

# 实验报告

实验名称	实验一 Linux 常用命令（一）		
实验教室	丹青 922	实验日期	2023 年 3 月 8 日
学 号	2021223123	姓 名	于心洁
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

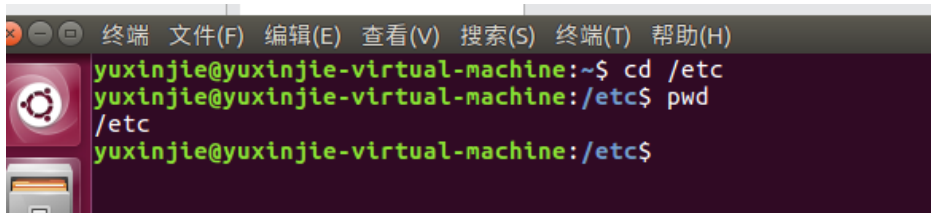
- 1、掌握Linux下文件和目录操作命令：cd、ls、mkdir、rmdir、rm
- 2、掌握Linux下文件信息显示命令：cat、more、head、tail
- 3、掌握Linux下文件复制、删除及移动命令：cp、mv
- 4、掌握 Linux 的文件排序命令：sort

## 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。

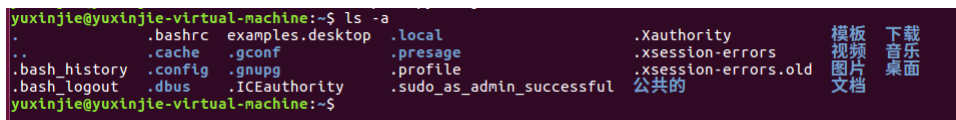
### 三、 实验内容及结果

#### 1. 使用命令切换到/etc 目录，并显示当前工作目录路径



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



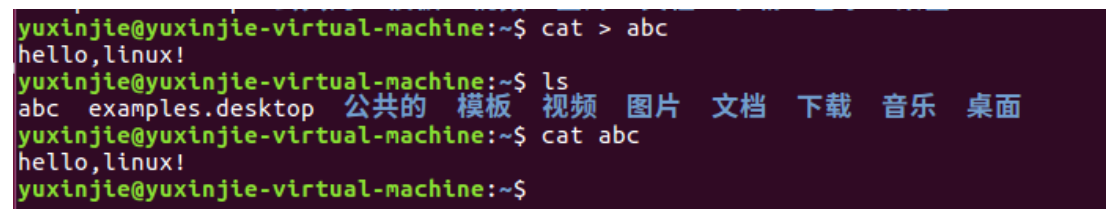
```
yuxinjie@yuxinjie-virtual-machine:~$ ls -la
.          .bashrc  examples.desktop  .local          .Xauthority      模板      下载
..         .cache   .gconf            .presage        .xsession-errors  视频      音乐
.bash_history .config  .gnupg            .profile        .xsession-errors.old 图片      桌面
.bash_logout .dbus    .ICEauthority     .sudo_as_admin_successful 公共的     文档
yuxinjie@yuxinjie-virtual-machine:~$
```

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



```
yuxinjie@yuxinjie-virtual-machine:~$ mkdir /home/yuxinjie/linux
yuxinjie@yuxinjie-virtual-machine:~$ ls
examples.desktop  linux  公共的  模板  视频  图片  文档  下载  音乐  桌面
yuxinjie@yuxinjie-virtual-machine:~$ rm linux
rm: 无法删除'linux': 是一个目录
yuxinjie@yuxinjie-virtual-machine:~$ rmdir linux
yuxinjie@yuxinjie-virtual-machine:~$ ls
examples.desktop  公共的  模板  视频  图片  文档  下载  音乐  桌面
yuxinjie@yuxinjie-virtual-machine:~$
```

#### 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



```
yuxinjie@yuxinjie-virtual-machine:~$ cat > abc
hello,linux!
yuxinjie@yuxinjie-virtual-machine:~$ ls
abc  examples.desktop  公共的  模板  视频  图片  文档  下载  音乐  桌面
yuxinjie@yuxinjie-virtual-machine:~$ cat abc
hello,linux!
yuxinjie@yuxinjie-virtual-machine:~$
```

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

6、查看文件 `/etc/adduser.conf` 的前 3 行内容，查看文件 `/etc/adduser.conf` 的最后 5 行内容。

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

yuxinjie@yuxinjie-virtual-machine:/etc$ tail -n 5 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
yuxinjie@yuxinjie-virtual-machine:/etc$
```

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 UIDs
# for dynamically allocated administrative and system accounts/groups.
# Please note that system software, such as the users allocated by the base-pass
# 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 dynamically
# allocated user accounts/groups.
FIRST_UID=1000
LAST_UID=29999

FIRST_GID=1000
LAST_GID=29999

# The USERGROUPS variable can be either "yes" or "no". If "yes" each
# created user will be given their own group to use as a default. If
# "no", each created user will be placed in the group whose gid is
# USERS_GID (see below).
USERGROUPS=yes

# If USERGROUPS is "no", then USERS_GID should be the GID of the group
# 'users' (or the equivalent group) on your system.
USERS_GID=100

# If DIR_MODE is set, directories will be created with the specified
# mode. Otherwise the default mode 0755 will be used.
DIR_MODE=0755

# If SETGID_HOME is "yes" home directories for users with their own
# group the setgid bit will be set. This was the default for
# versions <= 3.13 of adduser. Because it has some bad side effects we
# no longer do this per default. If you want it nevertheless you can
# still set it here.
SETGID_HOME=no

# If QUOTAUSER is set, a default quota will be set from that user with
# 'edquota -p QUOTAUSER newuser'
QUOTAUSER=""

# If SKEL_IGNORE_REGEX is set, adduser will ignore files matching this
# regular expression when creating a new home directory
SKEL_IGNORE_REGEX="dpkg-(old|new|dist|save)"

# Set this if you want the --add_extra_groups option to adduser to add
# new users to other groups.
# This is the list of groups that new non-system users will be added to
# Default:
#EXTRA_GROUPS="dialout cdrom floppy audio video plugdev users"

# If ADD_EXTRA_GROUPS is set to something non-zero, the EXTRA_GROUPS
# option above will be default behavior for adding new, non-system users
#ADD_EXTRA_GROUPS=1

# check user and group names also against this regular expression.
--更多--(97%)

```

## 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

```
yuxinjie@yuxinjie-virtual-machine:/$ sudo sh -c "sudo cat > facebook.txt"
[sudo] yuxinjie 的密码:
google 110 5000
baidu 100 5000
guge 50 3000
sohu 100 4500
yuxinjie@yuxinjie-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
- (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。

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

#### 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 3 月 15 日
学 号	2021223123	姓 名	于心洁
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;

```
yuxinjie@yuxinjie-virtual-machine:~$ mkdir baz
yuxinjie@yuxinjie-virtual-machine:~$ cd baz
yuxinjie@yuxinjie-virtual-machine:~/baz$ cat > qux
hi
a
bb
ccc
^Z
```

- (2) 在用户目录下查找文件 qux, 并显示该文件位置信息;

```
yuxinjie@yuxinjie-virtual-machine:~/baz$ find ~ -name qux
/home/yuxinjie/baz/qux
```

- (3) 统计文件 qux 中所包含内容的行数、字数和字节数;

```
yuxinjie@yuxinjie-virtual-machine:~/baz$ wc qux
4  4 12 qux
```

- (4) 在用户目录下查找文件 qux, 并删除该文件;

```
yuxinjie@yuxinjie-virtual-machine:~/baz$ find ~ -name qux -delete
yuxinjie@yuxinjie-virtual-machine:~/baz$ ll
总用量 8
drwxrwxr-x  2 yuxinjie yuxinjie 4096 5月  22 19:05 ./
drwxr-xr-x 18 yuxinjie yuxinjie 4096 5月  22 19:02 ../
```

(5) 查看文件夹 baz 内容，看一下是否删除了文件 qux。

```
yuxinjie@yuxinjie-virtual-machine:~/baz$ find ~ -name qux -delete
yuxinjie@yuxinjie-virtual-machine:~/baz$ ll
总用量 8
drwxrwxr-x 2 yuxinjie yuxinjie 4096 5月 22 19:05 ./
drwxr-xr-x 18 yuxinjie yuxinjie 4096 5月 22 19:02 ../
```

## 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;

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

```
总用量 8
drwxrwxr-x 2 yuxinjie yuxinjie 4096 5月 22 19:07 ./
drwxr-xr-x 19 yuxinjie yuxinjie 4096 5月 22 19:06 ../
-rw-rw-r-- 1 yuxinjie yuxinjie 0 5月 22 19:07 file1
-rw-rw-r-- 1 yuxinjie yuxinjie 0 5月 22 19:07 file2
yuxinjie@yuxinjie-virtual-machine:~/path1$ cd ~
```

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

```
yuxinjie@yuxinjie-virtual-machine:~$ mkdir path2
yuxinjie@yuxinjie-virtual-machine:~$ cd path2
yuxinjie@yuxinjie-virtual-machine:~/path2$ touch file3
yuxinjie@yuxinjie-virtual-machine:~/path2$ ll
总用量 8
drwxrwxr-x 2 yuxinjie yuxinjie 4096 5月 22 19:08 ./
drwxr-xr-x 20 yuxinjie yuxinjie 4096 5月 22 19:08 ../
-rw-rw-r-- 1 yuxinjie yuxinjie 0 5月 22 19:08 file3
```

(3) 在用户目录下新建文件 file4;

```
yuxinjie@yuxinjie-virtual-machine:~/path2$ cd ~
yuxinjie@yuxinjie-virtual-machine:~$ touch file4
yuxinjie@yuxinjie-virtual-machine:~$ ll
总用量 128
drwxr-xr-x 20 yuxinjie yuxinjie 4096 5月 22 19:09 ./
drwxr-xr-x 3 root root 4096 5月 7 20:30 ../
-rw-rw-r-- 1 yuxinjie yuxinjie 13 5月 12 19:56 abc
-rw-rw-r-- 1 yuxinjie yuxinjie 52 5月 7 21:21 .bash_history
-rw-rw-r-- 1 yuxinjie yuxinjie 220 5月 7 20:30 .bash_logout
-rw-rw-r-- 1 yuxinjie yuxinjie 3771 5月 7 20:30 .bashrc
drwxrwxr-x 2 yuxinjie yuxinjie 4096 5月 22 19:05 baz/
drwxr-xr-x 12 yuxinjie yuxinjie 4096 5月 7 21:08 .cache/
drwxr-xr-x 18 yuxinjie yuxinjie 4096 5月 7 21:05 .config/
drwxr-xr-x 3 yuxinjie yuxinjie 4096 5月 7 20:51 .dbus/
-rw-rw-r-- 1 yuxinjie yuxinjie 8980 5月 7 20:30 examples.desktop
-rw-rw-r-- 1 yuxinjie yuxinjie 0 5月 22 19:09 file4
drwxr-xr-x 2 yuxinjie yuxinjie 4096 5月 12 19:54 .gconf/
drwxr-xr-x 3 yuxinjie yuxinjie 4096 5月 12 19:39 .gnupg/
-rw-rw-r-- 1 yuxinjie yuxinjie 1170 5月 12 19:39 .ICEauthority
drwxr-xr-x 3 yuxinjie yuxinjie 4096 5月 7 20:51 .local/
drwxrwxr-x 2 yuxinjie yuxinjie 4096 5月 22 19:07 path1/
drwxrwxr-x 2 yuxinjie yuxinjie 4096 5月 22 19:08 path2/
drwxr-xr-x 2 yuxinjie yuxinjie 4096 5月 7 20:51 .presage/
-rw-rw-r-- 1 yuxinjie yuxinjie 655 5月 7 20:30 .profile
-rw-rw-r-- 1 yuxinjie yuxinjie 0 5月 7 21:20 .sudo_as_admin_successful
-rw-rw-r-- 1 yuxinjie yuxinjie 69 5月 12 19:39 .Xauthority
-rw-rw-r-- 1 yuxinjie yuxinjie 84 5月 12 19:39 .xsession-errors
-rw-rw-r-- 1 yuxinjie yuxinjie 1460 5月 7 21:45 .xsession-errors.old
drwxr-xr-x 2 yuxinjie yuxinjie 4096 5月 7 20:51 公共的/
drwxr-xr-x 2 yuxinjie yuxinjie 4096 5月 7 20:51 模板/
drwxr-xr-x 2 yuxinjie yuxinjie 4096 5月 7 20:51 视频/
drwxr-xr-x 2 yuxinjie yuxinjie 4096 5月 7 20:51 图片/
drwxr-xr-x 2 yuxinjie yuxinjie 4096 5月 7 20:51 文档/
drwxr-xr-x 3 yuxinjie yuxinjie 4096 5月 7 21:29 下载/
drwxr-xr-x 2 yuxinjie yuxinjie 4096 5月 7 20:51 音乐/
drwxr-xr-x 2 yuxinjie yuxinjie 4096 5月 7 20:51 桌面/
```

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

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

(5) 查看包 package.tar 的内容;

```
yuxinjie@yuxinjie-virtual-machine:~$ tar -tvf package.tar
drwxrwxr-x yuxinjie/yuxinjie 0 2023-05-22 19:07 path1/
-rw-rw-r-- yuxinjie/yuxinjie 0 2023-05-22 19:07 path1/file1
-rw-rw-r-- yuxinjie/yuxinjie 0 2023-05-22 19:07 path1/file2
-rw-rw-r-- yuxinjie/yuxinjie 0 2023-05-22 19:09 file4
```

(6) 向包 package.tar 里添加文件夹 path2 的内容;

```
yuxinjie@yuxinjie-virtual-machine:~$ tar -cvf package.tar path2
path2/
path2/file3
```

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

```
yuxinjie@yuxinjie-virtual-machine:~$ mkdir path3
yuxinjie@yuxinjie-virtual-machine:~$ cp package.tar path3
```

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

```
yuxinjie@yuxinjie-virtual-machine:~$ cd path3
yuxinjie@yuxinjie-virtual-machine:~/path3$ tar -xvf package.tar
path2/
path2/file3
```

### 3. 符号链接内容

(1) 新建文件 foo.txt, 内容为 123;

```
yuxinjie@yuxinjie-virtual-machine:~/path3$ cd ~
yuxinjie@yuxinjie-virtual-machine:~$ echo "123" > foo.txt
```

(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;

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

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

```
yuxinjie@yuxinjie-virtual-machine:~$ ll -li foo.txt bar.txt
418042 -rw-rw-r-- 2 yuxinjie yuxinjie 4 5月 22 19:13 bar.txt
418042 -rw-rw-r-- 2 yuxinjie yuxinjie 4 5月 22 19:13 foo.txt
yuxinjie@yuxinjie-virtual-machine:~$ echo "abc" > bar.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;

```
yuxinjie@yuxinjie-virtual-machine:~$ echo "abc" > bar.txt
yuxinjie@yuxinjie-virtual-machine:~$ diff foo.txt bar.txt
yuxinjie@yuxinjie-virtual-machine:~$ cp foo.txt
```

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

```
yuxinjie@yuxinjie-virtual-machine:~$ rm foo.txt
yuxinjie@yuxinjie-virtual-machine:~$ ll -i bar.txt
418042 -rw-rw-r-- 1 yuxinjie yuxinjie 4 5月 22 19:14 bar.txt
yuxinjie@yuxinjie-virtual-machine:~$ 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；

```
yuxinjie@yuxinjie-virtual-machine:~$ ln -s bar.txt baz.txt
yuxinjie@yuxinjie-virtual-machine:~$ ls -li bar.txt baz.txt
418042 bar.txt 418043 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。

```
yuxinjie@yuxinjie-virtual-machine:~$ rm bar.txt
yuxinjie@yuxinjie-virtual-machine:~$ ls -li 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；

(2) 为文件 qux.txt 增加执行权限(所有用户都可以执行)。

```
yuxinjie@yuxinjie-virtual-machine:~$ touch qux.txt
yuxinjie@yuxinjie-virtual-machine:~$ chmod +x qux.txt
yuxinjie@yuxinjie-virtual-machine:~$ ll
总用量 148
drwxr-xr-x 21 yuxinjie yuxinjie 4096 5月 22 19:21 ./
drwxr-xr-x 3 root root 4096 5月 7 20:30 ../
-rw-rw-r-- 1 yuxinjie yuxinjie 13 5月 12 19:56 abc
-rw-rw-r-- 1 yuxinjie yuxinjie 4 5月 22 19:14 bar.txt
-rw-r--r-- 1 yuxinjie yuxinjie 52 5月 7 21:21 .bash_history
-rw-r--r-- 1 yuxinjie yuxinjie 220 5月 7 20:30 .bash_logout
-rw-r--r-- 1 yuxinjie yuxinjie 3771 5月 7 20:30 .bashrc
drwxrwxr-x 2 yuxinjie yuxinjie 4096 5月 22 19:05 baz/
lrwxrwxrwx 1 yuxinjie yuxinjie 7 5月 22 19:17 baz.txt -> bar.txt
drwx----- 12 yuxinjie yuxinjie 4096 5月 7 21:08 .cache/
drwx----- 18 yuxinjie yuxinjie 4096 5月 7 21:05 .config/
drwx----- 3 yuxinjie yuxinjie 4096 5月 7 20:51 .dbus/
-rw-r--r-- 1 yuxinjie yuxinjie 8980 5月 7 20:30 examples.desktop
-rw-rw-r-- 1 yuxinjie yuxinjie 0 5月 22 19:09 file4
drwx----- 2 yuxinjie yuxinjie 4096 5月 12 19:54 .gconf/
drwx----- 3 yuxinjie yuxinjie 4096 5月 12 19:39 .gnupg/
-rw----- 1 yuxinjie yuxinjie 1170 5月 12 19:39 .ICEauthority
drwx----- 3 yuxinjie yuxinjie 4096 5月 7 20:51 .local/
-rw-rw-r-- 1 yuxinjie yuxinjie 10240 5月 22 19:10 package.tar
drwxrwxr-x 2 yuxinjie yuxinjie 4096 5月 22 19:07 path1/
drwxrwxr-x 2 yuxinjie yuxinjie 4096 5月 22 19:08 path2/
drwxrwxr-x 3 yuxinjie yuxinjie 4096 5月 22 19:12 path3/
drwx----- 2 yuxinjie yuxinjie 4096 5月 7 20:51 .presage/
-rw-r--r-- 1 yuxinjie yuxinjie 655 5月 7 20:30 .profile
-rwxrwxr-x 1 yuxinjie yuxinjie 0 5月 22 19:21 qux.txt*
-rw-r--r-- 1 yuxinjie yuxinjie 0 5月 7 21:20 .sudo_as_admin_successful
-rw----- 1 yuxinjie yuxinjie 69 5月 12 19:39 .Xauthority
-rw----- 1 yuxinjie yuxinjie 84 5月 12 19:39 .xsession-errors
-rw----- 1 yuxinjie yuxinjie 1460 5月 7 21:45 .xsession-errors.old
drwxr-xr-x 2 yuxinjie yuxinjie 4096 5月 7 20:51 公共的/
drwxr-xr-x 2 yuxinjie yuxinjie 4096 5月 7 20:51 模板/
drwxr-xr-x 2 yuxinjie yuxinjie 4096 5月 7 20:51 视频/
drwxr-xr-x 2 yuxinjie yuxinjie 4096 5月 7 20:51 图片/
drwxr-xr-x 2 yuxinjie yuxinjie 4096 5月 7 20:51 文档/
drwxr-xr-x 3 yuxinjie yuxinjie 4096 5月 7 21:29 下载/
drwxr-xr-x 2 yuxinjie yuxinjie 4096 5月 7 20:51 音乐/
drwxr-xr-x 2 yuxinjie yuxinjie 4096 5月 7 20:51 桌面/
```

#### 四、 实验过程分析与讨论

本次实验学习了打包文件，链接文件和修改文件权限的练习。

#### 五、 指导教师意见

指导教师签字：卢洋

# 实验报告

实验名称	实验三 vim 编辑器及 gcc 编译器的使用		
实验教室	丹青 922	实验日期	2023 年 3 月 22 日
学 号	2021223123	姓 名	于心洁
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 ；

```
root@yuxinjie: ~/workspace1  
yuxinjie@yuxinjie-virtual-machine:~$ mkdir workspace1  
yuxinjie@yuxinjie-virtual-machine:~$ cd workspace1
```

- (2) 进入目录 workspace1 ；

```
root@yuxinjie: ~/workspace1  
yuxinjie@yuxinjie-virtual-machine:~$ mkdir workspace1  
yuxinjie@yuxinjie-virtual-machine:~$ cd 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：

```
yuxinjie@yuxinjie-virtual-machine:~$ vim test.c  
yuxinjie@yuxinjie-virtual-machine:~$ gcc -o test test.c
```

- (4) 保存 test.c 的内容，并退出；

```
#include <stdio.h>

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

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

```
yuxinjie@yuxinjie-virtual-machine:~$ gcc -o test test.c
yuxinjie@yuxinjie-virtual-machine:~$ ./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;

```
netto@netto:~$  
yuxinjie@yuxinjie-virtual-machine:~$ mkdir workspace2  
yuxinjie@yuxinjie-virtual-machine:~$ cd workspace2
```

(2) 进入 workspace2 目录;

```
netto@netto:~$  
yuxinjie@yuxinjie-virtual-machine:~$ mkdir workspace2  
yuxinjie@yuxinjie-virtual-machine:~$ cd 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;

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

(4) 使用 vim 编辑当前目录下的 gai.conf ;

```
yuxinjie@yuxinjie-virtual-machine:~/workspace2$ vim gai.conf
```

```

e@yuxinjie-virtual-machine: ~/workspace2
# 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>
#   Add another rule to the RFC 3484 precedence table. See section 2.1
#   and 10.3 in RFC 3484. The default is:
#
#precedence ::1/128      50
#precedence ::/0         40
#precedence 2002::/16    30
#precedence ::/96        20
#precedence ::ffff:0:0/96 10
#
#   For sites which prefer IPv4 connections change the last line to
#
#precedence ::ffff:0:0/96 100
#
# scopev4 <mask> <value>
#   Add another rule to the RFC 6724 scope table for IPv4 addresses.
#   By default the scope IDs described in section 3.2 in RFC 6724 are
#   used. Changing these defaults should hardly ever be necessary.
#   The defaults are equivalent to:
#
#scopev4 ::ffff:169.254.0.0/112 2
#scopev4 ::ffff:127.0.0.0/104 2
#scopev4 ::ffff:0.0.0.0/96 14
~
~
~

```

(5) 将光标移到第 18 行;

```

e@yuxinjie-virtual-machine: ~/workspace2
# 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>
#   Add another rule to the RFC 3484 precedence table. See section 2.1
#   and 10.3 in RFC 3484. The default is:
#
#precedence ::1/128      50
#precedence ::/0         40
#precedence 2002::/16    30
#precedence ::/96        20
#precedence ::ffff:0:0/96 10
#
#   For sites which prefer IPv4 connections change the last line to
#
#precedence ::ffff:0:0/96 100
#
# scopev4 <mask> <value>
#   Add another rule to the RFC 6724 scope table for IPv4 addresses.
#   By default the scope IDs described in section 3.2 in RFC 6724 are
#   used. Changing these defaults should hardly ever be necessary.
#   The defaults are equivalent to:
#
#scopev4 ::ffff:169.254.0.0/112 2
#scopev4 ::ffff:127.0.0.0/104 2
#scopev4 ::ffff:0.0.0.0/96 14
~
~
~

```

(6) 复制该行内容;

```

终端 文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(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>
#   Add another rule to the RFC 3484 precedence table. See section 2.1
#   and 10.3 in RFC 3484. The default is:
#
#precedence ::1/128      50
#precedence ::/0         40
#precedence 2002::/16    30
#precedence ::/96        20
#precedence ::ffff:0:0/96 10
#
#   For sites which prefer IPv4 connections change the last line to
#
#precedence ::ffff:0:0/96 100
#
# scopev4 <mask> <value>
#   Add another rule to the RFC 6724 scope table for IPv4 addresses.
#   By default the scope IDs described in section 3.2 in RFC 6724 are
#   used. Changing these defaults should hardly ever be necessary.
#   The defaults are equivalent to:
#
#scopev4 ::ffff:169.254.0.0/112 2
#scopev4 ::ffff:127.0.0.0/104 2
#scopev4 ::ffff:0.0.0.0/96 14
~

```

(7) 将光标移到最后一行行首;

```

yuxinjie@yuxinjie-virtual-machine: ~/workspace
# 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>
#   Add another rule to the RFC 3484 precedence table. See section 2.1
#   and 10.3 in RFC 3484. The default is:
#
#precedence ::1/128      50
#precedence ::/0         40
#precedence 2002::/16    30
#precedence ::/96        20
#precedence ::ffff:0:0/96 10
#
#   For sites which prefer IPv4 connections change the last line to
#
#precedence ::ffff:0:0/96 100
#
# scopev4 <mask> <value>
#   Add another rule to the RFC 6724 scope table for IPv4 addresses.
#   By default the scope IDs described in section 3.2 in RFC 6724 are
#   used. Changing these defaults should hardly ever be necessary.
#   The defaults are equivalent to:
#
#scopev4 ::ffff:169.254.0.0/112 2
#scopev4 ::ffff:127.0.0.0/104 2
#scopev4 ::ffff:0.0.0.0/96 14
~

```

(8) 粘贴复制行的内容;

```
yuxinjie@yuxinjie-virtual-machine: ~/workspace2
# 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>
# 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>
# Add another rule to the RFC 3484 precedence table. See section 2.1
# and 10.3 in RFC 3484. The default is:
#
#precedence ::1/128 50
#precedence ::/0 40
#precedence 2002::/16 30
#precedence ::/96 20
#precedence ::ffff:0:0/96 10
#
# For sites which prefer IPv4 connections change the last line to
#
#precedence ::ffff:0:0/96 100
#
# scopev4 <mask> <value>
# Add another rule to the RFC 6724 scope table for IPv4 addresses.
# By default the scope IDs described in section 3.2 in RFC 6724 are
# used. Changing these defaults should hardly ever be necessary.
# The defaults are equivalent to:
#
#scopev4 ::ffff:169.254.0.0/112 2
#scopev4 ::ffff:127.0.0.0/104 2
#scopev4 ::ffff:0.0.0.0/96 14
#scopev4 ::ffff:0.0.0.0/96 14
~
~
~
```

(9) 撤销第 8 步的动作；

```

@yuxinjie-virtual-machine: ~/workspace2
# 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>
#   Add another rule to the RFC 3484 precedence table. See section 2.1
#   and 10.3 in RFC 3484. The default is:
#
#precedence ::1/128      50
#precedence ::/0         40
#precedence 2002::/16    30
#precedence ::/96        20
#precedence ::ffff:0:0/96 10
#
#   For sites which prefer IPv4 connections change the last line to
#
#precedence ::ffff:0:0/96 100
#
# scopev4 <mask> <value>
#   Add another rule to the RFC 6724 scope table for IPv4 addresses.
#   By default the scope IDs described in section 3.2 in RFC 6724 are
#   used. Changing these defaults should hardly ever be necessary.
#   The defaults are equivalent to:
#
#scopev4 ::ffff:169.254.0.0/112 2
#scopev4 ::ffff:127.0.0.0/104 2

```

(10) 存盘但不退出;





[illegible]

(11) 将光标移到首行;



```

jle@yuxinjie-virtual-machine: ~/workspace2
Hello, this is a vim world! 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>
# 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>
#   Add another rule to the RFC 3484 precedence table. See section 2.1
#   and 10.3 in RFC 3484. The default is:
#
#precedence ::1/128      50
#precedence ::/0         40
#precedence 2002::/16    30
#precedence ::/96        20
#precedence ::ffff:0:0/96 10
#
#   For sites which prefer IPv4 connections change the last line to
#
#precedence ::ffff:0:0/96 100
#
# scopev4 <mask> <value>
#   Add another rule to the RFC 6724 scope table for IPv4 addresses.
#   By default the scope IDs described in section 3.2 in RFC 6724 are
#   used. Changing these defaults should hardly ever be necessary.
#   The defaults are equivalent to:
#
#scopev4 ::ffff:169.254.0.0/112 2
#scopev4 ::ffff:127.0.0.0/104 2
~
~
~
~
~

```

(13) 删除字符串 “this” ；

```
# @yuxinjie-virtual-machine: ~/workspace2
Hello,this is a vim world!# 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
LibreOffice Calc re are possible runtime problems. The default is no.
# label <mask> <value>
# 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>
Add another rule to the RFC 3484 precedence table. See section 2.1
and 10.3 in RFC 3484. The default is:
#
#precedence ::1/128      50
#precedence :::/0       40
#precedence 2002::/16    30
#precedence ::/96        20
#precedence :ffff:0:0/96 10
#
For sites which prefer IPv4 connections change the last line to
#
#precedence ::ffff:0:0/96 100
#
scopev4 <mask> <value>
Add another rule to the RFC 6724 scope table for IPv4 addresses.
By default the scope IDs described in section 3.2 in RFC 6724 are
used. Changing these defaults should hardly ever be necessary.
The defaults are equivalent to:
#
#scopev4 :ffff:169.254.0.0/112 2
#scopev4 :ffff:127.0.0.0/104 2
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
```



#### 四、 实验过程分析与讨论

练习了 vim 编辑器的基本操作。

#### 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 3 月 29 日
学 号	2021223123	姓 名	于心洁
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` ；

```
yuxinjie@yuxinjie-virtual-machine:~$ sudo useradd -m -s /bin/sh -p $(openssl passwd -1 123456) -c "bar" foo
yuxinjie@yuxinjie-virtual-machine:~$ ll /home
总用量 16
drwxr-xr-x  4 root    root    4096 5月 22 22:53 ./
drwxr-xr-x 24 root    root    4096 5月 12 20:14 ../
drwxr-xr-x  2 foo     foo     4096 5月 22 22:53 foo/
drwxr-xr-x 24 yuxinjie yuxinjie 4096 5月 22 22:11 yuxinjie/
```

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` ；

```
yuxinjie@yuxinjie-virtual-machine:~$ sudo su - foo
```

```
$ usermod -u 2000 -s /bin/csh foo
usermod: user foo is currently used by process 55905
$
```

3. 从用户 `foo` 切换到 `root` ；

```
$ su
密码:
root@yuxinjie-virtual-machine:/home/foo#
```



4. 删除 foo 用户，并在删除该用户的同时一并删除其家目录；

```
root@yuxinjie-virtual-machine:/home/foo# su yuxinjie
yuxinjie@yuxinjie-virtual-machine:/home/foo$ cd ~
yuxinjie@yuxinjie-virtual-machine:~$ sudo userdel -r foo
userdel: user foo is currently used by process 56019
yuxinjie@yuxinjie-virtual-machine:~$
```

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；

```
yuxinjie@yuxinjie-virtual-machine:~$ vim user.txt
yuxinjie@yuxinjie-virtual-machine:~$ sudo newusers < user.txt
newusers: 第 1 行: 无法更新密码
newusers: 第 2 行: 无法更新密码
newusers: 发现错误, 忽略改动
```

6. 创建用户组 group1，并在创建时设置其 GID 为 3000；

```
yuxinjie@yuxinjie-virtual-machine:~$ sudo groupadd -g 3000 group1
[sudo] yuxinjie 的密码:
groupadd: group 'group1' already exists
```

7. 在用户组 group1 中添加两个之前批量创建的用户；

```
yuxinjie@yuxinjie-virtual-machine:~$ sudo usermod -a -G group1 user1
yuxinjie@yuxinjie-virtual-machine:~$ sudo usermod -a -G group1 user2
yuxinjie@yuxinjie-virtual-machine:~$
```

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。

```
yuxinjie@yuxinjie-virtual-machine:~$ sudo -u user1 -i
$ sudo cat /etc/shadow
[sudo] user1 的密码:
user1 不在 sudoers 文件中。此事将被报告。
$
```

## 实验过程分析与讨论

在创建用户组时遇到了很多问题。

## 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 5 日
学 号	2021223123	姓 名	于心洁
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);

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

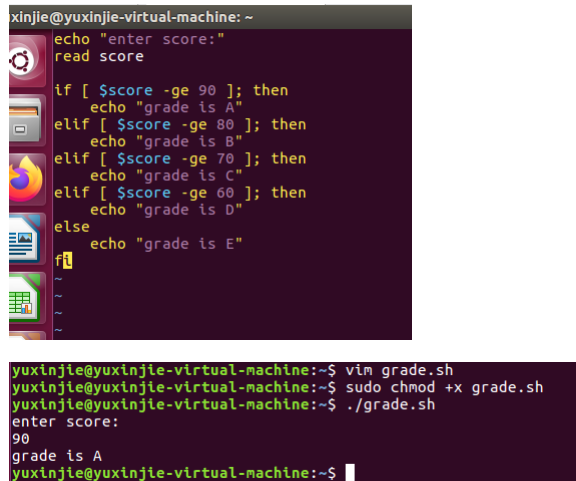
2. 定义变量 bar 的值为 100 , 并使用 test 命令比较其值是否大于 150 , 并显示 test 命令的退出码 (终端上执行);

```
yuxinjie@yuxinjie-virtual-machine:~$ bar=100
yuxinjie@yuxinjie-virtual-machine:~$ test $bar -gt 150
yuxinjie@yuxinjie-virtual-machine:~$ echo $?
1
yuxinjie@yuxinjie-virtual-machine:~$
```

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 );



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 实现。



```
xinjie@yuxinjie-virtual-machine: ~  
echo "enter score:"  
read score  
  
if [ $score -ge 90 ]; then  
    echo "grade is A"  
elif [ $score -ge 80 ]; then  
    echo "grade is B"  
elif [ $score -ge 70 ]; then  
    echo "grade is C"  
elif [ $score -ge 60 ]; then  
    echo "grade is D"  
else  
    echo "grade is E"  
fi  
  
yuxinjie@yuxinjie-virtual-machine:~$ vim grade.sh  
yuxinjie@yuxinjie-virtual-machine:~$ sudo chmod +x grade.sh  
yuxinjie@yuxinjie-virtual-machine:~$ ./grade.sh  
enter score:  
90  
grade is A  
yuxinjie@yuxinjie-virtual-machine:~$
```

#### 四、 实验过程分析与讨论

\$\$

Shell 本身的 PID (ProcessID)

\$!

Shell 最后运行的后台 Process 的 PID

\$?

最后运行的命令的结束代码 (返回值)

\$-

使用 Set 命令设定的 Flag 一览

\$\*

所有参数列表。如"\$\*"“用「」”括起来的情况、以"\$1 \$2 … \$n"的形式输出所有参数。

`$@`

所有参数列表。如"\$@"“用「」”括起来的情况、以"\$1" "\$2" … "\$n"的形式输出所有参数。

`$#`

添加到 Shell 的参数个数

`$0`

Shell 本身的文件名

`$1~$n`

添加到 Shell 的各参数值。`$1` 是第 1 参数、`$2` 是第 2 参数…。

条件判断

1、基本语法：[ condition ] （condition 前后有空格）

非空返回 true， 可使用 \$? 验证（0 为 true， >1 为 false）

2、常用判断条件

（1）= 字符串比较

（2）两个整数的比较

-lt 小于

-le 小于等于

-eq 等于

-gt 大于等于

-ne 不等于

（3）按照文件权限进行判断

-r 有读的权限

-w 有写的权限

-x 有执行的权限

（4）按照文件类型进行判断

-f 文件存在并且是一个常规的文件



-e 文件存在

-d 文件存在并是一个目录

五、指导教师意见

指导教师签字：卢洋

# 实验报告

实验名称	实验六 Shell 循环控制语句		
实验教室	丹青 922	实验日期	2023 年 4 月 12 日
学 号	2021223123	姓 名	于心洁
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。

```
jie@yuxinjie-virtual-machine: ~  
target_dir=~/workspace  
for file in *.c;do  
    cp "$file" "target_dir"  
done  
echo "ok"  
~  
~  
~  
~  
yuxinjie@yuxinjie-virtual-machine:~$ vim cc.sh  
yuxinjie@yuxinjie-virtual-machine:~$ sh cc.sh  
ok  
yuxinjie@yuxinjie-virtual-machine:~$
```

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

```
yuxinjie@yuxinjie-virtual-machine:~$ vim ou.sh  
yuxinjie@yuxinjie-virtual-machine:~$ bash ou.txt  
bash: ou.txt: 没有那个文件或目录  
yuxinjie@yuxinjie-virtual-machine:~$ bash ou.sh  
110  
yuxinjie@yuxinjie-virtual-machine:~$
```

```
jie@yuxinjie-virtual-machine: ~
#!/bin/bash
n=1
sum=0
while((n<=20))
do
    if((n%2 == 0));then
        sum=$((sum+n))
    fi
    n=$((n+1))
done
echo "$sum"
```

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

```
yuxinjie@yuxinjie-virtual-machine: ~
#!/bin/bash
i=1
s=0
until((i>10))
do
    squ=$((i*i))
    s=$((s+squ))
    i=$((i+1))
done
echo "$s"
```

```
yuxinjie@yuxinjie-virtual-machine:~$ vim sq.sh
yuxinjie@yuxinjie-virtual-machine:~$ bash sq.sh
385
yuxinjie@yuxinjie-virtual-machine:~$
```

3. 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。

```
yuxinjie@yuxinjie-virtual-machine:~$ bash abcd.sh
a1234
b1234
c1234
d1234
yuxinjie@yuxinjie-virtual-machine:~$
```

```
yuxinjie@yuxinjie-virtual-machine:~$ bash abcd.sh
a1234yuxinjie@yuxinjie-virtual-machine:~$
```

```
yuxinjie@yuxinjie-virtual-machine:~$ bash abcd.sh
a1234678910
b1234678910
c1234678910
d1234678910
yuxinjie@yuxinjie-virtual-machine:~$
```

```
yuxinjie@yuxinjie-virtual-machine:~$ vim abcd.sh
yuxinjie@yuxinjie-virtual-machine:~$ bash abcd.sh
a1234b1234c1234d1234yuxinjie@yuxinjie-virtual-machine:~$
```

#### 四、 实验过程分析与讨论

练习掌握了循环语句：while, until, for

#### 五、 指导教师意见

指导教师签字：卢洋

# 实验报告

实验名称	实验七 Shell 函数		
实验教室	丹青 922	实验日期	2023 年 4 月 19 日
学 号	2021223123	姓 名	于心洁
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

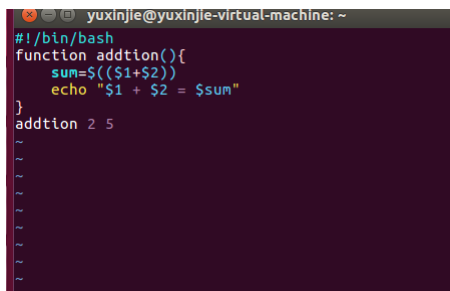
1. 掌握 Shell 函数的定义方法；
2. 掌握 Shell 函数的参数传递、调用和返回值；
3. 掌握 Shell 函数的递归调用方法；
4. 理解 Shell 函数的嵌套。

## 二、 实验环境

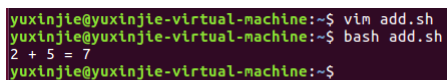
- (1) 计算机的硬件配置 PC 系列微机。
- (2) 计算机的软件配置 VMware 虚拟机软件及 Ubuntu 虚拟机。

## 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；



```
yuxinjie@yuxinjie-virtual-machine: ~  
#!/bin/bash  
function addition(){  
    sum=$(( $1+$2 ))  
    echo "$1 + $2 = $sum"  
}  
addition 2 5  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~
```



```
yuxinjie@yuxinjie-virtual-machine:~$ vim add.sh  
yuxinjie@yuxinjie-virtual-machine:~$ bash add.sh  
2 + 5 = 7  
yuxinjie@yuxinjie-virtual-machine:~$
```

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；

```

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

```

```

yuxinjie@yuxinjie-virtual-machine:~$ vim n.sh
yuxinjie@yuxinjie-virtual-machine:~$ bash n.sh
:7
5040
yuxinjie@yuxinjie-virtual-machine:~$

```

3. 一个Shell脚本的内容如下所示： 试运行该程序，并观察程序运行结果，理解函数嵌套的含义。

```

#!/bin/bash
function first() {
    function second() {
        function third() {
            echo "-3- here is in the third func."
        }
        echo "-2- here is in the second func."
    }
    echo "-1- here is in the first func."
}
echo "starting..."
yuxinjie@yuxinjie-virtual-machine:~$ bash f.sh
starting...

```



#### 四、 实验过程分析与讨论

- 1.掌握了 Shell 函数的定义方法；
2. 掌握了 Shell 函数的参数传递、调用和返回值；
3. 掌握了 Shell 函数的递归调用方法；
4. 理解了 Shell 函数的嵌套。

#### 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 4 月 26 日
学 号	2021223123	姓 名	于心洁
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：试使用 sed 命令实现如下功能：

- (1) 删除 \$ 符号；

```
yuxinjie@yuxinjie-virtual-machine:~$ cat > quote.txt
The honeysuckle band played all night long for only $90.
It was an evening of splendid music and company.
Too bad the disco floor fell through at 23:10.
The local nurse Miss P.Neave was in attendance.
yuxinjie@yuxinjie-virtual-machine:~$ 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；

```
yuxinjie@yuxinjie-virtual-machine:~$ cat quote.txt | sed -n '/music/p'
It was an evening of splendid music and company.
```

- (3) 在第 4 行后面追加内容： "hello world!" ；

```
yuxinjie@yuxinjie-virtual-machine:~$ cat quote.txt | sed '4a hello world!'
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" ；

```
yuxinjie@yuxinjie-virtual-machine:~$ cat quote.txt | sed 's/The/Quod/g'
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." ;

```
yuxinjie@yuxinjie-virtual-machine:~$ cat quote.txt | sed '3c This is the third line'
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;

```
yuxinjie@yuxinjie-virtual-machine:~$ cat quote.txt | sed '2d'
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。

```
yuxinjie@yuxinjie-virtual-machine:~$ cat quote.txt | sed -n "/$var/p"
The local nurse Miss P.Neave was in attendance.
```

2. 文件 numbers.txt 的内容如下所示: 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;

```
yuxinjie@yuxinjie-virtual-machine:~$ cat > numbers.txt
one : two : three
four : five : six
yuxinjie@yuxinjie-virtual-machine:~$ cat numbers.txt | awk '{FS=" "}{print $2}'
cat: numbers.txt: 没有那个文件或目录
yuxinjie@yuxinjie-virtual-machine:~$ cat numbers.txt | awk '{FS=" "}{print $2}'
:
:
:
yuxinjie@yuxinjie-virtual-machine:~$ cat numbers.txt | awk '{FS=":"}{print $2}'
:
:
:
five
yuxinjie@yuxinjie-virtual-machine:~$
```

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。

```
yuxinjie@yuxinjie-virtual-machine:~$ awk '{for(i=1;i<=NF;i++) if($i%2==0) {print $i; count++;}} END{print "The number is:" count}' foo.txt
2
4
46
The number is:3
yuxinjie@yuxinjie-virtual-machine:~$
```

4. 脚本的内容如下所示: 试运行该脚本, 并理解该脚本实现的功能。

```
yuxinjie@yuxinjie-virtual-machine:~$ vim fuc.sh
yuxinjie@yuxinjie-virtual-machine:~$ cat fuc.sh
#!/bin/bash
read -p "enter search pattern: " pattern
awk "/$pattern/{ nmatches++; print } END { print nmatches, "found." }" info.txt
yuxinjie@yuxinjie-virtual-machine:~$ bash fuc.sh
enter search pattern: abc
awk: cannot open info.txt (No such file or directory)
```

该脚本的功能是匹配字符串，首先输入字符串，然后在info.txt中查找，如果能找到，那nmatches加1，最后输出。

#### 四. 实验过程分析与讨论

1. 掌握了 sed 基本编辑命令的使用方法；
2. 掌握了 sed 与 Shell 变量的交互方法；
3. 掌握了 awk 命令的使用方法；
4. 掌握了 awk 与 Shell 变量的交互方法。

## 五、指导教师意见

指导教师签字：卢洋