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 9 日
学　　号	2021213097	姓　　名	代尊闯
专业班级	计算机科学与技术 02 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

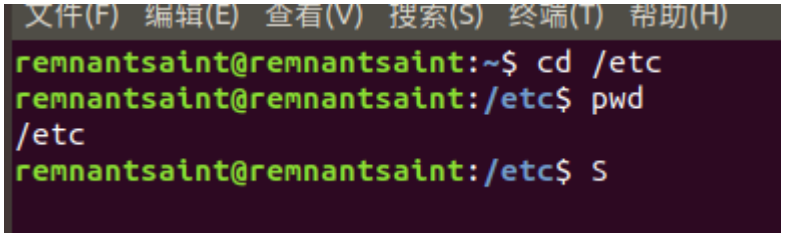
二、 实验环境

（1）计算机的硬件配置 PC 系列微机。

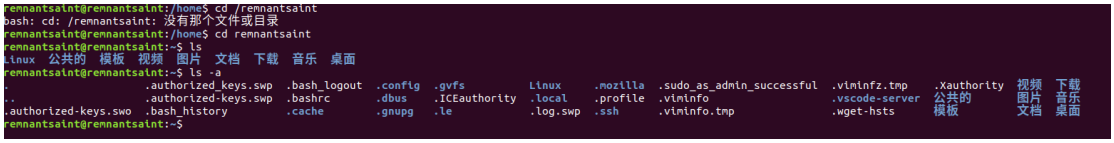
（2）计算机的软件配置 VMware 虚拟机软件及 Ubuntu 虚拟机。

三、 实验内容及结果

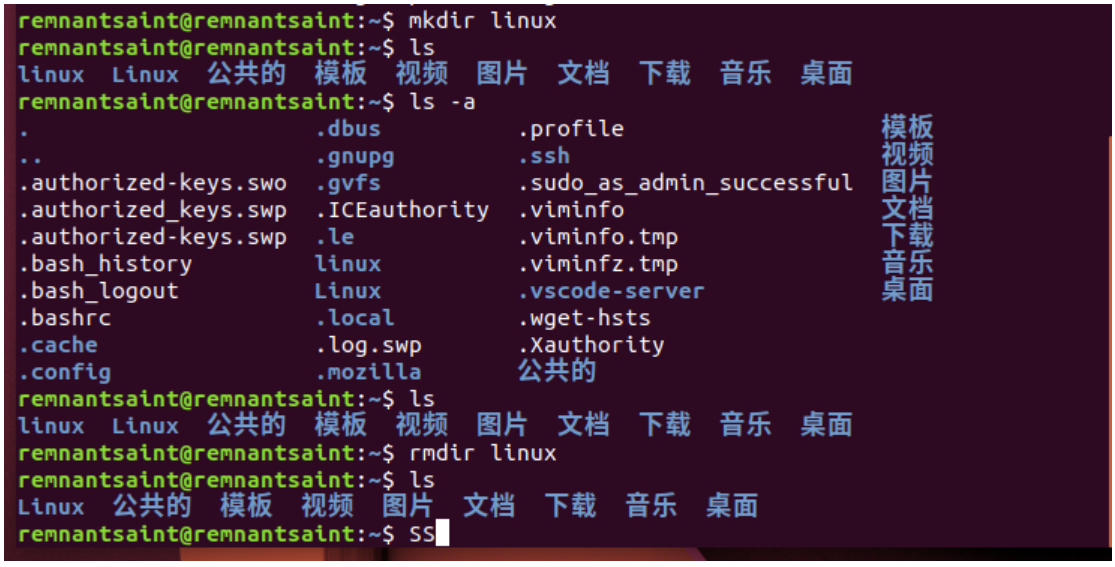
1. 使用命令切换到/etc 目录，并显示当前工作目录路径



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。



3、使用命令创建目录/home/lyj/linux，然后删除该目录。



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

```
remnantsaint@remnantsaint:~$ cat >abc
hello linux!
remnantsaint@remnantsaint:~$ cat abc
hello linux!
remnantsaint@remnantsaint:~$
```

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。

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

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。

```
remnantsaint@remnantsaint:/etc$ cat -n adduser.conf
     1  # /etc/adduser.conf: `adduser' configuration.
     2  # See adduser(8) and adduser.conf(5) for full documentation.
     3
```

7、/etc/adduser.conf 的内容。

```
remnantsaint@remnantsaint:/etc$ more adduser.conf
# /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
```

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

```
remnantsaint@remnantsaint:~$ cat >facebook.txt
google 110 5000
baidu 100 5000
guge 50 3000
sohu 100 4500
```

9．第一列为公司名称，第2列为公司人数，第3列为员工平均工资。

利用sort命令完成下列排

（1）按公司字母顺序排序

```
remnantsaint@remnantsaint:~$ sort facebook.txt
baidu 100 5000
google 110 5000
guge 50 3000
sohu 100 4500
```

（2）按公司人数排序

```
remnantsaint@remnantsaint:~$ sort -n -k 2 facebook.txt
guge 50 3000
baidu 100 5000
sohu 100 4500
google 110 5000
```

（3）

（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

```
remnantsaint@remnantsaint:~$ sort -n -t ' ' -k 2 -k 3 facebook.txt
guge 50 3000
sohu 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

```
remnantsaint@remnantsaint:~$ sort -n -t ' ' -k 3r -k 2 facebook.txt
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。

```
remnantsaint@remnantsaint:~$ sort -k 1.2 facebook.txt
baidu 100 5000
sohu 100 4500
google 110 5000
guge 50 3000
```

四、 实验过程分析与讨论

学会了以下知识：

ls -a 显示隐藏文件

mkdir 建立目录 rmdir 删除目录

cat >name 创建文件，用 ctrl+d 结束操作

cp -r a/ b 将 a 目录下的文件全部复制到 b 目录下

rm+文件名删除文件，不能删除目录

rm -i 删除前逐一询问确认。

rm -f 即使原档案属性设为唯读，亦直接删除，无需逐一确认。

rm -r 将目录及以下之档案亦逐一删除

sort -n 依照数值大小排序 -k 2 依照第二列排序 -t<分割字符> 指定排序时所用的栏位分隔字符 例：sort -n -t ' ' -k 2 -k 3 face.txt 为按数值排序 2 列，2 列中相同的按以' '分开的第三列排序 sort -k 3r 代表按第三列降序排列 sort -k 1.2 表示按第一列的第二个字母排序

五、指导教师意见

指导教师签字：卢洋

实验报告

实验名称	实验一 Linux 常用命令（二）		
实验教室	丹青 922	实验日期	2023 年 3 月 16 日
学　　号	2021213097	姓　　名	代尊闯
专业班级	计算机科学与技术 02 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

五、 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 。

六、 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。

七、 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

```
remnantsaint@remnantsaint:~$ rmdir baz
remnantsaint@remnantsaint:~$ mkdir baz
remnantsaint@remnantsaint:~$ cd baz
remnantsaint@remnantsaint:~/baz$ cat qux
cat: qux: 没有那个文件或目录
remnantsaint@remnantsaint:~/baz$ cat >qux
adsf
jlk
xzcv^Z
[1]+  已停止               cat > qux
remnantsaint@remnantsaint:~/baz$ ls
qux
remnantsaint@remnantsaint:~/baz$
```

（2）在用户目录下查找文件 qux ，并显示该文件位置信息

```
baz  Linux  uu  公共的  模板  视频  图片  文档  下载  音乐  桌面
remnantsaint@remnantsaint:~$ find . -name "qux" 2>/dev/null
./baz/qux
```

3）统计文件 qux 中所包含内容的行数、字数和字节数；

```
remnantsaint@remnantsaint:~/baz$ wc -lwc qux
2 2 9 qux
```

4）在用户目录下查找文件 qux ，并删除该文件；

```
remnantsaint@remnantsaint:/$ find . -name "qux" 2>/dev/null
./home/remnantsaint/baz/qux
remnantsaint@remnantsaint:/$ rm ./home/remnantsaint/baz/qux
remnantsaint@remnantsaint:/$ find . -name "qux" 2>/dev/null
```

5）查看文件夹 baz 内容，看一下是否删除了文件 qux .

```
remnantsaint@remnantsaint:~/baz$ cat qux
cat: qux: 没有那个文件或目录
```

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 ；

```
remnantsaint@remnantsaint:~$ mkdir path1
remnantsaint@remnantsaint:~$ cd path1
remnantsaint@remnantsaint:~/path1$ cat >file1
^Z
[2]+  已停止               cat > file1
remnantsaint@remnantsaint:~/path1$ cat file2
cat: file2: 没有那个文件或目录
remnantsaint@remnantsaint:~/path1$ cat >file2
^Z
[3]+  已停止               cat > file2
```

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

```
remnantsaint@remnantsaint:~$ mkdir path2
remnantsaint@remnantsaint:~$ cd path2
remnantsaint@remnantsaint:~/path2$ cat file3
cat: file3: 没有那个文件或目录
remnantsaint@remnantsaint:~/path2$ cat >file3
^Z
[4]+  已停止               cat > file3
```

3）在用户目录下新建文件 file4 ；

```
remnantsaint@remnantsaint:~$ cat > file4
^Z
[5]+  已停止               cat > 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 ；

```
remnantsaint@remnantsaint:~$ tar -cf package.tar path1 file4
remnantsaint@remnantsaint:~$ ls
file4  package.tar  path2  公共的  视频  文档  音乐
Linux  path1        uu     模板    图片  下载  桌面
```

5）查看包 package.tar 的内容；

```
remnantsaint@remnantsaint:~$ tar -tf package.tar
path1/
path1/file1
path1/file2
file4
```

6）向包 package.tar 里添加文件夹 path2 的内容；

```
remnantsaint@remnantsaint:~$ tar -rf package.tar path2
```

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

```
remnantsaint@remnantsaint:~$ mkdir path3
remnantsaint@remnantsaint:~$ ls
file4  package.tar  path2  uu      模板  图片  下载
Linux  path1        path3  公共的  视频  文档  音乐
remnantsaint@remnantsaint:~$ cp package.tar path3
```

8）进入 path3 文件夹，并还原包 package.tar 的内容。

```
remnantsaint@remnantsaint:~/path3$ tar -xvf package.tar
path1/
path1/file1
path1/file2
file4
path2/
path2/file3
```

3. 符号链接内容

1）新建文件 foo.txt ，内容为 123 ；

```
remnantsaint@remnantsaint:~$ vim 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；

```
remnantsaint@remnantsaint:~$ ln foo.txt bar.txt
remnantsaint@remnantsaint:~$ diff foo.txt bar.txt
```

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

```
remnantsaint@remnantsaint:~$ diff foo.txt bar.txt -e
```

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；

（

```
remnantsaint@remnantsaint:~$ cat >bar.txt
abc
^Z
[6]+  已停止               cat > bar.txt
remnantsaint@remnantsaint:~$ diff foo.txt bar.txt
```

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

7）

```
remnantsaint@remnantsaint:~$ ls -i bar.txt
789407 bar.txt
remnantsaint@remnantsaint:~$ 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；

（

```
remnantsaint@remnantsaint:~$ ls -i bar.txt baz.txt
789407 bar.txt  789403 baz.txt
```

```
remnantsaint@remnantsaint:~$ ln -s bar.txt baz.txt
```

```
remnantsaint@remnantsaint:~$ diff baz.txt bar.txt
```

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

9）

```
remnantsaint@remnantsaint:~$ rm bar.txt
remnantsaint@remnantsaint:~$ cat baz.txt
cat: baz.txt: 没有那个文件或目录
```

4. 权限管理

（

1）新建文件 qux.txt ；

（

```
remnantsaint@remnantsaint:~$ cat >qux.txt
123^Z
[7]+  已停止               cat > qux.txt
remnantsaint@remnantsaint:~$ chmod 777 qux.txt
```

2）为文件 qux.txt 增加执行权限（所有用户都可以执行）。

```
remnantsaint@remnantsaint:~$ ls -l qux.txt
-rwxrwxrwx 1 remnantsaint remnantsaint 0 3月  30 18:53 qux.txt
```

八、 实验过程分析与讨论

ls -a 显示隐藏文件

mkdir 建立目录 rmdir 删除目录

cat >name 创建文件，用 ctrl+d 结束操作

cp -r a/ b 将 a 目录下的文件全部复制到 b 目录下

rm+文件名删除文件，不能删除目录

rm -i 删除前逐一询问确认。

rm -f 即使原档案属性设为唯读，亦直接删除，无需逐一确认。

rm -r 将目录及以下之档案亦逐一删除

sort -n 依照数值大小排序 -k 2 依照第二列排序 -t<分割字符>指定

排序时所用的栏位分隔字符 例：sort -n -t ' ' -k 2 -k 3 face.txt 为按数值排序 2 列，2 列中相同的按以' '分开的第三列排序 sort -k 3r 代表按第三列降序排列 sort -k 1.2 表示按第一列的第二个字母排序

find . -name "*.c" 在当前目录及子目录查找后缀为.c 的文件 -name 前的.是当前目录及子目录 /是系统全部文件

find / -type f -iname qux 2>/dev/null 可以不找权限不够的文件

uniq 删除相邻的重复行，只保留一行 -c 删除行前显示重复次数 -d 仅显示重复行 -u 仅显示不重复行

wc [option] filename -c 统计字节数 -l 统计行数 -w 统计字数

comm [option] file1 file2 对两个已经排好序的文件比较 -12:表示 1，2 列不显示，默认都显示 输出三列：仅在 file1 中出现的行、仅在 file2 中出现的行、都出现的行

diff [option] file1 file2 逐行比较两个文件，列出不同，不要求已排序 -b 忽略空格影响 -c 采用三行上下文输出格式 -C n 采用 N 行上下文输出格式 -e 残圣一个合法的 ed 脚本作为输出

sort [option] filename -n 按整数排序 -u 去除重复行 -o filename 将排序结果输出文件 -k "n" -t "c"以 c 符号分割的第 n 个来排序 -r 从大到小排序

cp [option] 源 目标 cp *.c /home/remnantsaint/linuxtest -i 询问 -f 强制 -r 复制目录

tar [option] 包名 file-list(文件列表） -cf 所有文件打包 -cvf 打包同

时列出包里文件 file-list 写成*时就是所有文件 tar -tf .tar 查看包内容 -xvf 还原包内容 tar -rvf .tar file 添加新文件 -czvf 用 gzip 压缩包打包文件-zcvf 查看压缩内容 tar -zxvf .tar.gz -C ./b 解压缩.tar.gz 包到当前目录下的目录 b

硬链接：同一个文件有多个别名，改变内容共享 ln [option] file link 不能对目录建立硬链接

软链接（符号链接）该档案的内容是指向另一个档案的位置 -b 删除链接 -s 软链接

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 23 日
学　　号	2021213097	姓　　名	代尊闯
专业班级	计算机科学与技术 02 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

九、 实验目的

掌握 vim 编辑器及 gcc 编译器的使用方法。

十、 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。

十一、 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 ；

（

```
remnantsaint@remnantsaint:~$ mkdir workspace1
```

2）进入目录 workspace1 ；

（

```
remnantsaint@remnantsaint:~$ cd workspace1
remnantsaint@remnantsaint:~/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("hellow 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。

```
remnantsaint@remnantsaint:~/workspace1$ gcc test.c -o test
remnantsaint@remnantsaint:~/workspace1$ ls
test  test.c
remnantsaint@remnantsaint:~/workspace1$ ./test
hellow 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；

（

```
remnantsaint@remnantsaint:~$ mkdir workspace2
```

2）进入 workspace2 目录；

（

```
remnantsaint@remnantsaint:~$ cd workspace2
remnantsaint@remnantsaint:~/workspace2$
```

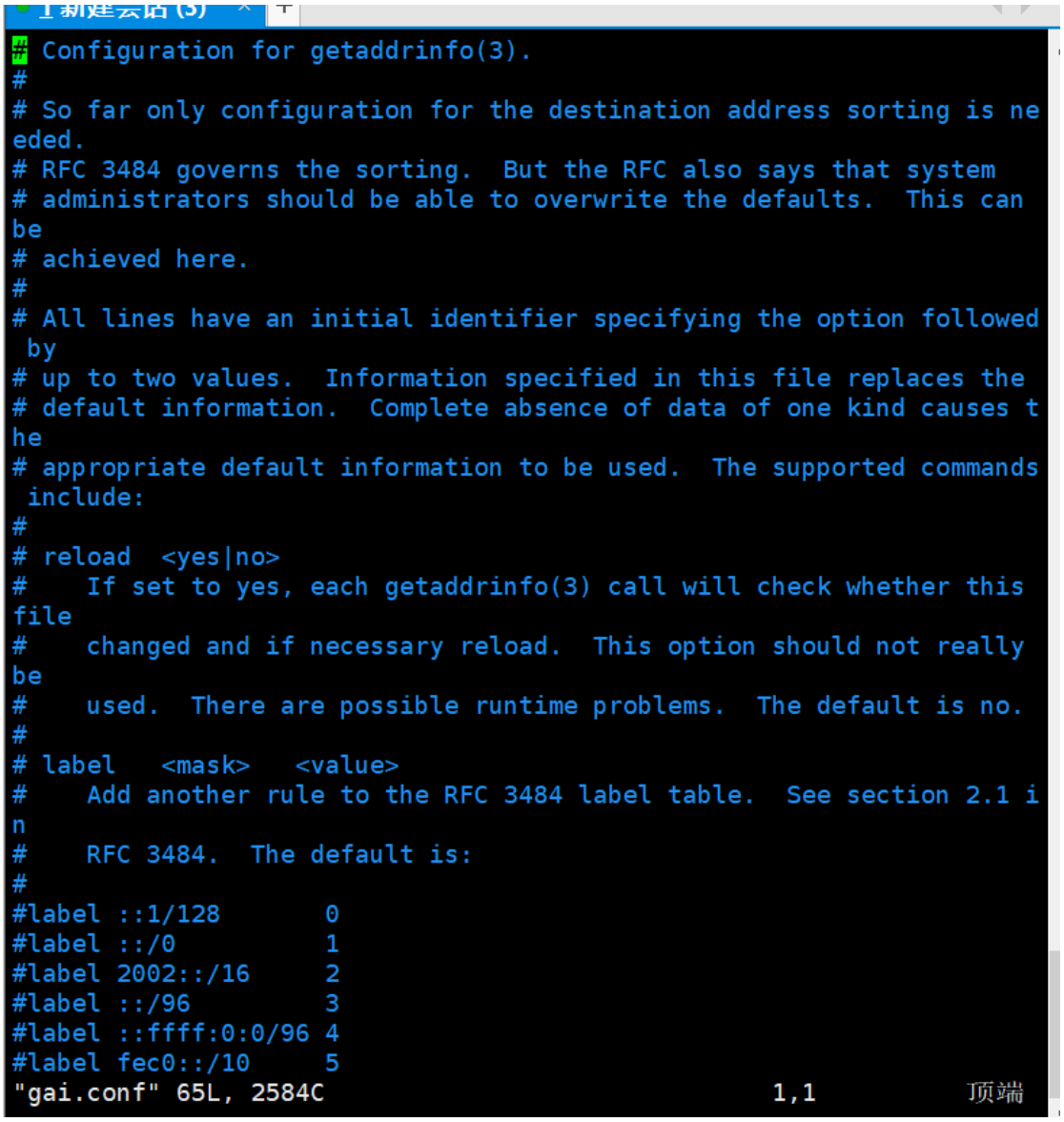
3）使用以下命令：

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；

（

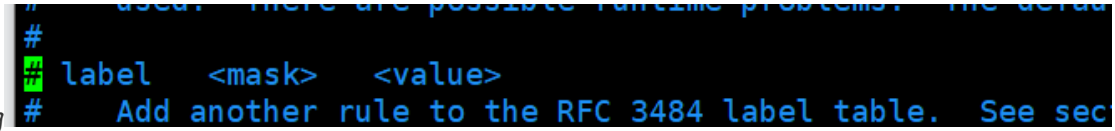
```
remnantsaint@remnantsaint:~/workspace2$ cp /etc/gai.conf gai.conf
```

4）使用 vim 编辑当前目录下的 gai.conf ；



5）将光标移到第 18 行；

（18gg



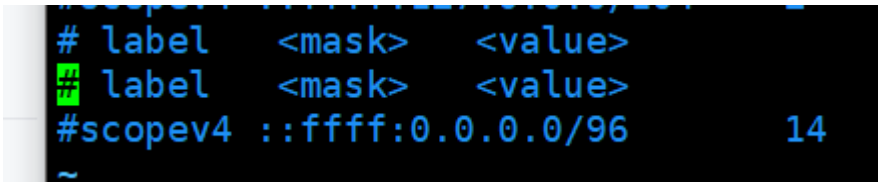
6）复制该行内容；

（yy

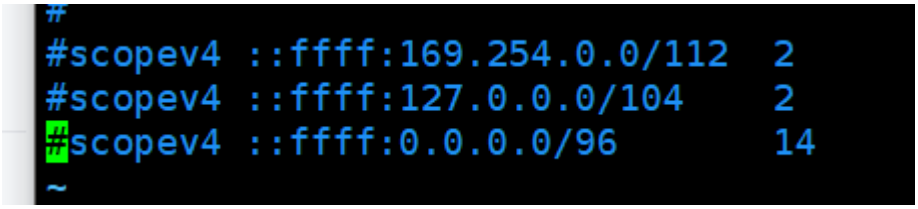
7）将光标移到最后一行行首；

（G

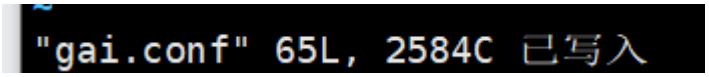
8）粘贴复制行的内容；



9）撤销第 8 步的动作；

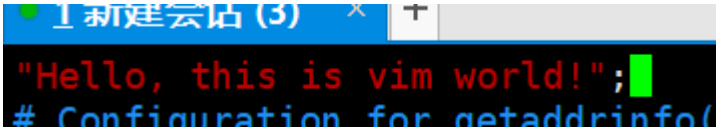


（10）存盘但不退出；



（11）将光标移到首行；

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



（13）删除字符串 "this" ；

（14）强制退出 vim ，不存盘。

十二、　　实验过程分析与讨论

要熟记 vim 的各种操作，每种操作之间有关联之处。

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 30 日
学　　号	2021213097	姓　　名	代尊闯
专业班级	计算机科学与技术 02 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

十三、　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 。

十四、 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。

十五、 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@remnantsaint:/home# useradd -c bar -s /bin/sh -m foo
root@remnantsaint:/home# ls
foo  liuziqi
root@remnantsaint:/home# passwd foo
输入新的 UNIX 密码：
重新输入新的 UNIX 密码：
passwd：已成功更新密码
```

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 ；

```
foo:x:2000:1001:bar:/home/foo:/bin/csh
```

3. 从用户 foo 切换到 root ；

```
$ exit
```

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@remnantsaint:/home# userdel -rf foo
userdel: foo 邮件池 (/var/mail/foo) 未找到
```

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@remnantsaint:/home/liuziqi# newusers 1
root@remnantsaint:/home/liuziqi# cat 2 | chpasswd
```

```
z1:x:1099:1099::/home/z1:/bin/bash
z2:x:1100:1100::/home/z2:/bin/bash
```

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

```
root@remnantsaint:/home/liuziqi# groupadd -g 3000 group1
```

7. 在用户组 group1 中添加两个之前批量创建的用户；

```
root@remnantsaint:/home/liuziqi# usermod -G group1 z1
root@remnantsaint:/home/liuziqi# usermod -G group1 z2
```

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‘

```
# User privilege specification
root    ALL=(ALL:ALL) ALL
z1      ALL=(ALL:ALL) ALL
z2      ALL=(ALL:ALL) ALL
```

```
z1@remnantsaint:~$ sudo cat /etc/shadow
[sudo] z1 的密码：
root:$6$1ucuFd38$GpkMuwteBABLZYNBm1RT.WZGEyRgFDMzzEmZekRq9VU
wTOAvqrznwbG.umk8qze6nl0UtLyOpOr/AsWuecJI9.:19111:0:99999:7:
::
daemon:*:18480:0:99999:7:::
bin:*:18480:0:99999:7:::
sys:*:18480:0:99999:7:::
sync:*:18480:0:99999:7:::
games:*:18480:0:99999:7:::
man:*:18480:0:99999:7:::
```

十六、　　实验过程分析与讨论

创建用户的各种信息需要牢记。

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 6 日
学　　号	2021213097	姓　　名	代尊闯
专业班级	计算机科学与技术 02 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

十七、　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。

十八、 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。

十九、 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@remnantsaint:/home# foo=200
root@remnantsaint:/home# 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@remnantsaint:/home# bar=100
root@remnantsaint:/home# test $bar -gt 150
root@remnantsaint:/home# echo $?
1
root@remnantsaint:/home#
```

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

echo $HOSTNAME
date
```

4. 创建一个Shell程序，要求可以处理一个输入参数，判断该输入参数是否为水仙花数；

所谓水仙花数是指一个 3 位数，该数字每位数字的 3 次幂之和等于其本身，例如：

```
153 == 1^3+3^3+5^3
```

根据上述定义 153 是水仙花数。编写程序时要求首先进行输入参数个数判断，判断是否有输入参数存在：如果没有则给出提示信息；否则给出该数是否是水仙花数。要求对 153 、 124 和 370 进行测试判断。

```
#!/bin/bash
read -p "请输入测试数据个数： " n
for (( i=1; i<=n; i=i+1))
do
        read -p "请输入测试数据： " y
        x=y
        sum=0
        n1=$(($x%10))
        n1=$(($n1*$n1*$n1))
        sum=$(($sum+$n1))
        x=$(($x/10))
        n1=$(($x%10))
        n1=$(($n1*$n1*$n1))
        sum=$(($sum+$n1))
        x=$(($x/10))
        n1=$x
        n1=$(($n1*$n1*$n1))
        sum=$(($sum+$n1))
        echo "结果为： $sum"
        test $sum -eq $y && echo "是水仙花数" || echo "不是>
水仙花数"
done
```

```
root@remnantsaint:/home/liuziqi/work# ./1.sh
请输入测试数据个数： 3
请输入测试数据： 153
结果为： 153
是水仙花数
请输入测试数据： 124
结果为： 73
不是水仙花数
请输入测试数据： 370
结果为： 370
是水仙花数
```

5. 创建一个Shell程序，输入 3 个参数，计算 3 个输入变量的和并输出；

```
#!/bin/bash

echo $(($1+$2+$3))
~
```

```
root@remnantsaint:/home/liuziqi/work# ./2.sh 1 2 3
6
```

6. 创建一个Shell程序，输入学生成绩，给出该成绩对应的等级： 90 分以上为 A ， 80-90 为 B ， 70-80 为 C ， 60-70 为 D ，小于 60 分为 E 。要求使用if elif else fi 实现

```
#!/bin/bash

read -p "输入成绩: " x

if [ "$x" -gt "90" ]; then
        echo A
elif [ "$x" -ge "80" ]; then
        echo B
elif [ "$x" -ge "70" ]; then
        echo C
elif [ "$x" -ge "60" ]; then
        echo D
else
        echo E
fi
```

（注意双引号都可以不加）

```
root@remnantsaint:/home/liuziqi/work# ./3.sh
输入成绩: 77
C
```

二十、　　实验过程分析与讨论

学习 shell 编程的使用，包括条件判断和循环，学会用 shell 编程中数据比较和数据运算

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 13 日
学　　号	2021213097	姓　　名	代尊闯
专业班级	计算机科学与技术 02 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

二十一、 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

二十二、 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。

二十三、 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。

```
#!/bin/bash

for i in $(ls)
do
        test ${i##*.}  = "c" && cp $i "/home/liuziqi/workpla
ce"
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；

```
#!/bin/bash

sum=0
cnt=0
a=0
while true
do
        sum=$((sum+a))
        a=$((a+2))
        cnt=$((cnt+1))
        if [ $cnt == 10 ]; then
                break
        fi
done
echo $sum
```

```
root@remnantsaint:/home/liuziqi/work# ./5.sh
90
```

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

```
#!/bin/bash

sum=0
cnt=1
a=0
until [ $cnt == 11 ]
do
        sum=$((sum+cnt*cnt))
        cnt=$((cnt+1))
done
echo $sum
```

```
root@remnantsaint:/home/liuziqi/work# ./5.sh
385
```

4. 运行下列程序，并观察程序的运行结果。将程序中的 `---` 分别替换为 `break`、`break 2`、`continue`、`continue 2`，并观察四种情况下的实验结果。

```
#!/bin/bash

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

Break 2和break的区别就是跳出两层循环，外层也不运行了，直接结束，continue 2同理

二十四、 实验过程分析与讨论

。加深了对 if 判断和 while\until\for 循环的使用，同时强化了对各种括号用法的理解，还有脚本中对文件的操作和判断

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 20 日
学　　号	2021213097	姓　　名	代尊闯
专业班级	计算机科学与技术 02 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

二十五、 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。

二十六、 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。

二十七、 实验内容及结果

1. 编写 Shell 脚本，实现一个函数，对两个数的和进行求解，并输出结果；

```
#!/bin/bash

function sum(){
        echo $(($1+$2))
}
read a b
sum a b
```

其中 sum $a $b 可能更标准

```
root@remnantsaint:/home/liuziqi/workplace# ./1.sh
2 7
9
```

2. 编写 Shell 脚本，在脚本中定义一个递归函数，实现 n 的阶乘的求解；

```
#!/bin/bash

read q
function sum(){
        local n="$1"
        if [ "$n" -eq 1 ]; then
                result=1
        else
                let "m=n-1"
        sum "$m"
        let "result=$n * $?"
        fi
        return $result
}
sum "$q"
echo "$?"
```

```
root@remnantsaint:/home/liuziqi/workplace# ./2.sh
5
120
```

3. 一个 Shell 脚本的内容如下所示：

```
#!/bin/bash
function first() {
 function second() {
 function third() {
 echo "-3- here is in the third func."
 }
 echo "-2- here is in the second func."
 third
 }
```

```
    echo "-1- here is in the first func."
    second
}
echo "starting..."
first
```

试运行该程序，并观察程序运行结果，理解函数嵌套的含义。

```
root@remnantsaint:/home/liuziqi/workplace# ./3.sh
starting...
-1- here is in the first func.
-2- here is in the second func.
-3- here is in the third func.
```

Shell编程中允许函数内定义函数，循环创建函数，在第一个函数外也可以运行

二十八、 实验过程分析与讨论

加深了对函数的理解和运用，更强化了循环的 if 判断

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 27 日
学　　号	2021213097	姓　　名	代尊闯
专业班级	计算机科学与技术 02 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

二十九、 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。

三十、 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。

三十一、 实验内容及结果

1. 文件 quote.txt 的内容如下所示：

```
The honeysuckle band played all night long for only $90.
It was an evening of splendid music and company.
Too bad the disco floor fell through at 23:10.
The local nurse Miss P.Neave was in attendance.
```

试使用 sed 命令实现如下功能：

（1）删除 $ 符号；

```
root@remnantsaint:/home/liuziqi/workplace# sed -i s/'\$'//g quote.txt
root@remnantsaint:/home/liuziqi/workplace# 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.
```

（2）显示包含 music 文字的行内容及行号；

```
root@remnantsaint:/home/liuziqi/workplace# cat -n quote.txt | sed -n '/m
usic/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@remnantsaint:/home/liuziqi/workplace# sed -i '4a hello world!' quot
e.txt
root@remnantsaint:/home/liuziqi/workplace# 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.
hello world!
```

（4）将文本 "The" 替换为 "Quod" ；

```
root@remnantsaint:/home/liuziqi/workplace# sed -i s/'The'/'Quod'/g quote
.txt
root@remnantsaint:/home/liuziqi/workplace# cat 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.
hello world!
```

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

```
root@remnantsaint:/home/liuziqi/workplace# sed -ip '3c This is the third
 line.' quote.txt
root@remnantsaint:/home/liuziqi/workplace# cat quote.txt
Quod honeysuckle band played all night long for only 90.
It was an evening of splendid music and company.
This is the third line.
Quod local nurse Miss P.Neave was in attendance.
hello world!
```

（6）删除第 2 行内容；

```
root@remnantsaint:/home/liuziqi/workplace# sed -i '2d' quote.txt
root@remnantsaint:/home/liuziqi/workplace# cat quote.txt
Quod honeysuckle band played all night long for only 90.
This is the third line.
Quod local nurse Miss P.Neave was in attendance.
hello world!
```

（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@remnantsaint:/home/liuziqi/workplace# cat quote.txt | sed -n "/${va
r}/p"
Quod honeysuckle band played all night long for only 90.
Quod local nurse Miss P.Neave was in attendance.
```

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@remnantsaint:/home/liuziqi/workplace# cat numbers.txt | awk -F ' '
'{print $3}
> '
two
five

root@remnantsaint:/home/liuziqi/workplace# cat numbers.txt | awk -F ':'
'{print $2}
> '
 two
 five
```

可以看出来，在以:为分割符的时候空格也算在了字符串内

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
```

```
#!/bin/bash

numbers=0
echo "even: "
for i in $(cat /home/liuziqi/workplace/foo.txt)
do
        if [ $((i%2)) -eq 0 ]; then
                echo $i
                numbers=$((numbers+1))
        fi
done
echo "numbers: "
echo $numbers
```

```
root@remnantsaint:/home/liuziqi/workplace# ./1.sh
even:
2
4
46
numbers:
3
```

4. 脚本的内容如下所示：

```
#!/bin/bash

read –p "enter search pattern: " pattern

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

试运行该脚本，并理解该脚本实现的功能

该脚本是分析info.txt文件，查找输入的字符串匹配的行数，输出出现的行和出现的次数

三十二、 实验过程分析与讨论

对文件处理命令 sed 和 awk 加深了理解，还有 shell 脚本的编写。

五、指导教师意见

指导教师签字：卢洋