

《Linux 操作系统》实验指导

实验六：Shell 编程基础

一、实验目的

- 1、掌握 Shell 的基础语法；
- 2、掌握 Bash Shell 的基础语法；
- 3、掌握 Shell 编程的基本思路；
- 4、掌握正则表达式。

二、实验学时

2 学时

三、实验类型

综合性

四、实验需求

1、硬件

单机虚拟化：每人配备计算机 1 台，计算机接入实验室局域网，且使用固定 IP 地址。

云服务器：每人配备计算机 1 台、云服务器 1 台，云服务器提供 SSH 方式的远程管理。

2、软件

Windows 7 操作系统，安装 VMware Workstation 10+、PuTTY 软件。

3、网络

支持对互联网的访问。

4、工具

无

五、实验理论

- 1、正则表达式的基本原理；
- 2、Linux Shell 的基本内容；
- 3、Bash Shell 的基本内容；
- 4、Shell 编程工具的基本内容。

六、预备知识

- 1、Linux Shell 下常用的命令；
- 2、Linux 下常用文本编辑器的使用；
- 3、Linux 操作系统中常见的 Shell 类型有哪些？请查阅相关资料，完成表 1-1 常见的 Shell 类型对比分析一览表。

表 1-1 常见的 Shell 类型对比分析一览表

Shell 类型	优点	Shell 类型	优点

要求：

请查阅相关资料，完成表 1-1 的填写，至少需要填写 4 种常见的 Shell 类型，并将表格填写到实验报告册中。

- 4、请阐述 Shell 的基本概念、Shell 的工作原理。

要求：

请查阅相关资料，回答上述问题，并将具体的答案填写到实验报告册中。

七、实验任务

- 1、掌握 Shell 的常量、变量、运算符、表达式、控制语句等使用；
- 2、掌握函数的定义与使用方法；
- 3、掌握常见的 Shell 工具的使用；
- 4、掌握正则表达式的基本内容。

八、实验步骤

1、实验场景与实验环境准备

本实验可根据实验室的具体条件，在单机虚拟化和云服务器两种场景下开展实验教学。

(1) 单机虚拟化场景下开展实验教学

实验场景：

每位学生通过物理主机上的虚拟化软件 VMware workstation 安装 Ubuntu Server 虚拟操作系统，虚拟操作系统 Ubuntu Server 配备固定 IP 地址，并开启 SSH 远程管理服务。

实验准备：

- ①启动虚拟化软件 VMware workstation 并附加上 Ubuntu Server 操作系统；
- ②配置 Ubuntu Server 操作系统的网络，并在虚拟主机上使用 Ping 命令进行网络连通性测试；
- ③在虚拟主机上安装 openssh, ufw。开启 openssh 服务，并设置 ufw 的规则为允许 172.16.2.x/24 的主机访问，开放 22 端口。
- ④虚拟主机后台运行，通过物理主机上的 PuTTY 软件，连接 Ubuntu Server 操作系统。

(2) 云服务器场景下开展实验教学

实验场景：

在云服务器为学生安装部署 Linux 操作系统，每个操作系统配备固定 IP 地址，并开启 SSH 远程管理服务，并把 root 用户权限和远程访问信息提供给学生。

实验准备：

学生使用 Windows 7 操作系统开展实验学习，使用 PuTTY 软件根据教师提供的云服务器访问信息远程访问实验用的 Linux 操作系统。

2、Shell 编程基础

(1) 文本信息输出

编写 Shell 程序，执行时在屏幕输出“Hello World”。程序文件名为：PrintInfo.sh。

参考程序：

```
vi /home/student/PrintInfo.sh          #使用 vi 编辑器打开文件
#!/bin/bash                            #在文件中编写 Shell 脚本
PrintHello="Hello World"
echo $PrintHello
chmod +x PrintInfo.sh                  #给文件 PrintInfo.sh 添加执行权限
sh ./PrintInfo.sh                       #执行文件 PrintInfo.sh
```

(2) 数值计算

编写 Shell 程序，分别使用\$[]、\$(())、let 命令对数值进行计算，执行时在屏幕输出“1 + 4 = 5”。程序文件名为：NumberCalculate.sh。

参考程序：

```
vi /home/student/ NumberCalculate.sh
#!/bin/bash
a=1
b=4
c=${a + $b}
d=$(( $a + $b ))
let e=a+$b
echo $a + $b = $c
echo $a + $b = $d
echo $a + $b = $e
chmod +x NumberCalculate.sh
sh ./ NumberCalculate.sh
```

(3) 获取 IP 地址

编写 Shell 程序，获取系统中的 eth0 网卡的 IP 地址。程序文件名为：GetIPAddress.sh。

参考程序：

```
vi /home/student/ GetIPAddress.sh
#!/bin/bash
IPAddr=`ifconfig | egrep 'inet addr' | grep -v '127.0.0.1' | cut -d: -f2 | cut -d' ' -f1`
echo "The IP Address is $IPAddr"
```

(4) 文件操作

编写 Shell 程序，读取用户输入的文件路径，判断目录文件是否存在。程序文件名为：FileExist.sh。

参考程序：

```
#!/bin/bash
echo "Please input the file path"
read path
if [ -e $path ]
then
```

```
    echo "The file exists"
else
    echo "Sorry, your input is wrong"
fi
```

要求：

(1) 请编写 Shell 脚本，读取用户输入的数据的长度，并倒序输出用户输入的数据。

(2) 编写 Shell 脚本，列出/usr 目录下权限为 755 的文件名。

请根据上述 2 个要求，进行具体实验，并将 Shell 脚本填写到实验报告册中。

3、Shell 控制语句

(1) if 选择结构

编写 Shell 程序，实现用户权限的认证。当用户输入的用户名、密码均为 student 时，提示用户权限信息正确。程序文件名为: IsLogin.sh。

参考程序

```
vi /home/student/IsLogin.sh
#!/bin/bash
echo "Enter your username"
read username
if [ "$username" = "student" ]
then
    echo "Enter your password"
    read password
    if [ "$password" = "student" ]
    then
        echo "welcome student"
    else
        echo "Your password is error"
    fi
else
    echo "Your username is error"
fi
```

(2) until 语句

编写 Shell 程序，使用 until 语句计算 1~100 的和。程序文件名为: CalculateSum1.sh。

参考程序

```
vi /home/student/ CalculateSum1.sh
#!/bin/bash
sum=0
number=1
until ! test $number -le 100
do
    sum=$(( $sum + $number )
    let number=$number+1
done
echo "The summary is $sum"
```

(3) while 语句

编写 Shell 程序，使用 while 语句计算 1~100 的和。程序文件名为: CalculateSum2.sh。

参考程序：

```
vi /home/student/CalculateSum2.sh
#!/bin/bash
i=1
sum=0
while((i<=100))
do
    sum=$((sum+i))
    i=$((i+1))
done
echo "The Summary is $sum"
```

(4) for 语句

编写 Shell 程序，统计当前目录下文件的个数。程序文件名为：FileCount.sh。

参考程序：

```
#!/bin/bash
cd /home/student
for file in `ls`
do
    if ! [ -d $file ]
    then
        let count=$count+1
    fi
done
echo "There are $count files"
```

要求：

- (1) 请编写 Shell 脚本，判断用户输入的年份是否为闰年。
 - (2) 请编写 Shell 脚本，打印给定的行数的*号。第一行打印 1 个，第二行打印 2 个，以此类推。行数由用户输入确定。
- 请根据上述 2 个要求，进行具体实验，并将 Shell 脚本填写到实验报告册中。

4、函数定义及其使用

Shell 有自定义函数的功能，通过定义函数可以使 Shell 脚本结构性更为清晰且易于维护。

(1) 编写 Shell 程序，读取用户输入的数据并进行计算。程序文件名为：ReadCalculate.sh。

参考程序：

```
vi /home/student/ReadCalculate.sh
#!/bin/bash
Sum()
{
    if [ $# -eq 2 ]
    then
        for var in $*;
        do
            let "sum += $var"
        done
        echo "sum is $sum"
    elif
```

```

        echo "please input 2 parameter"
        exit
    fi
}
Sum $*

```

调用该函数的方式:

```

student@ubuntu-labs-student:/home/student$bash ReadCalculate.sh 13 26
sum is 39

```

(2) 编写 Shell 程序, 读取用户输入的数据, 并取出数据的最大值。程序文件名为: ReadMax.sh。

参考程序:

```

vi /home/student/ReadMax.sh
#!/bin/bash
Max()
{
    while test $1
    do
        if test $maxvalue
        then
            if test $1 -gt $maxvalue
            then
                maxvalue=$1
            fi
        else
            maxvalue=$1
        fi
        shift      #函数参数左移一位
    done
    return $maxvalue
}
Max $*
echo "maxvalue is:$maxvalue"

```

调用该函数的方式:

```

student@ubuntu-labs-student:/home/student$bash ReadMax.sh 23 25 19 13 39 9
maxvalue is:39

```

要求:

(1) 请编写 Shell 脚本, 读取用户输入的数据, 并计算平均数。

(2) 请编写 Shell 脚本, 输出九九乘法表。

请根据上述 2 个要求, 进行具体实验, 并将 Shell 脚本填写到实验报告册中。

5、正则表达式

(1) 正则表达式的基本内容

① 字符集、单词、字符类

常见的用于匹配字符集、单词、字符类的元字符如表 1-2 所示:

表 1-2 用于匹配字符集、单词、字符类的元字符一览表

元字符	描述	元字符	描述
.	匹配除换行符之外的任意一个字符	[]	指定[]内字符集中的某一个字符
-	描述一个范围	[:upper:]	匹配所有的大写字母
[:alnum:]	匹配文字、数字字符	[:alpha:]	匹配字母字符
[:lower:]	匹配小写字母	[:digit:]	匹配小数
[:xdigit:]	匹配十六进制数字	[:punct:]	匹配标点符号
[:blank:]	匹配制表符和空格	[:space:]	匹配空格
[:cntrl:]	匹配所有控制符	[:print:]	匹配所有可打印的字符

②位置匹配

字符“^”和“\$”分别用于匹配行首和行尾。

^a	#用于匹配以 a 开头的字符
t\$	#用于匹配以 t 结尾的字符

③字符转义

字符转义用于匹配特殊字符，常见的转义字符如表 1-3 所示：

表 1-3 常见的转义字符一览表

转义字符	特殊字符	转义字符	特殊字符
*	*	\+	+
\.	.	\[[
\\	\	\?	?

④重复

用户希望某个字符能够不止一次出现，需要用到重复。常见的用于重复模式的元字符如表 1-4 所示：

表 1-4 常见的用于重复模式的元字符一览表

元字符	描述	元字符	描述
*	重复 0 次或更多次	+	重复 1 次或更多次
?	重复 0 次或一次	{n}	重复 n 次
{n,}	重复 n 次或更多次	{n,m}	重复大于 n 次，小于 m 次

⑤反义、分支

^[a]	#用于匹配除字母 a 之外的任何字符
^[aeiou]	#用于匹配除字母 a、e、i、o、u 之外的任何字符
^h t\$	#用于匹配以字母 h 开头，或者以字母 t 结尾的行

(2) 通过正则表达式进行更为灵活的文件目录操作

①列出/usr 目录下所有以 d 开头的文件或目录。

ls d*

②统计当前目录下的文件个数及目录个数。

```
ls -l * | grep "^-" | wc -l  
ls -l * | grep "^d" | wc -l
```

③编写 Shell 程序，备份 apache 的所有日志，并将文件扩展名修改为 bak。程序文件名为：FileBankup.sh。

```
vi /home/student/FileBankup.sh  
#!/bin/bash  
Dir=/var/log/apache2/  
for file in $Dir  
do  
    base=${file%.*}  
    mv $file ${base}.bak  
done
```

(3) 通过正则表达式实现信息定制化验证

编写 Shell 程序，对用户输入的手机号码验证其合法性。程序文件名为：CheckPhone。

参考程序：

```
vi /home/student/CheckPhone.sh  
#!/bin/bash  
read -p "Please input your telephone number:" numbers  
a=`echo $no | wc -c`  
if [ $a -eq 12 ]  
then  
    echo $a | grep "^1" > /dev/null  
    if [ $? -eq 0 ]  
    then  
        if [[ $numbers == +([0-9]) ]]  
        then  
            echo "Your telephone numbers are $numbers"  
        else  
            echo "Your input is wrong"  
        fi  
    else  
        echo "Your input is wrong"  
    fi  
else  
    echo "Your input is wrong"  
fi
```

要求：

- (1) 编写 Shell 程序，列出/home/student 文件夹下的所有扩展名为 sh 的文件名；
 - (2) 编写 Shell 程序，读取用户输入的身份证号和出生年月，并验证其结构合法性和出生日期信息的一致性。
- 请按照上述 (1) (2) 2 个要求，进行具体实验，并将 Shell 脚本，实验结果填写到实验报告册中。

九、实验分析

1、Shell 的数据类型、数据结构

- (1) Shell 是否有数据类型？能否将字符型数据转换为整型数据？若能，请阐述实现方法。
- (2) Shell 能否处理小数位的运算？若能，请阐述实现方法。
- (3) Shell 中常见的数据结构有哪些？请举例说明。

要求：

请查阅相关资料，回答上述 (1)(2)(3) 3 个问题，并将答案填写到实验报告册中。

2、内存管理机制

- (1) Linux 内存管理机制是什么？
- (2) 在程序运行过程中，内存如何进行分配与回收？

要求：

请查阅相关资料，回答上述问题，并将答案填写到实验报告册中。

3、Shell 安全

- (1) Shell 脚本是否可以编译、封装、加密？
- (2) Shell 脚本程序如何进行知识产权保护？

要求：

请查阅相关资料，回答上述 (1)(2) 2 个问题，并将答案填写到实验报告册中。

十、课外自主实验

1、文件操作

- (1) 在/home/student 目录中新建目录 myshelldir1、myshelldir2；
- (2) 请编写 Shell 脚本，在/home/student/myshelldir1 目录下分别使用 while 语句、for 语句创建文件 myshell1、myshell2、...、myshell20；
- (3) 请编写 Shell 脚本，将/home/student/myshelldir1 目录中文件名为 myshell1、myshell3、...、myshell19 的文件（文件名后缀数字为奇数的），拷贝到/home/student/myshelldir2 目录下。

要求：

请按照上述 3 个要求，进行具体实验，并将 Shell 脚本填到实验报告册中。

2、Shell 工具

- (1) 使用 cut 命令分别获取系统时间、系统用户名、UID、用户组名、GID；
- (2) 使用 sort 命令、uniq 命令统计访问 apache 的 IP 地址（过滤掉重复的 IP 地址）；
- (3) 对/etc/passwd 文件进行备份，备份文件为/home/student/userbankup；
- (4) 在操作系统中添加一个用户为 student2，使用 diff 命令比较/etc/passwd 文件和备份文件/home/student/userbankup，并记录对比结果。

十一、实验扩展资源

1、图书

- (1) 《鸟哥的 Linux 私房菜》 鸟哥 人民邮电出版社；
- (2) 《Linux 命令行大全》 William E.Shotts 人民邮电出版社；
- (3) 《Ubuntu Linux 从入门到精通》 陶松 人民邮电出版社。

2、文章

- (1) Linux 正则表达式详解: <http://www.jb51.net/article/42989.htm>;
- (2) Linux Shell 编程基础: <http://www.cnblogs.com/suyang/archive/2008/05/18/1201990.html>;
- (3) Shell 函数方法简介: <http://blog.chinaunix.net/uid-1924596-id-3048863.html>;
- (4) Shell 工具的使用: http://blog.sina.com.cn/s/blog_731bd0f40100ohuj.html。

3、互联网资源

- (1) Linux 中国社区: <http://linux.chinaunix.net/>
- (2) Linux 中文交流社区: <http://linux.cn/>
- (3) 中国 Linux 论坛: <http://www.linuxvedio.cn/>

4、电子资源下载

<http://ke.51xueweb.cn/Linux.html>