

# 北京科技大学实验报告

学院： 计算机与通信工程学院

专业： 计算机科学与技术

班级： 计 184

---

姓名： 王丹琳

学号： 41824179

实验日期： 2021 年 4 月 15 日

---

## 实验名称：

实验四 Shell 编程实践

## 实验目的：

结合课程讲授第九讲《Linux Shell 编程》的复习和理解，熟悉掌握有关 Linux Shell 编程、数学运算、关系符运算、分支（多分支）选择、循环语句等程序流程具体的实现方法，完成 Linux Shell 编程实验任务,能熟悉 Linux 系统高级管理及日常任务的自动化脚本执行模式，掌握 Shell 编程实践使用技巧。

## 实验仪器：

PC 机一台：ThinkPad T480

## 实验环境：

VMware 虚拟机

Linux Ubuntu 64 位

## 实验原理：

Linux，全称 GNU/Linux，是一种免费使用和自由传播的类 UNIX 操作系统，其内核由林纳斯·本纳第克特·托瓦兹于 1991 年 10 月 5 日首次发布，它主要受到 Minix 和 Unix 思想的启发，是一个基于 POSIX 的多用户、多任务、支持多线程和多 CPU 的操作系统。它能运行主要的 Unix 工具软件、应用程序和网络协议。它支持 32 位和 64 位硬件。Linux 继承了 Unix 以网络为核心的设计思想，是一个性能稳定的多用户网络操作系统。

## 实验内容与步骤：

- 1) 逐个学习实践使用课堂中提到的具体的 Shell 编程的那些实例。
- 2) 编写 Shell 程序，实现每天 12:30 将主目录下的所有目录和文件归档并压缩为 mybackup.tar.gz, 然后让该脚本开机运行。

## 实验数据及处理:

1) 逐个学习实践使用课堂中提到的具体的 Shell 编程的那些实例。

实例 1, bash 本身不支持简单的数学运算

```
loongson@loongson-VirtualBox:~$ cat ./example0.sh
x=8
y=$x
z=$x+$y
m=$x-$y
n=$x\*$y
echo $y
echo $z
echo $m
echo $n
```

```
loongson@loongson-VirtualBox:~$ bash example0.sh
8
8+8
8-8
8*8
```

实例 2, 可以使用 `$[]` 表达式来进行数学运算, 如下图

```
loongson@loongson-VirtualBox:~$ vi example1.sh
loongson@loongson-VirtualBox:~$ bash example1.sh
8
16
0
8*8
loongson@loongson-VirtualBox:~$ cat ./example1.sh
x=8
y=$x
z=$[$x+$y]
m=$[$x-$y]
n=`expr $x \*$y`
echo $y
echo $z
echo $m
echo $n
loongson@loongson-VirtualBox:~$
```

实例 3, 使用 `expr` 能够完成表达式的求值操作

```

loongson@loongson-VirtualBox:~$ vi example2.sh
loongson@loongson-VirtualBox:~$ bash example2.sh
8
16
0
64
loongson@loongson-VirtualBox:~$ cat ./example2.sh
x=8
y=$x
z=$((x+y))
let m=x-y
n=$((x*y))
echo $y
echo $z
echo $m
echo $n

```

实例 4，逻辑表达式主要用于条件判断，值为 `true`（或 0）表示结果为真；值为 `false`（非零值）表示结果为假，当使用左方括号而非 `test` 时，其后必须跟着一个空格，逻辑表达式两边的空格是必需的。

```

loongson@loongson-VirtualBox:~$ vim example3.sh
loongson@loongson-VirtualBox:~$ bash example3.sh
10 -eq 20 :a != b
loongson@loongson-VirtualBox:~$ cat ./example3.sh
#!/bin/bash
a=10
b=20
if [ $a -eq $b ]
then
    echo "$a -eq $b :a = b"
else
    echo "$a -eq $b :a != b"
fi

```

实例 5，字符串检测运算符，`=`检测两个字符串是否相等，`-z` 检测字符串长度是否为 0，若为 0 返回 `true`

```

loongson@loongson-VirtualBox:~$ cat ./example4.sh
#!/bin/bash
a="abc"
b="xyz"
if [ $a = $b ]
then
    echo "$a = $b :a = b"
else
    echo "$a = $b :a != b"
fi
if [ -z $a ]
then
    echo "-z $a :字符串长度为0 "
else
    echo "-z $a :字符串长度不为0 "
fi
if [ $a ]
then echo "$a :字符串非空"
else echo "$a :字符串为空"
fi
~
loongson@loongson-VirtualBox:~$ bash example4.sh
abc = xyz :a != b
-z abc :字符串长度不为0
abc :字符串非空

```

实例 6，文件测试运算符，`-r` 检测文件是否可读，如果是，则返回 `true`

```

loongson@loongson-VirtualBox:~$ vim example5.sh
loongson@loongson-VirtualBox:~$ bash example5.sh
文件具有读取权限
loongson@loongson-VirtualBox:~$ cat ./example5.sh
#!/bin/bash
file="./example4.sh"
if [ -r $file ]
then echo "文件具有读取权限"
else echo "文件并不具有读取权限"
fi

```

```

-rw-rw-r-- 1 loongson loongson 67 4月 15 10:06 example0.sh
-rw-rw-r-- 1 loongson loongson 81 4月 15 10:25 example1.sh
-rw-rw-r-- 1 loongson loongson 83 4月 15 10:26 example2.sh
-rw-rw-r-- 1 loongson loongson 107 4月 15 10:32 example3.sh
-rw-rw-r-- 1 loongson loongson 295 4月 15 10:41 example4.sh
-rw-rw-r-- 1 loongson loongson 133 4月 15 10:43 example5.sh

```

实例 7，布尔运算符，`-a` 与运算。两个表达式都为 `true` 才返回 `true`

```
loongson@loongson-VirtualBox:~$ vim example6.sh
loongson@loongson-VirtualBox:~$ cat ./example6.sh
#!/bin/bash
a=5
b=10
if [ $a -lt 10 -a $b -gt 15 ]
then echo "两个条件都满足"
else echo "不是两个条件都满足"
fi

loongson@loongson-VirtualBox:~$ bash example6.sh
不是两个条件都满足
```

实例 8，条件语句（忘记截图了）

普通用户执行命令 `bash example7.sh`，程序输出 `Welcome Teacher!`

超级用户执行命令 `bash example7.sh`，程序输出 `Welcome Super administrator!`

实例 9，循环结构（忘记截图了）

`While` 循环用于不断执行一系列命令，直到测试条件为假（`false`）

执行 `bash example8.sh` 命令，程序输出 `result=5050`

实例 10，循环语句 `until`，`until` 循环用来执行一系列命令，直到所指定条件为真时才终止循环。

执行命令 `bash example9.sh`，程序结果同实例 9：`result=5050`

实例 11 循环结构 `for`

```

loongson@loongson-VirtualBox:~$ vim example10.sh
loongson@loongson-VirtualBox:~$ bash example10.sh
/home/loongson/apr-1.7.0
/home/loongson/apr-1.7.0.tar.gz
/home/loongson/apr-util-1.6.1
/home/loongson/apr-util-1.6.1.tar.gz
/home/loongson/example10.sh
/home/loongson/example10.sh
/home/loongson/example1.sh
/home/loongson/example2.sh
/home/loongson/example3.sh
/home/loongson/example4.sh
/home/loongson/example5.sh
/home/loongson/example6.sh
/home/loongson/example7.sh
/home/loongson/example8.sh
/home/loongson/example9.sh
/home/loongson/examples.desktop
/home/loongson/expat-2.0.1
/home/loongson/expat_2.0.1.orig.tar.gz
/home/loongson/httpd-2.4.46
/home/loongson/httpd-2.4.46.tar.gz
/home/loongson/linux-3.10
/home/loongson/pcre-8.35
/home/loongson/pcre-8.35.tar.gz
loongson@loongson-VirtualBox:~$ cat ./example10.sh

#!/bin/bash
for FILE in $HOME/*.
do
    echo $FILE
done

```

实例 12，函数的返回值，定义一个变量，用来接收函数的计算结果，脚本在需要的时候访问这个变量来获得函数的返回值

```

loongson@loongson-VirtualBox:~$ vim example11.sh
loongson@loongson-VirtualBox:~$ bash example11.sh
Hello World
loongson@loongson-VirtualBox:~$ cat ./example11.sh
#!/bin/bash
Hello(){
mystr='Hello World'
}
Hello
echo $mystr

```

实例 13，将 shell 函数作为一个子程序调用，将返回值写到子程序的标准输出。

```
loongson@loongson-VirtualBox:~$ vim example12.sh
loongson@loongson-VirtualBox:~$ bash example12.sh
Hello World
loongson@loongson-VirtualBox:~$ cat ./example12.sh
#!/bin/bash
Hello(){
mystr='Hello World'
echo $mystr
}
result=$(Hello)
echo $result
```

实例 14，在函数中使用 **return** 返回整数值，**\$?** 是一个特殊的内部变量，可用于获取上一个命令执行后的返回结果，可以通过函数 **return** 语句来接受返回值

```
loongson@loongson-VirtualBox:~$ vim example13.sh
loongson@loongson-VirtualBox:~$ vim example13.sh
loongson@loongson-VirtualBox:~$ bash example13.sh
the sum is 9
loongson@loongson-VirtualBox:~$ cat ./example13.sh
#!/bin/bash
addnum(){
val=`expr $1 + $2`
return $val
}
addnum 4 5
ret=$?
echo "the sum is $ret"
```

实例 15，输出当前目录下所有文件，并输出文件总数和目录总数

```

loongson@loongson-VirtualBox:~$ vim example15.sh
loongson@loongson-VirtualBox:~$ bash example15.sh
=.:                                cannot open `=.' (No such file or
directory)
..:                                directory
apr-1.7.0:                         directory
apr-1.7.0.tar.gz:                  gzip compressed data, was "apr-1.
7.0.tar", last modified: Mon Apr  1 17:56:23 2019, max compres
sion, from Unix
apr-util-1.6.1:                   directory
apr-util-1.6.1.tar.gz:            gzip compressed data, was "apr-ut
il-1.6.1.tar", last modified: Wed Oct 18 14:51:43 2017, max co
m from Unix
.bash_history:                     UTF-8 Unicode text
.bash_logout:                     ASCII text
.bashrc:                           ASCII text
.cache:                            directory
cloud-lab:                         directory
.config:                           directory
.dbus:                             directory
.dnrc:                             ASCII text
example0.sh:                       ASCII text
example10.sh:                     ASCII text

```

2) 编写 Shell 程序, 实现每天 12: 30 将主目录下的所有目录和文件归档并压缩为 mybackup.tar.gz, 然后让该脚本开机运行。

1. 创建脚本

```

root@loongson-VirtualBox:/home/mou_1# touch /home/backup.sh

```

2. 写入命令

首先进入脚本: vi 命令 vi /home/backup.sh

```

#!/bin/bash
mkdir /home/beifen
#创建一个临时文件 (要保存备份的路径)
cp -r /home/mou_1 /home/beifen
#数据存在loongson目录下, 备份到beifen目录下, 所以先将数据拷过来
tar -zcvf mybackup.tar.gz /home/beifen
#将数据所在文件夹beifen打包
rm -rf /home/beifen/
#删除临时文件内容

find ./ -mtime +30 -name "*.tar.gz" -exec rm -rf {} \;
#删除该文件夹下超过30天的文件

```

3. 执行脚本文件



```
root@loongson-VirtualBox:/home# bash /home/backup.sh
/home/beifen/
/home/beifen/mou_1/
/home/beifen/mou_1/.profile
/home/beifen/mou_1/.bash_logout
/home/beifen/mou_1/examples.desktop
/home/beifen/mou_1/.bashrc
```

```
root@loongson-VirtualBox:/home# ls -a
.   ad_2      loongson    mou_2      test_user1
..  backup.sh mou_1       mybackup.tar.gz test_user2
root@loongson-VirtualBox:/home#
```

### 实验结果与分析:

通过本次实验，我对第 9 讲的内容进行了复习和理解。通过练习，掌握了有关 Linux Shell 编程、数学运算、关系符运算、分支（多分支）选择、循环语句等程序流程具体的实现方法。同时，通过查询资料，对 LINUX 日常任务的自动化脚本执行模式有了认识。

Shell 是一个用 C 语言编写的程序，它是用户使用 Linux 的桥梁。Shell 既是一种命令语言，又是一种程序设计语言。在实验中，我熟悉了 shell 的使用，为今后的 shell 使用打下扎实的基础。但要真正掌握 shell，还要在实际操作中多去使用。