

shell 脚本与 vim 编辑实验报告

张宇鑫 22090032057

2024 年 9 月 11 日

1 实验目的

了解 shell 如何编写并且掌握基本编写技能，学会使用 vim 编辑文本。

2 实验内容

基于一个 first.sh 展开实验内容

man 查看 ls 的所有操作后，举例使用按时间排序查看，查看所有文件等

```
yangdog@yangdog-virtual-machine:~$ man ls
yangdog@yangdog-virtual-machine:~$ ls -a
.          kjv12.txt          .viminfo.tmp
..         .local          .viminfz.tmp
.bash_history  mapreduce_3.py      .Xauthority
.bash_logout  .mozilla            .xsession-errors
               pig.c              .xsession-errors.old
               .presage          公共的
               .profile         模板
               program          视频
examples.desktop program_protected 图片
.gconf        .ssh              文档
.gitconfig    .sudo_as_admin_successful 下载
.gnupg        test              音乐
.ICEauthority .viminfo
yangdog@yangdog-virtual-machine:~$ ls -h
examples.desktop pig.c          test      视频  下载
kjev12.txt       program        公共的   图片  音乐
mapreduce_3.py   program_protected 模板     文档  桌面
yangdog@yangdog-virtual-machine:~$ ls -t
下载  图片  pig.c          公共的  文档
桌面  program_protected kjv12.txt   模板    音乐
test  program        mapreduce_3.py 视频     examples.desktop
yangdog@yangdog-virtual-machine:~$ ls -color
```

查看当前根目录，cd 切换目录

```
yangdog@yangdog-virtual-machine:~$ pwd
/home/yangdog
yangdog@yangdog-virtual-machine:~$ cd
```

编写 first.sh:具体功能是给出一个随机数用户输入一个数判断是否相等，直到猜对结束

```
angdog@yangdog-virtual-machine:~$ touch first.sh
log-virtual-machine:~$ vim first.sh
```

```
#!/bin/bash
num=$((RANDOM%100+1))
echo "$num"

while :
do
    read -p cai
    if [$cai -eq $num]
    then
        echo "YES"
        exit
    elif [$cai -gt $num]
    then
        echo "bigger"
    else
        echo "smaller"
    fi
done
```

运行 first.sh

```
yangdog@yangdog-virtual-machine:~$ vim first.sh
yangdog@yangdog-virtual-machine:~$ . first.sh
60
```

给出数为 60

```
yangdog@yangdog-virtual-machine:~$ . first.sh
60
18
smaller
99
bigger
60
YES
```

输入 60 判断相等，结束

复制一个 first1.sh 文件与 first.sh 内容一致，然后 rm 删除

```
#!/bin/bash
num=$((RANDOM%100+1))
echo "$num"

while :
do
    read -p cai
    if [$cai -eq $num]
    then
        echo "YES"
        exit
    elif [$cai -gt $num]
    then
        echo "bigger"
    else
        echo "smaller"
    fi
done
```

LibreOffice Impress

"first1.sh" 18L, 191C

```
yangdog@yangdog-virtual-machine:~$ rm first1.sh
yangdog@yangdog-virtual-machine:~$ vim first1.sh
```

使用 cat 查看是否删除成功

```
yangdog@yangdog-virtual-machine:~$ cat first.sh
#!/bin/bash
num=$((RANDOM%100+1))
echo "$num"

while :
do
    read cai
    if [ $cai -eq $num ]
    then
        echo "YES"
    elif [ $cai -gt $num ]
    then
        echo "bigger"
    else
        echo "smaller"
    fi
done
yangdog@yangdog-virtual-machine:~$ cat first1.sh
```

使用 head 查看前五五行内容

```
yangdog@yangdog-virtual-machine:~$ head -5 first.sh
#!/bin/bash
num=$((RANDOM%100+1))
while :
yangdog@yangdog-virtual-machine:~$
```

使用 stat 查看文件信息

```
yangdog@yangdog-virtual-machine:~$ stat first.sh
 文件: 'first.sh'
 大小: 188          块: 8          IO 块: 4096   普通文件
设备: 801h/2049d   Inode: 927641   硬链接: 1
权限: (0664/-rw-rw-r--)  Uid: ( 1000/ yangdog)   Gid: ( 1000/ yangdog)
最近访问: 2024-08-30 10:27:30.625665880 +0800
最近更改: 2024-08-30 10:27:24.569505001 +0800
最近改动: 2024-08-30 10:27:24.569505001 +0800
创建时间: -
```

使用 wc -w 查看单词数

```
yangdog@yangdog-virtual-machine:~$ wc -w first.sh
32 first.sh
```

使用 grep 查看 while 出现的次数

```
yangdog@yangdog-virtual-machine:~$ grep -c while first.sh
1
```

编写一个 hello 文本将其内容复制 yp 到 first.sh 中

```
yangdog@yangdog-virtual-machine:~$ touch hello.txt
yangdog@yangdog-virtual-machine:~$ vim hello.txt
```

```
sdfasfeaf666
sdfasfeaf666
sdfasfeaf666

yangdog@yangdog-Virtual-Machine:~$ cat first.sh
#!/bin/bash
num=$((RANDOM%100+1))
echo "$num"

while :
do
    read cai
    if [ $cai -eq $num ]
    then
        echo "YES"
    elif [ $cai -gt $num ]
    then
        echo "bigger"
    else
        echo "smaller"
    fi
done
sdfasfeaf666
sdfasfeaf666
sdfasfeaf666
```

3 实验结果

成功实现了 first.sh 脚本，实现了其功能：给出终端一个随机数进行猜测，同时进行了 vim 的基本内容应用，使用 touch 创建文本并将文本内容拷贝到了 first.sh 文件中来。

4 实验感悟

通过这次实验，我对于 shell 脚本的编写有了基本的了解与掌握，明白了一些基本语法内容的编写，同时也更加熟悉了使用 vim 编辑一些文本内容，并且学会了使用更多的 vim 指令去进行一些基本的操作来完成想达到的要求。