实验报告

|  |  |  |  |
| --- | --- | --- | --- |
| 实验名称 | **实验六 Shell程序的创建及条件判断语句** | | |
| 实验教室 |  | 实验日期 | 2021年5月19日 |
| 学 号 | 2019224278 | 姓 名 | 张泽晨 |
| 专业班级 | 计算机科学与技术7班 | | |
| 指导教师 | 卢洋 | | |

东北林业大学

信息与计算机科学技术实验中心

|  |
| --- |
| 1. 实验目的   （1）熟练掌握Shell循环语句：for、while、until  （2）熟练掌握Shell循环控制语句：break、continue |
| 1. 实验环境   （1）计算机的硬件配置PC系列微机。  （2）计算机的软件配置VMware虚拟机软件及Ubuntu虚拟机。 |
| 1. 实验内容及结果 2. 编写一个shell脚本，利用for循环把当前目录下的所有\*.sh文件复制到指定的目录中，并为没有执行权限的文件添加执行权限。（可以在当前目录下先建立几个\*.sh文件，用来测试，复制到的指定目录可以自己建立一个）      1. 编写shell脚本，利用while循环求前10个偶数之和。  1. 编写shell脚本，利用until循环求1到10的平方和。     （4）运行下列程序，观察程序的运行结果。红色的语句分别为break，break 2，continue，continue2，观察四种情况下的实验结果。  #!/bin/sh  for i in a b c d   do  echo -n $i  　for j in 1 2 3 4 5 6 7 8 9 10      do        if [ $j -eq 5 ];then          break或continue        fi        echo –n “ $j”      done  echo $j   done  1.当为break时，外层for循环只有在内层for循环中的j==5时跳出当前循环，j只输出5     1. 当为break 2时，内层for循环中j==5时，直接跳出内外两层循环，因此只输出外层循环的第一个值a         3.当为continue时，内层for循环中的j==5时，直接跳过当前循环体内剩余的语句，直接进行下一次循环，因为只有当整个内层循环全部执行完之后才输出j，因为最后一次循环j=10，故输出j=10  4.当为continue 2时，内层for循环中j==5时，直接跳过两层循环体剩余的语句，因此不会执行第一层for循环内的 echo $j，而只会执行echo $i |

|  |
| --- |
| 1. 实验过程分析与讨论   for循环，while循环的使用  1、for循环  （1）for循环有三种结构：一种是列表for循环，第二种是不带列表for循环。第三种是类C风格的for循环。  （2）列表for循环  do和done之间的命令称为循环体，执行次数和list列表中常数或字符串的个数相同。for循环，首先将in后list列表的第一个常数或字符串赋值给循环变量，然后执行循环体，以此执行list，最后执行done命令后的命令序列。  Sheel支持列表for循环使用略写的计数方式，1～5的范围用{1..5}表示（大括号不能去掉，否则会当作一个字符串处理）。  Sheel中还支持按规定的步数进行跳跃的方式实现列表for循环  for循环对字符串进行操作，也可一使用for file in \*，通配符\*产生文件名扩展，匹配当前目录下的所有文件。  （3）不带列表for循环  由用户制定参数和参数的个数，与上述的for循环列表参数功能相同。  #!/bin/bash    echo "number of arguments is $#"    echo "What you input is: "    for argument  do  echo "$argument"  done  比上述代码少了$@参数列表，$\*参数字符串。  2、while循环         也称为前测试循环语句，重复次数是利用一个条件来控制是否继续重复执行这个语句。为了避免死循环，必须保证循环体中包含循环出口条件即表达式存在退出状态为非0的情况。  （1）计数器控制的while循环  #!/bin/bash    sum=0    i=1    while(( i <= 100 ))  do  let "sum+=i"  let "i += 2"  done    echo "sum=$sum"        指定了循环的次数500，初始化计数器值为1，不断测试循环条件i是否小于等于100。在循环条件中设置了计数器加2来计算1～100内所有的奇数之和。  （2）结束标记控制的while循环  设置一个特殊的数据值（结束标记）来结束while循环。  #!/bin/bash    echo "Please input the num(1-10) "  read num    while [[ "$num" != 4 ]]  do  if [ "$num" -lt 4 ]  then  echo "Too small. Try again!"  read num  elif [ "$num" -gt 4 ]  then  echo "To high. Try again"  read num  else  exit 0  fi  done    echo "Congratulation, you are right! "  （4）命令行控制的while循环        使用命令行来指定输出参数和参数个数，通常与shift结合使用，shift命令使位置变量下移一位（$2代替$1、$3代替$2，并使$#变量递减），当最后一个参数显示给用户，$#会等于0，$\*也等于空。  #!/bin/bash    echo "number of arguments is $#"    echo "What you input is: "    while [[ "$\*" != "" ]]  do  echo "$1"  shift  done  循环条件可以改写为while[[ "$#" -ne 0 ]]，等于0时退出while循环  3、until循环        until命令和while命令类似，while能实现的脚本until同样也可以实现，但区别是until循环的退出状态是不为0，退出状态是为0（与while刚好相反），即whie循环在条件为真时继续执行循环而until则在条件为假时执行循环。  #!/bin/bash    i=0    until [[ "$i" -gt 5 ]]    #大于5  do  let "square=i\*i"  echo "$i \* $i = $square"  let "i++"  done |
| 五、指导教师意见  指导教师签字： |