

# Shell 函数

2022年10月21日 17:36

## 1. 自定义函数

linux shell 可以用用户定义函数，然后在shell脚本中可以随便调用。

- shell中函数的定义格式如下：

```
[ function]funname [()]
{
action:[return int;]
}
```

- 说明：

1. 可以带function fun() 定义，也可以直接fun() 定义,不带任何参数。
2. 参数返回，可以显示加：return 返回，如果不加，将以最后一条命令运行结果，作为返回值。 return后跟数值n(0-255)

- 下面的例子定义了一个函数并进行调用：

```
#!/bin/bashdemoFun(){
    echo"这是我的第一个 shell 函数!"}
echo"-----函数开始执行-----"demoFun
echo"-----函数执行完毕-----"#输出结果：#-----函数开始执行-----#这是我的第一个 shell 函数!#-----函数执行完毕-----
    • 定义一个带有return语句的函数：
```

```
#!/bin/bashfunWithReturn(){
    echo"这个函数会对输入的两个数字进行相加运算..."echo"输入第一个数字: "readaNuM
    echo"输入第二个数字: "readanotherNum
    echo"两个数字分别为 $aNuM和 $anotherNum!"return$((($aNuM+$anotherNum))}
funWithReturn
echo"输入的两个数字之和为 $?!"#输出类似下面：#这个函数会对输入的两个数字进行相加运算...#输入第一个数字:#1#输入第二个数字:#2#两个数字分别为 1 和 2 !#输入的两个数字之和为 3 !
    • 函数返回值在调用该函数后通过 $? 来获得。
    • 注意：所有函数在使用前必须定义。这意味着必须将函数放在脚本开始部分，直至shell解释器首次发现它时，才可以使用。调用函数仅使用其函数名即可。
```

## 2. 函数参数

- 在Shell中，调用函数时可以向其传递参数。在函数体内部，通过 \$n 的形式来获取参数的值，例如，\$1表示第一个参数，\$2表示第二个参数...带参数的函数示例：

```
#!/bin/bash#author:菜鸟教程#url:www.runoob.comfunWithParam(){
    echo"第一个参数为 $1!"echo"第二个参数为 $2!"echo"第十个参数为 $10 !"echo"第十个参数为 ${10}!"echo"第十一个参数为 ${11}!"echo"参数总数有 $#个!"echo"作为一个字符串输出所有参数 $*" }
funWithParam 1 2 3 4 5 6 7 8 9 34 73
#输出结果：#第一个参数为 1 !#第二个参数为 2 !#第十个参数为 10 !#第十个参数为 34 !#第十一个参数为 73 !#参数总数有 11 个!#作为一个字符串输出所有参数 1 2 3 4 5 6 7 8 9 34 73 !
    • 注意，$10 不能获取第十个参数，获取第十个参数需要${10}。当n>=10时，需要使用${n} 来获取参数。
    • 另外，还有几个特殊字符用来处理参数：
```

参数处理	说明
##	传递到脚本的参数个数
#\$	以一个单字符串显示所有向脚本传递的参数
\$\$	脚本运行的当前进程ID号
#!	后台运行的最后一个进程的ID号
@	与\$*相同，但是使用时加引号，并在引号中返回每个参数。
-	显示Shell使用的当前选项，与set命令功能相同。
?	显示最后命令的退出状态。0表示没有错误，其他任何值表明有错误。

## 3. 系统函数

### 3.1 basename

- basename: 返回完整路径最后/的部分，常用于获取文件名
- basename [pathname] [suffix]
- basename [string] [suffix]
- basename命令会删掉所有的前缀包括最后一个/ 选项：suffix为后缀，如果suffix被指定了，basename会将pathname或string中的suffix去掉
- dirname: 返回完整路径最后/的前面的部分，常用于返回路径部分
- dirname 文件绝对路径：从给定的包含绝对路径的文件名中出去文件名（非目录部分），然后返回剩下的路径（目录部分）
- 反正两个系统函数都不要最后一个/