山东大学 软件 学院

Linux应用 课程实验报告

|  |  |  |  |
| --- | --- | --- | --- |
| 学号：202000300125 | 姓名：贾星宇 | | 班级：2020级5班 |
| 实验题目： Shell编程基础 | | | |
| 实验学时：4 | | 实验日期：2020年12月14日 | |
| 实验目的：  1. 掌握引号的使用。  2. 理解什么是位置变量参数和特定变量参数。  3. 掌握简单的表达式赋值方法，包括expr命令和变量扩展$(())。  4. 掌握test命令的用法，熟悉test命令的字符串操作符、整数操作符、文件操作符和逻辑操作符。  5. 了解shell控制语句的基本用法。 | | | |
| 硬件环境：  lntel(R) Core(TM) i5-10210U CPU @ 1.60G Hz 2 .1 1 G Hz | | | |
| 软件环境：  VMware Workstation Pro ubuntu Linux | | | |
| 实验过程报告：  (一)引号的使用  1. 双引号和单引号的使用。依次输入下列命令：  (1) $string =“ hello world!”  (2)$echo “$string”  (3)$echo ‘$string’    比较(2)和(3)的结果，为什么不同？  ***（2）是变量，（3）是字符串。***  2. 反引号的使用。进入你的主目录，输入:  $echo “the current work directory is: `pwd`”  会显示什么结果？解释反引号的作用。    转移符号，会把之转化为Linux命令。  (二) shell编程  3.位置变量参数和特定变量参数.验证讲义中Param脚本的运行结果。  (1)在你的主目录/home/student下建立目录bin,并进入到bin目录。  用vi编辑器编辑Param脚本，内容如下：  echo " script name: "  echo $0  echo "show arguments:"  echo $\*  echo "my process id:"  echo $$  echo "Did my script go with any errors?"  echo $?  (2) 执行Param 脚本, 显示什么内容？  $Param hello world    (3)结合执行结果，解释脚本中$0,$\*,$$,$?的意义.  ***$0,脚本的名称***  ***$\*,参数***  ***$$,进程号***  ***$?，错误***  4. 表达式。依次进行以下操作：  (1) 给变量i赋值为10，变量j赋值为5.  (2) 分别用算术扩展$(())和expr两种方式求表达式i\*5-j的值。    5.test命令。  (1)字符串操作符。依次输入以下命令,并解释执行结果。  $str1=abcd  $str2="abcd "  $test "$str1" = "$str2"  $echo $?    ***两个字符串相等***  (2)整数操作符。依次输入以下命令,并解释执行结果。  $str1=1234  $str2=01234  $[ "$str1" –eq "$str2" ]  $echo $?    ***两个数值不相等***  (3)文件操作符。依次输入以下命令,并解释执行结果。  $touch aa  $ [ -w aa ]  $ test -w aa  $ echo $?  $[ ! -d aa ]  $ echo $?  用ls -l命令进行查看，验证上述输出的正确性。    (4)逻辑操作符。利用（3）中的文件aa执行如下命令，并解释结果。  $[ -r aa -a -x aa ]  $echo $?    6. shell控制语句. 编写一个脚本，要求如下：  (1)把用户名作为输入参数(位置变量参数),每隔60秒检查一次用户是否登陆,直至该用户登陆。  (2)如果执行脚本时输入的参数数目不为1，则输出"Incorrect number of arguments"提示信息，并退出执行脚本。  (3)该用户登陆后，显示"the user 用户名 has logged on"。其中用户名对应于用户执行时输入的参数信息。      提示：  (1) 用[ "$# " –ne 1 ] 测试输入参数个数是否为1  (2) 用 who|grep "^$user "> /dev/null 判断用户$user是否登陆  (3) 用 sleep 60 表示每隔60秒进行某项操作。 | | | |
| 结论分析与体会：  1. Shell是一个命令行解释器，是内核与用户的接口，是互动式的解释和执行用户输入的命令，并将其传给系统。Shell又是一种程序设计语言，Shell script--变量和参数控制结构。Shell在用户登陆后启动。  2.Shell和用户对话的基本单位: Linux命令。  3. exit是Shell的一个内建命令，使用它可以立即终止Shell程序的执行  格式：  exit [n]  其中，n为一数字，表示希望返回的退出状态。如果不指定，就采用exit之前最后所执行命令的退出状态  0，则成功执行；非0则执行失败或某处有错误  如果直接在终端上执行exit命令，会导致从系统注销，因为它会终止登陆Shell的执行。  4.与和或：命令1 && 命令2：  只有命令1为真(即成功被执行，返回0），命令2才能够被执行。  $ ls s\* && rm s\*  命令1 || 命令2：  如果命令1执行失败，那么就执行命令2。  $cd /dir1 || echo cannot change to /dir1  5. 若要计算表达式2\*i+1的值，则为：  $i=2  $echo 2\*$i+1  进行变量的整数运算：  $echo $((2\*i+1))  6. 算数扩展提供变量的整数运算机制：  Shell的内建命令，老版本的Shell可能不支持  形式：  $((expression))  expression：算术表达式，由变量和运算符组成，运算符源于c语言。  用法：  显示输出：  echo $((expression))  变量赋值：  variable=$((expression))  7. expr 用于进行整数运算  格式：  expr 表达式  表达式由操作数和运算符组成。操作数一般是整数，也可以是字符串。  表达式的各部分(操作数和运算符之间)必须以空格分隔。  表达式中某些对shell有特殊意义的字符必须转义。  运算符可为：  + 加法  - 减法  \\* 乘法  / 整除  % 求模(余数)  =  8. 多个算术表达式可以组合在一起  $expr 5 + 7 / 3  还可以使用反引号改变计算次序：  $expr `expr 5 + 7` / 3  expr命令一般用于整数值，此外，还可用于字符串测试。  $s1=“hello”  $expr $s1 = “hello”  9.test命令：用于测试一种或几种条件  格式：  test expression  [ expression ]  使用方括号时，要注意在条件两边加上空格。涉及到变量值（$variable）的比较时最好把变量取值用双引号括起来，以免Shell误解变量值。  测试结果反映在退出状态中,而不是直接显示输出：  条件为真 =〉退出状态为0  条件为假 =〉退出状态非0  10. 测试的条件分为4类:  测试两个字符串之间的关系 、测试两个整数之间关系、测试文件是否存在或是否具有某种状态或属性。 、测试多个条件的与(and)或(or)组合。  分别对应4种不同的操作符：  字符串操作符、 整数操作符、文件操作符、逻辑操作符 | | | |

1. 当次实验结束后一周内按班组织上交实验报告。

2. 实验报告文件命名为：“学号+姓名”，格式为WORD文档。