

实验报告

| | | | |
|------|-------------------|------|----------------|
| 实验名称 | 实验一 Linux 常用命令（一） | | |
| 实验教室 | 丹青 922 | 实验日期 | 2023 年 4 月 5 日 |
| 学 号 | 2021223147 | 姓 名 | 黄洋洋 |
| 专业班级 | 计算机科学与技术 05 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

```
[yang@yang ~]$ cd /etc
[yang@yang etc]$ pwd
/etc
```

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

```
[yang@yang etc]$ cd /home/yang
[yang@yang ~]$ pwd
/home/yang
[yang@yang ~]$ ls -a
.          .bash_profile  .config      .ICEauthority path2      公共      下载
..         .bashrc       .dbus        .local        path3      模板      音乐
bar.txt    baz           .esd_auth    .mozilla      qux.txt    视频      桌面
.bash_history  baz.txt      file2        package.tar   vfile      图片
.bash_logout .cache       file4        path1         .viminfo   文档
```

3. 使用命令创建目录/home/yang/linux，然后删除该目录。

```
[yang@yang ~]$ ls
bar.txt  baz.txt  file4      path1  path3  vfile  模板  图片  下载  桌面
baz      file2    package.tar path2  qux.txt 公共  视频  文档  音乐
[yang@yang ~]$ mkdir /home/yang/linux
[yang@yang ~]$ ls
bar.txt  baz.txt  file4  package.tar path2  qux.txt 公共  视频  文档  音乐
baz      file2    linux  path1      path3  vfile   模板  图片  下载  桌面
[yang@yang ~]$ pwd
/home/yang
[yang@yang ~]$ cd linux
[yang@yang linux]$ cd
[yang@yang ~]$ rmdir linux
[yang@yang ~]$ ls
bar.txt  baz.txt  file4      path1  path3  vfile  模板  图片  下载  桌面
baz      file2    package.tar path2  qux.txt 公共  视频  文档  音乐
[yang@yang ~]$ cd linux
bash: cd: linux: 没有那个文件或目录
```

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

```
[yang@yang ~]$ ls
bar.txt baz.txt file4 path1 path3 vfile 模板 图片 下载 桌面
baz file2 package.tar path2 qux.txt 公共 视频 文档 音乐
[yang@yang ~]$ cat > /home/yang/foo << EOF
> "hello,linux!"
> EOF
[yang@yang ~]$ LS
bash: LS: 未找到命令...
相似命令是: 'ls'
[yang@yang ~]$ ls
bar.txt baz.txt file4 package.tar path2 qux.txt 公共 视频 文档 音乐
baz file2 foo path1 path3 vfile 模板 图片 下载 桌面
[yang@yang ~]$ cat foo
"hello,linux!"
```

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

```
[yang@yang ~]$ ls
bar.txt baz.txt file4 package.tar path2 qux.txt 公共 视频 文档 音乐
baz file2 foo path1 path3 vfile 模板 图片 下载 桌面
[yang@yang ~]$ mkdir /home/yang/foo.bak
[yang@yang ~]$ ls
bar.txt file2 foo.bak path2 vfile 视频 下载
baz file4 package.tar path3 公共 图片 音乐
baz.txt foo path1 qux.txt 模板 文档 桌面
[yang@yang ~]$ cp /home/yang/foo /home/yang/foo.bak
[yang@yang ~]$ ls
bar.txt file2 foo.bak path2 vfile 视频 下载
baz file4 package.tar path3 公共 图片 音乐
baz.txt foo path1 qux.txt 模板 文档 桌面
[yang@yang ~]$ cd /home/yang/foo.bak
[yang@yang foo.bak]$ ls -a
. .. foo
[yang@yang ~]$ rm -rf /home/yang/foo.bak
[yang@yang ~]$ ls
bar.txt baz.txt file4 package.tar path2 qux.txt 公共 视频 文档 音乐
baz file2 foo path1 path3 vfile 模板 图片 下载 桌面
```

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。

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

huangyangyang@huangyangyang-virtual-machine:~$ tail -5 /etc/adduser.conf
# check user and group names also against this regular expression.
#NAME_REGEX="^[a-z][-a-z0-9_]*$"

# use extrausers by default
#USE_EXTRAUSERS=1
```

7、分屏查看文件/etc/adduser.conf 的内容。

```
# /etc/adduser.conf: `adduser' configuration.
# See adduser(8) and adduser.conf(5) for full documentation.

# 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

# If GROUPHOMES is "yes", then the home directories will be created as
# /home/groupname/user.
GROUPHOMES=no

# If LETTERHOMES is "yes", then the created home directories will have
# an extra directory - the first letter of the user name. For example:
# /home/u/user.
LETTERHOMES=no

# The SKEL variable specifies the directory containing "skeletal" user
# files; in other words, files such as a sample .profile that will be
# copied to the new user's home directory when it is created.
SKEL=/etc/skel

# FIRST_SYSTEM_[GU]ID to LAST_SYSTEM_[GU]ID inclusive is the range for UID
#s
# for dynamically allocated administrative and system accounts/groups.
# Please note that system software, such as the users allocated by the bas
# e-passwd
# package, may assume that UIDs less than 100 are unallocated.
FIRST_SYSTEM_UID=100
LAST_SYSTEM_UID=999

FIRST_SYSTEM_GID=100
LAST_SYSTEM_GID=999

# FIRST_[GU]ID to LAST_[GU]ID inclusive is the range of UIDs of dynamicall
# y
# allocated user accounts/groups.
FIRST_UID=1000
LAST_UID=59999

/etc/adduser.conf
```

```
huangyangyang@huangyangyang-virtual-machine:~$ less /etc/adduser.conf
huangyangyang@huangyangyang-virtual-machine:~$ █
```

8、使用命令cat用输出重定向在/home/huangyangyang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

```
huangyangyang@huangyangyang-virtual-machine:~$ cat > /home/huangyangyang/bar.txt << EOF
> google 110 5000
> baidu 100 5000
> guge 50 3000
> sodu 100 4500
> EOF
huangyangyang@huangyangyang-virtual-machine:~$ ls
ak      Desktop  Downloads Music     Public  Templates
bar.txt Documents fruits   Pictures snap    Videos
huangyangyang@huangyangyang-virtual-machine:~$
```

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

```
huangyangyang@huangyangyang-virtual-machine:~$ sort bar.txt -k 1,1
baidu 100 5000
google 110 5000
guge 50 3000
sodu 100 4500
```

(2) 按公司人数排序

```
huangyangyang@huangyangyang-virtual-machine:~$ sort bar.txt -k 2n,2
guge 50 3000
baidu 100 5000
sodu 100 4500
google 110 5000
```

(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

```
huangyangyang@huangyangyang-virtual-machine:~$ sort -n -t ' ' bar.txt -k 2,2 -k 3,3
guge 50 3000
sodu 100 4500
baidu 100 5000
google 110 5000
```

(4) 按员工工资降序排序，如工资相同，则按公司人数升序排序

```
huangyangyang@huangyangyang-virtual-machine:~$ sort -t ' ' bar.txt -k 3r,3 -k 2,2
baidu 100 5000
google 110 5000
sohu 100 4500
guge 50 3000
```

(5) 从公司英文名称的第2个字母开始进行排序。

```
huangyangyang@huangyangyang-virtual-machine:~$ sort -t ' ' bar.txt -k 1.2,1.2
baidu 100 5000
google 110 5000
sodu 100 4500
guge 50 3000
```

四、 实验过程分析与讨论

通过查阅相关资料，我了解到 sort 是默认【以第一个】条信息来排序，而且默认是以【文字】形式来排序，要用-n 来告知 sort 以数字来排序。若想恢复文件内容，可以一直按【u】恢复或者使用不保存退出【:q!】之后，再重新读取文件。

五、指导教师意见

指导教师签字：卢洋

实验报告

| | | | |
|------|-------------------|------|-----------------|
| 实验名称 | 实验二 Linux 常用命令（二） | | |
| 实验教室 | 丹青 922 | 实验日期 | 2023 年 4 月 12 日 |
| 学 号 | 2021223147 | 姓 名 | 黄洋洋 |
| 专业班级 | 计算机科学与技术 05 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 ；
2. 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
写任意几行内容

```
huangyangyang@huangyangyang-virtual-machine:~$ mkdir -p baz
huangyangyang@huangyangyang-virtual-machine:~$ cd baz
huangyangyang@huangyangyang-virtual-machine:~/baz$ cat > qux << EOF
> love
> so fun so funny
> EOF
```

- (2) 在用户目录下查找文件qux，并显示该文件位置信息

```
huangyangyang@huangyangyang-virtual-machine:~/baz$ cd
huangyangyang@huangyangyang-virtual-machine:~$ find -name qux
./baz/qux
```

- (3) 统计文件qux种所包含内容的行数，字数和字节数

```
huangyangyang@huangyangyang-virtual-machine:~$ cd baz
huangyangyang@huangyangyang-virtual-machine:~/baz$ wc qux
2  5 21 qux
```

(4) 在用户目录下查找文件qux，并删除该文件

```
huangyangyang@huangyangyang-virtual-machine:~$ find -name qux
./baz/qux
huangyangyang@huangyangyang-virtual-machine:~$ rm ./baz/qux
huangyangyang@huangyangyang-virtual-machine:~$ find -name qux
huangyangyang@huangyangyang-virtual-machine:~$
```

(5) 查看文件夹baz内容，看一下是否删除了文件qux

```
huangyangyang@huangyangyang-virtual-machine:~$ find baz
baz
huangyangyang@huangyangyang-virtual-machine:~$
```

```
lqzhou@LAPTOP-BOMFC06B:~$ find baz
baz
```

2. 文件打包

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

```
huangyangyang@huangyangyang-virtual-machine:~$ mkdir path1
huangyangyang@huangyangyang-virtual-machine:~$ cd path1
huangyangyang@huangyangyang-virtual-machine:~/path1$ touch file1 file2
huangyangyang@huangyangyang-virtual-machine:~/path1$ cd
huangyangyang@huangyangyang-virtual-machine:~$ find path1
path1
path1/file2
path1/file1
huangyangyang@huangyangyang-virtual-machine:~$
```

(2) 在用户目录下新建目录path2，在path2下新建文件file3

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

(3) 在用户目录下新建文件file4

```
huangyangyang@huangyangyang-virtual-machine:~$ touch file4
huangyangyang@huangyangyang-virtual-machine:~$ find file4
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

```

huangyangyang@huangyangyang-virtual-machine:~$ tar -cvf package.tar file4
path1
file4
path1/
path1/file2
path1/file1
huangyangyang@huangyangyang-virtual-machine:~$ ls
ak      Desktop  file4    package.tar  Pictures  Templates
bar.txt Documents fruits    path1        Public    Videos
baz     Downloads Music     path2        snap      workspace1
huangyangyang@huangyangyang-virtual-machine:~$

```

(5) 查看包package.tar的内容

```

huangyangyang@huangyangyang-virtual-machine:~$ tar -tvf package.tar
-rw-rw-r-- huangyangyang/huangyangyang 0 2023-05-23 09:48 file4
drwxrwxr-x huangyangyang/huangyangyang 0 2023-05-23 09:46 path1/
-rw-rw-r-- huangyangyang/huangyangyang 0 2023-05-23 09:46 path1/file2
-rw-rw-r-- huangyangyang/huangyangyang 0 2023-05-23 09:46 path1/file1

```

(6) 向包package.tar里添加文件夹path2的内容

```

huangyangyang@huangyangyang-virtual-machine:~$ tar -rvf package.tar path2
path2/
path2/file3
huangyangyang@huangyangyang-virtual-machine:~$ tar -tvf package.tar
-rw-rw-r-- huangyangyang/huangyangyang 0 2023-05-23 09:48 file4
drwxrwxr-x huangyangyang/huangyangyang 0 2023-05-23 09:46 path1/
-rw-rw-r-- huangyangyang/huangyangyang 0 2023-05-23 09:46 path1/file2
-rw-rw-r-- huangyangyang/huangyangyang 0 2023-05-23 09:46 path1/file1
drwxrwxr-x huangyangyang/huangyangyang 0 2023-05-23 09:47 path2/
-rw-rw-r-- huangyangyang/huangyangyang 0 2023-05-23 09:47 path2/file3
huangyangyang@huangyangyang-virtual-machine:~$

```

(7) 将包package.tar复制到用户目录下的新建文件夹path3中

```

huangyangyang@huangyangyang-virtual-machine:~$ mkdir path3
huangyangyang@huangyangyang-virtual-machine:~$ cp package.tar ./path3
huangyangyang@huangyangyang-virtual-machine:~$ find path3
path3
path3/package.tar

```

(8) 进入path3文件夹，并还原包package.tar的内容

```

huangyangyang@huangyangyang-virtual-machine:~$ cd path3
huangyangyang@huangyangyang-virtual-machine:~/path3$ tar -xvf package.tar
file4
path1/
path1/file2
path1/file1
path2/
path2/file3

```



```

lqzhou@LAPTOP-BOMFC06B:~$ cd path3
lqzhou@LAPTOP-BOMFC06B:~/path3$ tar -xvf package.tar
path1/
path1/file1
path1/file2
file4
path2/
path2/file3
lqzhou@LAPTOP-BOMFC06B:~/path3$ ls
file4  package.tar  path1  path2

```

3. 符号链接内容

- (1) 新建文件foo.txt, 内容为123

```

huangyangyang@huangyangyang-virtual-machine:~$ cat > foo.txt << EOF
> 123
> EOF
huangyangyang@huangyangyang-virtual-machine:~$ cat foo.txt
123
huangyangyang@huangyangyang-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

```

huangyangyang@huangyangyang-virtual-machine:~$ ln foo.txt bar.txt
huangyangyang@huangyangyang-virtual-machine:~$ comm foo.txt bar.txt
123
huangyangyang@huangyangyang-virtual-machine:~$ diff foo.txt bar.txt
huangyangyang@huangyangyang-virtual-machine:~$

```

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

```

huangyangyang@huangyangyang-virtual-machine:~$ ls -li foo.txt bar.txt
266899 bar.txt 266899 foo.txt

```

- (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

```

huangyangyang@huangyangyang-virtual-machine:~$ vi bar.txt
huangyangyang@huangyangyang-virtual-machine:~$ comm foo.txt bar.txt
abc

```

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

```

huangyangyang@huangyangyang-virtual-machine:~$ rm foo.txt
huangyangyang@huangyangyang-virtual-machine:~$ cat bar.txt
abc
huangyangyang@huangyangyang-virtual-machine:~$ ls -li bar.txt
266899 bar.txt

```

- (6) 创建文件bar.txt的符号链接文件baz.txt，然后查看bar.txt和bat.txt的inode号，并观察是否相同，比较bar.txt和baz.txt的文件内容是否相同

```
huangyangyang@huangyangyang-virtual-machine:~$ ln -s bar.txt baz.txt
huangyangyang@huangyangyang-virtual-machine:~$ comm bar.txt baz.txt
abc
huangyangyang@huangyangyang-virtual-machine:~$ diff bar.txt baz.txt
huangyangyang@huangyangyang-virtual-machine:~$
huangyangyang@huangyangyang-virtual-machine:~$ ls -li bar.txt baz.txt
266899 bar.txt 266896 baz.txt
huangyangyang@huangyangyang-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

```
huangyangyang@huangyangyang-virtual-machine:~$ rm bar.txt
huangyangyang@huangyangyang-virtual-machine:~$ cat baz.txt
cat: baz.txt: No such file or directory
```

4. 权限管理

- (1) 新建文件qux.txt

```
huangyangyang@huangyangyang-virtual-machine:~$ touch qux.txt
huangyangyang@huangyangyang-virtual-machine:~$ ls -al qux.txt
-rw-rw-r-- 1 huangyangyang huangyangyang 0 5月 23 10:39 qux.txt
```

- (2) 为文件qux.txt增加执行权限(所有用户都可以执行)

```
huangyangyang@huangyangyang-virtual-machine:~$ chmod +x qux.txt
huangyangyang@huangyangyang-virtual-machine:~$ ls -al qux.txt
-rwxrwxr-x 1 huangyangyang huangyangyang 0 5月 23 10:39 qux.txt
```


四、 实验过程分析与讨论

不允许对目录用 `ln` 命令建立硬链接，分清在什么目录下进行操作。`-cvf` 是打包命令，`-xvf` 是解包命令。

五、指导教师意见

指导教师签字：卢洋

实验报告

| | | | |
|------|-------------------------|------|-----------------|
| 实验名称 | 实验三 vim 编辑器及 gcc 编译器的使用 | | |
| 实验教室 | 丹青 922 | 实验日期 | 2023 年 4 月 19 日 |
| 学 号 | 2021223147 | 姓 名 | 黄洋洋 |
| 专业班级 | 计算机科学与技术 05 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 ；

```
huangyangyang@huangyangyang-virtual-machine:~$ mkdir workspace1
```

(2) 进入目录 workspace1 ；

```
huangyangyang@huangyangyang-virtual-machine:~$ cd workspace1
huangyangyang@huangyangyang-virtual-machine:~/workspace1$ pwd
/home/huangyangyang/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；

```
: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。

```
[yang@yang workspace1]$ gcc test.c -o test
[yang@yang 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；

```
[yang@yang ~]$ find workspace2
workspace2
```

(2) 进入 workspace2 目录；

```
[yang@yang ~]$ cd workspace2
[yang@yang 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；

```
[yang@yang workspace2]$ cp /etc/gai.conf ./gai.conf
```

```
[yang@yang workspace2]$ ls
gai.conf
```

(4) 使用 vim 编辑当前目录下的 gai.conf ；

```
#fyikb hjkgiugt yreutio hyion jhtr6 fr675r
#13 26 689786 807 253 123
#
#ej gh yui yiu congiguration thr jdehf
#fdifr All default up tovalues
#
#appropriate default information to be used.
#so far only configuration for the destination address sort
#
#
#
#
#214 675 3425 68 8679 3412 41 86 79 13 4 u867 89 7 21 f e5 y f
#000 879 7 54 23 4 dx vdfbh g j fad fr sa dsb
#srh er he th b eqr we ghyry j yt betj r t ad f g
#fgery tr hri t bhtr i dw q yuk
#f wrty v rhju rt ngrj i t trj
#ncjkshgtjwei klehrin jaerhuoweh kljropqw klaehytwi
~
~
~
~
~
```

(5) 将光标移到第 18 行;

用 ngg 此时 n=18

```
18,1
```

(6) 复制该行内容;

用 yy 复制行

(7) 将光标移到最后一行行首;

用 G 直接跳到末行行首

(8) 粘贴复制行的内容;

用 p 命令即可粘贴

```
#fgery tr hri t bhtr i dw q yuk
#f wrty v rhju rt ngrj i t trj
#ncjkshgtjwei klehrin jaerhuoweh kljropqw klaehytwi
~
#f wrty v rhju rt ngrj i t trj
~
```

(9) 撤销第 8 步的动作;

```
~
~
1 行被去掉 ; before #1 63 seconds ago
```

(10) 存盘但不退出;

用:w 即可

```
||: w
```

(11) 将光标移到首行;

用 gg 移到首行

```
||#13 26 689786 807 253 123
```

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

```
||#"hello,this is vim world!"  
||#13 26 689786 807 253 123  
||#
```

(13) 删除字符串 "this";

可以用 x 删除或者在插入模式下删除。

```
||#"hello, is vim world!"  
||#13 26 689786 807 253 123
```

(14) 强制退出 vim , 不存盘。

```
||: q!
```


四、 实验过程分析与讨论

vim 的用法比较多，需要练习积累才能熟练使用。

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 26 日 |
| 学 号 | 2021223147 | 姓 名 | 黄洋洋 |
| 专业班级 | 计算机科学与技术 05 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@yang ~]# useradd -c 'bar' -s /bin/sh -d /home/foo -m foo
[root@yang ~]# passwd foo
更改用户 foo 的密码 。
新的 密码：
```

```
新的 密码：
重新输入新的 密码：
passwd：所有的身份验证令牌已经成功更新。
[root@yang ~]# █
```

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/sh ；

```
[root@yang ~]# usermod foo -u 2000 -s /bin/sh
[root@yang ~]# su foo
sh-4.2$ cat /etc/passwd | tail -n 3
tcpdump: x:72:72:::/sbin/nologin
yang: x:1000:1000:yang:/home/yang:/bin/bash
foo: x:2000:1001:bar:/home/foo:/bin/sh
sh-4.2$
```

3. 从用户 foo 切换到 root ；

```
- sh-4.2$ su
密码：
[root@yang foo]# cd
[root@yang ~]# █
```

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@yang ~]# userdel -r foo
[root@yang ~]# cat /etc/passwd | tail -n 3
postfix: x:89:89::/var/spool/postfix:/sbin/nologin
tcpdump: x:72:72:::/sbin/nologin
yang: x:1000:1000:yang:/home/yang:/bin/bash
[root@yang ~]#
```

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@yang ~] # vi userfile
```

内容如下

```
user1: x: 2001: 2001: user1: /home/user1: /bin/bash
user2: x: 2002: 2002: user2: /home/user2: /bin/bash
user3: x: 2003: 2003: user3: /home/user3: /bin/bash
```

```
[[ root@yang ~] # vi passwdfile
```

内容如下

```
user1:156450an
user2:156450an
user3:156450a
```

```
[[ root@yang ~] # newusers userfile
[[ root@yang ~] # chpasswd < passwdfile
[[ root@yang ~] # cat /etc/passwd | tail -n 3
user1: x: 2001: 2001: user1: /home/user1: /bin/bash
user2: x: 2002: 2002: user2: /home/user2: /bin/bash
user3: x: 2003: 2003: user3: /home/user3: /bin/bash
[[ root@yang ~] #
```

先建立一个 user.txt 保存要创建的用户

```
[[ root@yang ~] # touch user.txt
[[ root@yang ~] # newusers < user.txt
[[ root@yang ~] # pwunconv
[[ root@yang ~] # cat > passwd.txt << EOF
> user1:156450an
> user2:156450an
> user3:156450an
> EOF
```

```
[[ root@yang ~] # chpasswd < passwd.txt
[[ root@yang ~] # pwconv
[[ root@yang ~] # cat /etc/passwd | tail -n 3
user1: x: 2001: 2001: user1: /home/user1: /bin/bash
user2: x: 2002: 2002: user2: /home/user2: /bin/bash
user3: x: 2003: 2003: user3: /home/user3: /bin/bash
[[ root@yang ~] #
```

全都建立完毕，并且密码隐藏为 x

6. 创建用户组 group1，并在创建时设置其 GID 为 3000；

```
[[ root@yang ~] # groupadd group1 -g 3000
```

7. 在用户组 group1 中添加两个之前批量创建的用户；

```
[root@yang ~]# gpasswd -a user1 group1
正在将用户 "user1"加入到 "group1"组中
[root@yang ~]# gpasswd -a user2 group1
正在将用户 "user2"加入到 "group1"组中
[root@yang ~]# su user1
bash-4.2$ sudo cat /etc/shadow | tail -n 3
```

我们信任您已经从系统管理员那里了解了日常注意事项。
总结起来无外乎这三点：

- #1) 尊重别人的隐私。
- #2) 输入前要先考虑(后果和风险)。
- #3) 权力越大，责任越大。

[sudo] user1 的密码：

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。

'user1' 不在 sudoers 文件中。此事将被报告。
您在 /var/spool/mail/root 中有新邮件

解决办法如下：

```
[root@yang ~]# gpasswd -a user2 wheel
正在将用户 "user2"加入到 "wheel"组中
[root@yang ~]# su user2
bash-4.2$ sudo cat /etc/shadow | tail -n 3
```

我们信任您已经从系统管理员那里了解了日常注意事项。
总结起来无外乎这三点：

- #1) 尊重别人的隐私。
- #2) 输入前要先考虑(后果和风险)。
- #3) 权力越大，责任越大。

[sudo] user2 的密码：

```
user1: $6$0YQ/b1QNMA$KIk1E9WDvkv6lAnQagr9woFr2fmZ2Sx7LY6k61Ko2CGfz0mxswPGdQmoZzFw1
HDu5t53It9dMVnFb5ZD120K0:19500:0:99999:7:::
user2: $6$dE7xb/dP0R$RCKbcyHl9ZYA3mJQ/W4gxUx.bs10CNio/8BzR015xVoDSgEtF.eHbU6sw8N91
JUkbGpI.WtKw7r93GJk5Jr6.:19500:0:99999:7:::
user3: $6$UHjTC8RWoC/$3Xthr1LpJYOK00fGo70AJ1u/z/K2kDrzFMv8hBMI/eemgNEZTTggIZfkRosdD
veg/KwITvvS4jqbpS5xFY7KZl:19500:0:99999:7:::
bash-4.2$ █
```


四、 实验过程分析与讨论

用户权限部分知识还需巩固学习,对于如何运行命令的办法是将所有用户加上 `sudo`.

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 27 日 |
| 学 号 | 2021223147 | 姓 名 | 黄洋洋 |
| 专业班级 | 计算机科学与技术 05 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@yang ~]# foo=200
[root@yang ~]# 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@yang ~]# bar=100
[root@yang ~]# test $bar -gt 150
[root@yang ~]# 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 ）；

```
#!/bin/bash
x=$(hostname)
y=$(date)
echo $x
echo $y
~
~
```

```
[root@yang ~]# ./4.sh
yang
2023年 05月 23日 星期二 22:53:14 CST
```

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

```
#!/bin/bash
read -p "Input number: " num
num1=`expr $num % 10`
#echo $num1
num2=`expr $num / 10 % 10`
#echo $num2
num3=`expr $num / 100 % 10`
#echo $num3
num13=`expr $num1 \* $num1 \* $num1`
num23=`expr $num2 \* $num2 \* $num2`
num33=`expr $num3 \* $num3 \* $num3`
num4=`expr $num13 + $num23 + $num33`
if (($num4 == $num))
then
    echo "true"
else
    echo "false"
fi
~
[root@yang ~]# vim 5.sh
[root@yang ~]# ./5.sh
Input number:153
true
[root@yang ~]# ./5.sh
Input number:124
false
```

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@yang ~]# ./6.sh 1 2 3
sum= 6
```

```
#!/bin/bash
echo 'sum= `expr $1 + $2 + $3`'
~
```

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 。

```
[root@yang ~]# ./7.sh
grade:78
level is c
```

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

四、 实验过程分析与讨论

对于格式要求很严格，少一个空格也不行。。所以一定要仔细仔细再仔细

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 3 日 |
| 学 号 | 2021223147 | 姓 名 | 黄洋洋 |
| 专业班级 | 计算机科学与技术 05 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）；

```
[root@yang ~]# cat l1.sh
#!/bin/bash
for file in `find *.c`
do
    mv $file ./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；

```
[root@yang ~]# vim l2.sh
[root@yang ~]# bash ./l2.sh
110
[root@yang ~]# cat l2.sh
#!/bin/bash
count=0
sum=0
num=2
while (($count<10))
do
    sum=`expr $sum + $num`
    num=`expr $num + 2`
    let "count++"
done
echo $sum
[root@yang ~]#
```

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@yang ~]# bash ./l3.sh
385
[root@yang ~]# cat l3.sh
#!/bin/bash
count=1
sum=0
until (($count == 11))
do
    sum=`expr $sum + $count \* $count`
    let "count++"
done
echo $sum
[root@yang ~]#
```

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
```

break 时

```
[root@yang ~]# bash l4.sh
a1234
b1234
c1234
d1234
```

break 2 时

```
[root@yang ~]# bash l4.sh
a1234[ root@yang ~]#
```

continue

```
[root@yang ~]# bash l4.sh
a1234678910
b1234678910
c1234678910
d1234678910
```

continue 2

```
[root@yang ~]# bash l4.sh
a1234b1234c1234d1234[ root@yang ~]#
```

四、 实验过程分析与讨论

领悟了各种指令的意义，实践更加掌握 shell 的循环语句和循环控制语句。

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 10 日 |
| 学 号 | 2021223147 | 姓 名 | 黄洋洋 |
| 专业班级 | 计算机科学与技术 05 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;

```
[root@yang ~]# bash l5.sh
input number1:1234
input number2:654
sum is 1888
```

```
#!/bin/bash
function atwonum()
{
    s=`expr $1 + $2`
    echo "sum is $s"
}
read -p "input number1:" num1
read -p "input number2:" num2
atwonum $num1 $num2
~
~
~
~
```

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;

我看错了,以为是斐波那契,然后一直犹豫怎么返回参数呢,因为 return 只能返回 0-255,之后才了解到 echo 可以返回到调用的位置。

```
[root@yang ~]# bash l6.sh
number: 6
720
```

```
#!/bin/bash
fact(){
if [ $1 -eq 1 ];then
    echo "1"
elif [ $1 -eq 0 ];then
    echo "0"
else
t=$(( $1-1 ))
result=$(fact $t)
echo "$1*$result"
fi
}
read -p "number: " num
result=$(fact $num)
echo "$result"
~
~
~
~
```

3. 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
```

```
[root@yang ~]# bash l7.sh
starting...
-1- here is in the first func.
-2- here is in the second func.
-3- here is in the third func.
[root@yang ~]# █
```

bash 中的函数必须在其第一次调用之前被完成。没有如 c 语言

中那种函数声明的形式。即使提前定义了函数的变量也不可以。但是只要执行顺序中，函数的定义在函数的使用之前即可。可以看出，其实函数的定义其实只是一种命令。而 bash 脚本是一种线性执行的语言。因此只要保证函数使用的流程正确，就可以将函数应用在各种可能的地方。

四、 实验过程分析与讨论

通过本次实验，基本掌握了 shell 函数的定义方法、参数传递、调用和返回值，基本掌握了 shell 函数的递归调用方法。体会到了 shell 函数的嵌套使用。

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 17 日 |
| 学 号 | 2021223147 | 姓 名 | 黄洋洋 |
| 专业班级 | 计算机科学与技术 05 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：

```
[root@yang ~]# cat > quote.txt << EOF  
> 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.  
> EOF
```

(1) 删除 \$ 符号；

```
[root@yang ~]# cat quote.txt | sed 's/\$/g'  
The honeysuckle band played all night long for only 0.  
It was an evening of splendid music and company.  
Too bad the disco floor fell through at 23:10.  
The local nurse Miss P.Neave was in attendance.
```

(2) 显示包含 music 文字的行内容及行号；

```
[root@yang ~]# 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!”；

```
[root@yang ~]# sed -e 4a'hello world!' quote.txt  
The honeysuckle band played all night long for only 0.  
It was an evening of splendid music and company.  
Too bad the disco floor fell through at 23:10.  
The local nurse Miss P.Neave was in attendance.  
hello world!
```

(4) 将文本 “The” 替换为 “Quod”；

```
[root@yang ~]# sed 's/The/Quod/g' quote.txt  
Quod honeysuckle band played all night long for only 0.  
It was an evening of splendid music and company.  
Too bad the disco floor fell through at 23:10.  
Quod local nurse Miss P.Neave was in attendance.
```

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

```
[root@yang ~]# sed '3c This is the third line\.' quote.txt
The honeysuckle band played all night long for only 0.
It was an evening of splendid music and company.
This is the third line.
The lacial nurse Miss P. Neave was in attendance.
```

```
[root@yang ~]# nl quote.txt | sed '3c This is the third line\.'
1 The honeysuckle band played all night long for only 0.
2 It was an evening of splendid music and company.
This is the third line.
4 The lacial nurse Miss P. Neave was in attendance.
```

(6) 删除第 2 行内容;

```
[root@yang ~]# nl quote.txt | sed '2d'
1 The honeysuckle band played all night long for only 0.
3 Too bad the disco floor fell through at 23:10.
4 The lacial 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。

```
[root@yang ~]# var=evening
[root@yang ~]# echo $var
evening
[root@yang ~]# set | nl - | sed -n '/evening/p'
101 _=evening
122 var=evening
```

```
[root@yang ~]# cat > numbers.txt << EOF
> one : two : three
> four : five : six
> EOF
```

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@yang ~]# awk -F ' ' '{print $2}' numbers.txt
:
:
```

```
[root@yang ~]# awk -F ':' '{printf $2}' numbers.txt
two five [root@yang ~]#
```


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。

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

```
[root@yang ~]# vim l9.sh
[root@yang ~]# bash l9.sh
input filename: foo.txt
even:
8
6
6
8
92
numbers:
5
```

4. 脚本的内容如下所示：

:

```
#!/bin/bash

read -p "enter search pattern: " pattern

awk "/$pattern/" '{ nmatches++; print } END { print nmatches, "found." }' info.txt
```

试运行该脚本，并理解该脚本实现的功能。

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

该脚本对 info.txt 查找 pattern 只要该行存在 pattern 就输出该行且计数，在最后输出总数。如下所示：

```
[[yang@yang ~]$ bash l10.sh
enter search pattern: test
awk: fatal: cannot open file `info.txt' for reading (没有那个文件或目录)
```

四、 实验过程分析与讨论

基本掌握了 sed 基本编辑命令的使用方法，基本掌握了 sed 与 shell 变量的交互方法，awk 命令的使用方法等。

五、指导教师意见

指导教师签字：卢洋