

Shell基础知识

1.1 简介

Shell是一个C语言编写的脚本语言，它是用户与Linux的桥梁，用户输入命令交给Shell处理，Shell将相应的操作传递给内核（Kernel），内核把处理的结果输出给用户。

1.2 第一个Shell脚本

```
# vim test.sh
#!/bin/bash
echo "Hello world"
```

执行Shell脚本的方式

1. `bash test.sh` // 不需要可执行权限
2. `source test.sh` // 不需要可执行权限
3. `chmod +x test.sh` // 添加可执行权限
4. `./test.sh` // 需要可执行权限
5. `sh test.sh` // 需要可执行权限

1.3 Shell变量

系统变量

编写shell脚本常用的系统变量：

```
$SHELL : 默认Shell
$HOME : 当前用户家目录
$LANG : 默认语言
$PATH : 默认可执行程序路径
$PWD : 当前目录
$UID : 用户ID
$USER : 当前用户
```

普通变量与临时环境变量

普通变量定义: `VAR=value`
临时环境变量定义: `export VAR=value`
变量引用: `$VAR`

位置变量

位置变量指的是函数或脚本后跟的第n个参数。

`$1-$n`，需要注意的是从第10个开始要用花括号调用，例如`${10}`

特殊变量

`$0` 脚本自身名字
`$?` 返回上一条命令是否执行成功，0为执行成功，非0则为执行失败
`$#` 位置参数总数
`$*` 所有的位置参数被看做一个字符串

\$@ 每个位置参数被看做独立的字符串
\$\$ 当前进程PID

1.4 变量引用

Shell中所有变量引用使用\$符，后跟变量名，有时个别特殊字符会影响正常引用，那么需要使用\${VAR}

1.5 双引号和单引号

在变量赋值时，如果值有空格，Shell会把空格后面的字符串解释为命令：

```
# VAR=1 2 3
-bash: 2: command not found
# VAR="1 2 3"
# echo $VAR
1 2 3
# VAR='1 2 3'
# echo $VAR
1 2 3
```

单引号是告诉Shell忽略特殊字符，而双引号则解释特殊符号原有的意义，比如\$、!

```
# N=3
# VAR="1 2 $N"
# echo $VAR
1 2 3
# VAR='1 2 $N'
# echo $VAR
1 2 $N
```

1.6 注释

在每行前面加个#号，即表示Shell忽略解释。

参考连接：<http://lizhenliang.blog.51cto.com/7876557/1881437>