

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	2018214390	姓 名	高钰皓
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

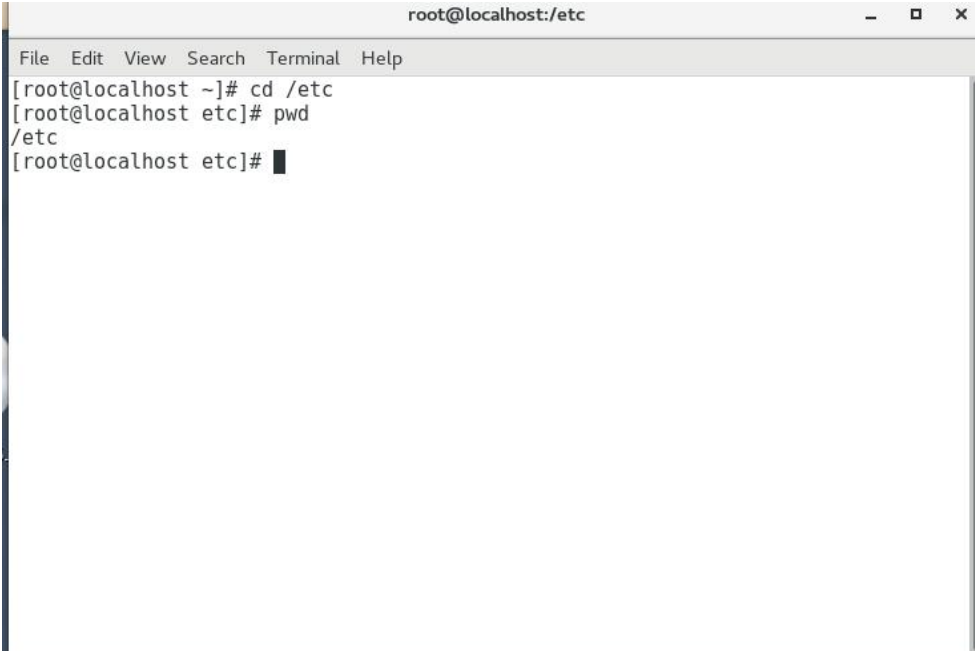
- 1、掌握Linux下文件和目录操作命令：cd、ls、mkdir、rmdir、rm
- 2、掌握Linux下文件信息显示命令：cat、more、head、tail
- 3、掌握Linux下文件复制、删除及移动命令：cp、mv
- 4、掌握 Linux 的文件排序命令：sort

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

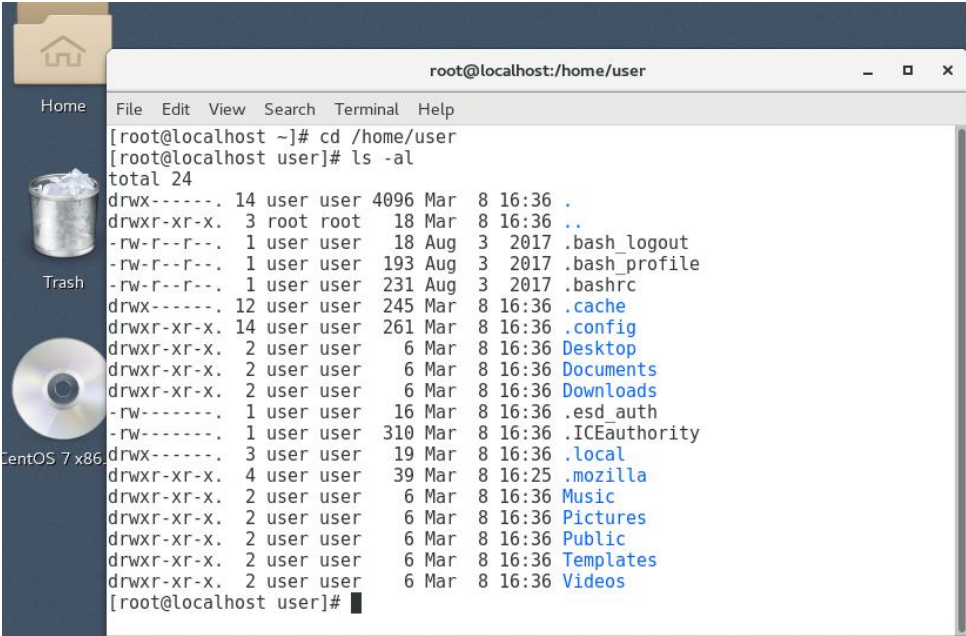
1. 使用命令切换到/etc 目录，并显示当前工作目录路径



A terminal window titled 'root@localhost:/etc' with a menu bar (File, Edit, View, Search, Terminal, Help). The command history shows: [root@localhost ~]# cd /etc, [root@localhost etc]# pwd, and the output /etc. The prompt is [root@localhost etc]#.

```
root@localhost:/etc
File Edit View Search Terminal Help
[root@localhost ~]# cd /etc
[root@localhost etc]# pwd
/etc
[root@localhost etc]#
```

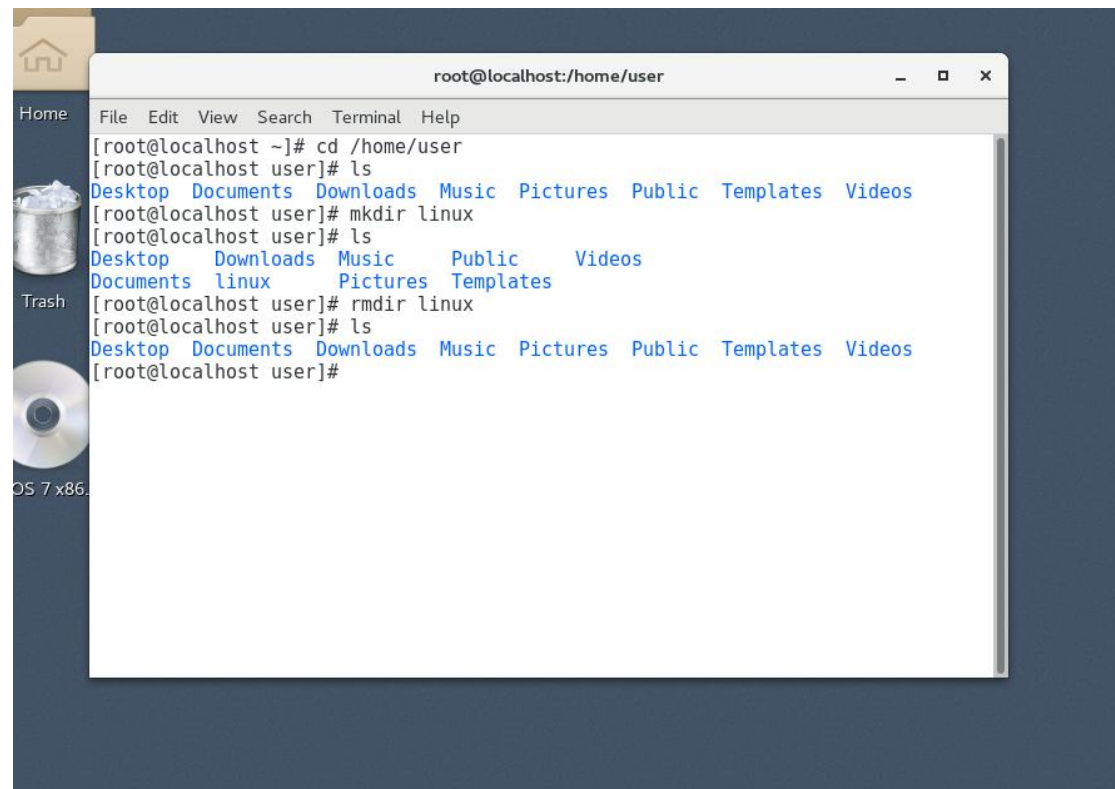
2、使用命令显示/home/user 目录下所有文件目录的详细信息，包括隐藏文件。



A desktop environment with a sidebar showing 'Home' and 'Trash' icons. A terminal window titled 'root@localhost:/home/user' is open, showing the command 'ls -al' and its output. The output lists files and directories with permissions, owner, group, size, date, and time.

```
root@localhost:/home/user
File Edit View Search Terminal Help
[root@localhost ~]# cd /home/user
[root@localhost user]# ls -al
total 24
drwx-----, 14 user user 4096 Mar  8 16:36 .
drwxr-xr-x,  3 root root  18 Mar  8 16:36 ..
-rw-r--r--,  1 user user  18 Aug  3 2017 .bash_logout
-rw-r--r--,  1 user user 193 Aug  3 2017 .bash_profile
-rw-r--r--,  1 user user 231 Aug  3 2017 .bashrc
drwx-----, 12 user user 245 Mar  8 16:36 .cache
drwxr-xr-x, 14 user user 261 Mar  8 16:36 .config
drwxr-xr-x,  2 user user   6 Mar  8 16:36 Desktop
drwxr-xr-x,  2 user user   6 Mar  8 16:36 Documents
drwxr-xr-x,  2 user user   6 Mar  8 16:36 Downloads
-rw-----,  1 user user  16 Mar  8 16:36 .esd_auth
-rw-----,  1 user user 310 Mar  8 16:36 .ICEauthority
drwx-----,  3 user user  19 Mar  8 16:36 .local
drwxr-xr-x,  4 user user  39 Mar  8 16:25 .mozilla
drwxr-xr-x,  2 user user   6 Mar  8 16:36 Music
drwxr-xr-x,  2 user user   6 Mar  8 16:36 Pictures
drwxr-xr-x,  2 user user   6 Mar  8 16:36 Public
drwxr-xr-x,  2 user user   6 Mar  8 16:36 Templates
drwxr-xr-x,  2 user user   6 Mar  8 16:36 Videos
[root@localhost user]#
```

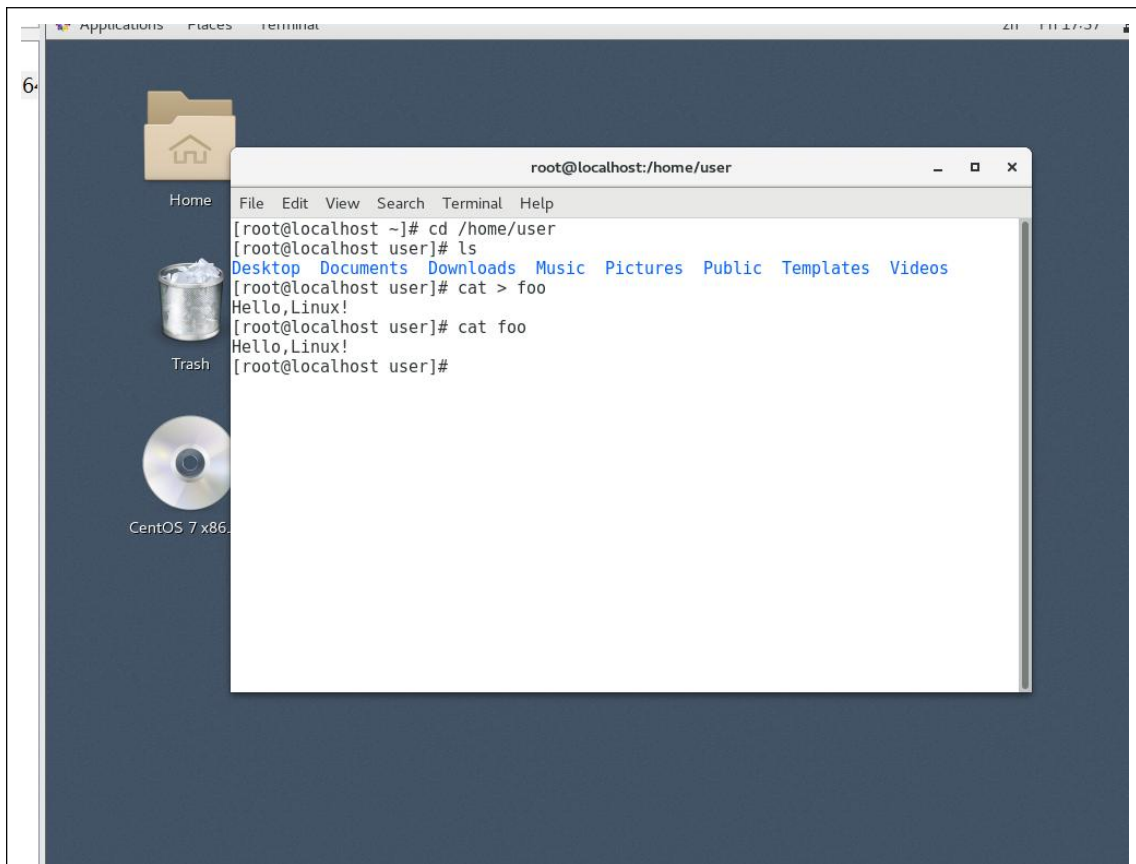
3、使用命令创建目录/home/user/linux，然后删除该目录。



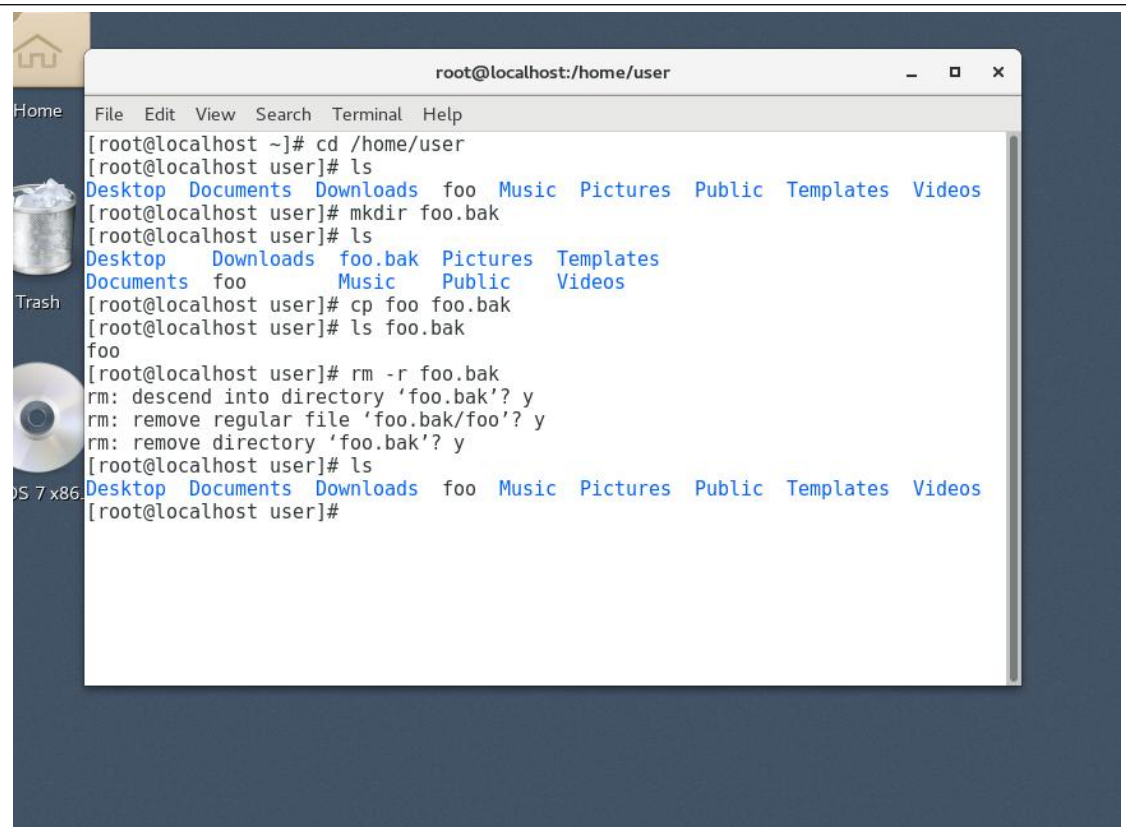
The screenshot shows a Linux desktop environment with a dark blue background. On the left side, there is a sidebar with icons for 'Home' (a house icon), 'Trash' (a trash can icon), and 'OS 7 x86_64' (a CD icon). A terminal window is open in the center, titled 'root@localhost:/home/user'. The terminal shows the following commands and output:

```
root@localhost: ~/# cd /home/user
root@localhost user# ls
Desktop Documents Downloads Music Pictures Public Templates Videos
root@localhost user# mkdir linux
root@localhost user# ls
Desktop Downloads Music Public Videos
Documents linux Pictures Templates
root@localhost user# rmdir linux
root@localhost user# ls
Desktop Documents Downloads Music Pictures Public Templates Videos
root@localhost user#
```

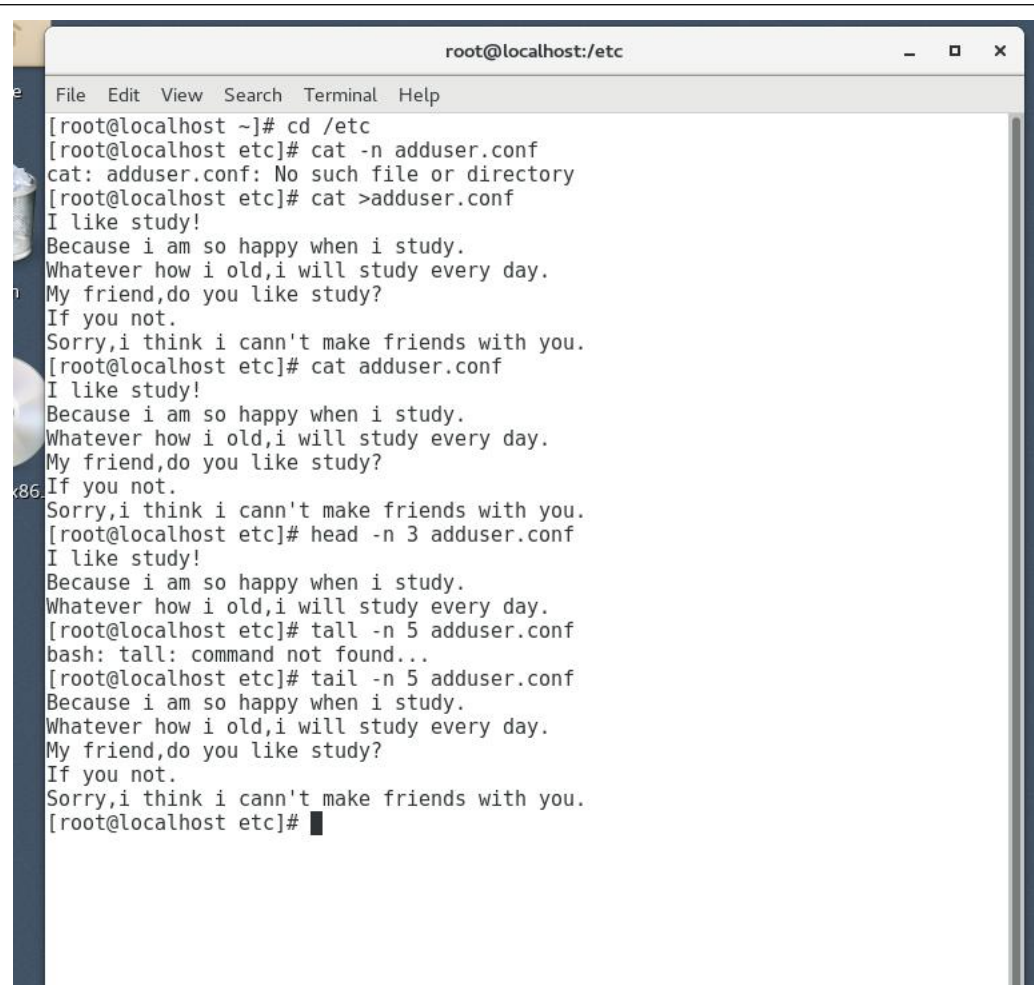
4、使用命令 cat 用输出重定向在/home/user 目录下创建文件 foo，文件内容为“Hello, Linux!”，并查看该文件的内容



5、使用命令创建目录 `/home/user/foo.bak`, 然后将 `/home/user/foo` 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。

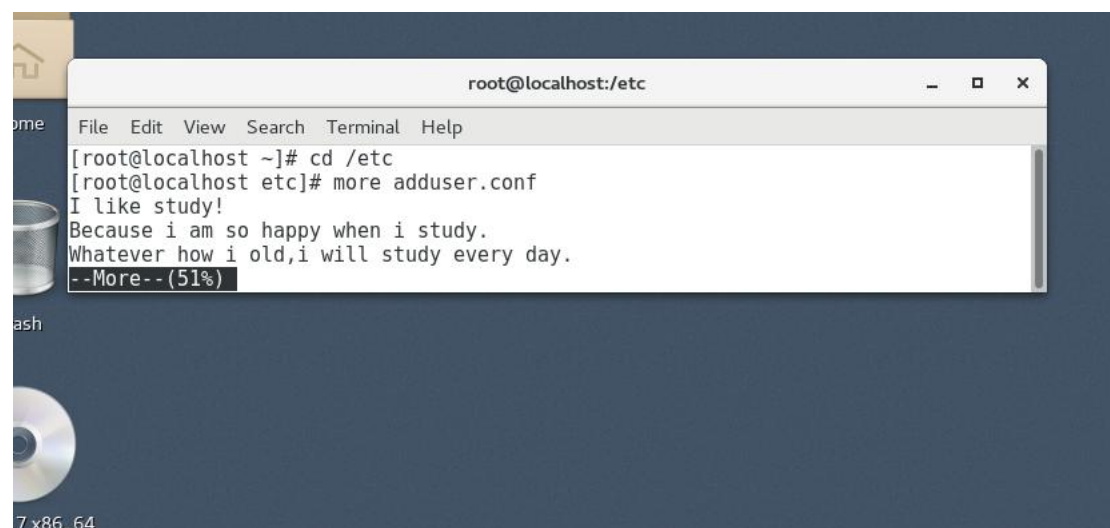


6、查看文件 `/etc/adduser.conf` 的前 3 行内容，查看文件 `/etc/adduser.conf` 的最后 5 行内容。不存在该文件，所以使用 `cat` 重定向输入文件。

A terminal window titled 'root@localhost:/etc' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows a series of commands and their outputs. The user navigates to /etc and attempts to cat 'adduser.conf', which fails with 'No such file or directory'. Then, they use '>' to create the file and enter several lines of text. Subsequent 'cat' and 'head' commands successfully display the file's content. A 'tall' command fails as it's not found, while a 'tail' command successfully shows the last five lines of the file.

```
root@localhost:/etc
File Edit View Search Terminal Help
[root@localhost ~]# cd /etc
[root@localhost etc]# cat -n adduser.conf
cat: adduser.conf: No such file or directory
[root@localhost etc]# cat >adduser.conf
I like study!
Because i am so happy when i study.
Whatever how i old,i will study every day.
My friend,do you like study?
If you not.
Sorry,i think i cann't make friends with you.
[root@localhost etc]# cat adduser.conf
I like study!
Because i am so happy when i study.
Whatever how i old,i will study every day.
My friend,do you like study?
If you not.
Sorry,i think i cann't make friends with you.
[root@localhost etc]# head -n 3 adduser.conf
I like study!
Because i am so happy when i study.
Whatever how i old,i will study every day.
[root@localhost etc]# tall -n 5 adduser.conf
bash: tall: command not found...
[root@localhost etc]# tail -n 5 adduser.conf
Because i am so happy when i study.
Whatever how i old,i will study every day.
My friend,do you like study?
If you not.
Sorry,i think i cann't make friends with you.
[root@localhost etc]#
```

7、分屏查看文件/etc/adduser.conf 的内容。

A terminal window titled 'root@localhost:/etc' with a menu bar (File, Edit, View, Search, Terminal, Help). The user navigates to /etc and uses the 'more' command to view 'adduser.conf'. The terminal shows the first three lines of the file, followed by a prompt '--More-- (51%)' indicating that the file is being paginated.

```
root@localhost:/etc
File Edit View Search Terminal Help
[root@localhost ~]# cd /etc
[root@localhost etc]# more adduser.conf
I like study!
Because i am so happy when i study.
Whatever how i old,i will study every day.
--More-- (51%)
```

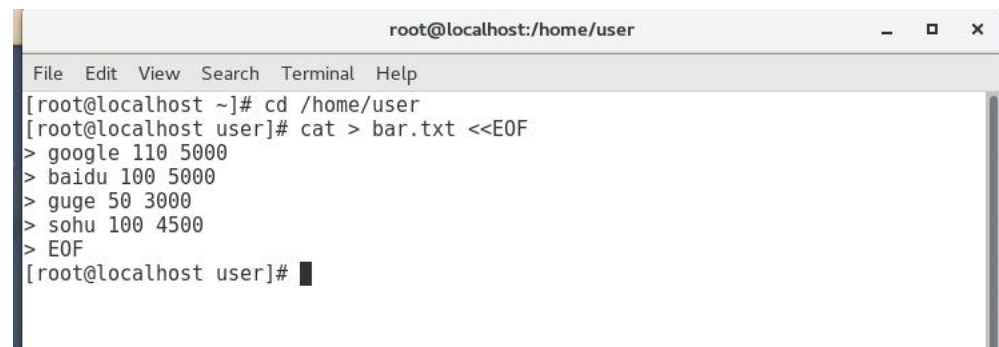
8、使用命令cat用输出重定向在/home/user目录下创建文件bar.txt，文件内容为：

google 110 5000

baidu 100 5000

guge 50 3000

sohu 100 4500



```
root@localhost:/home/user
File Edit View Search Terminal Help
[root@localhost ~]# cd /home/user
[root@localhost user]# cat > bar.txt <<EOF
> google 110 5000
> baidu 100 5000
> guge 50 3000
> sohu 100 4500
> EOF
[root@localhost user]#
```

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。


```
root@localhost:/home/user
File Edit View Search Terminal Help
> sohu 100 4500
> EOF
[root@localhost user]# sort bar.txt
baidu 100 5000
google 110 5000
guge 50 3000
sohu 100 4500
[root@localhost user]# sort -n -k 2 bar.txt
guge 50 3000
baidu 100 5000
sohu 100 4500
google 110 5000
[root@localhost user]# sort -n -k 2 -k 3 bar.txt
guge 50 3000
sohu 100 4500
baidu 100 5000
google 110 5000
[root@localhost user]# sort -n -k 3r -k 2 bar.txt
baidu 100 5000
google 110 5000
sohu 100 4500
guge 50 3000
[root@localhost user]# sort -k 1.2
^C
[root@localhost user]# sort -k 1.2 bar.txt
baidu 100 5000
sohu 100 4500
google 110 5000
guge 50 3000
[root@localhost user]#
```

四、 实验过程分析与讨论

刚开始队 Linux 系统完全不了解，对在 shell 窗口中进行文件的操作也无从下手，但好在对各个命令进行了尝试，在一次次看学习视频、使用、纠错、网上查资料等一系列学习活动后，渐渐熟悉了文件的操作，在图书馆借阅了相关图书后，更是对 Linux 系统有了更进一步的了解，对于各种文件操作逐渐得心应手。

但是，仍然存在的问题是：对于命令，还记不熟练，仅掌握了实验用到的这些操作，对于其他文件操作，仍需要一边看书查找，一边完成。

在这一过程中，我认识到，Linux 系统的诸多优点与广阔的发展

前景。而对于本门课程，我个人理解的是，它更倾向于一门工具课，想要学好，必须要在不断实践练习中学习，不能纸上谈兵，而在实践时抱有目标的学习，也使我更有学习的动力，并且在每次成功掌握一个命令操作后都感受到了一种满足与快乐。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 14 日
学 号	2018214390	姓 名	高钰皓
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

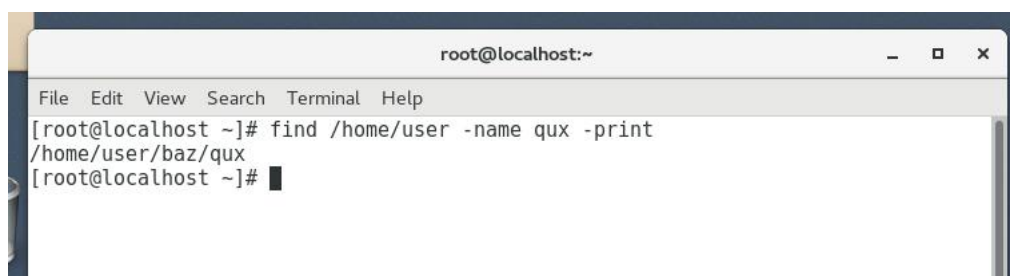
1. 查找指定文件

(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；



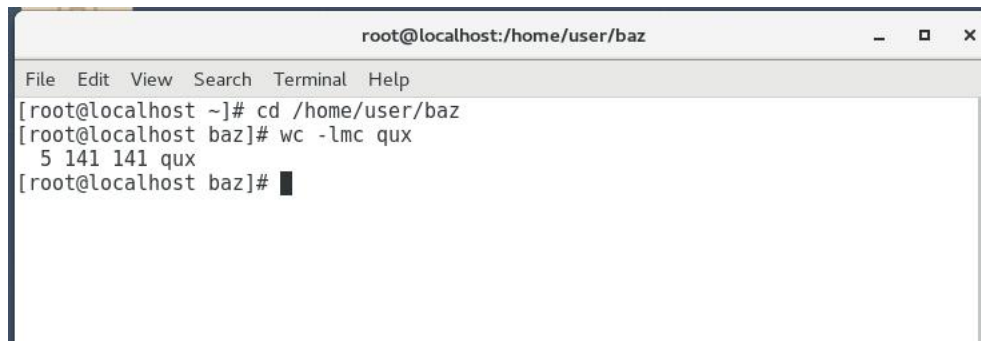
```
root@localhost:/home/user/baz
File Edit View Search Terminal Help
[root@localhost ~]# cd /home/user
[root@localhost user]# mkdir baz
[root@localhost user]# cd baz
[root@localhost baz]# cat > qux
I like study!
I study everyday!
I like study because it makes me happy!
My friend, do you like study?
If you not,sorry,we cann't be friends.
[root@localhost baz]#
```

(2) 在用户目录下查找文件 qux，并显示该文件位置信息；



```
root@localhost:~
File Edit View Search Terminal Help
[root@localhost ~]# find /home/user -name qux -print
/home/user/baz/qux
[root@localhost ~]#
```

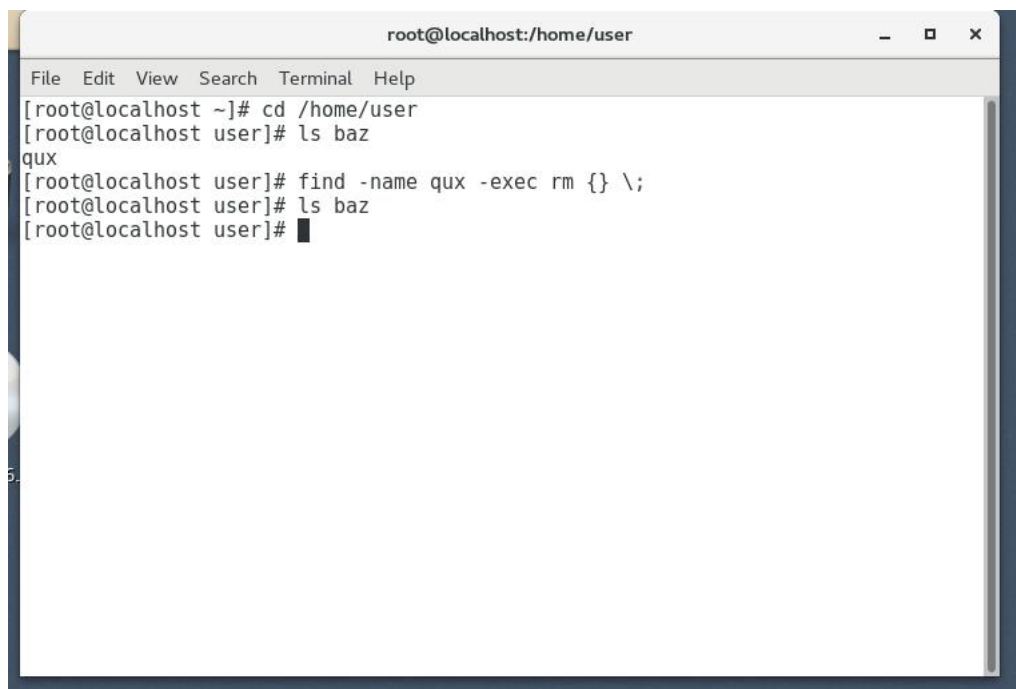
(3)统计文件 qux 中所包含内容的行数、字数和字节数;



```
root@localhost:/home/user/baz
File Edit View Search Terminal Help
[root@localhost ~]# cd /home/user/baz
[root@localhost baz]# wc -lmc qux
 5 141 141 qux
[root@localhost baz]#
```

(4)在用户目录下查找文件 qux , 并删除该文件;

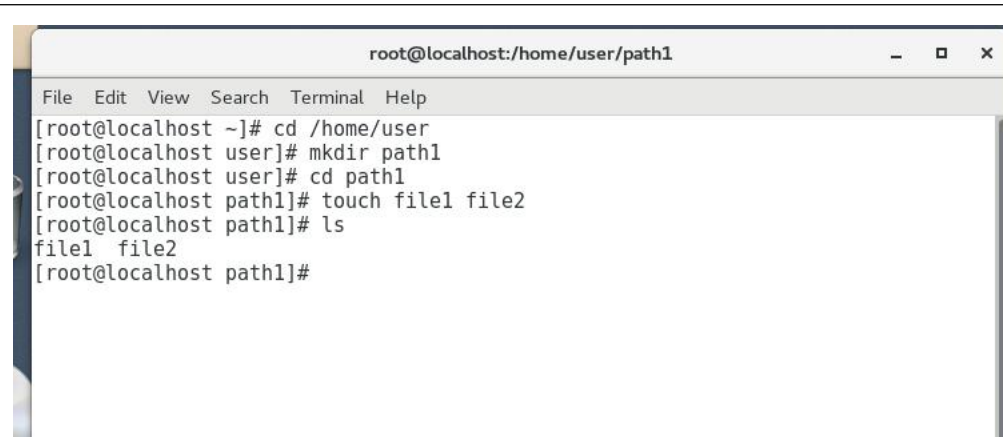
(5)查看文件夹 baz 内容, 看一下是否删除了文件 qux 。



```
root@localhost:/home/user
File Edit View Search Terminal Help
[root@localhost ~]# cd /home/user
[root@localhost user]# ls baz
qux
[root@localhost user]# find -name qux -exec rm {} \;
[root@localhost user]# ls baz
[root@localhost user]#
```

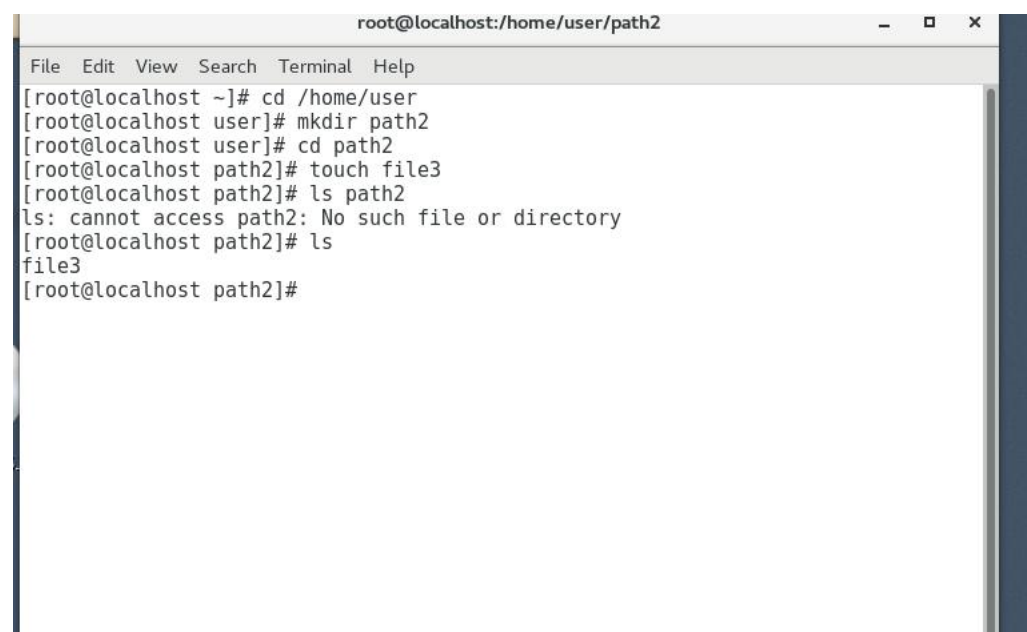
2、文件打包

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



```
root@localhost:/home/user/path1
File Edit View Search Terminal Help
[root@localhost ~]# cd /home/user
[root@localhost user]# mkdir path1
[root@localhost user]# cd path1
[root@localhost path1]# touch file1 file2
[root@localhost path1]# ls
file1  file2
[root@localhost path1]#
```

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



```
root@localhost:/home/user/path2
File Edit View Search Terminal Help
[root@localhost ~]# cd /home/user
[root@localhost user]# mkdir path2
[root@localhost user]# cd path2
[root@localhost path2]# touch file3
[root@localhost path2]# ls path2
ls: cannot access path2: No such file or directory
[root@localhost path2]# ls
file3
[root@localhost path2]#
```

(3) 在用户目录下新建文件 file4；

```
root@localhost:/home/user
File Edit View Search Terminal Help
[root@localhost ~]# cd /home/user
[root@localhost user]# touch file4
[root@localhost user]# ls
Desktop  Downloads  Music  path2  Public  Videos
Documents file4      path1  Pictures Templates
[root@localhost user]#
```

(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@localhost:/home/user
File Edit View Search Terminal Help
[root@localhost ~]# cd /home/user
[root@localhost user]# tar -cvf package.tar path1 file4
path1/
path1/file1
path1/file2
file4
[root@localhost user]# ls
Desktop  Downloads  Music  path1  Pictures  Templates
Documents file4      package.tar path2  Public  Videos
[root@localhost user]#
```

(5) 查看包 package.tar 的内容;

```
root@localhost:/home/user
File Edit View Search Terminal Help
[root@localhost ~]# cd /home/user
[root@localhost user]# tar -tvf package.tar
drwxr-xr-x root/root      0 2023-03-20 14:24 path1/
-rw-r--r-- root/root      0 2023-03-20 14:24 path1/file1
-rw-r--r-- root/root      0 2023-03-20 14:24 path1/file2
-rw-r--r-- root/root      0 2023-03-20 14:26 file4
[root@localhost user]#
```

(6) 向包 package.tar 里添加文件夹 path2 的内容;

```
root@localhost:/home/user
File Edit View Search Terminal Help
[root@localhost ~]# cd /home/user
[root@localhost user]# tar -rvf package.tar path2
path2/
path2/file3
```

(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。

```
root@localhost:/home/user/path3
File Edit View Search Terminal Help
[root@localhost ~]# cd /home/user
[root@localhost user]# mkdir path3
[root@localhost user]# cp package.tar path3
[root@localhost user]# ls
Desktop  Downloads  Music      path1  path3  Public  Videos
Documents file4      package.tar path2  Pictures Templates
[root@localhost user]# ls path3
package.tar
[root@localhost user]# cd path3
[root@localhost path3]# tar -xvf package.tar
path1/
path1/file1
path1/file2
file4
path2/
path2/file3
[root@localhost path3]# ls
file4 package.tar path1 path2
[root@localhost path3]#
```

3、符号链接内容

(1) 新建文件 foo.txt , 内容为 123 ;

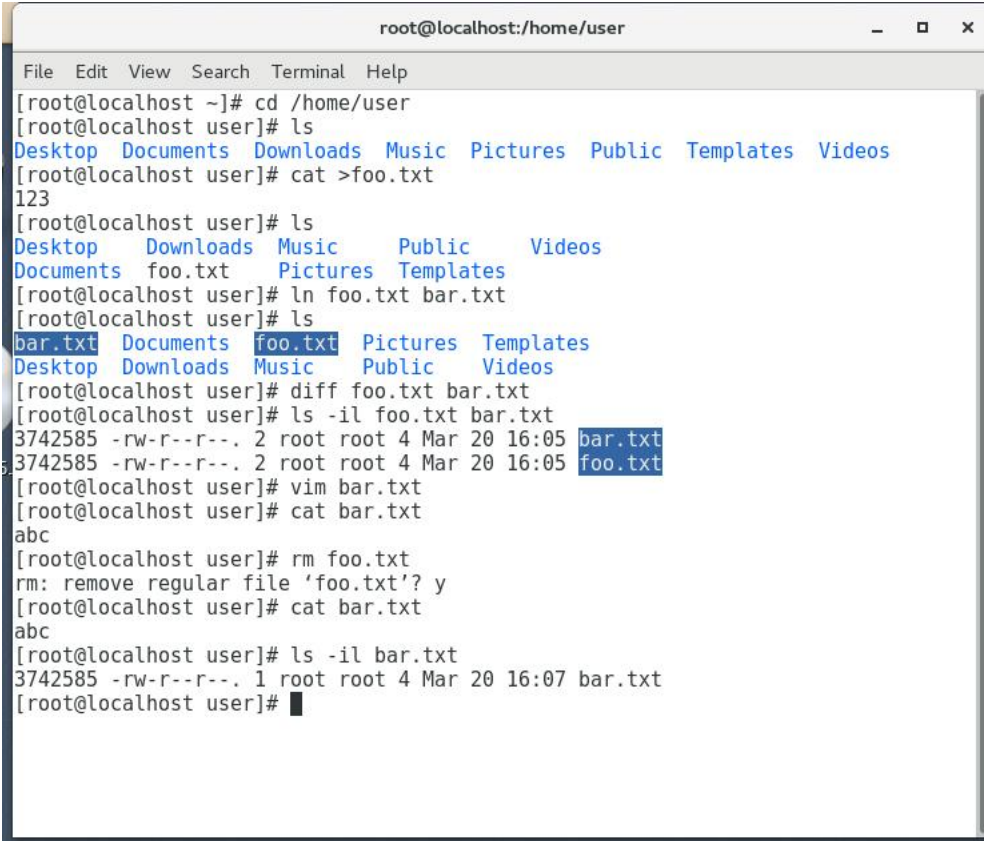
(2) 建立 foo.txt 的硬链接文件 bar.txt , 并比较 bar.txt

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；



```
root@localhost:/home/user
File Edit View Search Terminal Help
[root@localhost ~]# cd /home/user
[root@localhost user]# ls
Desktop Documents Downloads Music Pictures Public Templates Videos
[root@localhost user]# cat >foo.txt
123
[root@localhost user]# ls
Desktop Downloads Music Public Videos
Documents foo.txt Pictures Templates
[root@localhost user]# ln foo.txt bar.txt
[root@localhost user]# ls
bar.txt Documents foo.txt Pictures Templates
Desktop Downloads Music Public Videos
[root@localhost user]# diff foo.txt bar.txt
[root@localhost user]# ls -il foo.txt bar.txt
3742585 -rw-r--r--. 2 root root 4 Mar 20 16:05 bar.txt
3742585 -rw-r--r--. 2 root root 4 Mar 20 16:05 foo.txt
[root@localhost user]# vim bar.txt
[root@localhost user]# cat bar.txt
abc
[root@localhost user]# rm foo.txt
rm: remove regular file 'foo.txt'? y
[root@localhost user]# cat bar.txt
abc
[root@localhost user]# ls -il bar.txt
3742585 -rw-r--r--. 1 root root 4 Mar 20 16:07 bar.txt
[root@localhost user]#
```

(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。

```
root@localhost:/home/user
File Edit View Search Terminal Help
[root@localhost ~]# cd /home/user
[root@localhost user]# ls
bar.txt  Desktop  Downloads  Music  Pictures  Public  Templates  Videos
[root@localhost user]# ln -s bar.txt baz.txt
[root@localhost user]# ls -il bar.txt baz.txt
3742585 -rw-r--r--. 1 root root 4 Mar 20 16:07 bar.txt
2396720 lrwxrwxrwx. 1 root root 7 Mar 20 16:11 baz.txt -> bar.txt
[root@localhost user]# diff bar.txt baz.txt
[root@localhost user]# rm -f bar.txt
[root@localhost user]# cat baz.txt
cat: baz.txt: No such file or directory
[root@localhost user]#
```

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)。

```
root@localhost:/home/user
File Edit View Search Terminal Help
[root@localhost ~]# cd /home/user
[root@localhost user]# ls
Desktop  Documents  Downloads  Music  Pictures  Public  Templates  Videos
[root@localhost user]# touch qux.txt
[root@localhost user]# ls
Desktop  Downloads  Pictures  qux.txt  Videos
Documents  Music  Public  Templates
[root@localhost user]# ls -l qux.txt
-rw-r--r--. 1 root root 0 Mar 20 16:36 qux.txt
[root@localhost user]# chmod 755 qux.txt
[root@localhost user]# ls -l qux.txt
-rwxr-xr-x. 1 root root 0 Mar 20 16:36 qux.txt
[root@localhost user]#
```

四、实验过程分析与讨论

在这次的实验中，学到了文件的查找、内容行数等的统计、打包等，也学习了链接与硬链接的原理与操作，并且还学会了管理文件的权限。期间还接触到了 vim 的使用，root，与用户群组等方面的知识。但仅从此次实验来看，知识掌握的并不牢固，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 21 日
学 号	2018214390	姓 名	高钰皓
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。

a) 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：

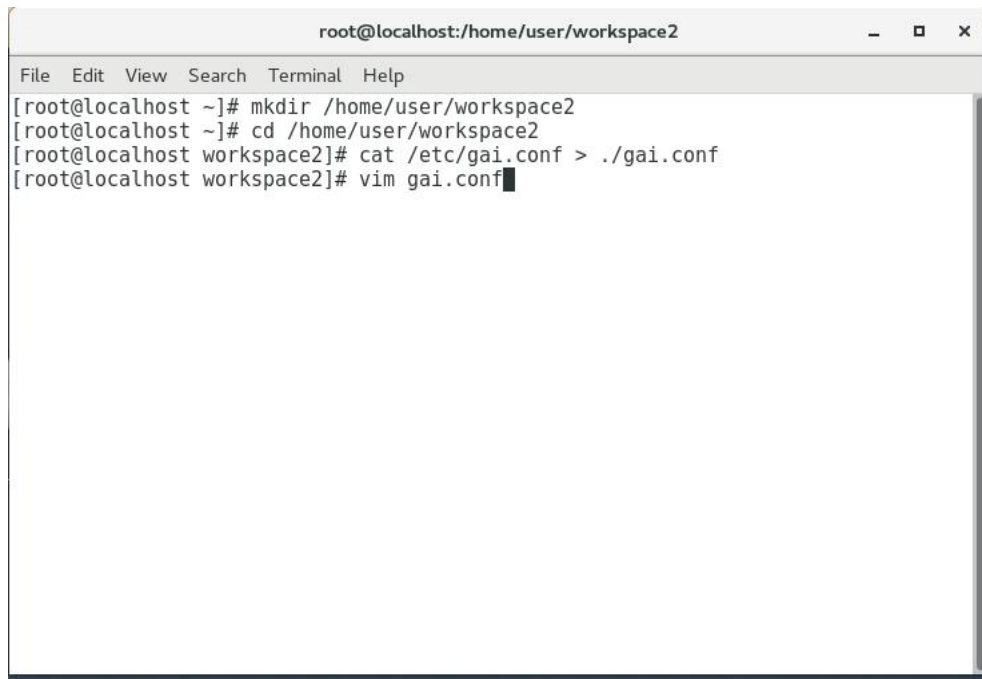
```
#include <stdio.h>

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。

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` ;



```
root@localhost:/home/user/workspace2
File Edit View Search Terminal Help
[root@localhost ~]# mkdir /home/user/workspace2
[root@localhost ~]# cd /home/user/workspace2
[root@localhost workspace2]# cat /etc/gai.conf > ./gai.conf
[root@localhost workspace2]# vim gai.conf
```

(5) 将光标移到第 18 行;

输入: `18G`

(6) 复制该行内容;

输入: `yy`

(7) 将光标移到最后一行行首;

输入: `G`

(8) 粘贴复制行的内容;

输入: `p`

(9) 撤销第 8 步的动作;

输入: `u`

(10) 存盘但不退出;

输入: :w

(11) 将光标移到首行;

输入: gg

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

输入: i, 进入编辑模式, 然后输入 "Hello, this is vim world!", 然后按 "Esc" 退出编辑模式。

(13) 删除字符串 "this";

将光标移植 "this" 前, 然后输入 "dw"

(14) 强制退出 vim, 不存盘。

输入: : q!

四、实验过程分析与讨论

本次实验，主要尝试了编译 C 文件与进行 vim 编译器的操作。

编译的过程使用了 `gcc -o` 这一指令，但其中实际上是包括了四个步骤的，第一，预处理，惊醒头文件展开，宏替换，去掉注释；第二，就是将 C 文件编译为汇编文件；第三，将汇编文件编译为机器可识别的二进制文件；第四，将函数中的相应代码组合到目标文件中。

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 4 月 21 日
学 号	2018214390	姓 名	高钰皓
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

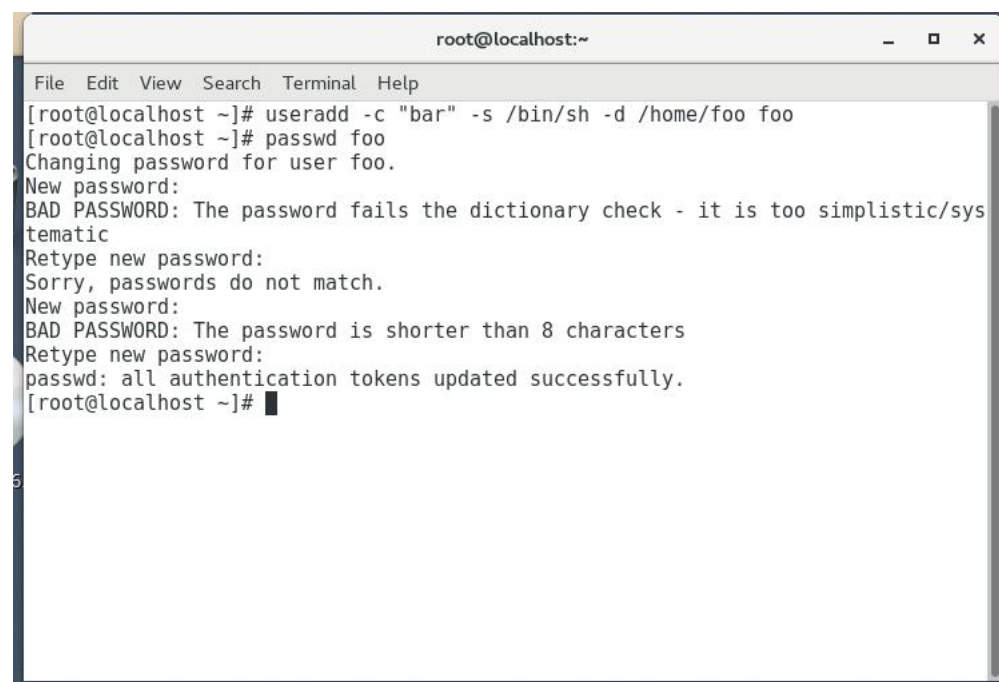
1. 掌握用户管理命令，包括命令 `useradd` 、 `usermod` 、 `userdel` 、 `newusers` ；
2. 掌握用户组管理命令，包括命令 `groupadd` 、 `groupdel` 、 `gpasswd` ；
3. 掌握用户和用户组维护命令，包括命令 `passwd` 、 `su` 、 `sudo` 。

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；



```
root@localhost:~  
File Edit View Search Terminal Help  
[root@localhost ~]# useradd -c "bar" -s /bin/sh -d /home/foo foo  
[root@localhost ~]# passwd foo  
Changing password for user foo.  
New password:  
BAD PASSWORD: The password fails the dictionary check - it is too simplistic/systematic  
Retype new password:  
Sorry, passwords do not match.  
New password:  
BAD PASSWORD: The password is shorter than 8 characters  
Retype new password:  
passwd: all authentication tokens updated successfully.  
[root@localhost ~]#
```

(注：第一次打密码时，因为密码是不显示的，导致我以为电脑卡了，所以第一次设置密码，两次输入密码不匹配。)

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@localhost:~  
File Edit View Search Terminal Help  
[root@localhost ~]# su foo  
sh-4.2$ usermod -u 2000 -s /bin/csh foo  
usermod: user foo is currently used by process 2453  
sh-4.2$
```

显示进程占用，所以改到root权限，修改uid

```
foo@localhost:~  
File Edit View Search Terminal Help  
[root@localhost ~]# usermod -u 2000 -s /bin/csh foo  
[root@localhost ~]# su - foo  
Last login: Tue May 23 16:47:16 CST 2023 on pts/0  
[foo@localhost ~]$ id  
uid=2000(foo) gid=1001(foo) groups=1001(foo) context=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023  
[foo@localhost ~]$
```

显示修改成功。

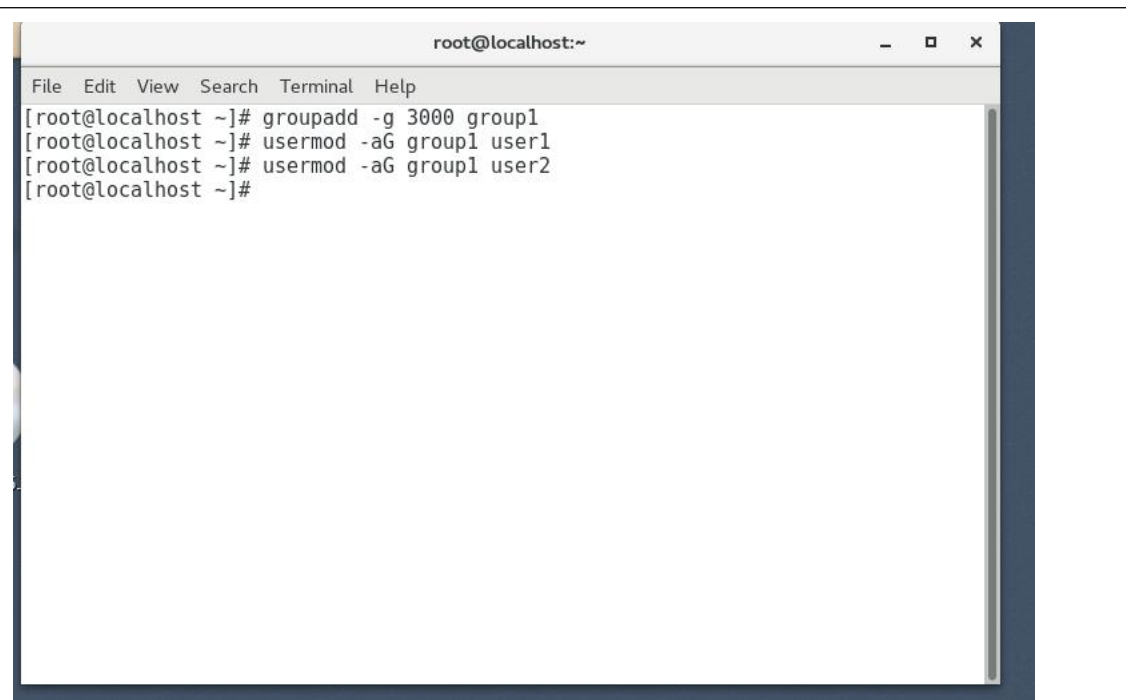
3. 从用户 foo 切换到 root ；

```
[foo@localhost ~]$ su -  
Password:  
Last login: Tue May 23 16:45:45 CST 2023 on pts/0
```

4. 删除 foo 用户，并在删除该用户的同时一并删除其家目录；

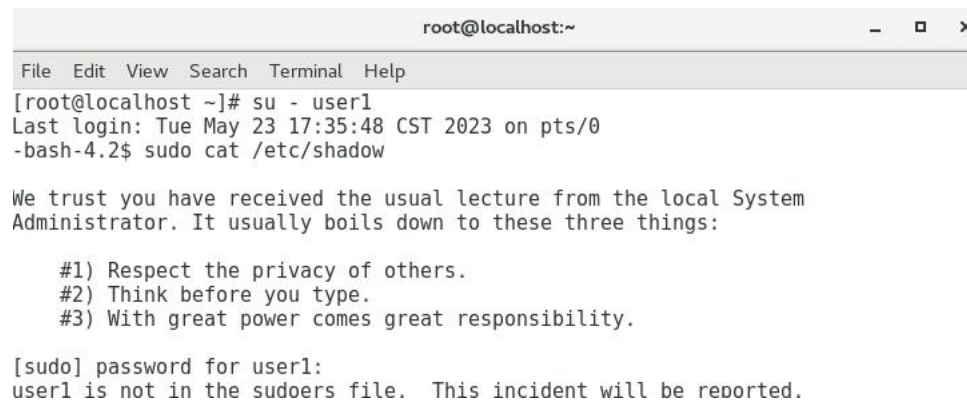
```
[root@localhost ~]# userdel -r foo  
[root@localhost ~]#
```

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；

A terminal window titled 'root@localhost:~' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the following commands and output:

```
[root@localhost ~]# groupadd -g 3000 group1
[root@localhost ~]# usermod -aG group1 user1
[root@localhost ~]# usermod -aG group1 user2
[root@localhost ~]#
```

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。

A terminal window titled 'root@localhost:~' with a menu bar (File, Edit, View, Search, Terminal, Help). The terminal shows the following commands and output:

```
[root@localhost ~]# su - user1
Last login: Tue May 23 17:35:48 CST 2023 on pts/0
-bash-4.2$ sudo cat /etc/shadow

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.

[sudo] password for user1:
user1 is not in the sudoers file. This incident will be reported.
```

不可执行, 所以修改权限如下:

```
root@localhost:~  
File Edit View Search Terminal Help  
## Next comes the main part: which users can run what software on  
## which machines (the sudoers file can be shared between multiple  
## systems).  
## Syntax:  
##  
##     user    MACHINE=COMMANDS  
##  
## The COMMANDS section may have other options added to it.  
##  
## Allow root to run any commands anywhere  
root    ALL=(ALL)    ALL  
user1 ALL=(ALL:ALL) ALL  
## Allows members of the 'sys' group to run networking, software,  
## service management apps and more.  
# %sys ALL = NETWORKING, SOFTWARE, SERVICES, STORAGE, DELEGATING, PROCESSES, LOC  
ATE, DRIVERS  
  
## Allows people in group wheel to run all commands  
%wheel  ALL=(ALL)    ALL  
  
## Same thing without a password  
# %wheel    ALL=(ALL)    NOPASSWD: ALL  
  
:wq!
```

然后执行成功，如下。

```
root@localhost:~  
File Edit View Search Terminal Help  
[root@localhost ~]# visudo  
[root@localhost ~]# su - user1  
Last login: Tue May 23 17:50:04 CST 2023 on pts/0  
-bash-4.2$ sudo cat /etc/shadow  
[sudo] password for user1:  
root:$6$q4gWZmvMB2o1tzgD$xGcn.et0WPYSYo84tWi/3UW4TCdDFfeR5.bJjcD/1kFzfMfocallI6M  
1jEn2UjlIb3Fsc6r2g2XJp/4HXH2kI.:0:99999:7:::  
bin:!:17110:0:99999:7:::  
daemon:!:17110:0:99999:7:::  
adm:!:17110:0:99999:7:::  
lp:!:17110:0:99999:7:::  
sync:!:17110:0:99999:7:::  
shutdown:!:17110:0:99999:7:::  
halt:!:17110:0:99999:7:::  
mail:!:17110:0:99999:7:::  
operator:!:17110:0:99999:7:::  
games:!:17110:0:99999:7:::  
ftp:!:17110:0:99999:7:::  
nobody:!:17110:0:99999:7:::  
systemd-network:!!:19424::::::::  
dbus:!!:19424::::::::  
polkitd:!!:19424::::::::  
abrt:!!:19424::::::::  
libstoragemgmt:!!:19424::::::::
```


四、实验过程分析与讨论

开始时，在修改权限是不太会，在 CSDN 查了相关知识后，学会了。但是，在从根权限转到某用户权限后修改其 uid，发现出现进程占用无法修改。从根权限修改某用户 uid，十分顺利。关于进程的冲突，这里不太明白。

关于用户与组的东西，做的时候要小心，尤其在修改一些文件的时候，或者权限的时候。

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	2018214390	姓 名	高钰皓
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

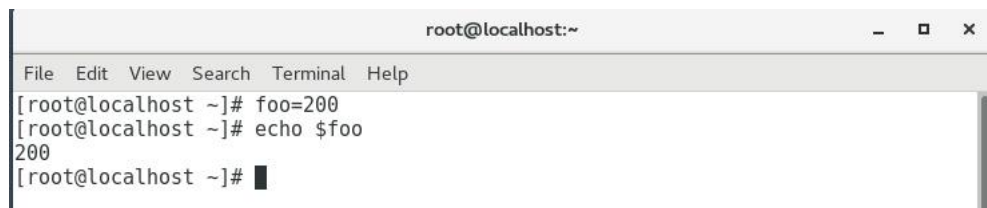
1. 掌握Shell程序的创建过程及Shell程序的执行方法；
2. 掌握Shell变量的定义方法，及用户定义变量、参数位置等；
3. 掌握变量表达式，包括字符串比较、数字比较、逻辑测试、文件测试；
4. 掌握条件判断语句，如 if 语句、case 语句。

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。

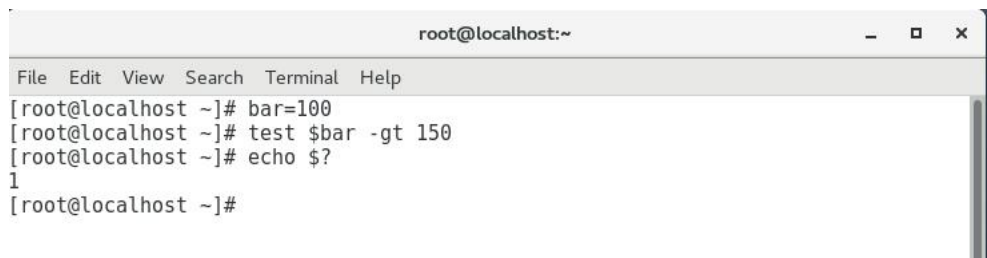
三、 实验内容及结果

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



```
root@localhost:~  
File Edit View Search Terminal Help  
[root@localhost ~]# foo=200  
[root@localhost ~]# echo $foo  
200  
[root@localhost ~]#
```

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)；



```
root@localhost:~  
File Edit View Search Terminal Help  
[root@localhost ~]# bar=100  
[root@localhost ~]# test $bar -gt 150  
[root@localhost ~]# echo $?  
1  
[root@localhost ~]#
```

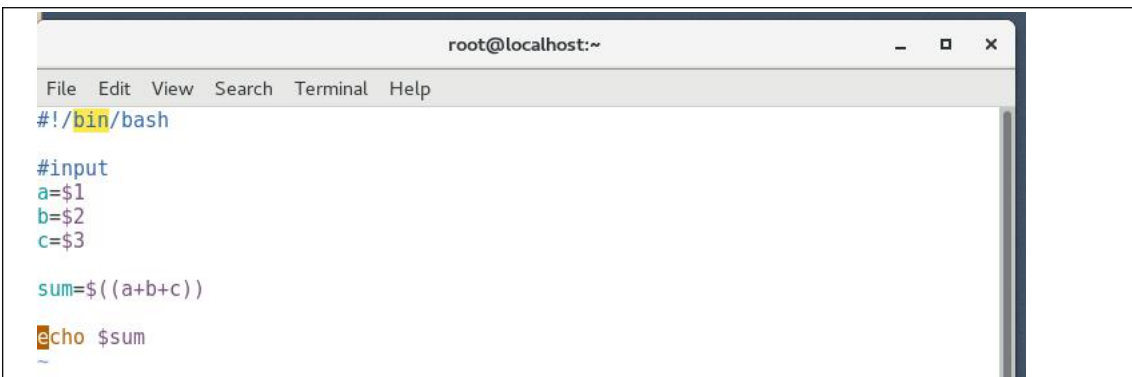
3. 创建一个Shell程序，其功能为显示计算机主机名(`hostname`)和系统时间(`date`)；


```
root@localhost:~  
File Edit View Search Terminal Help  
[root@localhost ~]# vim test1.sh  
[root@localhost ~]# chmod +x test1.sh  
[root@localhost ~]# ./test1.sh  
请输入一个数字作为参数  
[root@localhost ~]# ./test1.sh 153  
153 yes  
[root@localhost ~]# ./test1.sh 124  
124 no  
[root@localhost ~]# ./test1.sh 370  
370 yes  
[root@localhost ~]#
```

```
root@localhost:~  
File Edit View Search Terminal Help  
#!/bin/bash  
  
#input  
num=$1  
  
# 判断输入参数是否为空  
if [ -z "$1" ]  
then  
    echo "请输入一个数字作为参数"  
    exit 1  
fi  
  
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 yes"  
else  
    echo "$num no"  
fi  
  
~  
~
```

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；

```
root@localhost:~  
File Edit View Search Terminal Help  
[root@localhost ~]# vim test2.sh  
[root@localhost ~]# bash test2.sh 1 2 3  
6  
[root@localhost ~]#
```

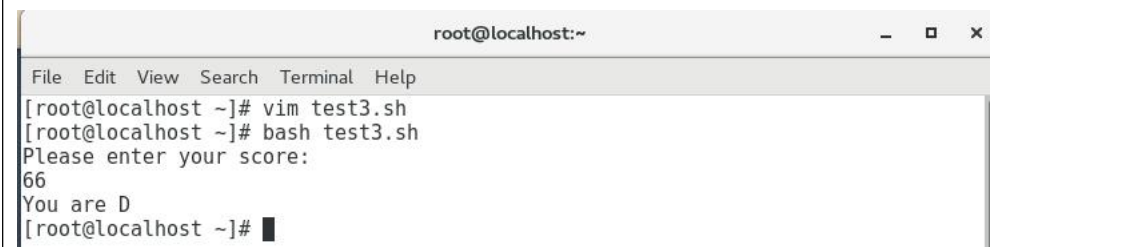


```
root@localhost:~  
File Edit View Search Terminal Help  
#!/bin/bash  
  
#input  
a=$1  
b=$2  
c=$3  
  
sum=$((a+b+c))  
echo $sum
```

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。



```
root@localhost:~  
File Edit View Search Terminal Help  
[root@localhost ~]# vim test3.sh  
[root@localhost ~]# bash test3.sh  
Please enter your score:  
66  
You are D  
[root@localhost ~]#
```

```
root@localhost:~  
File Edit View Search Terminal Help  
#!/bin/bash  
  
echo "Please enter your score:"  
read score  
  
if [ $score -ge 90 ];  
then  
    echo "You are A"  
elif [ $score -ge 80 ];  
then  
    echo "You are B"  
elif [ $score -ge 70 ];  
then  
    echo "You are C"  
elif [ $score -ge 60 ];  
then  
    echo "You are D"  
else  
    echo "You are E"  
fi
```

四、 实验过程分析与讨论

实验过程比较顺利，不过是在工具书的帮助下做的。相关的知识还不熟练，需要多练习掌握。

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	2018214390	姓 名	高钰皓
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

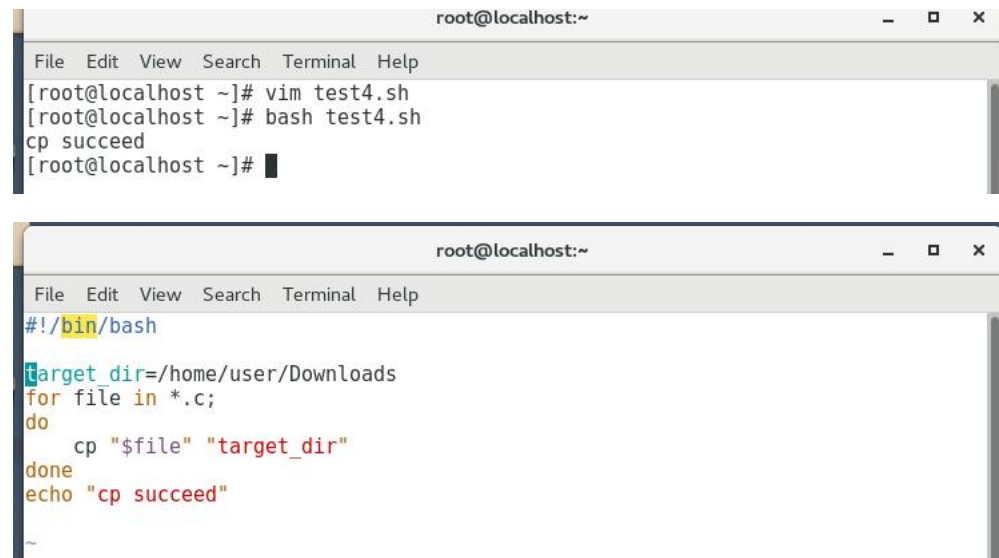
1. 熟练掌握Shell循环语句：for 、 while 、 until ；
2. 熟练掌握Shell循环控制语句：break 、 continue 。

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

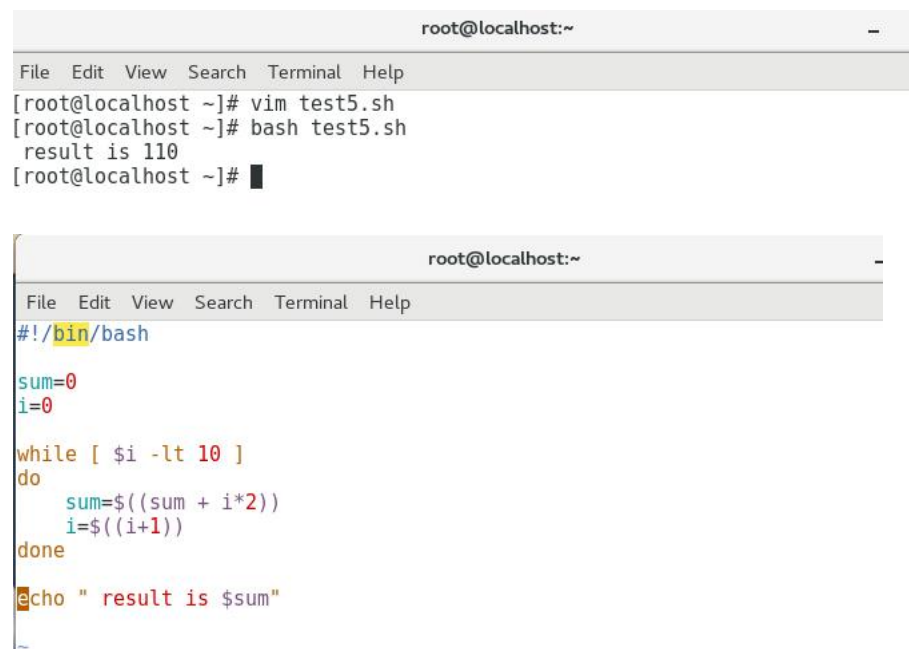
1. 编写一个Shell脚本，利用 for 循环把当前目录下的所有 *.c 文件复制到指定的目录中(如 ~/workspace)；可以事先在当前目录下建立若干 *.c 文件用于测试。



The first terminal screenshot shows the execution of a script named test4.sh. The user runs 'vim test4.sh' and then 'bash test4.sh'. The output is 'cp succeed'. The second terminal screenshot shows the content of the script test4.sh, which uses a for loop to copy all *.c files from the current directory to the directory ~/Downloads. The script content is as follows:

```
#!/bin/bash
target_dir=/home/user/Downloads
for file in *.c;
do
    cp "$file" "$target_dir"
done
echo "cp succeed"
```

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

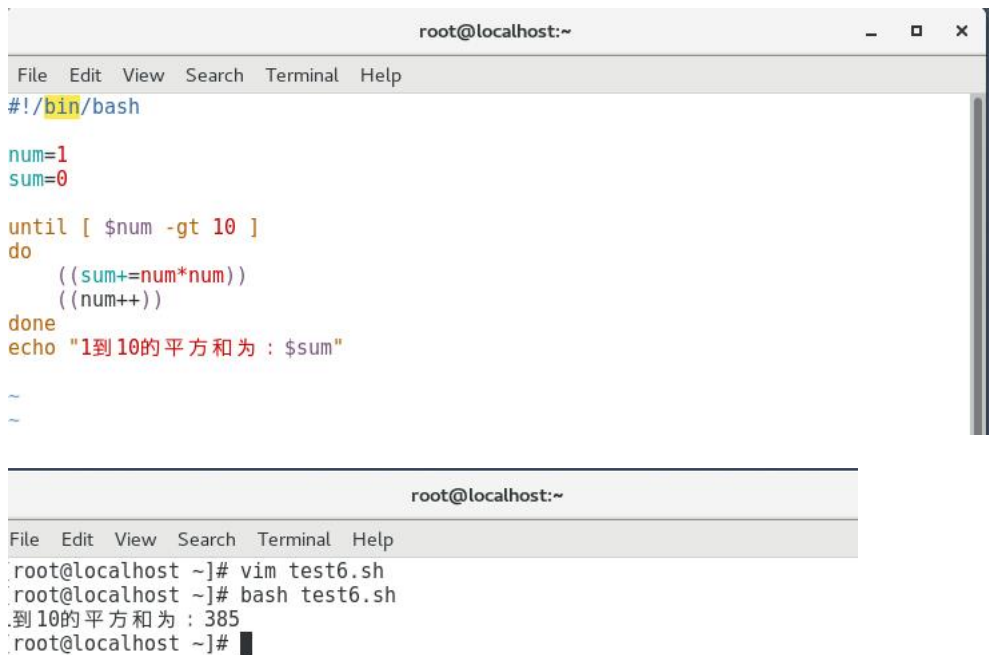


The first terminal screenshot shows the execution of a script named test5.sh. The user runs 'vim test5.sh' and then 'bash test5.sh'. The output is 'result is 110'. The second terminal screenshot shows the content of the script test5.sh, which uses a while loop to calculate the sum of the first 10 even numbers. The script content is as follows:

```
#!/bin/bash
sum=0
i=0
while [ $i -lt 10 ]
do
    sum=$((sum + i*2))
    i=$((i+1))
done
echo "result is $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;



```
root@localhost:~  
File Edit View Search Terminal Help  
#!/bin/bash  
  
num=1  
sum=0  
  
until [ $num -gt 10 ]  
do  
    ((sum+=num*num))  
    ((num++))  
done  
echo "1到10的平方和为 : $sum"  
  
~  
~  
  
root@localhost:~  
File Edit View Search Terminal Help  
root@localhost ~]# vim test6.sh  
root@localhost ~]# bash test6.sh  
1到10的平方和为 : 385  
root@localhost ~]#
```

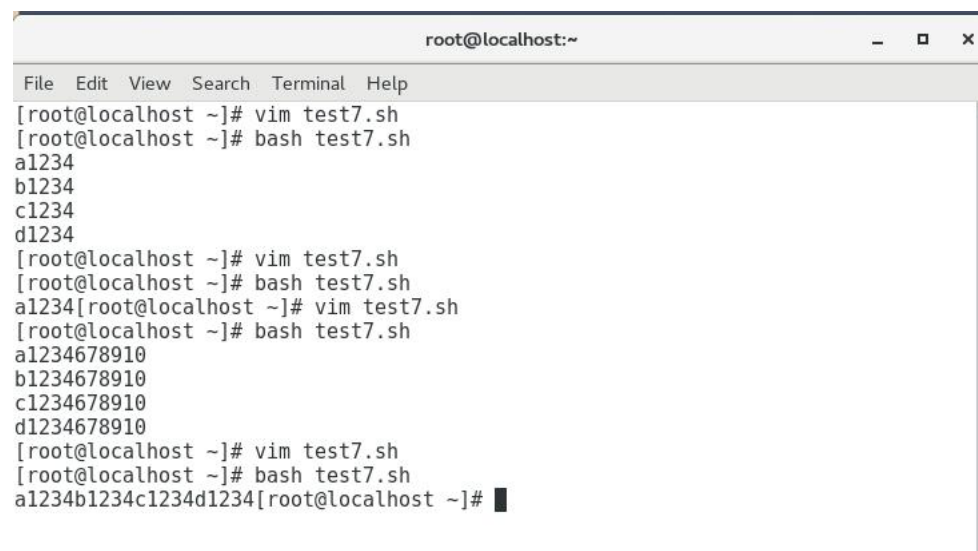
4. 运行下列程序，并观察程序的运行结果。将程序中的---分别替换为 break、break2、continue、continue2，并观察四种情况下的 实验结果

```
#!/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

A terminal window titled 'root@localhost:~' with standard window controls. It shows the execution of a script named 'test7.sh'. The script contains three identical blocks of code, each consisting of a 'vim' command followed by a 'bash' command. The output of the script shows the concatenation of the outputs of these three blocks. The first block outputs 'a1234', 'b1234', 'c1234', and 'd1234'. The second block outputs 'a1234678910', 'b1234678910', 'c1234678910', and 'd1234678910'. The third block outputs 'a1234b1234c1234d1234'.

```
root@localhost:~  
File Edit View Search Terminal Help  
[root@localhost ~]# vim test7.sh  
[root@localhost ~]# bash test7.sh  
a1234  
b1234  
c1234  
d1234  
[root@localhost ~]# vim test7.sh  
[root@localhost ~]# bash test7.sh  
a1234[root@localhost ~]# vim test7.sh  
[root@localhost ~]# bash test7.sh  
a1234678910  
b1234678910  
c1234678910  
d1234678910  
[root@localhost ~]# vim test7.sh  
[root@localhost ~]# bash test7.sh  
a1234b1234c1234d1234[root@localhost ~]#
```

四、实验过程分析与讨论

这次实验后, 明显比上一次编写 shell 更熟悉了, 逐渐习惯了 shell 编程。

五、指导教师意见

指导教师签字：卢洋

实验报告

实验名称	实验七 Shell 函数		
实验教室	丹青 922	实验日期	2023 年 5 月 5 日
学 号	2018214390	姓 名	高钰皓
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；



The first screenshot shows a terminal window with the following content:

```
root@localhost:~  
File Edit View Search Terminal Help  
#!/bin/bash  
function sum(){  
    result=$(( $1+$2 ))  
    echo "Sum of $1 and $2 is $result"  
}  
sum 3 5  
~  
~
```

The second screenshot shows the same terminal window after running the script:

```
root@localhost:~  
File Edit View Search Terminal Help  
[root@localhost ~]# vim test8.sh  
[root@localhost ~]# bash test8.sh  
Sum of 3 and 5 is 8  
[root@localhost ~]#
```

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 image displays two terminal windows from a Linux system, showing the creation and execution of a shell script named test9.sh.

The top terminal window shows the script's content:

```
root@localhost:~  
File Edit View Search Terminal Help  
#!/bin/bash  
  
function factorial {  
    if [ $1 -eq 1 ]; then  
        echo 1  
    else  
        local temp=$(factorial $(( $1 - 1 )) )  
        echo $(( $1 * $temp ))  
    fi  
}  
# 输入要求阶乘的数字  
echo "请输入一个正整数 : "  
read num  
  
# 调用递归函数  
result=$(factorial $num)  
  
# 输出结果  
echo "$num 的阶乘为 : $result"  
~  
~  
~  
~  
:wq
```

The bottom terminal window shows the script being executed:

```
root@localhost:~  
File Edit View Search Terminal Help  
[root@localhost ~]# vim test9.sh  
[root@localhost ~]# bash test9.sh  
请输入一个正整数 :  
5  
5 的阶乘为 : 120  
[root@localhost ~]#
```

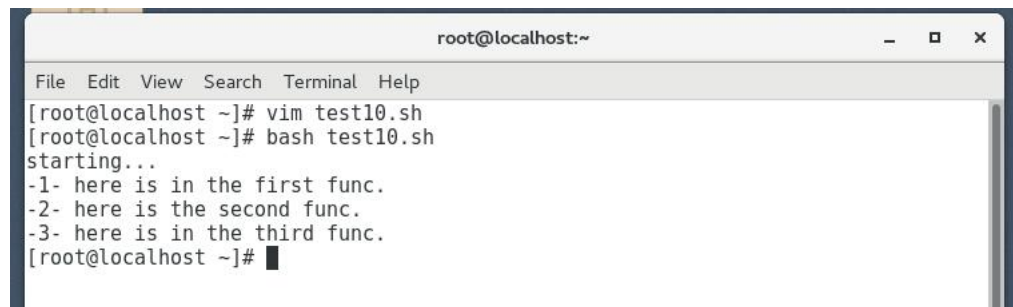
3. 一个Shell脚本的内容如下所示:

```
#!/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。



```
root@localhost:~  
File Edit View Search Terminal Help  
[root@localhost ~]# vim test10.sh  
[root@localhost ~]# bash test10.sh  
starting...  
-1- here is in the first func.  
-2- here is the second func.  
-3- here is in the third func.  
[root@localhost ~]#
```

四、实验过程分析与讨论

本次实验时间了 shenll 编程里自定义函数的编写，过程比较顺利，同其他语言编程的思想都是大差不差的，只在参数传递与返回值等形式上不太一样。

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 7 日
学 号	2018214390	姓 名	高钰皓
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。

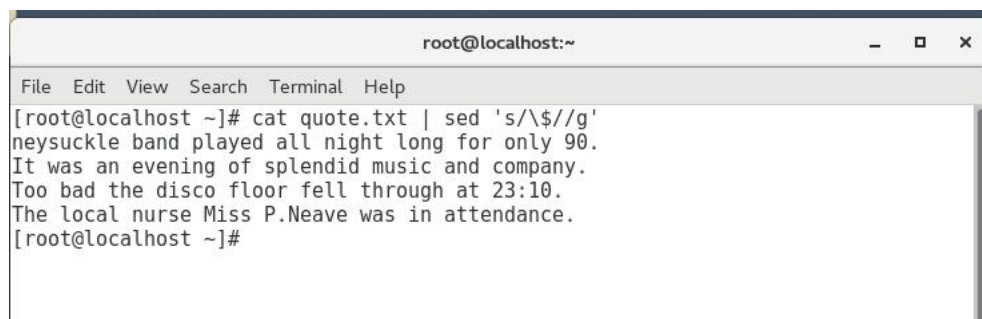
三、实验内容及结果

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.
```

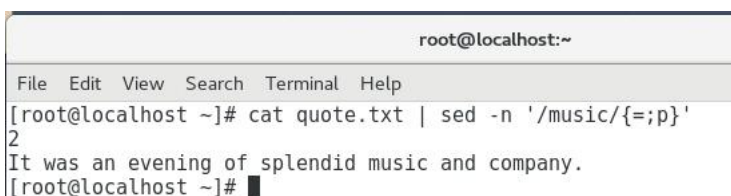
试使用 sed 命令实现如下功能:

(1) 删除 \$ 符号;



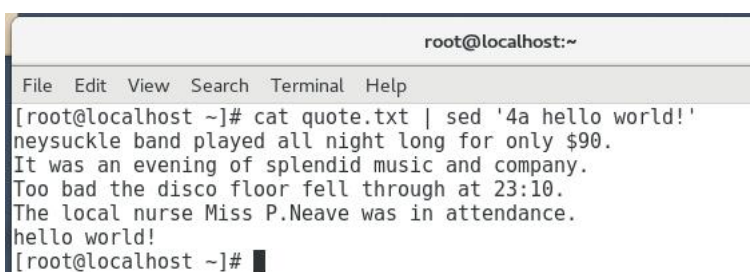
```
root@localhost:~  
File Edit View Search Terminal Help  
[root@localhost ~]# cat quote.txt | sed 's/\$//g'  
neysuckle band played all night long for only 90.  
It was an evening of splendid music and company.  
Too bad the disco floor fell through at 23:10.  
The local nurse Miss P.Neave was in attendance.  
[root@localhost ~]#
```

(2) 显示包含music 文字的行内容及行号;



```
root@localhost:~  
File Edit View Search Terminal Help  
[root@localhost ~]# cat quote.txt | sed -n '/music/{=;p}'  
2  
It was an evening of splendid music and company.  
[root@localhost ~]#
```

(3) 在第 4 行后面追加内容: "hello world!";



```
root@localhost:~  
File Edit View Search Terminal Help  
[root@localhost ~]# cat quote.txt | sed '4a hello world!'  
neysuckle 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!  
[root@localhost ~]#
```

(4) 将文本 "The" 替换为 "Quod" ;

```

root@localhost:~
File Edit View Search Terminal Help
[root@localhost ~]# cat quote.txt | sed 's/The/Quod/g'
neysuckle 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.
[root@localhost ~]#

```

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

```

root@localhost:~
File Edit View Search Terminal Help
[root@localhost ~]# cat quote.txt | sed '3c This is third line.'
neysuckle band played all night long for only $90.
This is third line.
The local nurse Miss P.Neave was in attendance.
[root@localhost ~]#

```

(6) 删除第 2 行内容;

```

root@localhost:~
File Edit View Search Terminal Help
[root@localhost ~]# cat quote.txt | sed '2d'
neysuckle 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.
[root@localhost ~]#

```

(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。

```

root@localhost:~
File Edit View Search Terminal Help
[root@localhost ~]# var="evening"
[root@localhost ~]# cat quote.txt | sed -n "/$var/p"
It was an evening of splendid music and company.
[root@localhost ~]#

```

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;

```
root@localhost:~  
File Edit View Search Terminal Help  
[root@localhost ~]# awk '{print $2}' numbers.txt  
:  
:  
:  
[root@localhost ~]# awk -F: '{print $2}' numbers.txt  
two  
five  
[root@localhost ~]#
```

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
```

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。

这个脚本实现的功能是从一个名为info.txt的文件中搜索用户输入的模式（pattern），并输出匹配到的行数及匹配到的内容。具体内容如下：

首先，通过read命令获取用户输入的模式，赋值给变量模式。然后，使用awk命令在info.txt文件中搜索包含pattern的行，并打印出来。

在搜索过程中，如果匹配成功，则将变量nmatches加1。最后，输出nmatches的值，表示匹配到的行数。

需要注意的是，使用awk命令时，需要将单引号和双引号同时使用。因为awk命令中包含单引号，如果只使用单引号，则会导致命令执行错误。

四、实验过程分析与讨论

本次实验，个人感觉，sed 还好，awk 使用起来要难一些。需要查阅资料才能做。

五、指导教师意见

指导教师签字：卢洋