# 实验八 shell编程

一、实验目的

1、了解shell的概念

2、了解shell变量的种类及其使用

3、理解和掌握条件测试的原理和方法

4、掌握用VI创建shell脚本的方法

5、掌握shell脚本的执行方法

6、掌握if、case等条件语句的使用

7、掌握for、while、until等循环语句的使用方法

8、理解和掌握shift及其它语句的使用方法

二、预备知识

     （1）Linux系统的Shell

使用Linux系统时，用户多数时间是通过Shell与系统进行对话的。Shell是一个接收用户输入命令并将其转换成指令的程序，它是用户与Linux系统之间的界面之一。

Linux系统中可以使用的Shell包括：

bash：Bourne Again Shell。这是Linux系统上最常用的Shell，它合乎POSIX标准且相容于Bourne Shell，提供了编辑命令行的功能。

csh：C Shell。由Berkeley (伯克利大学) 开发，与Bourne Shell在交互式使用上很多是相兼容的，但在程序设计界面上却有很大的不同。它不提供编辑命令行的功能，但它有类似的功能叫命令历史记录

ksh：Korn Shell。在UNIX系统下得到普遍使用，并最早将现代Shell技术 (包括取自C Shell的部分) 引入Bourne Shell，提供了编辑命令行功能

sh：Bourne Shell。是原始的Shell，不提供编辑命令行功能

tcsh：C Shell的加强版，提供了编辑命令行的功能

zsh：Z Shell。最新的Shell，与Bourne Shell兼容，提供编辑命令行的功能。

默认shell—bash Shell

三、实验内容

1.Shell变量的定义及其使用。

2.位置参数和环境变量。

3.Shell中的特殊字符。

四、实验步骤

（一）Shell变量

Shell有两种变量：内部变量、环境变量和临时变量。在Shell脚本中临时变量又分为两类：用户定义的变量和位置参数。

1回值为0代表为真，其他非0值均为假。、常用的内部变量:

# 传给shell程序位置参数的个数

？存储上一个执行命令的返回值，返

0 存储shell程序的名称

\* 存储shell程序的所有参数

例如：

echo "Number of parameters is:"$#

echo "Program name is:"$0

echo "Parameters as a single string is:"$\*

2、常用的环境变量:

HOME 用户的主目录

PATH 执行命令时所搜索的路径

PS1 shell命令行的一级提示符

PS2 shell命令行的二级提示符

用 printenv 命令查看当前起作用的环境变量列表

$printenv

列出的环境变量中最重要的是$PATH，它指定了查找可执行程序的路径顺序。如果运行ifconfig检查某个连接状态，而提示命令没找到，其原因是shell的环境变量里没有设置这条路径名。解决办法是:

$whereis ifconfig

$/sbin/ifconfig

在命令行中使用export命令把/sbin目录加到$PATH变量中:

$PATH=$PATH:/sbin;export PATH

$ifconfig

但这是暂时的，仅维持本次登陆执行了export有效。要长期有效必须修改/etc/profile或主目录下.bash\_profile文件中的环境变量。

3、用户定义的变量

用户定义的变量是最普遍的Shell变量，变量名是以字母或下化线打头的字母、数字和下线符序列，并且大小写字母意义不同。变量名的长度不受限制。定义变量并赋值的一般形式是：变量名=字符串

例如，myfile=/usr/meng/ff/m1.c

   在程序中使用变量的值时，要在变量名前面加上一个符号“$”。这个符号告诉Shell要读取该变量的值。

4、位置参数

执行Shell脚本时可以使用参数。由出现命令行上的位置确定的参数称做位置参数。在sh中总共有十个位置参数，其对应的名称依次是$0,$1,$2,...$9。其中$0始终表示命令名或Shell脚本名，对于一个命令行，必然有命令名，也就必定有$0；而其它位置参数依据实际需求，可有可无。

（二） SHELL中的特殊字符

1、通配符

    通常的通配符有三种：

      \* 星号，它匹配任意字符的0次或多次出现。但注意，文件名前面的圆点（.）和路径名中的斜线（/）必须显示匹配。

      ？ 问号，它匹配任意一个字符。

      [] 一对方括号，其中有一个字符组。其作用是匹配该字符组所限定的任意一个字符。

      应该注意：字符 \* 和 ？在一对方括号外面是通配符，若出现在其内部，它们就失去通配符的能力了。

      ！ 叹号，若它紧跟在一对方括号的左方括号[之后，则表示不在一对方括号中所列出的字符。

2、引号

   在SHELL中引号分为三种：单引号、双引号、和倒引号。

（1）双引号

    由双引号括起来的字符，除$、倒引号和反斜线（\）仍保留其功能外，其余字符通常作为普通字符对待。

（2）单引号

    由单引号括起来的字符都作为普通字符出现。

（3）倒引号

  例如：

$ today=`date`

$echo Today is $ today

Today is Thu May 04 10 ：56 ：20 CST 2000

(4）反斜线

转义字符，若想在字符串中使用反斜线本身，则必须采用（\\）的形式，其中第一个反斜线作为转义字符，而把第二个反斜线变为普通字符。

（三）条件测试

1.字符串比较

= 比较两个字符串是否相等

!= 比较两个字符串是否不相等

-n 判断字符长度是否大于零

-z 判断字符长度是否等于零

2．数字比较

-eq 比较两个数是否相等

-ge 比较一个数是否大于或是等于另一个数

-le 比较一个数是否小于或是等于另一个数

-ne 比较两个数是否不等

-gt 比较一个数是否大于另一个数

-lt 比较一个数是否小于另一个数

3．文件操作符

-d 确定文件是否为目录

-f 确定文件是否为普通文件

-r 确定是否对文件设置了读许可

-s 确定文件名是否具有大于零的长度

-w 确定文件是否设置了写许可

-x 确定文件是否设置了执行许可

测试语句：

    有两种常用形式：一种是用test命令，如上所示。另一种是用一对方括号将测试条件括起来。两种形式完全等价。例如，测试位置参数$1是否是已存在的普通文件，可写成：test -f " $1"，也完全可写成：[ -f $ l ]

在格式上应注意，如果在test语句中使用shell变量，为表示完整、避免造成歧异起见最好用双引号将变量括起来。利用一对方括号表示条件测试时，在左方括号[之后、右方括号]之前各应有空格。

1. if 语句

if语句通过判断逻辑表达式来作出选择，在bash中的条件语句有如下的格式：

if [ expression ]; then

statements

elif [ expression ]; then

statements

else

statements

fi

if条件是可以嵌套的，

2. case语句

case语句是用来执行依赖于离散值或是匹配指定变量值的范围的语句。

格式如下：

case str in

str1 | str2) statements;;

str3 | str4) statements;;

\*) statements;;

esac

3.for语句

for curvar in list

do

statements

done

对list中的每个值需要执行一次statements时应利用这种格式。对每次循环，把list中的当前值赋给变量vcurvar. List可以使含有几个项的变量，或是用空格分隔的值表。

4. while语句

在pdksh和bash 中，利用的如下的格式：

while expression

do

statements

done

（四）创建简单shell程序

1、进入VI编辑器

2、编写程序

例：#!/bin/bash

hello()

{

echo "hello there today's date is:`date`"

}

echo "now going to the function hello"

hello

echo "back from the function"

#end

3. 求前五个偶数的和。

参考程序：

#!/bin/bash

loopcount=0

result=0

while [ $loopcount –lt 5 ]

do

loopcount=‘expr $loopcount +1’

increment=‘expr $loopcount \\*2’

result= ‘$result + $increment’

done

echo “result is $result”

4．设计一个shell程序，添加一个新组为class1，然后添加属于这个组的30个用户，用户名的形式为stdxx，其中xx从01到30。

参考程序：

#!/bin/sh

i=1

groupadd class1

while [ $i -le 30 ]

do

if [ $i -le 9 ] ;then

username=stu0${i}

else

username=stu${i}

fi

useradd $username

mkdir /home/$username

chown -r $username /home/$username

chgrp -r class1 /home/$username

i=$(($i+1))

done

5．编写shell程序，实现自动删除50个账号的功能。账号名为stud1至stud50。

参考程序：

#!/bin/sh

i=1

while [ $i -le 50 ]

do

userdel -r stud${i}

i=$(($i+1 ))

done

（五）执行shell程序的方法

方法一 : 加执行权后，程序完整路径引导。

方法二 : shell解释器名称引导。 bash 1.sh

方法三: 点空格引导。

方法四 : 使用命令替换。

./

sh 1.shell

五、实验报告要求

1．整理实验结果，使其分别与相应的命令对应

2．记录程序设计的流程

3．根据实验步骤，写出实验报告

评分标准：1.能够验证shell编程语法并按时提交实验报告为及格；

2.能使熟练掌握shell环境下变量的使用为良好；

3.能够综合应用Linux命令进行shell 编程为优秀。