系统开发工具基础实验报告

姓名: 李佳潼　学号: 22020007043　年级专业: 2022 级计算机科学与技术

实验名称: Shell&Vim& 数据整理　实验时间: 2024 年 8 月 30 日

1 实验目的

- 学习 Shell 工具和脚本、Vim 编辑器以及数据整理方面的相关基础知识
- 学习 Shell 工具使用的基本命令以及使用 bash 脚本语言编写基础脚本并运行的相关操作
- 学习使用 Vim 编辑器编写文件的基本操作以及常用扩展插件的安装与使用
- 掌握使用基本命令（sed 等）完成数据整理的相关操作，实现基本问题的解决

2 实验内容及结果

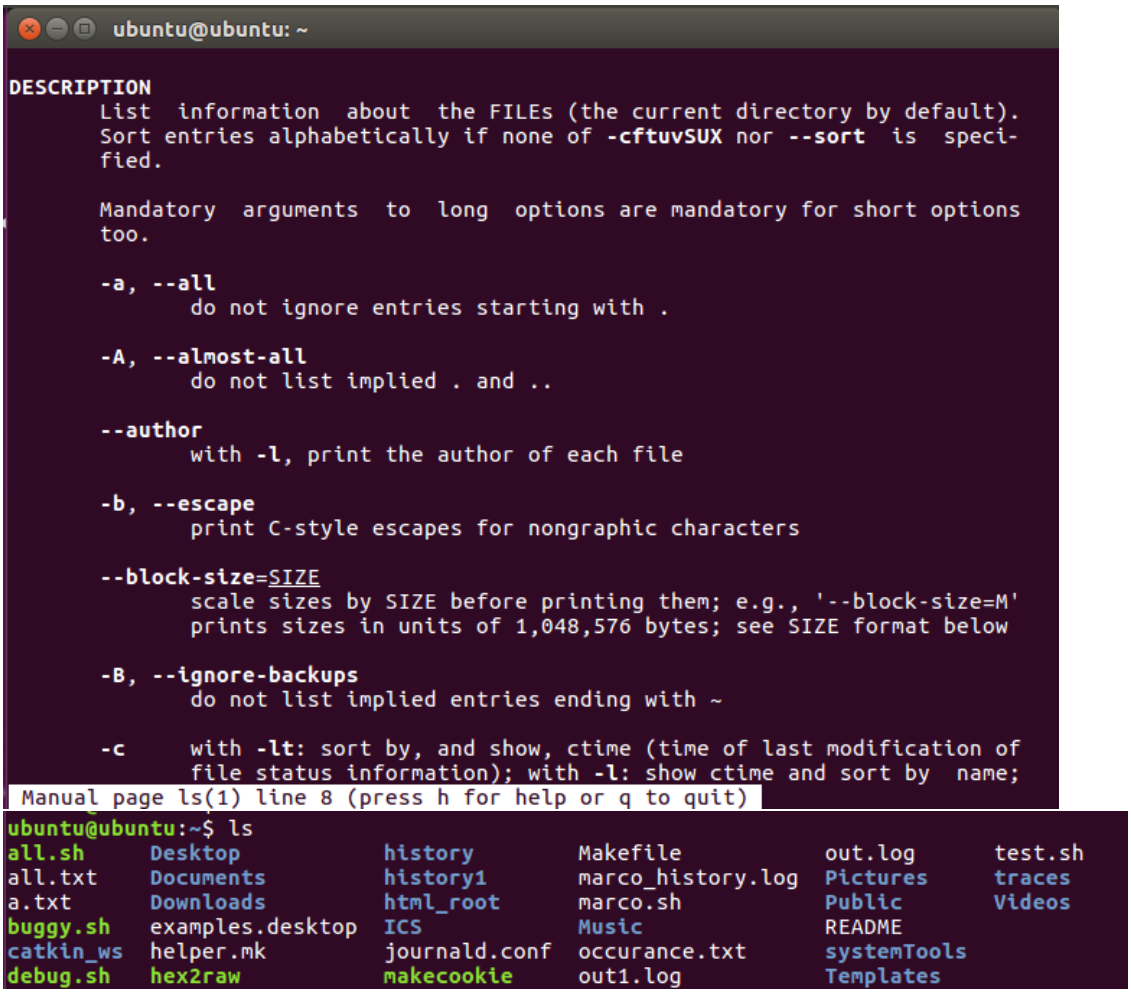
由于本次实验课下练习为课堂练习的延伸，故不拆分为【课上练习】与【课堂练习】两部分，仅根据 Shell、Vim、数据整理三个方面分别进行练习展示

- Shell

1. 阅读 man ls

```
man ls
```

```
ubuntu@ubuntu:~$ echo $SHELL
/bin/bash
ubuntu@ubuntu:~$ man ls
```



2. 查看所有文件（包括隐藏文件）

```
ls -l -a
```

```
ubuntu@ubuntu:~$ ls -l -a
total 272
drwxr-xr-x 23 ubuntu ubuntu  4096 Aug 29 17:39 .
drwxr-xr-x  3 root   root    4096 Sep 19  2023 ..
-rw-------  1 ubuntu ubuntu  8751 Jun 21 10:35 .bash_history
-rw-r--r--  1 ubuntu ubuntu   220 Sep 19  2023 .bash_logout
-rw-r--r--  1 ubuntu ubuntu  3842 Apr  6 07:32 .bashrc
drwx------ 18 ubuntu ubuntu  4096 Jan  2  2024 .cache
drwxrwxr-x  5 ubuntu ubuntu  4096 Apr  6 07:32 catkin_ws
-rw-r--r--  1 ubuntu ubuntu   735 Aug 25  2023 .clang-format
drwx------  3 ubuntu ubuntu  4096 Oct  9  2023 .compiz
drwx------ 16 ubuntu ubuntu  4096 Jan  2  2024 .config
drwxr-xr-x  2 ubuntu ubuntu  4096 Sep 19  2023 Desktop
-rw-r--r--  1 ubuntu ubuntu    25 Sep 19  2023 .dmrc
drwxr-xr-x  2 ubuntu ubuntu  4096 Sep 19  2023 Documents
drwxr-xr-x  2 ubuntu ubuntu  4096 Dec  4  2023 Downloads
-rw-r--r--  1 ubuntu ubuntu   347 Aug 25  2023 .editorconfig
-rw-r--r--  1 ubuntu ubuntu  8980 Sep 19  2023 examples.desktop
drwx------  2 ubuntu ubuntu  4096 Sep 19  2023 .gconf
-rw-r--r--  1 ubuntu ubuntu    93 Aug 25  2023 .gitignore
drwx------  3 ubuntu ubuntu  4096 Aug 29 17:36 .gnupg
-rw-------  1 ubuntu ubuntu 12288 Sep 20  2023 .HelloWorld.c.swm
-rw-------  1 ubuntu ubuntu 12288 Sep 19  2023 .HelloWorld.c.swn
-rw-------  1 ubuntu ubuntu 12288 Sep 19  2023 .HelloWorld.c.swo
-rw-------  1 ubuntu ubuntu 12288 Sep 19  2023 .HelloWorld.c.swp
-rw-r--r--  1 ubuntu ubuntu  5116 Aug 25  2023 helper.mk
-rwxr-xr-x  1 ubuntu ubuntu 12108 Dec 12  2018 hex2raw
drwxrwxr-x  2 ubuntu ubuntu  4096 Aug 29 17:39 history
-rw-------  1 ubuntu ubuntu 13992 Aug 29 17:36 .ICEauthority
drwxrwxr-x  2 ubuntu ubuntu  4096 Sep 20  2023 ICS
-rw-r--r--  1 ubuntu ubuntu    45 Aug 27  2023 .labname.mk
drwx------  3 ubuntu ubuntu  4096 Sep 19  2023 .local
-rwxr-xr-x  1 ubuntu ubuntu  7584 Dec 12  2018 makecookie
-rw-r--r--  1 ubuntu ubuntu  1015 Oct  9  2023 Makefile
drwx------  5 ubuntu ubuntu  4096 Sep 19  2023 .mozilla
drwxr-xr-x  2 ubuntu ubuntu  4096 Sep 19  2023 Music
drwxr-xr-x  2 ubuntu ubuntu  4096 Sep 19  2023 Pictures
-rw-r--r--  1 ubuntu ubuntu   655 Sep 19  2023 .profile
drwxr-xr-x  2 ubuntu ubuntu  4096 Sep 19  2023 Public
-rw-r--r--  1 ubuntu ubuntu  2961 Aug 25  2023 README
drwxrwxr-x  4 ubuntu ubuntu  4096 Apr  6 07:33 .ros
drwx------  2 ubuntu ubuntu  4096 Mar 21 04:22 .ssh
-rw-r--r--  1 root   root       0 Sep 19  2023 .sudo_as_admin_successful
drwxr-xr-x  2 ubuntu ubuntu  4096 Sep 19  2023 Templates
drwxrwxr-x  2 ubuntu ubuntu  4096 Sep 20  2023 traces
```

3. 查看文件，打印以人类可以理解的格式输出

```
ls -l -h
```

```
ubuntu@ubuntu:~$ ls -l -h
total 156K
-rwxrwxrwx 1 ubuntu ubuntu   67 Sep  3 18:17 all.sh
-rw-rw-r-- 1 ubuntu ubuntu 2.0K Sep  3 18:18 all.txt
-rw-rw-r-- 1 ubuntu ubuntu    5 Sep  3 17:56 a.txt
-rwxrwxrwx 1 ubuntu ubuntu  190 Aug 29 18:18 buggy.sh
drwxrwxr-x 5 ubuntu ubuntu 4.0K Apr  6 07:32 catkin_ws
-rwxrwxrwx 1 ubuntu ubuntu  168 Aug 29 18:32 debug.sh
drwxr-xr-x 2 ubuntu ubuntu 4.0K Sep 19  2023 Desktop
drwxr-xr-x 2 ubuntu ubuntu 4.0K Sep 19  2023 Documents
drwxr-xr-x 2 ubuntu ubuntu 4.0K Dec  4  2023 Downloads
-rw-r--r-- 1 ubuntu ubuntu 8.8K Sep 19  2023 examples.desktop
-rw-r--r-- 1 ubuntu ubuntu 5.0K Aug 25  2023 helper.mk
-rwxr-xr-x 1 ubuntu ubuntu  12K Dec 12  2018 hex2raw
drwxrwxr-x 2 ubuntu ubuntu 4.0K Sep  3 17:39 history
drwxrwxr-x 2 ubuntu ubuntu 4.0K Sep  3 17:55 history1
drwxrwxr-x 3 ubuntu ubuntu 4.0K Aug 29 20:20 html_root
drwxrwxr-x 2 ubuntu ubuntu 4.0K Sep 20  2023 ICS
-rw-r--r-- 1 root   root      0 Sep  3 18:24 journald.conf
-rwxr-xr-x 1 ubuntu ubuntu 7.5K Dec 12  2018 makecookie
-rw-r--r-- 1 ubuntu ubuntu 1015 Oct  9  2023 Makefile
```

4. 查看文件，以最近访问顺序排序

```
ls -l -t
```

```
ubuntu@ubuntu:~$ ls -l -t
total 96
drwxrwxr-x 2 ubuntu ubuntu  4096 Aug 29 17:39 history
drwxrwxr-x 5 ubuntu ubuntu  4096 Apr  6 07:32 catkin_ws
drwxr-xr-x 2 ubuntu ubuntu  4096 Dec  4  2023 Downloads
-rw-r--r-- 1 ubuntu ubuntu  1015 Oct  9  2023 Makefile
drwxrwxr-x 2 ubuntu ubuntu  4096 Sep 20  2023 traces
drwxrwxr-x 2 ubuntu ubuntu  4096 Sep 20  2023 ICS
drwxr-xr-x 2 ubuntu ubuntu  4096 Sep 19  2023 Desktop
drwxr-xr-x 2 ubuntu ubuntu  4096 Sep 19  2023 Documents
drwxr-xr-x 2 ubuntu ubuntu  4096 Sep 19  2023 Music
drwxr-xr-x 2 ubuntu ubuntu  4096 Sep 19  2023 Pictures
drwxr-xr-x 2 ubuntu ubuntu  4096 Sep 19  2023 Public
drwxr-xr-x 2 ubuntu ubuntu  4096 Sep 19  2023 Templates
drwxr-xr-x 2 ubuntu ubuntu  4096 Sep 19  2023 Videos
-rw-r--r-- 1 ubuntu ubuntu  8980 Sep 19  2023 examples.desktop
-rw-r--r-- 1 ubuntu ubuntu  5116 Aug 25  2023 helper.mk
-rw-r--r-- 1 ubuntu ubuntu  2961 Aug 25  2023 README
-rwxr-xr-x 1 ubuntu ubuntu 12108 Dec 12  2018 hex2raw
-rwxr-xr-x 1 ubuntu ubuntu  7584 Dec 12  2018 makecookie
```

5. 查看文件，以彩色文本显示输出结果

```
ls -l --color=auto
```

```
ubuntu@ubuntu:~$ ls -l --color=auto
total 96
drwxrwxr-x 5 ubuntu ubuntu  4096 Apr  6 07:32 catkin_ws
drwxr-xr-x 2 ubuntu ubuntu  4096 Sep 19  2023 Desktop
drwxr-xr-x 2 ubuntu ubuntu  4096 Sep 19  2023 Documents
drwxr-xr-x 2 ubuntu ubuntu  4096 Dec  4  2023 Downloads
-rw-r--r-- 1 ubuntu ubuntu  8980 Sep 19  2023 examples.desktop
-rw-r--r-- 1 ubuntu ubuntu  5116 Aug 25  2023 helper.mk
-rwxr-xr-x 1 ubuntu ubuntu 12108 Dec 12  2018 hex2raw
drwxrwxr-x 2 ubuntu ubuntu  4096 Aug 29 17:39 history
drwxrwxr-x 2 ubuntu ubuntu  4096 Sep 20  2023 ICS
-rwxr-xr-x 1 ubuntu ubuntu  7584 Dec 12  2018 makecookie
-rw-r--r-- 1 ubuntu ubuntu  1015 Oct  9  2023 Makefile
drwxr-xr-x 2 ubuntu ubuntu  4096 Sep 19  2023 Music
drwxr-xr-x 2 ubuntu ubuntu  4096 Sep 19  2023 Pictures
drwxr-xr-x 2 ubuntu ubuntu  4096 Sep 19  2023 Public
-rw-r--r-- 1 ubuntu ubuntu  2961 Aug 25  2023 README
drwxr-xr-x 2 ubuntu ubuntu  4096 Sep 19  2023 Templates
drwxrwxr-x 2 ubuntu ubuntu  4096 Sep 20  2023 traces
drwxr-xr-x 2 ubuntu ubuntu  4096 Sep 19  2023 Videos
```

6. 编写两个 bash 函数 marco 和 polo 执行下面的操作。每当你执行 marco 时，当前的工作目录应当以某种形式保存，当执行 polo 时，无论现在处在什么目录下，都应当 cd 回到当时执行 marco 的目录。为了方便 debug，你可以把代码写在单独的文件 marco.sh 中，并通过 source marco.sh 命令，(重新) 加载函数

   编写 bash 脚本及运行如下

```
ubuntu@ubuntu: ~
#!/bin/bash

# 保存当前工作目录
marco() {
    echo "$(pwd)" > $HOME/marco_history.log
    echo "save pwd$(pwd)"
}

# 返回到保存的工作目录
polo() {
    cd "$(cat "$HOME/marco_history.log")"
}

~
~
~
~
```

```
ubuntu@ubuntu:~$ ls
catkin_ws  Downloads         hex2raw  makecookie  Music     README     Videos
Desktop    examples.desktop  history  Makefile    Pictures  Templates
Documents  helper.mk         ICS      marco.sh    Public    traces
ubuntu@ubuntu:~$ source marco.sh
ubuntu@ubuntu:~$ marco
save pwd/home/ubuntu
ubuntu@ubuntu:~$ cd
ubuntu@ubuntu:~$ polo
ubuntu@ubuntu:~$ cd /tmp
ubuntu@ubuntu:/tmp$ polo
ubuntu@ubuntu:~$
```

7. 编写一段 bash 脚本，运行如下的脚本直到它出错，将它的标准输出和标准错误流记录到文件，并在最后输出所有内容。加分项：报告脚本在失败前共运行了多少次

   分别编写 buggy.sh 和 debug.sh

```
ubuntu@ubuntu:~$ vim buggy.sh
ubuntu@ubuntu:~$ vim debug.sh
```

   buggy.sh 内容如下

```
ubuntu@ubuntu: /
n=$(( RANDOM % 100 ))

if [[ n -eq 42 ]]; then
    echo "Something went wrong"
    >&2 echo "The error was using magic numbers"
    exit 1
fi

echo "Everything went according to plan"
~
~
```

   首次 debug.sh

```
ubuntu@ubuntu: ~
count=0
 echo > out.log

 Files
 do
     ./buggy.sh &>> out.log
     if [[ $? -ne 0 ]]; then
         cat out.log
         echo "failed after $count times"
         break
     fi
     ((count++))

 done
~
```

```
Everything went according to plan
Everything went according to plan
Everything went according to plan
Everything went according to plan
Everything went according to plan
Everything went according to plan
Everything went according to plan
Everything went according to plan
Everything went according to plan
Everything went according to plan
Everything went according to plan
Everything went according to plan
Everything went according to plan
Everything went according to plan
Something went wrong
The error was using magic numbers
failed after 60 times
ubuntu@ubuntu:~$ cat out.log | grep Everything | wc -l
60
```

脚本在失败前共运行了 60 次

二次 debug.sh

```
ubuntu@ubuntu: ~
echo > out.log
 for ((count=0;;count++))
 do
     ./buggy.sh &>> out.log
     if [[ $? -ne 0 ]]; then
         echo "failed after $count times"
         break

     fi
 done
~
```

```
ubuntu@ubuntu:~$ ./debug.sh
failed after 59 times
ubuntu@ubuntu:~$ cat out.log | grep Everything | wc -l
59
```

脚本在失败前共运行了 59 次

三次 debug.sh

```
ubuntu@ubuntu: ~
count=0
 ./buggy.sh &>> out.log
 until [[ "$?" -ne 0 ]];
 do
     count=$((count+1))
     ./buggy.sh &>> out.log
 done

 echo "failed after $count runs"
~
~
ubuntu@ubuntu:~$ ./debug.sh
failed after 9 runs
ubuntu@ubuntu:~$ cat out.log | grep Everything | wc -l
9
```

脚本在失败前共运行了 9 次

8. 递归地查找文件夹中所有的 HTML 文件，并将它们压缩成 zip 文件。注意，即使文件名中包含空格，您的命令也应该能够正确执行

首先创建所需的文件

```
ubuntu@ubuntu:~$ mkdir html_root
ubuntu@ubuntu:~$ cd html_root
ubuntu@ubuntu:~/html_root$ touch {1..10}.html
ubuntu@ubuntu:~/html_root$ mkdir html
ubuntu@ubuntu:~/html_root$ cd html
ubuntu@ubuntu:~/html_root/html$ touch xxxx.html
```

执行 find 命令

```
ubuntu@ubuntu:~/html_root/html$ find . -type f -name "*.html" | xargs -d '\n'  t
ar -cvzf html.zip
./xxxx.html
ubuntu@ubuntu:~/html_root/html$ cd ../
ubuntu@ubuntu:~/html_root$ find . -type f -name "*.html" | xargs -d '\n'  tar -c
vzf html.zip
./7.html
./html/xxxx.html
./9.html
./2.html
./3.html
./8.html
./1.html
./4.html
./6.html
./10.html
./5.html
```

可以发现查找到.html 后缀的所有文件，并压缩为.zip 文件

9. 编写一个命令或脚本递归的查找文件夹中最近使用的文件。更通用的做法，你可以按照最近的使用时间列出文件吗?

```
ubuntu@ubuntu:~/html_root$ find . -type f -print0 | xargs -0 ls -lt | head -1
-rw-rw-r-- 1 ubuntu ubuntu 212 Aug 29 18:42 ./html.zip
ubuntu@ubuntu:~/html_root$ find . -type f -mmin -60 -print0 | xargs -0 ls -lt |
head -10
-rw-rw-r-- 1 ubuntu ubuntu 212 Aug 29 18:42 ./html.zip
-rw-rw-r-- 1 ubuntu ubuntu 116 Aug 29 18:42 ./html/html.zip
-rw-rw-r-- 1 ubuntu ubuntu   0 Aug 29 18:41 ./html/xxxx.html
-rw-rw-r-- 1 ubuntu ubuntu   0 Aug 29 18:40 ./10.html
-rw-rw-r-- 1 ubuntu ubuntu   0 Aug 29 18:40 ./1.html
-rw-rw-r-- 1 ubuntu ubuntu   0 Aug 29 18:40 ./2.html
-rw-rw-r-- 1 ubuntu ubuntu   0 Aug 29 18:40 ./3.html
-rw-rw-r-- 1 ubuntu ubuntu   0 Aug 29 18:40 ./4.html
-rw-rw-r-- 1 ubuntu ubuntu   0 Aug 29 18:40 ./5.html
-rw-rw-r-- 1 ubuntu ubuntu   0 Aug 29 18:40 ./6.html
```

10. 显示当前工作目录的路径

```
pwd
```

```
ubuntu@ubuntu:~$ pwd
/home/ubuntu
```

11. 列出当前目录中的文件和子目录

```
ls
```

```
ubuntu@ubuntu:~$ ls
all.sh      Desktop           history        Makefile           out.log      test.sh
all.txt     Documents         history1       marco_history.log  Pictures     traces
a.txt       Downloads         html_root      marco.sh           Public       Videos
buggy.sh    examples.desktop  ICS            Music              README
catkin_ws   helper.mk         journald.conf  occurance.txt      systemTools
debug.sh    hex2raw           makecookie     out1.log           Templates
```

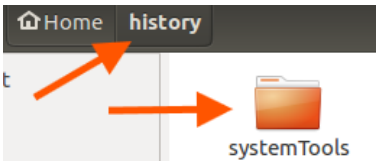
12. 更改当前工作目录

```
cd [directory]
```

```
ubuntu@ubuntu:~$ cd systemTools
ubuntu@ubuntu:~/systemTools$
```

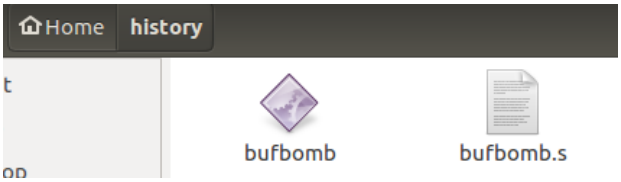
13. 创建一个新目录

```
mkdir [directory]
```



14. 删除一个空目录

```
rmdir [directory]
```

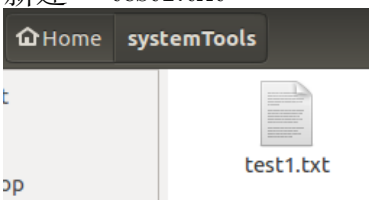


可以发现主目录下 systemTools 空目录已经被删除

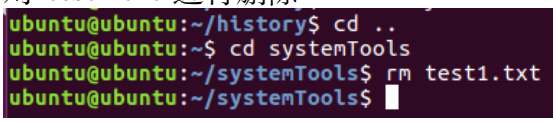
15. 删除文件

```
rm [file]
```

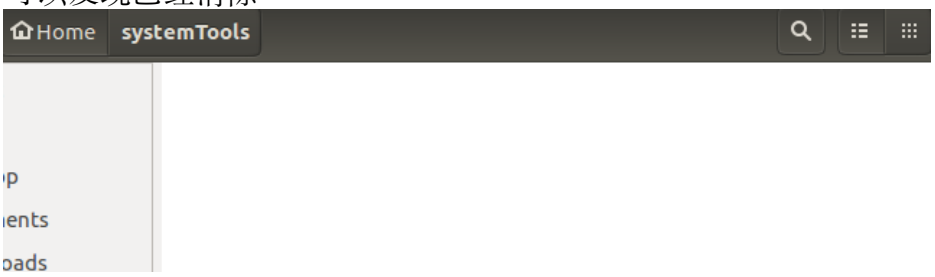
新建一 test1.txt



对 test1.txt 进行删除



可以发现已经清除



16. 显示文件内容

```
cat [file]
```

```
ubuntu@ubuntu:~/systemTools$ cat test2.txt
this is a test txt.
```

17. 检查与目标主机的连通性

```
ping [hostname]
```

```
ubuntu@ubuntu:~$ ping localhost
PING localhost (127.0.0.1) 56(84) bytes of data.
64 bytes from localhost (127.0.0.1): icmp_seq=1 ttl=64 time=0.058 ms
64 bytes from localhost (127.0.0.1): icmp_seq=2 ttl=64 time=0.072 ms
64 bytes from localhost (127.0.0.1): icmp_seq=3 ttl=64 time=0.047 ms
64 bytes from localhost (127.0.0.1): icmp_seq=4 ttl=64 time=0.050 ms
64 bytes from localhost (127.0.0.1): icmp_seq=5 ttl=64 time=0.047 ms
64 bytes from localhost (127.0.0.1): icmp_seq=6 ttl=64 time=0.050 ms
64 bytes from localhost (127.0.0.1): icmp_seq=7 ttl=64 time=0.044 ms
64 bytes from localhost (127.0.0.1): icmp_seq=8 ttl=64 time=0.049 ms
64 bytes from localhost (127.0.0.1): icmp_seq=9 ttl=64 time=0.067 ms
^C
--- localhost ping statistics ---
9 packets transmitted, 9 received, 0% packet loss, time 8198ms
rtt min/avg/max/mdev = 0.044/0.053/0.072/0.012 ms
```

18. 显示或配置网络接口（部分系统需用 ip addr 替代）

```
ifconfig
```

```
ubuntu@ubuntu:~$ ifconfig
ens33     Link encap:Ethernet  HWaddr 00:0c:29:f4:82:84
          inet addr:192.168.56.128  Bcast:192.168.56.255  Mask:255.255.255.0
          inet6 addr: fe80::49eb:9c07:991f:7c3/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:207 errors:0 dropped:0 overruns:0 frame:0
          TX packets:218 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:108613 (108.6 KB)  TX bytes:23208 (23.2 KB)
          Interrupt:19 Base address:0x2000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:449 errors:0 dropped:0 overruns:0 frame:0
          TX packets:449 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:65062 (65.0 KB)  TX bytes:65062 (65.0 KB)
```

19. 显示网络连接、路由表和接口状态

```
netstat
```

```
ubuntu@ubuntu:~$ netstat
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
udp        0      0 localhost:44608         localhost:44608         ESTABLISHED
Active UNIX domain sockets (w/o servers)
Proto RefCnt Flags       Type       State         I-Node   Path
unix  2      [ ]         DGRAM                    12916    /run/systemd/cgroups-agent
unix  2      [ ]         DGRAM                    23437    /run/user/1000/systemd/notif
y
unix  2      [ ]         DGRAM                    12922    /run/systemd/journal/syslog
unix  15     [ ]         DGRAM                    12926    /run/systemd/journal/dev-log
unix  7      [ ]         DGRAM                    12929    /run/systemd/journal/socket
unix  3      [ ]         DGRAM                    12915    /run/systemd/notify
unix  3      [ ]         STREAM     CONNECTED     24478    @/tmp/.X11-unix/X0
unix  3      [ ]         STREAM     CONNECTED     25124
unix  3      [ ]         STREAM     CONNECTED     26306
unix  3      [ ]         STREAM     CONNECTED     20626
unix  3      [ ]         STREAM     CONNECTED     27020    @/tmp/ibus/dbus-sJkBxcpF
unix  3      [ ]         STREAM     CONNECTED     20627    /var/run/dbus/system_bus_soc
ket
unix  3      [ ]         STREAM     CONNECTED     18510    /var/run/dbus/system_bus_soc
ket
```

20. 查找所有指定名称的文件夹

```
find . -name <名称> -type d
```

```
ubuntu@ubuntu:~$ find . -name history -type d
./history
```

21. 查找所有文件夹路径中包含 test 的 txt 文件

```
find . -path '*/test/*.txt -type f
```

```
ubuntu@ubuntu:~/systemTools$ find . -path '*/test/*.txt' -type f
ubuntu@ubuntu:~/systemTools$
```

22. 查找前一天修改的所有文件

```
find . -mtime -1
```

```
ubuntu@ubuntu:~$ find . -mtime -1
.
./.bash_history
./.ICEauthority
./.gnupg
./.gnupg/S.gpg-agent
./history
./.cache/compizconfig-1
./.cache/compizconfig-1/copytex.pb
./.cache/update-manager-core/meta-release-lts
./.cache/upstart
./.cache/upstart/unity7.log.1.gz
./.cache/upstart/hud.log
./.cache/upstart/unity-settings-daemon.log
./.cache/upstart/update-notifier-release.log.1.gz
./.cache/upstart/gnome-keyring-ssh.log.1.gz
./.cache/upstart/dbus.log.1.gz
./.cache/upstart/indicator-sound.log.1.gz
./.cache/upstart/unity-panel-service.log.1.gz
./.cache/upstart/upstart-event-bridge.log.1.gz
```

23. 查找所有大小在 500k 至 10M 的压缩文件

```
find . -size +500k -size -10M -name '*.tar.gz'
```

```
ubuntu@ubuntu:~/history$ find . -size +500k -size -10M -name '*.tar.gz'
ubuntu@ubuntu:~/history$
```

24. 显示简单文本

```
echo "Hello, World!"
```

```
ubuntu@ubuntu:~$ echo "Hello, world!"
Hello, world!
```

25. 使用转义字符

```
echo -e "Hello\nWorld!"
```

```
ubuntu@ubuntu:~$ echo -e "Hello\nWorld!"
Hello
World!
```

26. 显示变量的值

```
name="Alice"
echo "Hello, $name!"
```

```
ubuntu@ubuntu:~$ name="Alice"
ubuntu@ubuntu:~$ echo "Hello, $name!"
Hello, Alice!
```

27. 查看系统中正在运行的进程的输出

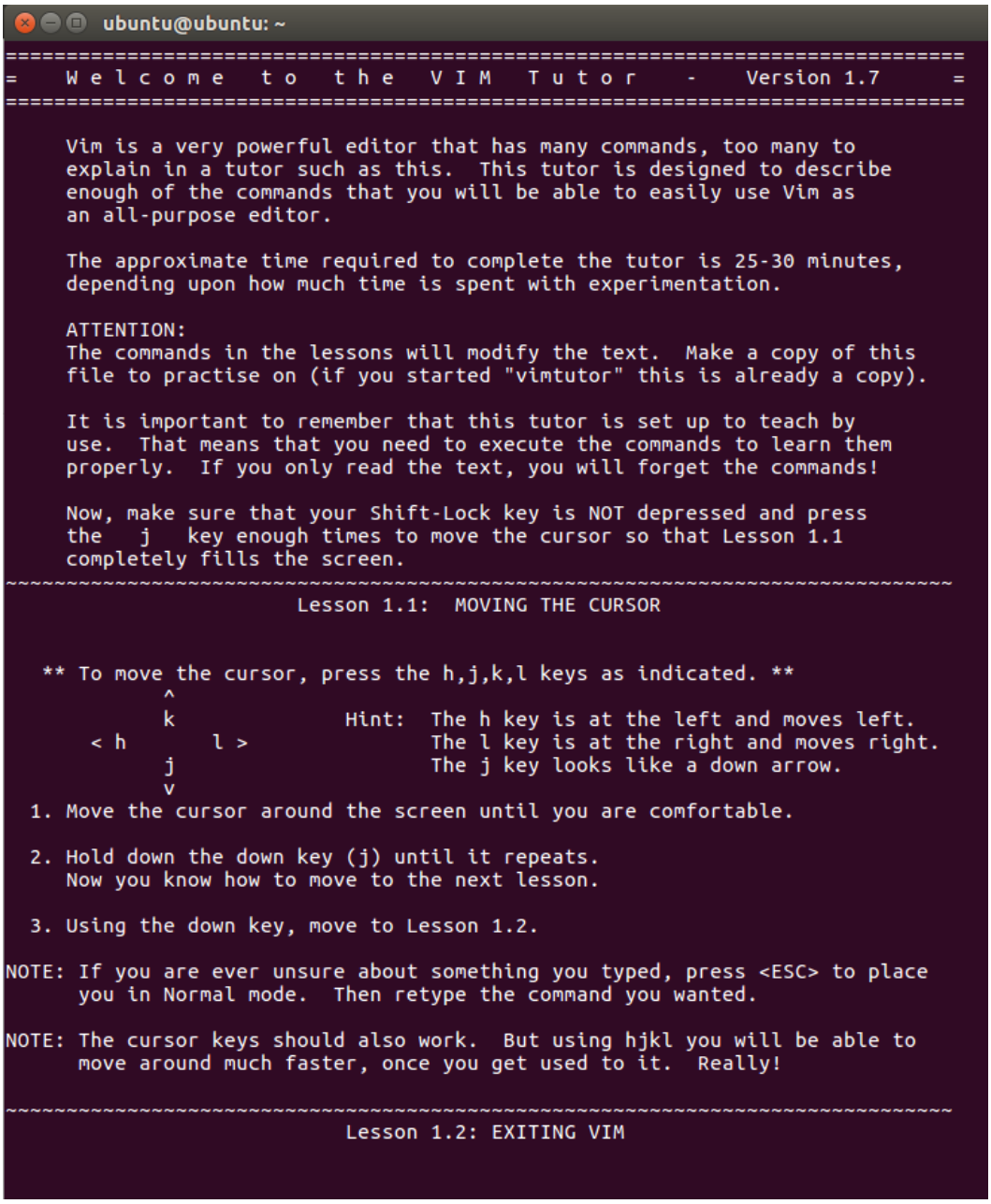
```
ps -aux
```

```
ubuntu@ubuntu:~/systemTools$ ps -aux
USER        PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root          1  0.2  0.1  24044  5036 ?        Ss   17:34   0:04 /sbin/init auto noprom
root          2  0.0  0.0      0     0 ?        S    17:34   0:00 [kthreadd]
root          4  0.0  0.0      0     0 ?        I<   17:34   0:00 [kworker/0:0H]
root          6  0.0  0.0      0     0 ?        I<   17:34   0:00 [mm_percpu_wq]
root          7  0.0  0.0      0     0 ?        S    17:34   0:00 [ksoftirqd/0]
root          8  0.0  0.0      0     0 ?        I    17:34   0:01 [rcu_sched]
root          9  0.0  0.0      0     0 ?        I    17:34   0:00 [rcu_bh]
root         10  0.0  0.0      0     0 ?        S    17:34   0:00 [migration/0]
root         11  0.0  0.0      0     0 ?        S    17:34   0:00 [watchdog/0]
root         12  0.0  0.0      0     0 ?        S    17:34   0:00 [cpuhp/0]
root         13  0.0  0.0      0     0 ?        S    17:34   0:00 [cpuhp/1]
root         14  0.0  0.0      0     0 ?        S    17:34   0:00 [watchdog/1]
root         15  0.0  0.0      0     0 ?        S    17:34   0:00 [migration/1]
root         16  0.0  0.0      0     0 ?        S    17:34   0:00 [ksoftirqd/1]
root         18  0.0  0.0      0     0 ?        I<   17:34   0:00 [kworker/1:0H]
root         19  0.0  0.0      0     0 ?        S    17:34   0:00 [kdevtmpfs]
root         20  0.0  0.0      0     0 ?        I<   17:34   0:00 [netns]
root         21  0.0  0.0      0     0 ?        S    17:34   0:00 [rcu_tasks_kthre]
root         22  0.0  0.0      0     0 ?        S    17:34   0:00 [kauditd]
```

- Vim

1. 完成 vimtutor

```
vimtutor
```



2. 下载我们提供的 vimrc，然后把它保存到 /.vimrc。

```
ubuntu@ubuntu:~$ mkdir -p ~/.vim/pack/vendor/start
ubuntu@ubuntu:~$ ls
buggy.sh    Downloads         html_root          marco.sh  Public       Videos
catkin_ws   examples.desktop  ICS                Music     README
debug.sh    helper.mk         makecookie         out1.log  systemTools
Desktop     hex2raw           Makefile           out.log   Templates
Documents   history           marco_history.log  Pictures  traces
ubuntu@ubuntu:~$ cd ~/.vim/pack/vendor/start
ubuntu@ubuntu:~/.vim/pack/vendor/start$ git clone https://github.com/ctrlpvim/ct
rlp.vim
Cloning into 'ctrlp.vim'...
remote: Enumerating objects: 4299, done.
remote: Counting objects: 100% (168/168), done.
remote: Compressing objects: 100% (105/105), done.
remote: Total 4299 (delta 71), reused 147 (delta 62), pack-reused 4131 (from 1)
Receiving objects: 100% (4299/4299), 1.70 MiB | 440.00 KiB/s, done.
Resolving deltas: 100% (1661/1661), done.
Checking connectivity... done.
```

```
nnoremap <Left>  :echoe "Use h"<CR>
nnoremap <Right> :echoe "Use l"<CR>
nnoremap <Up>    :echoe "Use k"<CR>
nnoremap <Down>  :echoe "Use j"<CR>
" ...and in insert mode
inoremap <Left>  <ESC>:echoe "Use h"<CR>
inoremap <Right> <ESC>:echoe "Use l"<CR>
inoremap <Up>    <ESC>:echoe "Use k"<CR>
inoremap <Down>  <ESC>:echoe "Use j"<CR>

set runtimepath^=~/.vim/pack/vendor/start/ctrlp.vim
-- INSERT --
```

3. 使用 vim-plug 安装插件

```
ubuntu@ubuntu:~$ curl -fLo ~/.vim/autoload/plug.vim --create-dirs https://raw.gi
thubusercontent.com/junegunn/vim-plug/master/plug.vim
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 84223  100 84223    0     0   3753       0  0:00:22  0:00:22 --:--:-- 20216
```

```
ubuntu@ubuntu: ~
[Plugins] [No Name]                                                        X
  1 Installing plugins (0/2)
2   [__]
  1
  2 + hardmode: Cloning into '/home/ubuntu/.vim/pack/vendor/start/ctrlp.vim/plug
  3 + NERDTree: Cloning into '/home/ubuntu/.vim/pack/vendor/start/ctrlp.vim/plug
~
```

- 数据整理

1. 统计 words 文件 (/usr/share/dict/words) 中包含至少三个 a 且不以 s 结尾的单词个数

   大小写转换

```
ubuntu@ubuntu:~$ cat /usr/share/dict/words | tr "[:upper:]" "[:lower:]" | grep -E "^([^
a]*a){3}.*$" | grep -v "'s$" | wc -l
833
```

2. 查找一个以 a 结尾的字符串三次

```
ubuntu@ubuntu:~$ cat /usr/share/dict/words | tr "[:upper:]" "[:lower:]" | grep -E "^([^
a]*a){3}.*$" | grep -v "'s$" | sed -E "s/.*([a-z]{2})$/\1/" | sort | uniq -c | sort | t
ail -n3
      8 am
      8 ce
      9 ca
```

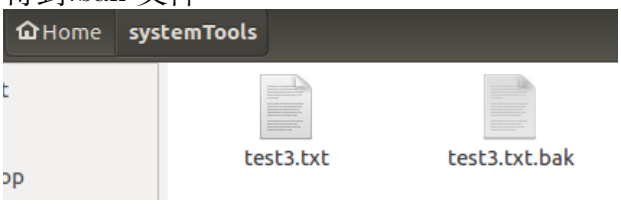
3. 匹配结尾为 s 的结果，然后取反

```
ubuntu@ubuntu:~$ cat /usr/share/dict/words | tr "[:upper:]" "[:lower:]" | grep -E "^([^
a]*a){3}.*$" | grep -v "'s$" | sed -E "s/.*([a-z]{2})$/\1/" | sort | uniq | wc -l
110
```

4. 得到没出现的组合

   编写 all.sh 文件并给权限

得到.bak 文件



3 实验总结与体会

通过学习 Shell 工具和脚本、Vim 编辑器以及数据整理方面的相关基础知识，了解了 Linux 系统基本终端命令以及编写文件的要求和注意事项，熟悉了查看文件夹、文件，编辑文件，数据查找整理等相关操作的基本流程。

同时通过练习也意识到，一些命令需要根据问题进行调整，灵活运用

附课程仓库地址：`https://github.com/Physical2/systemToolsBase`（点击跳转）