**Bash学习笔记**

# 函数

(1)函数定义：

函数名(){

执行体

}

(2)返回值

如果使用函数返回值，return只能返回一个整数（0~255），不能返回字符串，且返回值保存在$?变量中，不能直接赋值给其它变量。

有两种方式：

① 用$?赋值

func

i=$?

② 在函数中，用echo打印返回值，再赋值

function func () { echo 3; }

i=`func`

(3) 如果return没有指定参数，则是最后一行脚本的退出状态值

(4) 函数参数

① 向函数传递的参数被当作位置参量来处理，在函数中是本地变量

② 函数参数用$1, $2 ,..., $n来表示，但和通过命令行传递给脚本的参数不同。调用方式为：func param1 param2

③ 如果函数中要使用脚本的参数，只能将脚本的参数作为函数的参数传递给函数，例如，可以将脚本的第1个参数$1作为函数的第2个参数传给函数，那么函数则通过$2来访问脚本的第1个参数

|  |
| --- |
| #!/bin/bash  printArg(){  echo $1  }  # 判断是否有参数  if [ x$1 != x ];then  #...有参数  for arg in "$@"  do  echo `printArg $arg`  done  else  #...没有参数  echo '没有参数'  fi |

也可以直接向函数传递数据组

|  |
| --- |
| #!/bin/bash  printArg(){  echo "$@"  }  # 判断是否有参数  if [ x$1 != x ];then #...有参数  printArg "$@"  else #...没有参数  echo '没有参数'  fi |

# bash传入参数处理

格式：sh test.sh arg1 arg2 agr3......

新建一个test.sh，添加下面内容

|  |
| --- |
| #!/bin/bash  # 判断是否有参数  if [ x$1 != x ];then  #...有参数，处理参数  for arg in "$@"  do  echo $arg  done  else  #...没有参数  echo '没有参数'  fi |

执行命令：sh test.sh arg1 arg2 arg3

三种处理方式

(1) 手工处理方式

|  |
| --- |
| $0 表示./test.sh,即命令本身，相当于c/c++中的argv[0]  $1 表示第一个参数  $2,$3...... 表示第二、第三个参数，类推。  $# 表示参数的个数，不包括命令本身  $@ 表示参数本身的列表，也不包括命令本身，是一个参数数组  $\* 表示所有的参数解释成一个字符串 |