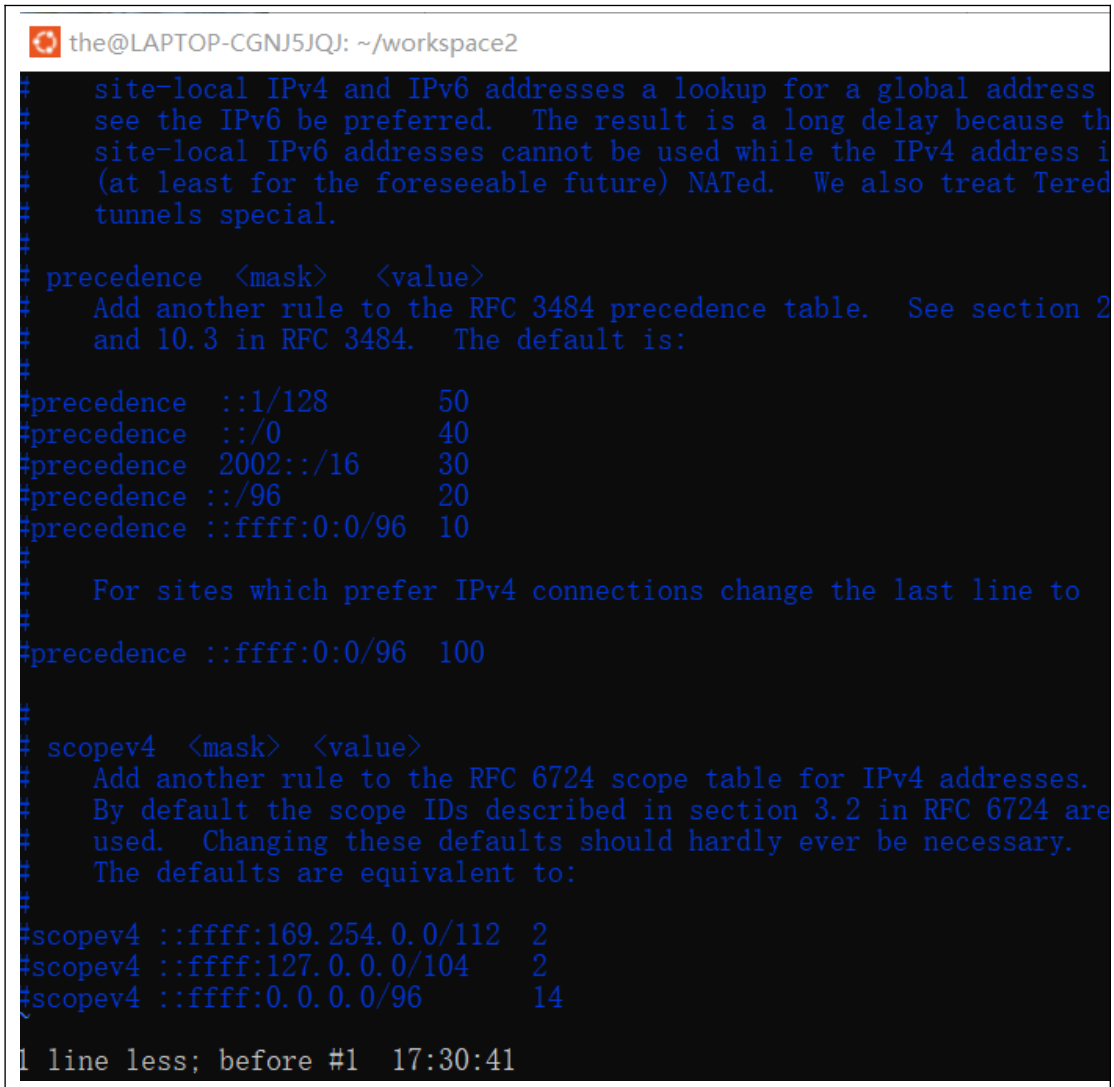
（8）粘贴复制行的内容;

p(默认粘贴在所在行的下一行)P（光标所在行之前）


the@LAPTOP-CGNJ5JQJ: ~/workspace2
site-local IPv4 and IPv6 addresses a lookup for a global address would
see the IPv6 be preferred. The result is a long delay because the
site-local IPv6 addresses cannot be used while the IPv4 address is
(at least for the foreseeable future) NATed. We also treat Teredo
tunnels special.
#
precedence <mask> <value>
Add another rule to the RFC 3484 precedence table. See section 2.1
and 10.3 in RFC 3484. The default is:
#
#precedence ::1/128   50
#precedence ::/0   40
#precedence 2002::/16   30
#precedence ::/96   20
#precedence ::ffff:0:0/96 10
#
For sites which prefer IPv4 connections change the last line to
#
#precedence ::ffff:0:0/96 100
#
scopev4 <mask> <value>
Add another rule to the RFC 6724 scope table for IPv4 addresses.
By default the scope IDs described in section 3.2 in RFC 6724 are
used. Changing these defaults should hardly ever be necessary.
The defaults are equivalent to:
#
#scopev4 ::ffff:169.254.0.0/112 2
#scopev4 ::ffff:127.0.0.0/104   2
label   <mask>   <value>
#scopev4 ::ffff:0.0.0.0/96   14

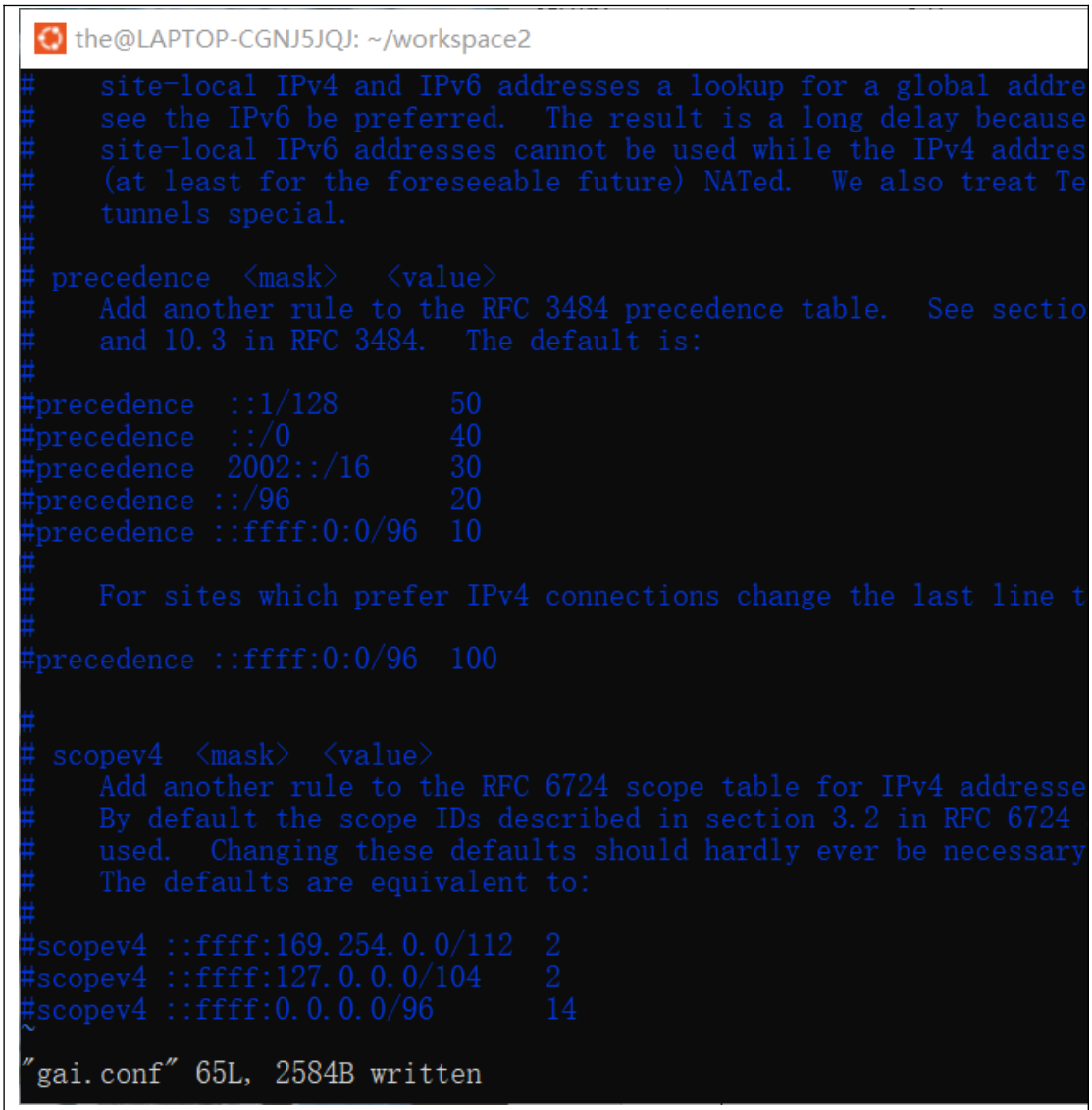
(9) 撤销第 8 步的动作;

(u)


the@LAPTOP-CGNJ5JQJ: ~/workspace2
site-local IPv4 and IPv6 addresses a lookup for a global address
see the IPv6 be preferred. The result is a long delay because th
site-local IPv6 addresses cannot be used while the IPv4 address i
(at least for the foreseeable future) NATed. We also treat Tered
tunnels special.
#
precedence <mask> <value>
Add another rule to the RFC 3484 precedence table. See section 2
and 10.3 in RFC 3484. The default is:
#
#precedence ::1/128   50
#precedence ::/0   40
#precedence 2002::/16   30
#precedence ::/96   20
#precedence ::ffff:0:0/96 10
#
For sites which prefer IPv4 connections change the last line to
#
#precedence ::ffff:0:0/96 100
#
scopev4 <mask> <value>
Add another rule to the RFC 6724 scope table for IPv4 addresses.
By default the scope IDs described in section 3.2 in RFC 6724 are
used. Changing these defaults should hardly ever be necessary.
The defaults are equivalent to:
#
#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 #1 17:30:41

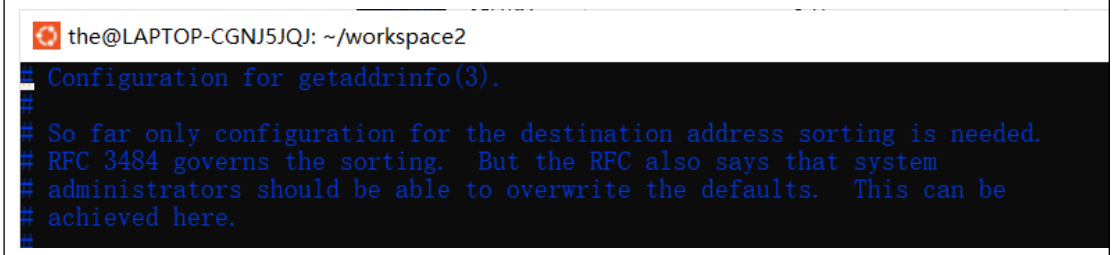
（10）存盘但不退出；

（：w）

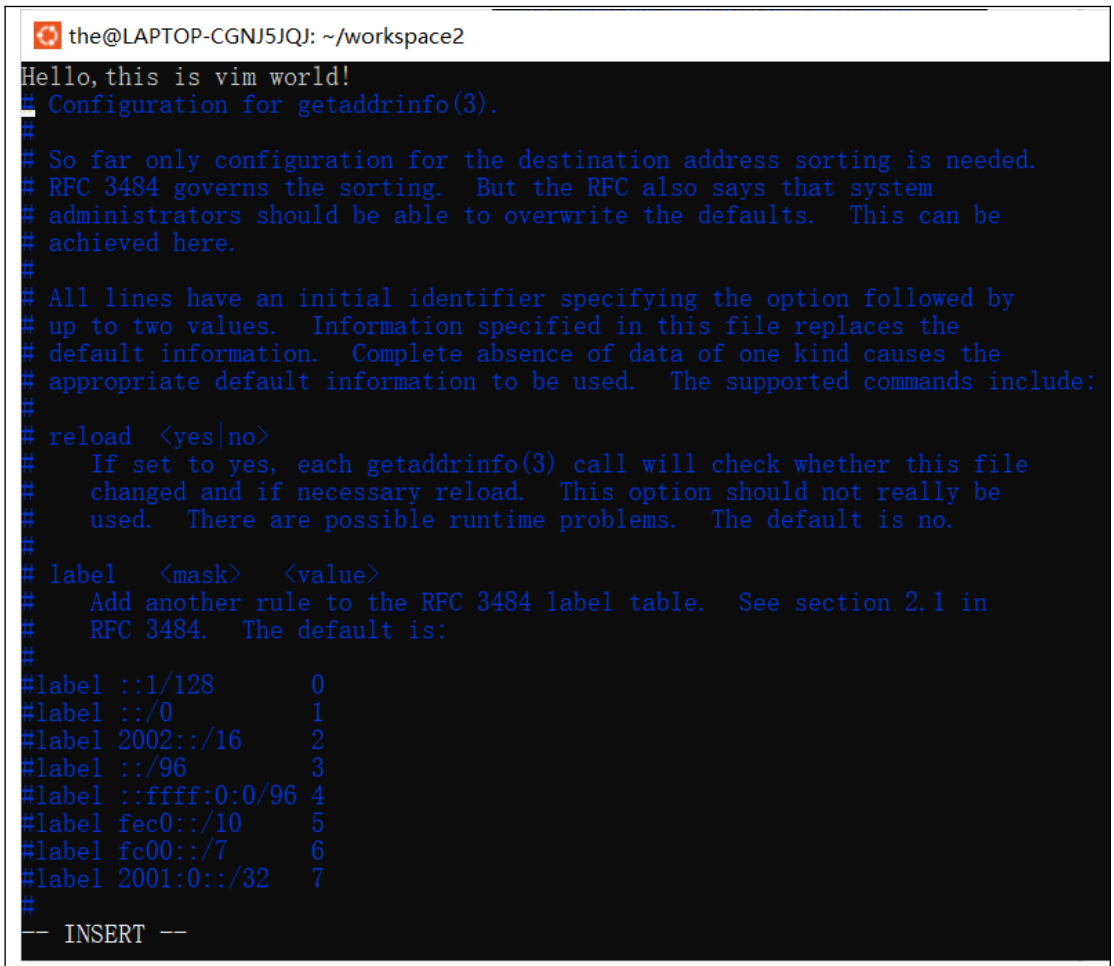


（11）将光标移到首行；

（gg）

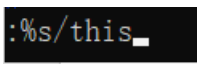


（12）插入模式下输入 "Hello, this is vim world!" ；



（13）删除字符串 "this"；

进入命令模式，输入：%s/this



（14）强制退出 vim ，不存盘

四、实验过程分析与讨论

在执行编译命令 gcc test.c -o test 时，提示 Command 'gcc' not found, but can be installed with:sudo apt install gcc，因此通过执行命令 sudo apt install gcc，并且在 install gcc 时通过 sudo apt-get update 顺利完成。

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	2021223280	姓　　名	刘嘉琦
专业班级	计算机科学与技术 06 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。

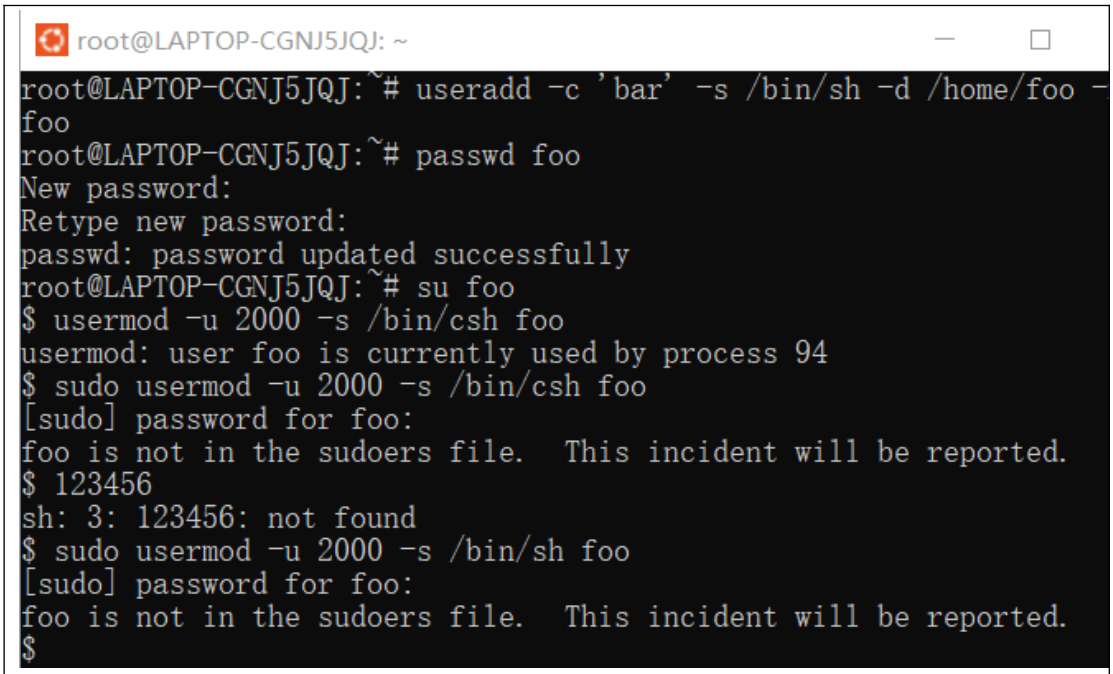
三、实验内容及结果

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@LAPTOP-CGNJ5JQJ:~# useradd -c 'bar' -s /bin/sh -d /home/foo -m
foo
root@LAPTOP-CGNJ5JQJ:~# passwd foo
New password:
Retype new password:
passwd: password updated successfully
```

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 ；

一直无法实现功能，代码截图如下



3. 从用户 foo 切换到 root ；

```
root@LAPTOP-CGNJ5JQJ:~# su foo
```

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@LAPTOP-CGNJ5JQJ:~# userdel foo -r
```

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@LAPTOP-CGNJ5JQJ:~# cat > user.txt <<EOF
> user001::600:100:user:/home/user001:/bin/bash
> user002::601:100:user:/home/user002:/bin/bash
> user003::602:100:user:/home/user003:/bin/bash
> user004::603:100:user:/home/user004:/bin/bash
> user005::604:100:user:/home/user005:/bin/bash
> EOF
root@LAPTOP-CGNJ5JQJ:~# cat user.txt
user001::600:100:user:/home/user001:/bin/bash
user002::601:100:user:/home/user002:/bin/bash
user003::602:100:user:/home/user003:/bin/bash

user004::603:100:user:/home/user004:/bin/bash
user005::604:100:user:/home/user005:/bin/bash
root@LAPTOP-CGNJ5JQJ:~#
```

```
root@LAPTOP-CGNJ5JQJ:~# newusers < user.txt
```

```
root@LAPTOP-CGNJ5JQJ:~# pwunconv
```

```
root@LAPTOP-CGNJ5JQJ:~# cat > passwd.txt <<EOF
> user001:123456
> user002:123456
> user003:123456
> user004:123456
> user005:123456
> user006:123456
> EOF
```

```
root@LAPTOP-CGNJ5JQJ:~# chpasswd < passwd.txt
```

```
root@LAPTOP-CGNJ5JQJ:~# pwunconv
```

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

```
root@LAPTOP-CGNJ5JQJ:~# groupadd group -g 3000
```

7. 在用户组 group1 中添加两个之前批量创建的用户;

```
root@LAPTOP-CGNJ5JQJ:~# usermod -g group user1
```

```
root@LAPTOP-CGNJ5JQJ:~# usermod -g group user2
```

四、实验过程分析与讨论

遇到的困难是转换用户 su userone 无法正常实现。

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 22 日
学　　号	2021223280	姓　　名	刘嘉琦
专业班级	计算机科学与技术 06 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);

```
the@LAPTOP-CGNJ5JQJ:~/scripts$ foo=200
the@LAPTOP-CGNJ5JQJ:~/scripts$ echo ${foo}
200
the@LAPTOP-CGNJ5JQJ:~/scripts$
```

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);

```
the@LAPTOP-CGNJ5JQJ:~/scripts$ bar=100
```

```
the@LAPTOP-CGNJ5JQJ:~/scripts$ echo ${bar}
100
the@LAPTOP-CGNJ5JQJ:~/scripts$ test ${bar} -gt 150
the@LAPTOP-CGNJ5JQJ:~/scripts$ echo $?
1
```

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 ）；

```
the@LAPTOP-CGNJ5JQJ:~/scripts$ vim test.sh
```

the@LAPTOP-CGNJ5JQJ: ~/scripts

```
# !/bin/bash\

hostname
date
~
~
~
~
~
~
~
~
~
~
"test.sh" 4L, 29B
```

```
the@LAPTOP-CGNJ5JQJ:~/scripts$ bash test.sh
LAPTOP-CGNJ5JQJ
Wed May 24 00:34:00 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；

the@LAPTOP-CGNJ5JQJ: ~/scripts

```
#! /bin/bash

read -p "请输入一个三位数" sum

let a=${sum}/100
let b=${sum}/10%10
let c=${sum}%10
d=${a} * ${a} * ${a} + ${b} * ${b} *${b} + ${c}*${c}*${c}
echo "${d}"
if (( ${sum} -eq ${d} ))
then
        echo "${sum}是水仙花数"
else
        echo "不是"
fi
```

```
the@LAPTOP-CGNJ5JQJ:~/scripts$ vim test2.sh
the@LAPTOP-CGNJ5JQJ:~/scripts$ bash test2.sh
请输入一个三位数153
test2.sh: line 8: teat: command not found

153是水仙花数
the@LAPTOP-CGNJ5JQJ:~/scripts$ bash test2.sh
请输入一个三位数124
test2.sh: line 8: teat: command not found

124是水仙花数
the@LAPTOP-CGNJ5JQJ:~/scripts$ bash test2.sh
请输入一个三位数370
test2.sh: line 8: teat: command not found

370是水仙花数
the@LAPTOP-CGNJ5JQJ:~/scripts$
```

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；

the@LAPTOP-CGNJ5JQJ: ~/scripts

```
#!/bin/bash
read a
read b
read c
let d=${a}+${b}+${c}
echo "sum= ${d}"
```

```
the@LAPTOP-CGNJ5JQJ:~/scripts$ vim test.sh
the@LAPTOP-CGNJ5JQJ:~/scripts$ bash test.sh
1
2
3
sum= 6
```

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。

```
the@LAPTOP-CGNJ5JQJ:~/scripts$ vim test5.sh
the@LAPTOP-CGNJ5JQJ:~/scripts$ bash test5.sh
input your grade:50
level is E
```

the@LAPTOP-CGNJ5JQJ: ~/scripts

```
#!/bin/bash
read -p "input your grade:" grade
if ((${grade} < 60))
then
        echo "level is E"
elif ((${grade} < 70))
then
        echo "level is D"
elif ((${grade} < 80))
then
        echo"C"
elif ((${grade} < 90))
then
        echo"B"
else echo"A"
fi
```

四、实验过程分析与讨论

在编写 shell 程序时遇到了很多困难，比如水仙花数成熟书写过程中各部分积一直无法正确表示。

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 25 日
学　　号	2021223280	姓　　名	刘嘉琦
专业班级	计算机科学与技术 06 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 。

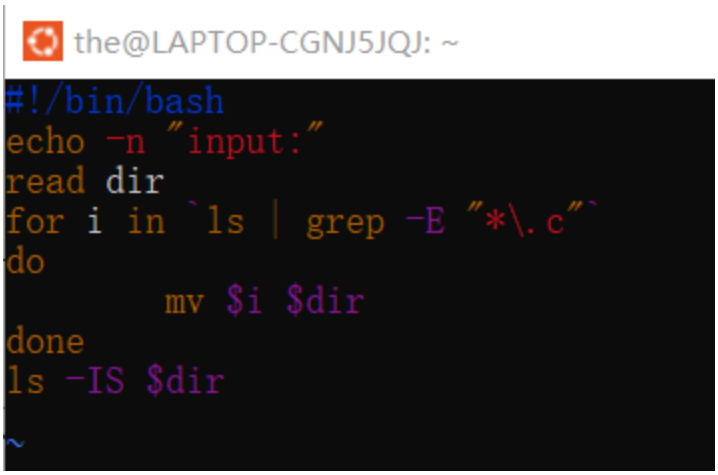
二、实验环境

（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。



```
the@LAPTOP-CGNJ5JQJ:~$ vim test61.sh
the@LAPTOP-CGNJ5JQJ:~$ bash test61.sh
input:workspace1
a.c   b.c   c.c   test   test.c
the@LAPTOP-CGNJ5JQJ:~$
```

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

the@LAPTOP-CGNJ5JQJ: ~/scripts

```
um=0;
i=2;
while ((($i <= 20 )) ; do
        sum=$((sum + i));
        ((i=i+2));
done;
echo $sum;
~
```

```
the@LAPTOP-CGNJ5JQJ:~$ cd scripts
the@LAPTOP-CGNJ5JQJ:~/scripts$ vim test62.sh
the@LAPTOP-CGNJ5JQJ:~/scripts$ bash test62.sh
110
the@LAPTOP-CGNJ5JQJ:~/scripts$
```

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

算出1-10的各个数的平方后，无法实现都加在一起

```
the@LAPTOP-CGNJ5JQJ:~/scripts$ bash test63.sh
test63.sh: line 2: [: -ge: unary operator expected
teat teat2.sh test test.sh test2.sh test5.sh test62.sh test63.sh test71.sh test72.sh test73.sh
1
4
9
16
25
36
49
64
81
100
```

the@LAPTOP-CGNJ5JQJ: ~/scripts

```
nt=1
sum=0
until [ $int -ge 11 ]
do
        sq=`expr $int \* $int`
        echo $sq
        int=`expr $int + 1`
        sum=`expr $sum + $sq`
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 、 break2 、 continue 、 continue 2 ，并观察四种情况下的实验结果。

1、break

```
the@LAPTOP-CGNJ5JQJ:~/scripts$ vim test64.sh
the@LAPTOP-CGNJ5JQJ:~/scripts$ bash test64.sh
a -n 1
-n 2
-n 3
-n 4

b -n 1
-n 2
-n 3
-n 4

c -n 1
-n 2
-n 3
-n 4

d -n 1
-n 2
-n 3
-n 4
```

2、break2

```
the@LAPTOP-CGNJ5JQJ:~/scripts$ vim test64.sh
the@LAPTOP-CGNJ5JQJ:~/scripts$ bash test64.sh
a -n 1
 -n 2
 -n 3
 -n 4
```

3、continue

```
the@LAPTOP-CGNJ5JQJ:~/scripts$ vim test64.sh
the@LAPTOP-CGNJ5JQJ:~/scripts$ bash test64.sh
a -n 1
 -n 2
 -n 3
 -n 4
 -n 6
 -n 7
 -n 8
 -n 9
 -n 10

b -n 1
 -n 2
 -n 3
 -n 4
 -n 6
 -n 7
 -n 8
 -n 9
 -n 10

c -n 1
 -n 2
 -n 3
 -n 4
 -n 6
 -n 7
 -n 8
 -n 9
 -n 10
```

```
d -n 1
 -n 2
 -n 3
 -n 4
 -n 6
 -n 7
 -n 8
 -n 9
 -n 10
```

4、continue 2

```
the@LAPTOP-CGNJ5JQJ:~/scripts$ vim test64.sh
the@LAPTOP-CGNJ5JQJ:~/scripts$ bash test64.sh
a -n 1
 -n 2
 -n 3
 -n 4
test64.sh: line 7: continuie: command not found
 -n 5
 -n 6
 -n 7
 -n 8
 -n 9
 -n 10

b -n 1
 -n 2
 -n 3
 -n 4
test64.sh: line 7: continuie: command not found
 -n 5
 -n 6
 -n 7
 -n 8
 -n 9
 -n 10

c -n 1
 -n 2
 -n 3
 -n 4
test64.sh: line 7: continuie: command not found
 -n 5
 -n 6
 -n 7
 -n 8
 -n 9
 -n 10
```

```
d -n 1
-n 2
-n 3
-n 4
test64.sh: line 7: continuie: command not found
-n 5
-n 6
-n 7
-n 8
-n 9
-n 10
```

四、实验过程分析与讨论

完成实验后对 until 和 while 有了更深的了解。

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 30 日
学　　号	2021223280	姓　　名	刘嘉琦
专业班级	计算机科学与技术 06 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

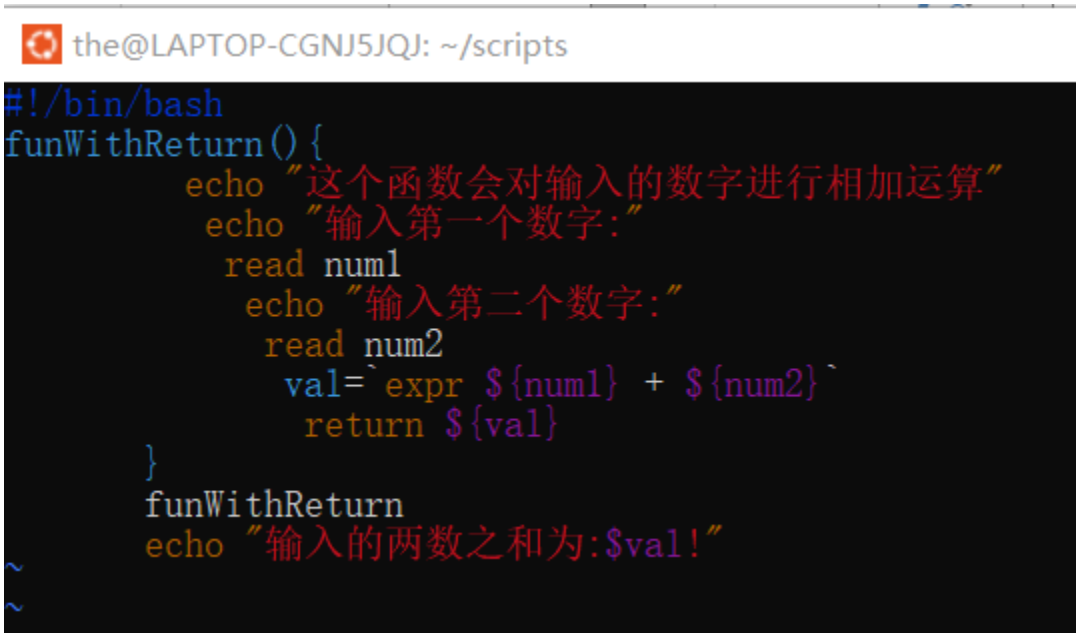
二、实验环境

（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;



```
the@LAPTOP-CGNJ5JQJ:~/scripts$ vim test71.sh
the@LAPTOP-CGNJ5JQJ:~/scripts$ bash test71.sh
这个函数会对输入的数字进行相加运算
输入第一个数字:
1
输入第二个数字:
2
输入的两数之和为:3!
```

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;

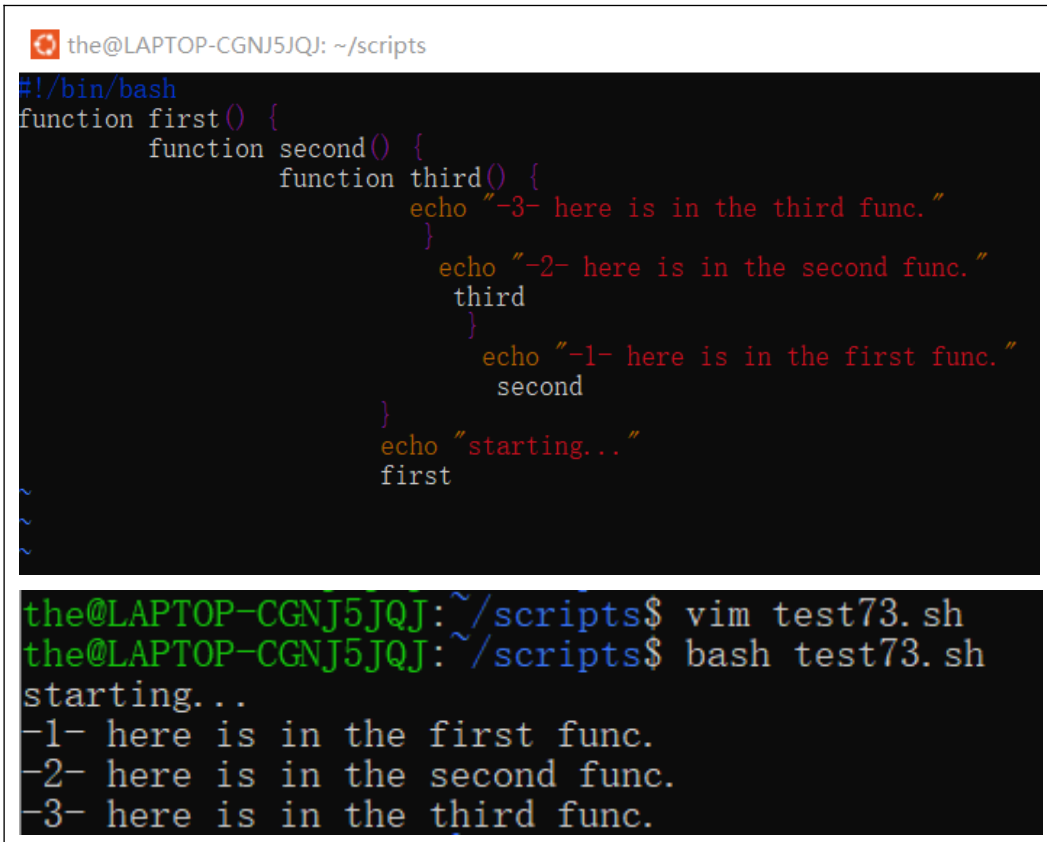
the@LAPTOP-CGNJ5JQJ: ~/scripts

```
#!/bin/bash
echo "请输入"
read n
echo
func ()
{
        local i="$1"
        if [ "$i" -eq 0 ]; then
                result=1
        else
                let "m=i-1"
                func "$m"
                let "result=$i * $?"
        fi
        return $result
}
func "$n"
echo "$n的阶乘为: $?"
```

```
the@LAPTOP-CGNJ5JQJ:~/scripts$ vim test72.sh
the@LAPTOP-CGNJ5JQJ:~/scripts$ bash test72.sh
请输入
3

3的阶乘为: 6
```

3. 一个Shell脚本的内容如下所示：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

理解了 shell 和子 shell 的关系，子 shell 和父 shell 的关系其实就是子进程和父进程的关系，只不过子 shell 和父 shell 是关联的进程是 bash 进程。

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 8 日
学　　号	2021223280	姓　　名	刘嘉琦
专业班级	计算机科学与技术 06 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。

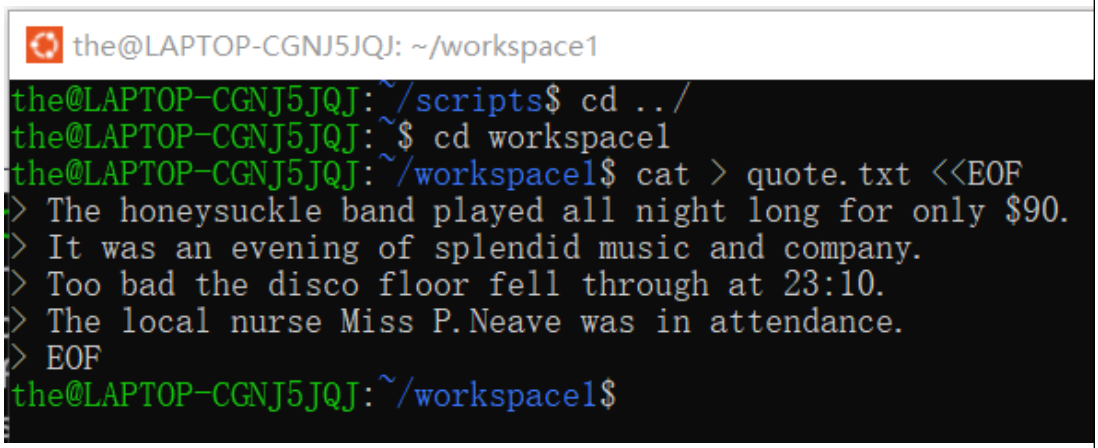
（2）计算机的软件配置 VMware 虚拟机软件及 Ubuntu 虚拟机。

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;

```
the@LAPTOP-CGNJ5JQJ:~/workspace1$ cat quote.txt | sed 's/\$//g'
The honeysuckle band played all night long for only 0.
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.
the@LAPTOP-CGNJ5JQJ:~/workspace1$
```

（2）显示包含 music 文字的行内容及行号;

```
the@LAPTOP-CGNJ5JQJ:~/workspace1$ nl quote.txt | sed -n '/music/p'
     2  It was an evening of splendid music and company.
the@LAPTOP-CGNJ5JQJ:~/workspace1$
```

（3）在第 4 行后面追加内容： "hello world!" ;

```
the@LAPTOP-CGNJ5JQJ:~/workspace1$ sed -e 4a'Hello world!' quote.txt
The honeysuckle band played all night long for only 0.
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" ;

```
the@LAPTOP-CGNJ5JQJ:~/workspace1$ sed 's/The/Quod/g' quote.txt
Quod honeysuckle band played all night long for only 0.
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." ;

```
the@LAPTOP-CGNJ5JQJ:~/workspace1$ nl quote.txt | sed '2d'
     1  The honeysuckle band played all night long for only 0.
     3  Too bad the disco floor fell through at 23:10.
     4  The local nurse Miss P.Neave was in attendance.
```

（6）删除第 2 行内容;

```
the@LAPTOP-CGNJ5JQJ:~/workspace1$ nl quote.txt | sed '2d'
     1  The honeysuckle band played all night long for only 0.
     3  Too bad the disco floor fell through at 23:10.
     4  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。

```
the@LAPTOP-CGNJ5JQJ:~/workspace1$ var='evening'
the@LAPTOP-CGNJ5JQJ:~/workspace1$ echo $var
evening
the@LAPTOP-CGNJ5JQJ:~/workspace1$ set | nl - | sed -n '/evening/p'
    59  _=evening
    65  var=evening
```

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；

the@LAPTOP-CGNJ5JQJ: ~/workspace1

```
the@LAPTOP-CGNJ5JQJ:~/workspace1$ cat > numbers.txt <<EOF
> one : two : three
> four : five : six
> EOF
the@LAPTOP-CGNJ5JQJ:~/workspace1$
```

```
the@LAPTOP-CGNJ5JQJ:~/workspace1$ awk -F ' ' '{print $2}' numbers.txt
:
:
the@LAPTOP-CGNJ5JQJ:~/workspace1$ awk -F ':' '{print $2}' numbers.txt
 two
 five
```

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。

the@LAPTOP-CGNJ5JQJ: ~/workspace1

```
#!/bin/bash
read -p "input filename: " filename
echo "even:"
export num=0
while read line
do
        a1='echo $line | awk '{print $1}' '
        a2='echo $line | awk '{print $2}' '
        a3='echo $line | awk '{print $3}' '
        if (($a1%2 == 0))
        then
                let num=$num+1
                echo $a1

        fi
        if (($a2%2 == 0))
        then
                let num=$num+1
                echo $a2
        fi
        if (($a3%2 == 0))
        then
                let num=$num+1
                echo $a3
        fi
done <$filename
echo $num
~
~
```

```
the@LAPTOP-CGNJ5JQJ:~/workspace1$ cat >foo.txt <<EOF
> 2 4 6
> 15 16 17
> 9 8 7
> EOF
```

```
input filename: foo.txt
2
4
6
16
8
5
```

4. 脚本的内容如下所示:

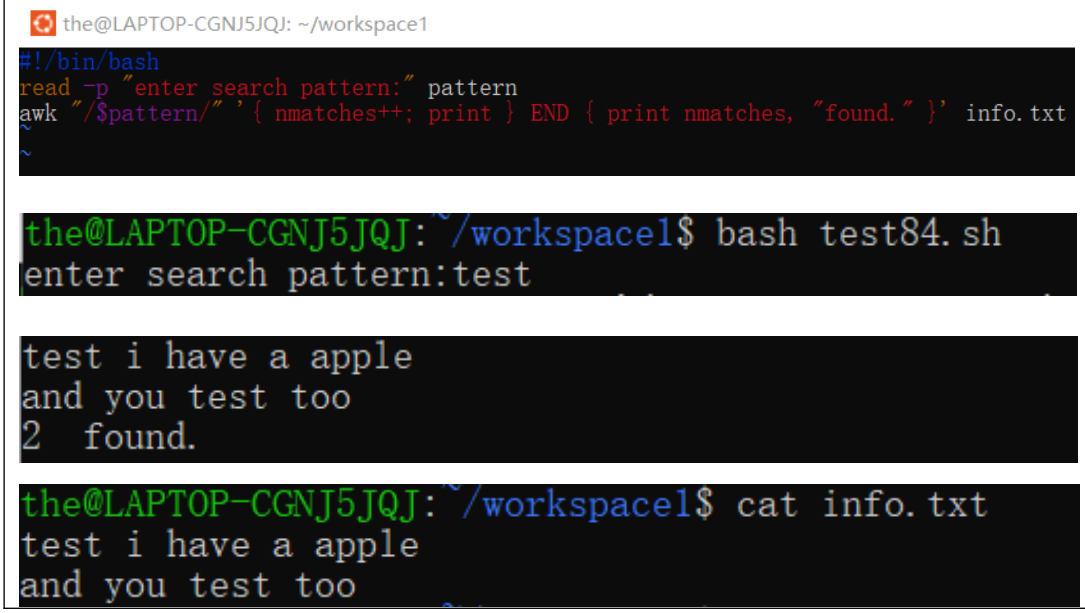
```
#!/bin/bash

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。

在test文件中查找存在pattern的行，并输出该行，同时计数。



四、实验过程分析与讨论

对 shell 和子 shell 的关系理解的更深入了，应用方面更加得心应手。

五、指导教师意见

指导教师签字：卢洋