

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 5 月 1 日
学 号	2021224827	姓 名	陈阳
专业班级	奥林计算机科学与技术 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

```
chenyang@localhost ~]$ cd /etc
chenyang@localhost etc]$ pwd
/etc
chenyang@localhost 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。

```
[chenyang@localhost home]$ cd chenyang
[chenyang@localhost ~]$ ls -a
.          .bash_profile  .dbus          .mozilla  视频  音乐
..         .bashrc       .esd_auth      .redhat   图片  桌面
.bash_history .cache        .ICEauthority  公共     文档
.bash_logout .config       .local         模板     下载
[chenyang@localhost ~]$
```

3、使用命令创建目录/home/lyj/linux， 然后删除该目录。

```
[chenyang@localhost ~]$ mkdir /home/chenyang/linux
[chenyang@localhost ~]$ ls
linux 公共 模板 视频 图片 文档 下载 音乐 桌面
[chenyang@localhost ~]$ rmdir linux
[chenyang@localhost ~]$ ls
公共 模板 视频 图片 文档 下载 音乐 桌面
```

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

```
[chenyang@localhost ~]$ cat > abc
hello linux
[chenyang@localhost ~]$ ls
abc 公共 模板 视频 图片 文档 下载 音乐 桌面
```

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。

```
chenyang@localhost ~]$ mkdir ak
chenyang@localhost ~]$ ls
abc ak 公共 模板 视频 图片 文档 下载 音乐 桌面
chenyang@localhost ~]$ cp -r abc ak
chenyang@localhost ~]$ ls
abc ak 公共 模板 视频 图片 文档 下载 音乐 桌面
chenyang@localhost ~]$ cd ak
chenyang@localhost ak]$ rm -i abc
rm: 是否删除普通文件 "abc"? y
chenyang@localhost ak]$ cd ..
chenyang@localhost ~]$ rmdir ak
```

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。

```
[chenyang@localhost ~]$ head -3 /etc/adduser.conf
chenyang@localhost ~]$ tail -5 /etc/adduser.conf
```

7、分屏查看文件/etc/adduser.conf 的内容。

```
[chenyang@localhost ~]$ more /etc/adduser.conf
```

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

```
[chenyang@localhost ~]$ cat > facebook.txt
google 110 5000
baidu 100 5000
guge 50 3000
sohu 100 4500
```

9. 第一列为公司名称，第2列为公司人数，第3列为员工平均工资。

利用sort命令完成下列排序：

(1) 按公司字母顺序排序

(2) 按公司人数排序

(3) 按公司人数排序，人数相同的按照员工平均工资升序排序

(4) 按员工工资降序排序，如工资相同，则按公司人数升序排序

(5) 从公司英文名称的第2个字母开始进行排序。

```
[chenyang@localhost ~]$ cat > facebook.txt
google 110 5000
baidu 100 5000
guge 50 3000
sohu 100 4500
[chenyang@localhost ~]$ sort -r facebook.txt
sohu 100 4500
guge 50 3000
google 110 5000
baidu 100 5000
[chenyang@localhost ~]$ sort -n facebook.txt
baidu 100 5000
google 110 5000
guge 50 3000
sohu 100 4500
[chenyang@localhost ~]$ sort -n -t ' ' -k 2 -k 3 facebook.txt
guge 50 3000
sohu 100 4500
baidu 100 5000
google 110 5000
[chenyang@localhost ~]$ sort -n -t ' ' -k 3r -k 2 facebook.txt
baidu 100 5000
google 110 5000
sohu 100 4500
guge 50 3000
[chenyang@localhost ~]$ sort -t ' ' -k 1,2 facebook.txt
baidu 100 5000
sohu 100 4500
google 110 5000
guge 50 3000
```

四、 实验过程分析与讨论

在寻找/etc/adduser.conf 文件遇到了困难在最后那个实验，对于一些命令还是不太熟悉，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 5 月 3 日
学 号	2021224827	姓 名	陈阳
专业班级	奥林计算机科学与技术 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；

```
[chenyang@localhost ~]$ mkdir baz
[chenyang@localhost ~]$ cd baz
[chenyang@localhost baz]$ cat > qux
cneruy
dsjiojdw
dcd
```

(2) 在用户目录下查找文件 qux ，并显示该文件位置信息；

```
[chenyang@localhost ~]$ find /home/chenyang -name qux
/home/chenyang/baz/qux
```

(3) 统计文件 qux 中所包含内容的行数、字数和字节数；

```
[chenyang@localhost ~]$ wc -lwc ./baz/qux
3 3 21 ./baz/qux
```

(4) 在用户目录下查找文件 qux ，并删除该文件；

```
[chenyang@localhost ~]$ find /home/chenyang -name qux -exec rm {} \;
```

(5) 查看文件夹 baz 内容，看一下是否删除了文件 qux 。

```
[chenyang@localhost ~]$ cd ./baz/
[chenyang@localhost baz]$ ls
[chenyang@localhost 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 ；

```
[chenyang@localhost ~]$ mkdir path1
[chenyang@localhost ~]$ cd path1
[chenyang@localhost path1]$ touch file1 file2
```

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

```
[chenyang@localhost ~]$ mkdir path2
[chenyang@localhost ~]$ cd path2
[chenyang@localhost path2]$ touch file3
```

(3) 在用户目录下新建文件 file4 ；

```
[chenyang@localhost ~]$ touch file4
```

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

```
[chenyang@localhost ~]$ tar -cvf package.tar path1 file4
path1/
path1/file1
path1/file2
file4
```

(5) 查看包 package.tar 的内容；

```
[chenyang@localhost ~]$ tar -tf package.tar
path1/
path1/file1
path1/file2
file4
```

(6) 向包 package.tar 里添加文件夹 path2 的内容；

```
[chenyang@localhost ~]$ tar -rf package.tar path2
```

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

```
[chenyang@localhost ~]$ mkdir path3
[chenyang@localhost ~]$ cp package.tar ./path3
```

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

```
[chenyang@localhost ~]$ 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；

```
[chenyang@localhost ~]$ vim foo.txt
[chenyang@localhost ~]$ ls
abc  facebook.txt  foo.txt  path1  path3  模板  图片  下载  桌面
baz  file4          package.tar  path2  公共  视频  文档  音乐
[chenyang@localhost ~]$ 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；

```
[chenyang@localhost ~]$ ln foo.txt bar.txt
[chenyang@localhost ~]$ diff foo.txt bar.txt
```

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

```
[chenyang@localhost ~]$ ls -li
70151 abc 70155 file4 70156 path2 134347217 视频 151 音乐
70161 bar.txt 70161 foo.txt 134348090 path3 67109012 图片 201326737 桌面
134348088 baz 70159 package.tar 134347216 公共 201326738 文档
70154 facebook.txt 67184530 path1 67109011 模板 150 下载
```

(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；

```
[chenyang@localhost ~]$ vim bar.txt
[chenyang@localhost ~]$ diff bar.txt foo.txt
```

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

```
[chenyang@localhost ~]$ rm foo.txt
[chenyang@localhost ~]$ ls -li bar.txt
70161 bar.txt
[chenyang@localhost ~]$ cat bar.txt
abc
```

(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；

```
[chenyang@localhost ~]$ ln -s bar.txt baz.txt
[chenyang@localhost ~]$ ls -li
70151 abc          70154 facebook.txt    70156 path2  134347217 视频      151 音乐
70161 bar.txt      70155 file4          134348090 path3  67109012 图片    201326737 桌面
134348088 baz          70159 package.tar    134347216 公共    201326738 文档
70158 baz.txt      67184530 path1              67109011 模板      150 下载
[chenyang@localhost ~]$ diff bar.txt baz.txt
```

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

```
[chenyang@localhost ~]$ rm bar.txt
[chenyang@localhost ~]$ cat baz.txt
cat: baz.txt: 没有那个文件或目录
```

4. 权限管理

(1) 新建文件 qux.txt；

```
[chenyang@localhost ~]$ touch qux.txt
```

(2) 为文件 qux.txt 增加执行权限（所有用户都可以执行）。

```
[chenyang@localhost ~]$ chmod a+x qux.txt
```

四、实验过程分析与讨论

对部分命令不太熟悉，需要查询 CSDN 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 5 月 5 日
学 号	2021224827	姓 名	陈阳
专业班级	奥林计算机科学与技术 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 ；

```
chenyang@localhost ~]$ mkdir workspace1
```

(2) 进入目录 workspace1 ；

```
chenyang@localhost ~]$ cd workspace1
chenyang@localhost 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( )
```

```
{
```

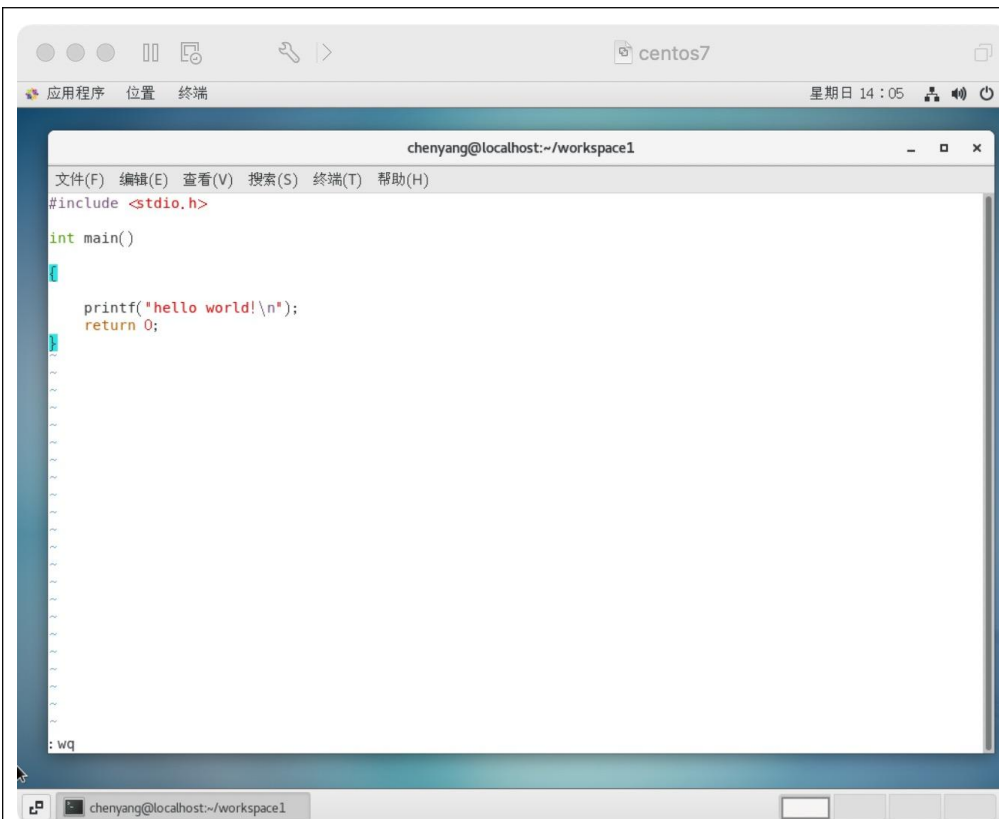
```
printf("hello world!\n");
```

```
return 0;
```

```
}
```

```
chenyang@localhost workspace1]$ vim test.c
```

(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。

```
[chenyang@localhost workspace1]$ gcc test.c -o test
[chenyang@localhost workspace1]$ ./test
hello world!
```

2. vim 编辑器的详细使用：

(1) 在用户目录下创建一个名为 workspace2 的目录；

```
[chenyang@localhost ~]$ mkdir workspace2
```

(2) 进入 workspace2 目录；

```
[chenyang@localhost ~]$ cd workspace2
[chenyang@localhost workspace2]$
```

(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；

```
[chenyang@localhost workspace2]$ cat /etc/gai.conf > ./gai.conf
```

(4) 使用 vim 编辑当前目录下的 gai.conf ；

```
chenyang@localhost workspace2]$ vim gai.conf
```


(5) 将光标移到第 18 行；

18G

(6) 复制该行内容；

yy

(7) 将光标移到最后一行行首；

G

(8) 粘贴复制行的内容；

p

(9) 撤销第 8 步的动作；

u

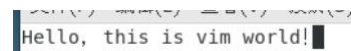
(10) 存盘但不退出；

:w

(11) 将光标移到首行；

gg

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



```
vim - 0.9.1_20190126
Hello, this is vim world!
```

(13) 删除字符串 "this" ；



```
vim - 0.9.1_20190126
Hello, is vim world!
```

(14) 强制退出 vim ， 不存盘。

:q!

四、实验过程分析与讨论

部分 vim 相关命令没有很好地掌握，查询 CSDN 并完成。

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 5 月 8 日
学 号	2021224827	姓 名	陈阳
专业班级	奥林计算机科学与技术 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` 、 `groupmod` 、 `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 chenyang]# useradd -d/home/foo -s/bin/sh -p 123456 foo
```

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 chenyang]# su foo
sh-4.2$ exit
exit
[root@localhost chenyang]# usermod -u 2000 foo
[root@localhost chenyang]# usermod -s /bin/csh foo
```

3. 从用户 `foo` 切换到 `root` ；

```
[root@localhost chenyang]# su root
```

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 chenyang] # userdel -r foo
userdel: 用户 'foo'不存在
```

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@localhost chenyang] # vim user.txt
[root@localhost chenyang] # newusers < user.txt
newusers: 第 1 行: 无效行
newusers: 第 2 行: 无效行
newusers: 第 3 行: 无效行
newusers: 第 4 行: 无效行
newusers: 发现错误，忽略改动
[root@localhost chenyang] # pwunconv
[root@localhost chenyang] # vim password.txt
[root@localhost chenyang] # sudo chpasswd < users.txt
bash: users.txt: 没有那个文件或目录
[root@localhost chenyang] # sudo chpasswd < user.txt
[root@localhost chenyang] # ll /home
总用量 4
drwx-----, 22 chenyang chenyang 4096 5月 28 14:50 chenyang
drwx-----, 3 cy cy 78 11月 18 2022 cy
drwxr-xr-x, 3 root root 19 11月 18 2022 linux
```

6. 创建用户组 group1，并在创建时设置其 GID 为 3000；

```
[root@localhost chenyang] # sudo groupadd -g 3000 group1
[root@localhost chenyang] # cat /etc/group | grep group1
group1:x:3000:
```

7. 在用户组 group1 中添加两个之前批量创建的用户；

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。

四、实验过程分析与讨论

在用户和用户组管理这一部分的知识点掌握不好。

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 5 月 10 日
学 号	2021224827	姓 名	陈阳
专业班级	奥林计算机科学与技术 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）；

```
[root@localhost chenyang] # foo=200
[root@localhost chenyang] # echo $foo
200
```

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 chenyang] # bar=100
[root@localhost chenyang] # test $bar -gt 150
[root@localhost chenyang] # 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）；

```
hostname=$(hostname)
current_date=$(date)
echo "Hostname: $hostname"
echo "System Time: $current_date"
~

[root@localhost chenyang] # vim dwx_script.sh
[root@localhost chenyang] # sh dwx_script.sh
Hostname: localhost.localdomain
System Time: 2023年 05月 28日 星期日 15:07:46 EDT
```

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。

```
number=$1

if [ ${#number} -ne 3 ] ; then
    echo "qingshuru3weishu"
    exit 1
fi

digit1=$((number / 100))
digit2=$((number / 10 % 10))
digit3=$((number % 10))

sum=$((digit1 ** 3 + digit2 ** 3 + digit3 ** 3))

if [ $sum -eq $number ] ; then
    echo "%number TRUE"
else
    echo "$number FALSE"
fi
```

```

false
[root@localhost chenyang] # vim shuixianhua.sh
[root@localhost chenyang] # sh shuixianhua.sh
qingshuru3weishu
[root@localhost chenyang] # sh shuixianhua.sh 153
qingshuru3weishu
[root@localhost chenyang] # sh shuixianhua.sh 124
qingshuru3weishu
[root@localhost chenyang] # sh shuixianhua.sh 370
qingshuru3weishu

```

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；

```

#!/bin/bash

numsum()
{
    read -p "please send me the first num: " num1
    read -p "please send me the second num: " num2
    read -p "please send me the third num: " num3
    echo "the sum is $(( ${num1} + ${num2} + ${num3} ))"
    return $(( ${num1} + ${num2} + ${num3} ))
}

numsum
echo $?

```

```

[root@localhost chenyang] # vim sum.sh
[root@localhost chenyang] # bash sum.sh
please send me the first num: 1
please send me the second num: 1
please send me the third num: 2
the sum is 4
4

```

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。

```

#!/bin/bash

if [[ "$1" -gt "90" ]]
then
    echo A
elif [[ "$1" -gt "80" && "$1" -le "90" ]]
then
    echo B
elif [[ "$1" -gt "70" && "$1" -le "80" ]]
then
    echo C
elif [[ "$1" -gt "60" && "$1" -le "70" ]]
then
    echo D
else
    echo E
fi

```

```
[root@localhost chenyang] # vim sum.sh
[root@localhost chenyang] # vim gradelevel.sh
[root@localhost chenyang] # sh gradelevel.sh 99
A
[root@localhost chenyang] # sh gradelevel.sh 85
B
[root@localhost chenyang] # sh gradelevel.sh 77
C
[root@localhost chenyang] # sh gradelevel.sh 62
D
[root@localhost chenyang] # sh gradelevel.sh 45
E
```

四、实验过程分析与讨论

在写代码的部分稍微有点困难，在 CSDN 学习并解决。

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 12 日
学 号	2021224827	姓 名	陈阳
专业班级	奥林计算机科学与技术 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。

```
source_dir=$(pwd)
target_dir=~/workspace
for file in *.c; do
    cp "$file" "$target_dir"
done
echo "Copied .c files from $source_dir to $target_dir"
```

~

```
[root@localhost chenyang]# ls /workspace
[root@localhost chenyang]# vim c.sh
[root@localhost chenyang]# sh c.sh
cp: 无法获取 "*.c" 的文件状态(stat): 没有那个文件或目录
Copied .c files from /home/chenyang to /root/workspace
[root@localhost chenyang]# ls ~/workspace
```

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

```
count=0
sum=0
number=0
while [ $count -lt 10 ]; do
    number=$((number + 2))
    sum=$((sum + number))
    count=$((count + 1))
done
echo "The sum of the first 10 even numbers is: $sum"
```

```
[root@localhost chenyang]# vim sumeven.sh
[root@localhost chenyang]# sh sumeven.sh
```

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

```
#!/bin/bash
num=1
count=1
until ((num>10))
do
    count=$(( ${count} + num**2 ))
    num=$(( ${num} +1 ))
done
echo $count
```

```
[chenyang@localhost ~]$ vim squaresum.sh
[chenyang@localhost ~]$ bash squaresum.sh \
>
386
[chenyang@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 、 break 2 、 continue 、 continue 2 ，并观察四种情况下的实验结果。

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

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

```
echo -n $i
```

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

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

```
---
```

```
fi
```

```
echo -n $j
```

```
done
```

```
echo "
```

```
done
```

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

```
[chenyang@localhost ~]$ vim test.sh
[chenyang@localhost ~]$ sh test.sh
a1234
b1234
c1234
d1234
```

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

```
[chenyang@localhost ~]$ vim test.sh
[chenyang@localhost ~]$ sh test.sh
a1234[chenyang@localhost ~]$
```

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

```
a1234[chenyang@localhost ~]$ vim test.sh
[chenyang@localhost ~]$ sh test.sh
a1234678910
b1234678910
c1234678910
d1234678910
```



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

```
[chenyang@localhost ~]$ vim test.sh
```

```
[chenyang@localhost ~]$ sh test.sh
```

```
a1234b1234c1234d1234[chenyang@localhost ~]$ █
```

四、实验过程分析与讨论

对代码不太熟悉，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	实验七 Shell 函数		
实验教室	丹青 922	实验日期	2023 年 5 月 12 日
学 号	2021224827	姓 名	陈阳
专业班级	奥林计算机科学与技术 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

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；

```
#!/bin/bash
numsum()
[
read -p "please send me a num: " num1
read -p "please send me another num: " num2
echo "the sum is $(( ${num1} + ${num2} ))"
return $(( ${num1} + ${num2} ))
]
numsum
echo $?
```

a1234b1234c1234d1234[chenyang@localhost ~]\$ vim sum2.sh
[chenyang@localhost ~]\$ sh sum2.sh
please send me a num: 4
please send me another num: 6
the sum is 10
10

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；

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

[chenyang@localhost ~]$ vim jc.sh
[chenyang@localhost ~]$ sh jc.sh
num
5
```

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。

```
#!/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
~
```

```
[chenyang@localhost ~]$ vim qt.sh
[chenyang@localhost ~]$ sh qt.sh
starting...
-1- here is in the first func.
-2- here is in the second func.
-3- here is in the third func.
[chenyang@localhost ~]$
```

四、实验过程分析与讨论

出现了一些问题，还需要进一步学习并掌握代码。

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 12 日
学 号	2021224827	姓 名	陈阳
专业班级	奥林计算机科学与技术 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 nurse Miss P.Neave was in attendance.

试使用 sed 命令实现如下功能：

(1) 删除 \$ 符号；

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

(2) 显示包含 music 文字的行内容及行号；

```
[chenyang@localhost ~]$ nl quote.txt | sed -n '/music/p'
2 It was an evening of splendid music and company.
```

(3) 在第 4 行后面追加内容： "hello world!" ；

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

(4) 将文本 "The" 替换为 "Quod" ；

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

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

```
[chenyang@localhost ~]$ sed '3c This is the third line.' quote.txt
The honeysuckle band played all night long for only $90.
This is the third line.
The local nurse Miss P.Neave was in attendance.
```

(6) 删除第 2 行内容；

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

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

```
[chenyang@localhost ~]$ var=evening
[chenyang@localhost ~]$ cat quote.txt | sed -n "/$var/p"
It was an evening of splendid music and company.
```

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；

```
[chenyang@localhost ~]$ vim numbers.txt
[chenyang@localhost ~]$ awk -F ':' '{print $2}' numbers.txt
two
five
[chenyang@localhost ~]$ awk -F ' ' '{print $2}' numbers.txt
:
:
[chenyang@localhost ~]$
```

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

```
[chenyang@localhost ~]$ vim foo.txt
[chenyang@localhost ~]$ awk 'BEGIN{count=0} {for(i=1; i<=NF; i++) {if($i%2==0){pri
nt $i; count +=1}} } END{print count}' foo.txt
2
4
46
3
[chenyang@localhost ~]$ █
```

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。

```
#!/bin/bash
read -p "enter search pattern: " pattern
awk "/$pattern/" '{ nmatcher++; print } END { print nmatcher, "found." }' info.txt

[chenyang@localhost ~]$ vim info.txt
[chenyang@localhost ~]$ sh info.txt
enter search pattern: a
#!/bin/bash
read -p "enter search pattern: " pattern
awk "/$pattern/" '{ nmatcher++; print } END { print nmatcher, "found." }' info.txt
3 found.
```

四、实验过程分析与讨论

对部分命令不太熟悉，查询 CSDN 并解决

五、指导教师意见

指导教师签字：卢洋

