

江南大学物联网工程学院实验报告

课程名称 《Linux 环境程序设计》 实验名称 实验 4 shell 编程 实验日期 2017.11.09

班级 计科 1404 姓名 阎覃 学号 1030414414

实验报告要求 1. 实验目的 2. 实验内容 3. 实验步骤 4. 运行情况 5. 实验体会

1 实验目的

- 了解 shell 的作用和主要分类;
- 掌握 bash 的建立和执行方式;
- 掌握 bash 的基本语法;
- 学会编写 shell 脚本。

2 实验内容

- shell 脚本的建立和执行
- shell 变量和位置参数;
- 一般控制结构。

3 实验步骤及运行情况

Problem 1 利用 vi 建立一个脚本文件, 其中包括 date、cal、pwd、ls 等常用命令。然后以不同方式执行该脚本。

Answer

首先进入 vim 创建名为 test1.sh 的脚本

```
1 vim ./test1.sh
```

之后输入以下脚本

test1.sh

```
1 #!/bin/bash
2
3 date
4 cal
5 pwd
6 ls
```

方法 1 直接执行脚本

```
1 chmod +x ./test1.sh #使脚本具有执行权限
2 ./test1.sh #执行脚本
```

方法 2 作为解释器参数

```
1 /bin/sh test1.sh
```

运行截图

```

1. bash
MacBook-Pro:ex4 ethan$ chmod +x ./test1.sh
MacBook-Pro:ex4 ethan$ ./test1.sh
2017年11月10日 星期五 16时31分17秒 CST
十一月 2017
日 一 二 三 四 五 六
          1  2  3  4
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30

/Volumes/macFile/Files/Google 云端硬盘/Study/Linux/ex4
doc      test1.sh      要求
MacBook-Pro:ex4 ethan$ /bin/sh test1.sh
2017年11月10日 星期五 16时31分29秒 CST
十一月 2017
日 一 二 三 四 五 六
          1  2  3  4
 5  6  7  8  9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30

/Volumes/macFile/Files/Google 云端硬盘/Study/Linux/ex4
doc      test1.sh      要求
MacBook-Pro:ex4 ethan$

```

Problem 2 利用变量赋值方法, 将字符串DOS file c:>\student*显示出来.

Answer

新建 test2.sh 输入以下脚本

test2.sh

```

1 #!/bin/bash
2
3 str='DOS file c:>\student\*'
4 echo $str

```

运行截图

```

MacBook-Pro:ex4 ethan$ ./test2.sh
DOS file c:>\student\*

```

Problem 3 分析下列 shell 脚本的功能:

```

1 count=$#
2 cmd=echo
3 while [ $count -gt 0 ]
4 do
5     cmd="$cmd \$$count"
6     count=`expr $count - 1`
7 done
8 eval $cmd

```

Answer

\$#代表的是传递给脚本参数个数;

\\$\$count 转义后即 \$2, \$1,..., 表示命令名后面的参数;

count=`expr \$count - 1`, 将变量 count 自减 1。

最后的 cmd 可以得到如下形式: echo \$2 \$1, 即将所有参输出。

Problem 4 编写一个 shell 脚本, 它把第二个位置参数及其以后的各个参数指定的文件复制到第一个位置参数指定的目录中.

Answer

```

test4.sh

1  #!/bin/bash
2
3  if [ ! -d $1 ]
4  then
5      echo "$1 is not a directory!"
6      exit 1
7  fi
8  count=$#
9  cmd=cp
10 while [ $count -gt 0 ]
11 do
12     cmd="$cmd \$$count"
13     count=`expr $count - 1`
14 done
15 eval $cmd

```

运行截图

```

MacBook-Pro:ex4 ethan$ touch 1
MacBook-Pro:ex4 ethan$ touch 2
MacBook-Pro:ex4 ethan$ bash test4.sh ./test 1 2
MacBook-Pro:ex4 ethan$ ls test
1      2

```

Problem 5 打印给定目录下的某些文件, 由第一个参数指出文件所在的目录, 其余参数是要打印的文件名.

Answer

```

test5.sh

1  #!/bin/bash
2
3  if [ ! -d $1 ]
4  then
5      echo "$1 is not a directory!"
6      exit 1
7  fi
8  count=2
9  cd $1
10 cmd=cat
11 while [ $count -le $# ]
12 do
13     cmd="$cmd \$$count"
14     count=`expr $count + 1`
15 done
16 eval $cmd

```

运行截图

```

MacBook-Pro:test ethan$ date>1
MacBook-Pro:test ethan$ date>2
MacBook-Pro:test ethan$ cd..
-bash: cd..: command not found
MacBook-Pro:test ethan$ cd ..
MacBook-Pro:ex4 ethan$ bash test5.sh test 1 2
2017年11月10日 星期五 23时44分15秒 CST
2017年11月10日 星期五 23时44分17秒 CST

```

Problem 6 利用 for 循环将当前目录下的.c 文件移到指定的目录下，并按文件大小排序，显示移动后指定目录的内容。

Answer

test6.sh

```

1  #!/bin/bash
2
3  if [ ! -d $1 ]
4  then
5      echo "$1 is not a directory!"
6      exit 1
7  fi
8  file='ls *.c'
9  for f in $file
10 do
11     if [ -f $f ]
12     then
13         cp $f $1
14     fi
15 done
16 ls -Sl $1

```

运行截图

```

MacBook-Pro:ex4 ethan$ bash test6.sh test
total 16
-rw-r--r--  1 ethan  admin  147 11 11 00:05 1.c
-rw-r--r--  1 ethan  admin   48 11 11 00:05 2.c

```

Problem 7 编写一个 shell 脚本, 求费波纳奇数列的前 10 项及总和.

Answer

test7.sh

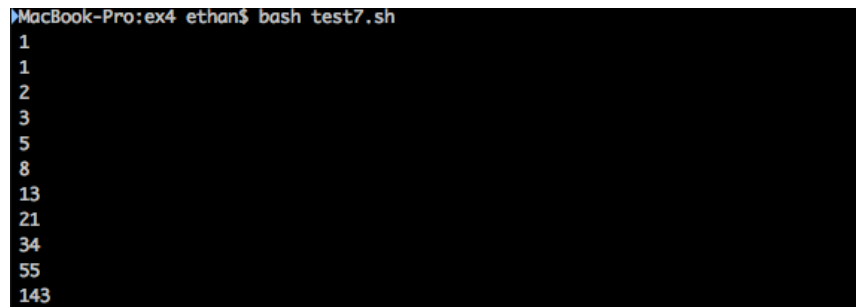
```

1  #!/bin/bash
2
3  array[0]=1
4  array[1]=1
5
6  max=10
7  i=0
8  sum=0
9  while [ $i -lt $max ]
10 do
11     if [ $i -ge 2 ]
12     then
13         let array[$i]=array[$i-1]+array[$i-2]
14     fi
15     let sum+=array[i]

```

```
16 echo ${array[$i]}
17 let i++
18 done
19
20 echo $sum
```

运行截图



```
MacBook-Pro:ex4 ethan$ bash test7.sh
1
1
2
3
5
8
13
21
34
55
143
```

4 实验体会

这次实验我学会了使用 shell 脚本做一些简单的操作，由于是第一次写 shell 脚本，难免会感觉不熟悉。比如有些关键字和语法结构与我们常用的 C++ 不同，不过还是能体会 shell 脚本语言的简单高效，因为他没有 C++ 或者 Java 那么严谨，十分灵活。

实验报告采用 L^AT_EX 编写，代码托管至 GitHub:
<https://github.com/Ethan0w0/JNU-Linux-exp>

教师评价	优		良		中		及格		不及格		教师签名		日期	
------	---	--	---	--	---	--	----	--	-----	--	------	--	----	--