Shell学习笔记

一、shell变量

命名规则：

首个字母必须为字母;

中间不能有空格，可以使用下划线(\_);

不能使用标点符号；

不能使用bash里的关键字

赋值方式:

可以显示地赋值，还可以用语句变量赋值： for file in `ls /etc`；

赋值时，变量名和等号之间不可以有空格

使用变量：

使用时在变量名前加 $ 即可， 如 $var， 使用变量时可以加 ｛｝，也可以不加 ｛｝，但从编程习惯来讲，推荐加上｛｝；

已经定义的变量可以被重新定义；

只读变量：

先给变量赋值，然后使用 readonly 声明变量即可

删除变量：

使用unset删除变量，但是unset无法删除只读变量

Eg： unset 变量名

变量类型：

1. 局部变量：尽在当前shell中有效，其他shell不能访问
2. 环境变量：
3. Shell变量：

二、shell字符串

Shell可以 用单引号，也可以用双引号，也可以不用引号。

单引号：

单引号里的任何字符都会原样输出，单引号字符串中的变量是无效的；

单引号字符串中不能出现单引号（对单引号使用转义字符也不行）。

双引号：

双引号里可以有变量；

双引号里可以出现转义字符；

拼接字符串：

可以使用双引号做拼接，也可以使用‘+’号做拼接。

获取字符串长度：

echo ${#string}: 输出String的长度

提取字符串：

Echo ${string:1:4} : 输出string变量中从位置1开始的4个字符，位置从0开始计算；

查找子字符串：

echo `expr index “string” is` : 输出字符‘i’或‘s’的位置

三、shell数组

Shell只支持一维数组，不支持多维数组，并且没有限定数组的大小。且数据下标从0开始。

定义数组：

Shell中数组用“（）”表示，数组元素用“空格”分割开，一般形式为

数组名=（值1 值2 值3）；

可以单独定义数组的各个分量：arr[0] = val0，可以不使用连续的下标，且下标的范围没有限制；

读取数组元素：

${数组名[下标]}；

获取数组中的所有元素：

${array[@]}

获取数组的长度：

echo ${#arr[@]} 或 echo ${#arr[\*]}

获取数组单个元素的长度：

Echo ${#arr[n]}

四、shell注释

#用来注释；

没有多行注释，可以采用加｛ ｝的方式，将其视为一个函数，不调用即可实现多行注释的效果

五、shell传递参数

执行shell脚本时，可以传递参数，脚本内获取参数的格式为： $n. n代表一个数字，1为执行脚本的第一个参数， 2为执行脚本的第二个参数，依次类推。

（1）$0 代表的是脚本本身的名称；

（2）处理参数的几个特殊字符：

$#：传递到脚本的参数的个数

$\*: 以一个单字符串显示所有向脚本传递的参数

$@: 以多个字符串显示所有向脚本传递的参数，每一个参数对应一个字符串

$$: 脚本运行的当前进程ID号

$?: 显示最后命令的退出状态。0表示没有错误，非零表示有错误。

$\*与$@的区别：

相同点：都是引用所有参数

不同点：只有在双引号中体现出来。假设脚本运行时写了3个参数1， 2， 3， \*等价于“1 2 3”，@ 等价于“1”，“2”，“3”。