

# 实验报告

## Linux 第一次实验

题 目：	Linux第一次实验
姓 名：	彭凯锋
学 号：	8208211317
班 级：	计科2105
日 期：	2023年3月18日

## 1 实验目的

- 掌握Linux Shell程序的基本语法
- 掌握Linux Shell程序编辑、运行、调试的方法
- 掌握Linux Shell环境变量及函数的使用
- 编写Linux shell程序

## 2 题目一：运行调试程序

### 2.1 实验内容

运行并且调试实验指导书给出的程序，给出运行结果。

### 2.2 实验设计

在vscode中编写对应的程序，使用shell终端进行运行，观察运行结果，使用基本的shell调试技术对程序进行Debug。

### 2.3 代码与运行结果

#### 2.3.1 fortest

```
1 #!/bin/bash
2
3 for a in x y z; do
4     echo now a=$a
5 done
```

```
● (base) → Lab0 chmod a+x fortest.sh
● (base) → Lab0 ./fortest.sh
    now a=x
    now a=y
    now a=z
○ (base) → Lab0 █
```

#### 2.3.2 fortest1

```
1 #!/bin/bash
2 for a
3 do
4     echo now a=$a
5 done
```

运行结果:

```
● (base) → Lab0 ./fortest1.sh
○ (base) → Lab0 █
```

#### 2.3.3 functest

```
1 #!/bin/bash
2
3 setup()
4 {
5     echo setup
```

```

6  }
7
8  do_date()
9  {
10     date
11 }
12
13 chgdir()
14 {
15     cd $1
16     pwd
17 }
18
19 do_data
20 setup
21 chgdir $1

```

#### 运行结果:

当前目录结构:

```

1  .
2  ├── argv.sh
3  ├── file.names
4  ├── fortest1.sh
5  ├── fortest.sh
6  ├── functest.sh
7  ├── next
8  |   ├── new_group.sh
9  |   ├── student_info.sh
10 |   └── student.txt
11 ├── paramtest.sh
12 ├── test_and_debug.sh
13 ├── untiltest.sh
14 └── whilettest.sh

```

其中next为我们的当前文件夹的子文件夹，是我们脚本传递的参数，用于切换目录。

经过调试注意到：使用 `./` 或者是 `bash` 的方式运行脚本会开启一个新的shell进程，通过pwd可以发现在子进程里面修改了我们的目录但是回到当前进程是没有修改目录的，所以采用点操作符或者是 `source` 的方法运行脚本，保证脚本在当前的进程运行，这样就会看到我们的目录发生了改变。

**注意：脚本位置参数和函数的位置参数的区别**

```

● (base) → Lab0 . ./functest.sh next
2023年 04月 04日 星期二 13:20:12 CST
setup
/home/peng/Codes/Scripts/Lab0/next
○ (base) → next █

```

### 2.3.4 paramtest

```
1 #!/bin/bash
2
3 echo filename:$0
4 echo arguments:$*
5 echo number arg:$#
6 echo arg2:$2
7 shift
8 echo number arg:$#
9 echo arg2:$2
10 set hello, everone
11 echo args:$*
12 echo arg2:$2
```

#### 运行结果:

当我们不传递参数的时候的运行结果

```
● (base) → Lab0 bash ./paramtest.sh
filename:./paramtest.sh
arguments:
number arg:0
arg2:
number arg:0
arg2:
args:hello, everone
arg2:everone
○ (base) → Lab0 █
```

可以看到,参数的内容为空同时参数的个数为0,通过 `shift` 命令左移参数同时使用 `set` 命令设置参数之后可以看到第二个参数变成了, everone。

当我传递参数 `arg1` 和 `arg2` 的时候的运行结果

```
● (base) → Lab0 bash ./paramtest.sh arg1 arg2
filename:./paramtest.sh
arguments:arg1 arg2
number arg:2
arg2:arg2
number arg:1
arg2:
args:hello, everone
arg2:everone
○ (base) → Lab0 █
```

可以看到,左移参数之后之后一个参数了,同时设置两个参数,这里 `,` 之后是有空格的,设定的是#1 #2两个参数,如果是没有空格的就是只是设置#1一个参数。

### 2.3.5 untiltest

```

1  #!/bin/bash
2
3  ans=yes
4
5  until [ "$ans" = "no" ]
6  do
7      echo Enter a name
8      read name
9      echo $name >> file.names
10     echo "Continue?"
11     echo Enter yes or no
12     read ans
13 done

```

运行结果:

```

• (base) → Lab0 ./untiltest.sh
Enter a name
peng
Continue?
Enter yes or no
yes
Enter a name
kai
Continue?
Enter yes or no
no
○ (base) → Lab0 █

```

查看输出的结果，结果保存在 `file.names` 文件里面

```

• (base) → Lab0 cat file.names
peng
kai
○ (base) → Lab0 █

```

### 2.3.6 whiletest

```

1  #!/bin/bash
2
3  while test -r abc.txt
4  do
5      echo "file abc.txt has not been deleted"
6      sleep 10
7  done
8
9  echo "file abc.txt has been deleted"

```

运行结果:

不断检查abc.txt文件是否被删除。

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

file abc.txt has not been deleted

file abc.txt has not been deleted  
file abc.txt has not been deleted  
file abc.txt has been deleted

(base) → Lab0

(base) → Lab0 rm -rf file.names

(base) → Lab0 rm -rf abc.txt

(base) → Lab0

zsh...

zsh...

英 · 拼

## 3 题目二：批量添加用户

### 3.1 实验内容

设计一个shell脚本，添加一个新的用户组为class，然后添加属于这个组的三十个用户，用户名的形式为stdxx，其中xx从01到30。

### 3.2 实验设计

先创建一个用户组，名字叫做class，然后使用循环创建三十个用户。

### 3.3 代码与运行结果

```
1 #!/bin/bash
2
3 groupadd class
4
5 for i in $(seq -w 01 30)
6 do
7     useradd -g class -m std"$i"
8 done
```

class 的用户组编号1001

```
class:x:1001:
```

通过查看 `/etc/passwd` 里面的信息，可以看到所有的用户都已经添加成功，并且都是属于class用户组。



```
• (base) → next grep "std[0-3][0-9]" "/etc/passwd"
std01:x:1001:1001::/home/std01:/bin/sh
std02:x:1002:1001::/home/std02:/bin/sh
std03:x:1003:1001::/home/std03:/bin/sh
std04:x:1004:1001::/home/std04:/bin/sh
std05:x:1005:1001::/home/std05:/bin/sh
std06:x:1006:1001::/home/std06:/bin/sh
std07:x:1007:1001::/home/std07:/bin/sh
std08:x:1008:1001::/home/std08:/bin/sh
std09:x:1009:1001::/home/std09:/bin/sh
std10:x:1010:1001::/home/std10:/bin/sh
std11:x:1011:1001::/home/std11:/bin/sh
std12:x:1012:1001::/home/std12:/bin/sh
std13:x:1013:1001::/home/std13:/bin/sh
std14:x:1014:1001::/home/std14:/bin/sh
std15:x:1015:1001::/home/std15:/bin/sh
std16:x:1016:1001::/home/std16:/bin/sh
std17:x:1017:1001::/home/std17:/bin/sh
std18:x:1018:1001::/home/std18:/bin/sh
std19:x:1019:1001::/home/std19:/bin/sh
std20:x:1020:1001::/home/std20:/bin/sh
std21:x:1021:1001::/home/std21:/bin/sh
std22:x:1022:1001::/home/std22:/bin/sh
std23:x:1023:1001::/home/std23:/bin/sh
std24:x:1024:1001::/home/std24:/bin/sh
std25:x:1025:1001::/home/std25:/bin/sh
std26:x:1026:1001::/home/std26:/bin/sh
std27:x:1027:1001::/home/std27:/bin/sh
std28:x:1028:1001::/home/std28:/bin/sh
std29:x:1029:1001::/home/std29:/bin/sh
std30:x:1030:1001::/home/std30:/bin/sh
○ (base) → next █
```

## 4 题目三：查找学生信息

### 4.1 实验内容

编写一段shell程序：根据从键盘输入的学生学号，到档案表文件中查找对应的学生信息，并进行显示。

### 4.2 实验设计

设计一个获取用户输入的函数，判断用户输入是否合法，如果是合法的，就从表单中获取用户信息并且输出。在脚本中不断循环运行该函数，知道用户输入结束提示符为止。

### 4.3 代码与运行结果

```
1  #!/bin/bash
2
3  file="student.txt"
4
5  query()
6  {
7      read -p "请输入学号" id
8
9      if [ -z "$id" ]; then
10         echo "请输入有效的学号"
11         return
12     fi
13
14     if ! [[ "$id" =~ ^[0-9]+$ ]]; then
15         echo "请输入有效的学号"
16         return
17     fi
18
19     if [ ! -f "$file" ]; then
20         echo "文件不存在"
21         return
22     fi
23
24     info=$(grep "^$id" "$file")
25
26     if [ -z "$info" ]; then
27         echo "没有找到对应学生的信息"
28         return
29     fi
30
31     array=($info)
32     echo "学号: ${array[0]}"
33     echo "姓名: ${array[1]}"
34 }
35
36 while true; do
37     query
38     read -p "按q或者是Q退出" choice
```

```

39     case $choice in
40         [qQ]) break;;
41         *) continue;;
42     esac
43 done
44
45 echo "SCRIPT DONE"

```

成绩表单

```

● (base) → next cat student.txt
1 s1
2 s2
3 s3
4 s4
5 s5
9 s9%
○ (base) → next █

```

查询结果

```

● (base) → next ./student_info.sh
请输入学号 1
学号： 1
姓名： s1
按q或者是Q退出
请输入学号 2
学号： 2
姓名： s2
按q或者是Q退出
请输入学号 asdf
请输入有效的学号
按q或者是Q退出q
SCRIPT DONE

```

## 5 总结与收获

通过此次实验，我锻炼了自己编写shell脚本的能力，熟悉了shell脚本的基本的知识。对变量、函数、流程控制的掌握更加熟练，能够对shell脚本进行调试分析。