

# 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 10 日 |
| 学 号  | 2021213104        | 姓 名  | 谭茗瑞             |
| 专业班级 | 计算机科学与技术 03 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

```
tanmingrui@tanmingrui-virtual-machine:~$ cd /etc
tanmingrui@tanmingrui-virtual-machine:/etc$ pwd
/etc
tanmingrui@tanmingrui-virtual-machine:/etc$
```

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

```
tanmingrui@tanmingrui-virtual-machine:~$ cd /home
tanmingrui@tanmingrui-virtual-machine:/home$ cd tanmingrui
tanmingrui@tanmingrui-virtual-machine:~$ ls -a
.      视频  音乐      .bashrc  .pam_environment
..     图片  桌面      .cache   .profile
公共的 文档  .bash_history .config  snap
模板   下载  .bash_logout .local   .sudo_as_admin_successful
tanmingrui@tanmingrui-virtual-machine:~$
```

#### 3、使用命令创建目录/home/tangmingrui/linux， 然后删除该目录。

```
tanmingrui@tanmingrui-virtual-machine:/home$ cd tanmingrui
tanmingrui@tanmingrui-virtual-machine:~$ mkdir linux
tanmingrui@tanmingrui-virtual-machine:~$ ls
公共的 模板 视频 图片 文档 下载 音乐 桌面 linux snap
tanmingrui@tanmingrui-virtual-machine:~$ rmdir linux
tanmingrui@tanmingrui-virtual-machine:~$ ls
公共的 模板 视频 图片 文档 下载 音乐 桌面 snap
tanmingrui@tanmingrui-virtual-machine:~$
```

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

```
tanmingrui@tanmingrui-virtual-machine:~$ cat > abc
Hello, Linux!
^C
tanmingrui@tanmingrui-virtual-machine:~$ ls
公共的 模板 视频 图片 文档 下载 音乐 桌面 abc snap
tanmingrui@tanmingrui-virtual-machine:~$ cat abc
Hello, Linux!
tanmingrui@tanmingrui-virtual-machine:~$
```

5、使用命令创建目录/home/tanmingrui/ak，然后将/home/tanmingrui/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。

```
tanmingrui@tanmingrui-virtual-machine:~$ mkdir ak
tanmingrui@tanmingrui-virtual-machine:~$ ls
公共的 模板 视频 图片 文档 下载 音乐 桌面 abc ak snap
tanmingrui@tanmingrui-virtual-machine:~$ cp -r abc ak
tanmingrui@tanmingrui-virtual-machine:~$ ls
公共的 模板 视频 图片 文档 下载 音乐 桌面 abc ak snap
tanmingrui@tanmingrui-virtual-machine:~$ cd ak
tanmingrui@tanmingrui-virtual-machine:~/ak$ rm -i abc
rm: 是否删除普通文件 'abc'? y
tanmingrui@tanmingrui-virtual-machine:~/ak$ cd ..
tanmingrui@tanmingrui-virtual-machine:~$ rmdir ak
tanmingrui@tanmingrui-virtual-machine:~$
```

cd

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。

```
tanmingrui@tanmingrui-virtual-machine:/etc$ head -n 3 adduser.conf
# /etc/adduser.conf: `adduser' configuration.
# See adduser(8) and adduser.conf(5) for full documentation.
```

```
tanmingrui@tanmingrui-virtual-machine:/etc$ tail -n 5 adduser.conf
#NAME_REGEX="^[a-z][-a-z0-9_]*$"

# use extrausers by default
#USE_EXTRAUSERS=1
DIR_MODE=0750
tanmingrui@tanmingrui-virtual-machine:/etc$
```

7、分屏查看文件/etc/adduser.conf 的内容。

```
tanmingrui@tanmingrui-virtual-machine:/etc$ more adduser.conf
# /etc/adduser.conf: `adduser' configuration.
# See adduser(8) and adduser.conf(5) for full documentation.

# A commented out setting indicates that this is the default in the
# code. If you need to change those settings, remove the comment and
# make your intended change.

# The DSHELL variable specifies the default login shell on your
# system.
#DSHELL=/bin/bash

# The DHOME variable specifies the directory containing users' home
# directories.
#DHOME=/home
```

8、使用命令cat用输出重定向在/home/tanmingrui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

```
tanmingrui@tanmingrui-virtual-machine:~$ cat > facebook.txt
goole 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。

```
tanmingrui@tanmingrui-virtual-machine:~$ sort -r facebook.txt
sohu 100 4500
guge 50 3000
goole 110 5000
baidu 100 5000
tanmingrui@tanmingrui-virtual-machine:~$ sort -n facebook.txt
baidu 100 5000
goole 110 5000
guge 50 3000
sohu 100 4500
```

```
tanmingrui@tanmingrui-virtual-machine:~$ sort -n -t ' ' -k 2 -k 3 facebook.txt
guge 50 3000
sohu 100 4500
baidu 100 5000
goole 110 5000
```

```
tanmingrui@tanmingrui-virtual-machine:~$ sort -n -t ' ' -k 3r -k 2 facebook.txt
baidu 100 5000
goole 110 5000
sohu 100 4500
guge 50 3000
tanmingrui@tanmingrui-virtual-machine:~$ sort -t ' ' -k 1.2 facebook.txt
baidu 100 5000
sohu 100 4500
goole 110 5000
guge 50 3000
```

#### 四、 实验过程分析与讨论

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

#### 五、指导教师意见

指导教师签字：卢洋

# 实验报告

|      |                   |      |                 |
|------|-------------------|------|-----------------|
| 实验名称 | 实验二 Linux 常用命令（二） |      |                 |
| 实验教室 | 丹青 922            | 实验日期 | 2023 年 5 月 11 日 |
| 学 号  | 2021213104        | 姓 名  | 谭茗瑞             |
| 专业班级 | 计算机科学与技术 03 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；

```
tanmingrui@tanmingrui-virtual-machine:~$ cd /home
tanmingrui@tanmingrui-virtual-machine:/home$ cd tanmingrui
tanmingrui@tanmingrui-virtual-machine:~$ mkdir baz
tanmingrui@tanmingrui-virtual-machine:~$ cd baz
tanmingrui@tanmingrui-virtual-machine:~/baz$ cat > qux
1
2
3
^C
```

(2) 在用户目录下查找文件qux，并显示该文件位置信息；

```
tanmingrui@tanmingrui-virtual-machine:~/baz$ ls
qux
tanmingrui@tanmingrui-virtual-machine:~/baz$ cd ..
tanmingrui@tanmingrui-virtual-machine:~$ find -name qux
./baz/qux
tanmingrui@tanmingrui-virtual-machine:~$ ls -li -F qux
ls: 无法访问 'qux': 没有那个文件或目录
tanmingrui@tanmingrui-virtual-machine:~$ cd baz
tanmingrui@tanmingrui-virtual-machine:~/baz$ ls -li -F qux
397764 qux
```

(3) 统计文件qux中所包含内容的行数、字数和字节数；

```
tanmingrui@tanmingrui-virtual-machine:~/baz$ wc -l qux
3 qux
tanmingrui@tanmingrui-virtual-machine:~/baz$ wc -m qux
6 qux
tanmingrui@tanmingrui-virtual-machine:~/baz$ wc -c qux
6 qux
tanmingrui@tanmingrui-virtual-machine:~/baz$ wc qux
3 3 6 qux
tanmingrui@tanmingrui-virtual-machine:~/baz$
```

(4) 在用户目录下查找文件qux，并删除该文件；

```
tanmingrui@tanmingrui-virtual-machine:~/baz$ cd ..
tanmingrui@tanmingrui-virtual-machine:~$ cd ..
tanmingrui@tanmingrui-virtual-machine:/home$ cd tangmingrui
bash: cd: tangmingrui: 没有那个文件或目录
tanmingrui@tanmingrui-virtual-machine:/home$ cd tanmingrui
tanmingrui@tanmingrui-virtual-machine:~$ find -name qux -exec rm -rf {} \;
```

(5) 查看文件夹baz内容，看一下是否删除了文件qux。

```
tanmingrui@tanmingrui-virtual-machine:~$ cd baz
tanmingrui@tanmingrui-virtual-machine:~/baz$ ls
tanmingrui@tanmingrui-virtual-machine:~/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;

```
tanmingrui@tanmingrui-virtual-machine:~/baz$ cd ..
tanmingrui@tanmingrui-virtual-machine:~$ mkdir path1
tanmingrui@tanmingrui-virtual-machine:~$ cd path1
tanmingrui@tanmingrui-virtual-machine:~/path1$ touch file1
tanmingrui@tanmingrui-virtual-machine:~/path1$ touch file2
tanmingrui@tanmingrui-virtual-machine:~/path1$ ls
file1 file2
tanmingrui@tanmingrui-virtual-machine:~/path1$
```

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

```
tanmingrui@tanmingrui-virtual-machine:~$ mkdir path2
tanmingrui@tanmingrui-virtual-machine:~$ cd path2
tanmingrui@tanmingrui-virtual-machine:~/path2$ touch file3
tanmingrui@tanmingrui-virtual-machine:~/path2$ ls
file3
tanmingrui@tanmingrui-virtual-machine:~/path2$
```

(3) 在用户目录下新建文件file4;

```
tanmingrui@tanmingrui-virtual-machine:~/path2$ cd ..
tanmingrui@tanmingrui-virtual-machine:~$ touch file4
tanmingrui@tanmingrui-virtual-machine:~$ ls
公共的 模板 视频 图片 文档 下载 音乐 桌面 abc baz facebook.txt file4 path1 path2 snap
tanmingrui@tanmingrui-virtual-machine:~$
```

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

```
tanmingrui@tanmingrui-virtual-machine:~$ tar -cvf package.tar file4 path1
file4
path1/
path1/file2
path1/file1
```

(5) 查看包package.tar的内容;

```
tanmingrui@tanmingrui-virtual-machine:~$ cat package.tar
file400000664000175000017500000000000014432647272013123 0ustar  tanmingruitanmingruipath1/0000775000175000017500000000000
14432647047013224 5ustar  tanmingruitanmingruipath1/file2000066400017500001750000000000014432647047014136 0ustar  tanmin
gruitanmingruipath1/file1000066400017500001750000000000014432647043014131 0ustar  tanmingruitanmingruitanmingrui@tanming
rui-virtual-machine:~$
```

(6) 向包package.tar里添加文件夹path2的内容;

```
tanmingrui@tanmingrui-virtual-machine:~$ tar -rf package.tar path2
tanmingrui@tanmingrui-virtual-machine:~$ tar -tf package.tar
file4
path1/
path1/file2
path1/file1
path2/
path2/file3
tanmingrui@tanmingrui-virtual-machine:~$
```

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

```
tanmingrui@tanmingrui-virtual-machine:~$ mkdir path3
tanmingrui@tanmingrui-virtual-machine:~$ cp package.tar path3
tanmingrui@tanmingrui-virtual-machine:~$ cd path3
tanmingrui@tanmingrui-virtual-machine:~/path3$ ls
package.tar
```

(8) 进入path3文件夹, 并还原包package.tar的内容。

```
tanmingrui@tanmingrui-virtual-machine:~/path3$ tar -xvf package.tar
file4
path1/
path1/file2
path1/file1
path2/
path2/file3
tanmingrui@tanmingrui-virtual-machine:~/path3$ ls
file4  package.tar  path1  path2
tanmingrui@tanmingrui-virtual-machine:~/path3$
```

### 3. 符号链接内容

(1) 新建文件foo.txt，内容为123;

```
tanmingrui@tanmingrui-virtual-machine:~$ echo 123 > foo.txt
tanmingrui@tanmingrui-virtual-machine:~$ ls
公共的  视频  文档  音乐  abc  facebook.txt  foo.txt  path1  path3
模板    图片  下载  桌面  baz  file4         package.tar  path2  snap
tanmingrui@tanmingrui-virtual-machine:~$ cat foo.txt
123
tanmingrui@tanmingrui-virtual-machine:~$
```

(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;

```
tanmingrui@tanmingrui-virtual-machine:~$ ln foo.txt bar.txt
tanmingrui@tanmingrui-virtual-machine:~$ diff foo.txt bar.txt
tanmingrui@tanmingrui-virtual-machine:~$ ls
公共的  视频  文档  音乐  abc  baz  file4  package.tar  path2  snap
模板    图片  下载  桌面  bar.txt  facebook.txt  foo.txt  path1  path3
tanmingrui@tanmingrui-virtual-machine:~$ comm foo.txt bar.txt
123
tanmingrui@tanmingrui-virtual-machine:~$
```

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

```
tanmingrui@tanmingrui-virtual-machine:~$ ls -li -F foo.txt
399133 foo.txt
tanmingrui@tanmingrui-virtual-machine:~$ ls -li -F bar.txt
399133 bar.txt
tanmingrui@tanmingrui-virtual-machine:~$
```

(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;

```
tanmingrui@tanmingrui-virtual-machine:~$ vi bar.txt
tanmingrui@tanmingrui-virtual-machine:~$ diff bar.txt foo.txt
tanmingrui@tanmingrui-virtual-machine:~$ comm bar.txt foo.txt
      abc
tanmingrui@tanmingrui-virtual-machine:~$ cat foo.txt
abc
tanmingrui@tanmingrui-virtual-machine:~$
```

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

```
tanmingrui@tanmingrui-virtual-machine:~$ rm foo.txt
tanmingrui@tanmingrui-virtual-machine:~$ ls -li -F bar.txt
395247 bar.txt
tanmingrui@tanmingrui-virtual-machine:~$ cat bar.txt
abc
tanmingrui@tanmingrui-virtual-machine:~$
```

(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;

```
tanmingrui@tanmingrui-virtual-machine:~$ ln -s bar.txt baz.txt
tanmingrui@tanmingrui-virtual-machine:~$ ls
公共的 图片 音乐 bar.txt facebook.txt path1 snap
模板 文档 桌面 baz file4 path2
视频 下载 abc baz.txt package.tar path3
tanmingrui@tanmingrui-virtual-machine:~$ ls -li -F bar.txt
395247 bar.txt
tanmingrui@tanmingrui-virtual-machine:~$ ls -li -F baz.txt
393435 baz.txt@
tanmingrui@tanmingrui-virtual-machine:~$ comm baz.txt bar.txt
      abc
tanmingrui@tanmingrui-virtual-machine:~$
```

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

```
tanmingrui@tanmingrui-virtual-machine:~$ rm bar.txt
tanmingrui@tanmingrui-virtual-machine:~$ cat baz.txt
cat: baz.txt: 没有那个文件或目录
tanmingrui@tanmingrui-virtual-machine:~$
```

## 4. 权限管理

(1) 新建文件qux.txt;

```
tanmingrui@tanmingrui-virtual-machine:~$ touch qux.txt
tanmingrui@tanmingrui-virtual-machine:~$ ls
公共的 视频 文档 音乐 abc baz.txt file4 path1 path3 snap
模板 图片 下载 桌面 baz facebook.txt package.tar path2 qux.txt
tanmingrui@tanmingrui-virtual-machine:~$
```

(2) 为文件qux.txt增加执行权限(所有用户都可以执行)。

```
tanmingrui@tanmingrui-virtual-machine:~$ touch qux.txt
tanmingrui@tanmingrui-virtual-machine:~$ ls
公共的 视频 文档 音乐 abc baz.txt file4 path1 path3 snap
模板 图片 下载 桌面 baz facebook.txt package.tar path2 qux.txt
tanmingrui@tanmingrui-virtual-machine:~$ chmod a+x qux.txt
tanmingrui@tanmingrui-virtual-machine:~$ ls -al qux.txt
-rwxrwxr-x 1 tanmingrui tanmingrui 0 5月 22 19:41 qux.txt
tanmingrui@tanmingrui-virtual-machine:~$
```

#### 四、实验过程分析与讨论

对于很多命令的不熟悉导致做实验需要花大量的时间，所以Linux 这门课需要不断地练习来达到熟能生巧的目的。往往一个命令拥有很多的参数，能够实现各种各样的功能。其中 ln 命令生成硬链接时是使当前文件指向该文件的索引号，这样删除另一个文件，通过硬链接仍然可以访问到以前的数据。而生成软连接时实际上是生成了一个包含另一文件的位置信息的文本文件，当源文件删除，该文件便不可访问内容。

## 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 12 日 |
| 学 号  | 2021213104              | 姓 名  | 谭茗瑞             |
| 专业班级 | 计算机科学与技术 03 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;

```
tanmingrui@tanmingrui-virtual-machine:~$ mkdir workspace1
tanmingrui@tanmingrui-virtual-machine:~$ ls
公共的  图片  音乐  baz          file4      path2     snap
模板    文档  桌面  baz.txt      package.tar path3     workspace1
视频    下载  abc   facebook.txt path1      qux.txt
```

- (2) 进入目录 workspace1;

```
tanmingrui@tanmingrui-virtual-machine:~$ cd workspace1
tanmingrui@tanmingrui-virtual-machine:~/workspace1$ ls
tanmingrui@tanmingrui-virtual-machine:~/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;

```
#include<stdio.h>

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

~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
```

: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。

```
tanmingrui@tanmingrui-virtual-machine:~/workspace$ gcc test.c -o test
tanmingrui@tanmingrui-virtual-machine:~/workspace$ ./test
hello world!
tanmingrui@tanmingrui-virtual-machine:~/workspace$
```

## 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;

```
tanmingrui@tanmingrui-virtual-machine:~$ mkdir workspace2
tanmingrui@tanmingrui-virtual-machine:~$ ls
公共的  图片  音乐  baz          file4      path2      snap
模板    文档  桌面  baz.txt      package.tar path3      workspace1
视频    下载  abc   facebook.txt path1      qux.txt   workspace2
tanmingrui@tanmingrui-virtual-machine:~$
```

(2) 进入workspace2目录;

```
tanmingrui@tanmingrui-virtual-machine:~$ cd workspace2
tanmingrui@tanmingrui-virtual-machine:~/workspace2$ ls
tanmingrui@tanmingrui-virtual-machine:~/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;

```
tanmingrui@tanmingrui-virtual-machine:~/workspace2$ cat /etc/gai.conf > ./gai.conf
tanmingrui@tanmingrui-virtual-machine:~/workspace2$ ls
gai.conf
tanmingrui@tanmingrui-virtual-machine:~/workspace2$
```

(4) 使用vim 编辑当前目录下的 gai.conf;

```

# Configuration for getaddrinfo(3).
#
# So far only configuration for the destination address sorting is needed.
# RFC 3484 governs the sorting. But the RFC also says that system
# administrators should be able to overwrite the defaults. This can be
# achieved here.
#
# All lines have an initial identifier specifying the option followed by
# up to two values. Information specified in this file replaces the
# default information. Complete absence of data of one kind causes the
# appropriate default information to be used. The supported commands include:
#
# help ad <yes|no>
# If set to yes, each getaddrinfo(3) call will check whether this file
# changed and if necessary reload. This option should not really be
# used. There are possible runtime problems. The default is no.
#
# label <mask> <value>
# Add another rule to the RFC 3484 label table. See section 2.1 in
# RFC 3484. The default is:
#
#label ::1/128 0
#label ::/0 1
#label 2002::/16 2
#label ::/96 3
#label ::ffff:0:0/96 4
#label fec0::/10 5
#label fc00::/7 6
#label 2001:0::/32 7
#
# This default differs from the tables given in RFC 3484 by handling
# (now obsolete) site-local IPv6 addresses and Unique Local Addresses.
# The reason for this difference is that these addresses are never
# NATed while IPv4 site-local addresses most probably are. Given
# the precedence of IPv6 over IPv4 (see below) on machines having only
# 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>
"gai.conf" 65L, 2584B

```

(5) 将光标移到第 18 行;

```
# Configuration for getaddrinfo(3).
#
# So far only configuration for the destination address sorting is needed.
# RFC 3484 governs the sorting. But the RFC also says that system
# administrators should be able to overwrite the defaults. This can be
# achieved here.
#
# All lines have an initial identifier specifying the option followed by
# up to two values. Information specified in this file replaces the
# default information. Complete absence of data of one kind causes the
# appropriate default information to be used. The supported commands include:
#
# reload <yes|no>
#   If set to yes, each getaddrinfo(3) call will check whether this file
#   changed and if necessary reload. This option should not really be
#   used. There are possible runtime problems. The default is no.
#
# label <mask> <value>
#   Add another rule to the RFC 3484 label table. See section 2.1 in
#   RFC 3484. The default is:
#
#label ::1/128      0
#label ::/0         1
#label 2002::/16    2
#label ::/96        3
#label ::ffff:0:0/96 4
#label fec0::/10    5
#label fc00::/7     6
#label 2001:0::/32  7
#
# This default differs from the tables given in RFC 3484 by handling
# (now obsolete) site-local IPv6 addresses and Unique Local Addresses.
# The reason for this difference is that these addresses are never
# NATed while IPv4 site-local addresses most probably are. Given
# the precedence of IPv6 over IPv4 (see below) on machines having only
# 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>
"gal.conf" 65L, 2584B
```

(6) 复制该行内容;

```
# label <mask> <value>
#   Add another rule to the RFC 3484
```

(7) 将光标移到最后一行行首;

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

(8) 粘贴复制行的内容;

```
# label <mask> <value>
scopev4 ::ffff:0.0.0.0/96         14
显示应用程序
```

(9) 撤销第 8 步的动作;

```
#scopev4 ::ffff:0.0.0.0/96         14
~
1 行被去掉; before #1 45 秒前
```

(10) 存盘但不退出;

```
#scopev4 ::ffff:0.0.0.0/96      14
~
:W
```

(11) 将光标移到首行;

H

```
Configuration for getaddrinfo(3).
#
# So far only configuration for the destination address sorting is needed.
# RFC 3484 governs the sorting. But the RFC also says that system
# administrators should be able to overwrite the defaults. This can be
# achieved here.
#
# All lines have an initial identifier specifying the option followed by
# up to two values. Information specified in this file replaces the
# default information. Complete absence of data of one kind causes the
# appropriate default information to be used. The supported commands include:
#
# reload <yes|no>
#   If set to yes, each getaddrinfo(3) call will check whether this file
#   changed and if necessary reload. This option should not really be
#   used. There are possible runtime problems. The default is no.
#
# label <mask> <value>
#   Add another rule to the RFC 3484 label table. See section 2.1 in
#   RFC 3484. The default is:
#
#label ::1/128      0
#label ::/0         1
#label 2002::/16    2
#label ::/96        3
#label ::ffff:0:0/96 4
#label fec0::/10    5
#label fc00::/7     6
#label 2001:0::/32  7
#
# This default differs from the tables given in RFC 3484 by handling
# (now obsolete) site-local IPv6 addresses and Unique Local Addresses.
# The reason for this difference is that these addresses are never
# NATed while IPv4 site-local addresses most probably are. Given
# the precedence of IPv6 over IPv4 (see below) on machines having only
# 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>
```

1,1

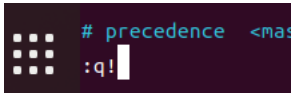
(12) 插入模式下输入"Hello, this is vim world!";

```
Hello, this is vim world!# Configuration for getaddrinfo(3).
#
```

(13) 删除字符串"this";

```
Hello, is vim world!# Configuration for getaddrinfo(3).
#
```

(14) 强制退出vim, 不存盘。



```
# precedence <mas
:q!
```

#### 四、 实验过程分析与讨论

Vim 的使用操作很多很杂，但整体上来将难度并不大，只要跟着笔记和资料都是能轻松完成的。只要能熟悉这些操作，就能多快好省的完成编辑工作



## 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 12 日 |
| 学 号  | 2021213104    | 姓 名  | 谭茗瑞             |
| 专业班级 | 计算机科学与技术 03 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@tanmingrui-virtual-machine:/home# useradd -d /home/foo -s /bin/sh -g users -m foo
root@tanmingrui-virtual-machine:/home# ll
总用量 16
drwxr-xr-x  4 root      root      4096  5月 22 21:03 ./
drwxr-xr-x 20 root      root      4096  3月  4 19:50 ../
drwxr-x---  2 foo       users    4096  5月 22 21:03 foo/
drwxr-x--- 20 tanmingrui tanmingrui 4096  5月 22 20:40 tanmingrui/
root@tanmingrui-virtual-machine:/home# usermod -c bar foo
root@tanmingrui-virtual-machine:/home# tail -1 /etc/passwd
foo:x:1001:100:bar:/home/foo:/bin/sh
root@tanmingrui-virtual-machine:/home# usermod -p 123456 foo
root@tanmingrui-virtual-machine:/home# tail -1 /etc/passwd
foo:x:1001:100:bar:/home/foo:/bin/sh
root@tanmingrui-virtual-machine:/home#
```

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

```
root@ubuntu:/home# usermod -u 2000 -s /bin/csh foo
usermod: no changes
root@ubuntu:/home# su foo
% tail -1 /etc/passwd
foo:x:2000:100:bar:/home/foo:/bin/csh
%
```

3. 从用户 foo 切换到 root;

```
% exit
% exit
root@ubuntu:/home#
```

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@tanmingrui-virtual-machine:/home# userdel -r foo
userdel: foo 邮件池 (/var/mail/foo) 未找到
root@tanmingrui-virtual-machine:/home# tail -1 /etc/passwd
tanmingrui:x:1000:1000:tanmingrui,,,:/home/tanmingrui:/bin/bash
root@tanmingrui-virtual-machine:/home# cd foo
bash: cd: foo: 没有那个文件或目录
root@tanmingrui-virtual-machine:/home# ll
总用量 12
drwxr-xr-x  3 root      root      4096  5月 22 21:09 ./
drwxr-xr-x 20 root      root      4096  3月  4 19:50 ../
drwxr-x--- 20 tanmingrui tanmingrui 4096  5月 22 20:40 tanmingrui/
root@tanmingrui-virtual-machine:/home#

```

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@ubuntu:/home# newusers manage
newusers: manage: No such file or directory
root@ubuntu:/home# vi manage
root@ubuntu:/home# newusers manage
root@ubuntu:/home# vi userpass
root@ubuntu:/home# chpasswd < userpass
chpasswd: (user 2025) pam_chauthtok() failed, error:
Authentication token manipulation error
chpasswd: (line 6, user 2025) password not changed
chpasswd: line 7: missing new password

```

```

gdm:x:121:125:Gnome Display Manager:/var/lib/gdm3:/bin/false
zfm:x:1000:1000:zfm,,,:/home/zfm:/bin/bash
2020:x:1002:1002::/home/2020:
2021:x:1003:1003::/home/2021:
2022:x:1004:1004::/home/2022:
2023:x:1005:1005::/home/2023:
2024:x:1006:1006::/home/2024:
root@ubuntu:/home#

```

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

```

root@tanmingrui-virtual-machine:/home# groupadd -g 3000 group1
root@tanmingrui-virtual-machine:/home# tail -1 /etc/group
group1:x:3000:
root@tanmingrui-virtual-machine:/home#

```

7. 在用户组 `group1` 中添加两个之前批量创建的用户；

```

root@ubuntu:/# usermod -G group1 2023
root@ubuntu:/# usermod -G group1 2024
usermod: user '2024' does not exist
root@ubuntu:/# usermod -G group1 2022
root@ubuntu:/# tail -1 /etc/group
group1:x:3000:2023,2022

```

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

```
root@ubuntu:/# cat /etc/sudoers
root@ubuntu:/# ls -al /etc/sudoers
-r--r----- 1 root root 755 Jan 17 2018 /etc/sudoers
root@ubuntu:/# chmod u+w /etc/sudoers
root@ubuntu:/# ls -al /etc/sudoers
-rw-r----- 1 root root 755 Jan 17 2018 /etc/sudoers
root@ubuntu:/# vi /etc/sudoers
root@ubuntu:/#
cat: /etc/shadow: No such file or directory
$ sudo cat /etc/shadow
root:$6$eGqWAqnU$HIW5w0XgSvWv.s3/yTm.9NdRM6PdMh4HX5seCRrvQknXA7jp
wZv2UWnxFWvd2CmyJ6ySZok1:19088:0:99999:7:::
daemon*:18885:0:99999:7:::
bin*:18885:0:99999:7:::
sys*:18885:0:99999:7:::
sync*:18885:0:99999:7:::
games*:18885:0:99999:7:::
man*:18885:0:99999:7:::
lp*:18885:0:99999:7:::
mail*:18885:0:99999:7:::
news*:18885:0:99999:7:::
uucp*:18885:0:99999:7:::
proxy*:18885:0:99999:7:::
www-data*:18885:0:99999:7:::
backup*:18885:0:99999:7:::
list*:18885:0:99999:7:::
irc*:18885:0:99999:7:::
gnats*:18885:0:99999:7:::
```

#### 四、 实验过程分析与讨论

用户与组的操作有点难度, 重点在于掌握有关用户与组的文件信息, 以及每个字段代表的是什么意思。第一次切换到 root 用户时显示密码错误, 但实际上我从未给 root 设置过密码, 之后上百度查阅才成功解决了问题。我在批量创建用户这一方面还并未彻底掌握, 实验后一定再仔细学一遍。

## 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 22 日 |
| 学 号  | 2021213104             | 姓 名  | 谭茗瑞             |
| 专业班级 | 计算机科学与技术 03 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);

```
tanmingrui@tanmingrui-virtual-machine:~$ foo=200
tanmingrui@tanmingrui-virtual-machine:~$ echo $foo
200
tanmingrui@tanmingrui-virtual-machine:~$
```

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);

```
tanmingrui@tanmingrui-virtual-machine:~/workspace$ bar=100
tanmingrui@tanmingrui-virtual-machine:~/workspace$ test $bar -gt 150 && echo yes || echo no
no
tanmingrui@tanmingrui-virtual-machine:~/workspace$ echo $?
0
tanmingrui@tanmingrui-virtual-machine:~/workspace$ test $bar -gt 150
tanmingrui@tanmingrui-virtual-machine:~/workspace$ echo $?
1
tanmingrui@tanmingrui-virtual-machine:~/workspace$
```

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);

```
tanmingrui@tanmingrui-virtual-machine:~/workspace$ vi pr-hn-date.sh
tanmingrui@tanmingrui-virtual-machine:~/workspace$ sh pr-hn-date.sh
tanmingrui-virtual-machine
2023年 05月 22日 星期一 21:34:34 CST
tanmingrui@tanmingrui-virtual-machine:~/workspace$ cat pr-hn-date.sh
#!/bin/bash
hn=$(cat /etc/hostname)
dt=$(date)
echo $hn
echo $dt
tanmingrui@tanmingrui-virtual-machine:~/workspace$
```

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：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。

```

tanmingrui@tanmingrui-virtual-machine:~/workspace$ vi judegne.sh
tanmingrui@tanmingrui-virtual-machine:~/workspace$ bash judegne.sh
please input the number:153
This number is daffodils
tanmingrui@tanmingrui-virtual-machine:~/workspace$ bash judegne.sh
please input the number:124
This number is not daffodils
tanmingrui@tanmingrui-virtual-machine:~/workspace$ cat judegne.sh
#!/bin/bash
read -p "please input the number:" num
numx=$num
num1=$((num%10))
num=$((num/10))
num2=$((num%10))
num=$((num/10))
num3=$((num%10))
num=$((num1**3+num2**3+num3**3))
if test $num -eq $numx
then
    echo 'This number is daffodils'
else
    echo 'This number is not daffodils'
fi
tanmingrui@tanmingrui-virtual-machine:~/workspace$ bash judegne.sh
please input the number:370
This number is daffodils
tanmingrui@tanmingrui-virtual-machine:~/workspace$

```

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；

```

tanmingrui@tanmingrui-virtual-machine:~/workspace$ vi sum.sh
tanmingrui@tanmingrui-virtual-machine:~/workspace$ chmod u+x sum.sh
tanmingrui@tanmingrui-virtual-machine:~/workspace$ ./sum.sh
Please input the first number:1
Please input the second number:2
Please input the third number:3
the sum= 6
tanmingrui@tanmingrui-virtual-machine:~/workspace$ cat sum.sh
#!/bin/bash
read -p "Please input the first number:" num1
read -p "Please input the second number:" num2
read -p "Please input the third number:" num3
num=$((num1+num2+num3))
echo "the sum= $num"
tanmingrui@tanmingrui-virtual-machine:~/workspace$

```

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。

```
tanmingrui@tanmingrui-virtual-machine:~/workspace$ vi grade.sh
tanmingrui@tanmingrui-virtual-machine:~/workspace$ sh grade.sh
please input the student's grade: 100
A
tanmingrui@tanmingrui-virtual-machine:~/workspace$ sh grade.sh
please input the student's grade: 50
E
tanmingrui@tanmingrui-virtual-machine:~/workspace$ sh grade.sh
please input the student's grade: 0
E
tanmingrui@tanmingrui-virtual-machine:~/workspace$ cat grade.sh
#!/bin/bash
read -p"please input the student's grade: " gra
if test $gra -ge 90
then echo A
elif test $gra -ge 80
then echo B
elif test $gra -ge 70
then echo C
elif test $gra -ge 60
then echo D
else
    echo E
fi
tanmingrui@tanmingrui-virtual-machine:~/workspace$
```

#### 四、 实验过程分析与讨论

运用 shell 脚本进行编程时,需要熟练地掌握 vim 的使用方法。另外在编程中如果想使用某个命令的返回值时,要么使用反引,要么使用美元符号加括号。在进行数字运算的时候,要用 `$(())` 的格式,或者使用命令。而且有些时候 sh 命令并不能解释带数字运算的语句,这个时候可以用 bash

#### 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 22 日 |
| 学 号  | 2021213104       | 姓 名  | 谭茗瑞             |
| 专业班级 | 计算机科学与技术 03 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。

```
zfm@ubuntu:~/workspace$ ls
zfm@ubuntu:~/workspace$ cd ../test
zfm@ubuntu:~/test$ vi copyC.sh
zfm@ubuntu:~/test$ ls
abc.c  b.c  copyC.sh  grade.sh  pr-hn-date.sh
a.c    c.c  evenFound.sh  judgenu.sh  sum.sh
zfm@ubuntu:~/test$ sh copyC.sh
zfm@ubuntu:~/test$ cd ../workspace
zfm@ubuntu:~/workspace$ ls
abc.c  a.c  b.c  c.c
zfm@ubuntu:~/workspace$ cat ../test/copyC.sh
#!/bin/bash

for i in `find . -name '*.c'`
do
    cp $i ../workspace
done
```

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

果;

```
tanmingrui@tanmingrui-virtual-machine:~/workspace$ vi evenSum.sh
tanmingrui@tanmingrui-virtual-machine:~/workspace$ bash evenSum.sh
110
tanmingrui@tanmingrui-virtual-machine:~/workspace$ cat evenSum.sh
#!/bin/bash

i=2
sum=0
while ((i<=20))
do
    sum=$((sum+i))
    i=$((i+2))
done
echo $sum
tanmingrui@tanmingrui-virtual-machine:~/workspace$
```

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

```
tanmingrui@tanmingrui-virtual-machine:~/workspace$ vi quater.sh
tanmingrui@tanmingrui-virtual-machine:~/workspace$ bash quater.sh
385
tanmingrui@tanmingrui-virtual-machine:~/workspace$ cat quater.sh
#!/bin/bash

i=1
sq=0
until [ $i -ge 11 ]
do
    temp=$((i*i))
    sq=$((sq+temp))
    i=$((i+1))
done
echo $sq
tanmingrui@tanmingrui-virtual-machine:~/workspace$
```

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。



```
#!/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
```

```
tanmingrui@tanmingrui-virtual-machine:~/workspace$ vi replace1.sh
tanmingrui@tanmingrui-virtual-machine:~/workspace$ sh replace1.sh
a1234
b1234
c1234
d1234
```

```
tanmingrui@tanmingrui-virtual-machine:~/workspace$ vi replace1.sh
tanmingrui@tanmingrui-virtual-machine:~/workspace$ sh replace1.sh
a1234tanmingrui@tanmingrui-virtual-machine:~/workspace$
```

```
a1234tanmingrui@tanmingrui-virtual-machine:~/workspace$ vi replace1.sh
tanmingrui@tanmingrui-virtual-machine:~/workspace$ sh replace1.sh
a1234678910
b1234678910
c1234678910
d1234678910
tanmingrui@tanmingrui-virtual-machine:~/workspace$
```

```
tanmingrui@tanmingrui-virtual-machine:~/workspace$ vi replace1.sh
tanmingrui@tanmingrui-virtual-machine:~/workspace$ sh replace1.sh
a1234b1234c1234d1234tanmingrui@tanmingrui-virtual-machine:~/workspace$
```

#### 四、 实验过程分析与讨论

Until 循环条件中若为假继续循环，若为真则跳出循环，这一点是我们平常容易混淆的地方。另外在引用变量时千万不要忘记加'\$'符号。Break 是终止一层循环，continue 是跳出当前循环；而 break 2 是终止两层循环，continue 2 是跳过两个循环。

#### 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 23 日 |
| 学 号  | 2021213104    | 姓 名  | 谭茗瑞             |
| 专业班级 | 计算机科学与技术 03 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;

```
tanmingrui@tanmingrui-virtual-machine:~/workspace$ vi fun_sum.sh
tanmingrui@tanmingrui-virtual-machine:~/workspace$ sh fun_sum.sh
please input the first number: 211
please input the second number: 985
1196
tanmingrui@tanmingrui-virtual-machine:~/workspace$ cat fun_sum.sh
#!/bin/bash

read -p "please input the first number: " a
read -p "please input the second number: " b
sum(){
    result=$((a+b))
    echo $result
}
sum
tanmingrui@tanmingrui-virtual-machine:~/workspace$
```

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;

```
tanmingrui@tanmingrui-virtual-machine:~/workspace$ vi factorial.sh
tanmingrui@tanmingrui-virtual-machine:~/workspace$ bash factorial.sh
please input a positive number: 20
2432902008176640000
tanmingrui@tanmingrui-virtual-machine:~/workspace$ bash factorial.sh
please input a positive number: 5
120
tanmingrui@tanmingrui-virtual-machine:~/workspace$ cat factorial.sh
#!/bin/bash

fact(){
    if [ $1 -eq 1 ]
    then echo 1
    else
        local temp=$(( $1 - 1 ))
        local num=$(( fact $temp ))
        echo=$(( $1 * $num ))
    fi
}

read -p "please input a positive number: " num
result=$(( fact $num ))
echo $result
```

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。

```
tanmingrui@tanmingrui-virtual-machine:~/workspace$ vi test.sh
tanmingrui@tanmingrui-virtual-machine:~/workspace$ bash test.sh
starting. . .
-1- here is in the first func.
-2- here is in the second func.
-3- here is in the third func.
```

#### 四、 实验过程分析与讨论

Shell 函数的括号里不可以像其它编程语言直接传参，但是可以通过函数名 再在后面加上参数，就可以在函数内部用\$1,\$2……的方式调用参数了。函数嵌套即在一个函数中可以定义另一个函数，如此往复，而且不论在那一层都可以调用定义过的函数。

## 五、指导教师意见

指导教师签字：卢洋

# 实验报告

|      |               |      |                 |
|------|---------------|------|-----------------|
| 实验名称 | 实验八 sed 和 awk |      |                 |
| 实验教室 | 丹青 922        | 实验日期 | 2023 年 5 月 23 日 |
| 学 号  | 2021213104    | 姓 名  | 谭茗瑞             |
| 专业班级 | 计算机科学与技术 03 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;

```
tanmingrui@tanmingrui-virtual-machine:~/workspace$ vi quote.txt
tanmingrui@tanmingrui-virtual-machine:~/workspace$ 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

```
tanmingrui@tanmingrui-virtual-machine:~/workspace$ sed -n '/music/{=;p}' quote.txt
2
was an evening of splendid music and company.Too bad the
tanmingrui@tanmingrui-virtual-machine:~/workspace$
```

(3) 在第 4 行后面追加内容: "hello world!"

```
tanmingrui@tanmingrui-virtual-machine:~/workspace$ sed '$ahello 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"

```
tanmingrui@tanmingrui-virtual-machine:~/workspace$ sed 's/The/Quod/' 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." ;

```
tanmingrui@tanmingrui-virtual-machine:~/workspace$ sed '3cThis is the third line.' quote.txt
The honeysuckle band played all night long for only $90.It
was an evening of splendid music and company.Too bad the
This is the third line.
P.Neave was in attendance.
```

(6) 删除第 2 行内容

```
tanmingrui@tanmingrui-virtual-machine:~/workspace$ sed '2d' quote.txt
The honeysuckle band played all night long for only $90.It
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。

```
tanmingrui@tanmingrui-virtual-machine:~/workspace$ var='evening'
tanmingrui@tanmingrui-virtual-machine:~/workspace$ sed -n '/$var'/p' quote.txt
was an evening of splendid music and company.Too bad the
tanmingrui@tanmingrui-virtual-machine:~/workspace$
```

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

```
tanmingrui@tanmingrui-virtual-machine:~/workspace$ vi numbers.txt
tanmingrui@tanmingrui-virtual-machine:~/workspace$ awk -F " " '{print $2}' numbers.txt
:
:
tanmingrui@tanmingrui-virtual-machine:~/workspace$ 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。

例如：foo.txt 内容为：

2 4 3

15 46 79

则输出为：

even:

2

4

46

numbers:3

```

tanmingrui@tanmingrui-virtual-machine:~/workspace$ vi judge.sh
tanmingrui@tanmingrui-virtual-machine:~/workspace$ bash judge.sh
even
2
4
46
numbers:
3
tanmingrui@tanmingrui-virtual-machine:~/workspace$ cat judge.sh
#!/bin/bash

cnt=0
echo even
for j in $(cat foo.txt)
do
    if [ $((j%2)) -eq 0 ]
    then echo $j
        cnt=$((cnt+1))
    fi
done
echo numbers:
echo $cnt
tanmingrui@tanmingrui-virtual-machine:~/workspace$

```

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。

```

tanmingrui@tanmingrui-virtual-machine:~/workspace$ bash pattern.sh
enter search pattern: 1
985 211
985 211
9 8 5 2 1 1
9 8 5 2 1 1
4 found.
tanmingrui@tanmingrui-virtual-machine:~/workspace$ cat info.txt
985 211
985 211
9 8 5 2 1 1
9 8 5 2 1 1
tanmingrui@tanmingrui-virtual-machine:~/workspace$

```

#### 四、 实验过程分析与讨论

第四问脚本实现功能是匹配当前字符在文件中出现了多少行。Sed 和 awk 实现的功能是非常强大的，利用好了可以解决很多问题。

#### 五、指导教师意见

指导教师签字：卢洋