bash 内部变量

echo eval exec expot readonly,read,shift,wait,exit ,点

echo 变量名

eval args

读入参数args，并将它们组成一个新的命令，然后执行。

Exec 命令参数

当shell执行到exec语句时，不会去创建新的子进程，而是转去执行制定的命令，当指定的命令执行完时，该进程（也就是最初的shell）就终止了，所以shell程序中exec后的语句将不再执行。

Export 变量名=value

Readonly 变量名

从标准输入读取字符串，传给指定变量

Local 变量名 局部变量

Shift 语句

Shift语句染着如下方式重新命名所有的位置参数变量，即$2成为$1,$3成为$2 在程序中每使用一次shift语句，都将所有的位置参数依次向左移动一个位置，并将位置参数$#减1，直至减到0为止

Shell

${#string} 字符串长度

-bash-3.00# OLDBOY="I am oldboy"

-bash-3.00# echo ${#OLDBOY}

11

-bash-3.00# echo $OLDBOY

I am oldboy

-bash-3.00#

${string:postion}

-bash-3.00# echo ${OLDBOY:2}

am oldboy

-bash-3.00#

${string:postion:length}

-bash-3.00# echo ${OLDBOY:2:2}

Am

${string#substring}

-bash-3.00# echo ${OLDBOY#am}

I am oldboy

-bash-3.00# echo ${OLDBOY#I am}

oldboy

-bash-3.00# echo ${OLDBOY#am}

I am oldboy

-bash-3.00# echo ${OLDBOY#I}

am oldboy

-bash-3.00#

-bash-3.00# echo ${OLDBOY%boy}

I am old

-bash-3.00#

变量替换

${value:-word}

变量的处理：计算变量的长度

-bash-3.00# chars=`seq -s " " 100`

-bash-3.00#

-bash-3.00# echo $chars

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100

-bash-3.00#

-bash-3.00# echo ${#chars}

291

-bash-3.00# echo $chars|wc -m

292

-bash-3.00# echo `expr length $chars`

expr: syntax error

-bash-3.00# echo `expr length "$chars"`

291

-bash-3.00#

-bash-3.00# time for i in $(seq 1111);do count=${#chars};done

real 0m0.202s

user 0m0.200s

sys 0m0.002s

-bash-3.00# echo $count

291

-bash-3.00# time for i in $(seq 1111);do count=`echo $chars|wc -m`;done

real 0m2.844s

user 0m0.968s

sys 0m2.812s

-bash-3.00# time for i in $(seq 1111);do count=`echo expr length "$chars"`;done

real 0m1.093s

user 0m0.368s

sys 0m0.725s

第一手资料：man bash

2015-02-02

变量的数值计算

(()),let,expr,bc $[]

各种（（））运算的shell脚本例子