**shell脚本**

1. **shell脚本介绍：**
2. linux的shell可以执行众多命令，shell脚本其实就是把这些命令放在一个文件中，然后依次执行，就可以完成对linux系统的各种复杂操作。所以对于运维人员，shell脚本是一定要会的，各种对系统的维护，对项目的部署都需要shell脚本来完成。对于开发人员，会一些简单的shell脚本就可以了。
3. **shell脚本的查看网站：shell脚本没有权威网站，只要搜索学习shell脚本的常用网站，随便一个都可以，都差不多。**

**在这里推荐一个中文shell脚本网站。阮一锋的bash脚本网站，对于开发人员来说，shell脚本简单的会一些就可以了。**

**网址：**[**https://wangdoc.com/bash/intro**](https://wangdoc.com/bash/intro)

1. shell脚本的基础知识：shell脚本就是将诸多命令放在一个文件中依次执行，只不过为了控制这些命令的执行，shell脚本必须提供诸多控制命令执行的功能。
2. shell脚本的启动方式
   * 1. 直接./的方式启动
     2. 指定bash来解析脚本

一般来说，直接./就可以了，简单，常用。

1. shell脚本的一些基础知识
   * 1. **shell脚本的第一行为#!/bin/bash**，这一行可以告诉bash使用bash来解释脚本，如果没有这一行，系统就会使用默认的shell来解释脚本，在大多数情况下不会出现错误，但少数情况下，bash和shell不大的区别就会导致错误。
     2. “#”表示注释，对于任何语言，注释都是非常重要的，shell脚本中“#”表示注释。
     3. 在执行脚本之前，需要给脚本添加执行权限。
     4. “”：使用双引号，变量将被当做变量解释
     5. ‘’： 使用单引号，变量将被当做字符串解释
     6. `` （数字键1左面的反引号）：反引号包含的变量将被解释器解释为系统命令，然后执行命令。
     7. 转义字符：主要就是\n，\t。echo命令使用转义字符需要加上-e参数。
     8. () ：执行“()”中的命令不会影响“()”外的变量。
     9. {} ：执行“{}”中的命令会影响“{}”外的变量

**2. shell脚本的基础语法**

1. 变量的分类：

(1) 变量可以划分为三类（不同的人划分方式，名称可能略有区别，但内容是一样的）

(2) 普通变量：

1. readonly标识符：代表变量不可更改
2. unset表示符：将num变量设置为空。
3. **在shell脚本中，变量在定义时默认是不指定类型的。在shell脚本中，变量简单的分为字符串类型，整数类型就可以了，默认变量为字符串类型。**
4. **如果我们想要将一个变量定义为整数类型，可以使用declare -i来显示定义**

(3) **内部变量，也可以叫预设变量，预定义变量，就是shell已经预定义的变量。**

1) $# 传给shell脚本参数的数量

2) $\* 传给shell脚本参数的内容

3) $1，$2，$3，$4... 表示第一个，第二个，第三个...命令行参数

4) **$? 用于检查上一个命令是否执行成功，如果执行成功，则返回0，否则返回非0的数。这个变量非常有用，一定要会!**

5) $0 当前进程名称

6) $$ 当前进程的进程号

(4) 环境变量：在shell脚本中可以直接使用bash的环境变量。比如可以定义一个新的环境变量，这样其它脚本也就可以看到该环境变量了，可以类比C语言的全局变量。或者修改PATH或LD\_LIBRARY\_PATH的值，以方便之后或其它脚本使用PATH或LD\_LIBRARY\_PATH

实际上，运维人员环境变量使用的比较多，只要写一些简单的脚本就可以了，环境变量用的极少。

2.变量的判断：

(1) ${num:-val}：如果num存在，则整个表达式的值为num，否则整个表达式的值为val。

(2) ${num:=val}: 如果num存在，则整个表达式的值为num，否则整个表达式的值为val，且将num赋值为val。

3.对字符串的操作：

(1) 获取字符串的长度：格式为“${#str}”

(2) 从字符串中提取字符串：格式为“${str:n1: n2}”，从n1的位置开始提取n2个字符

(3) 对字符串进行替换：格式为“${str/old/new}”用new来替换第一个匹配到的old字符串。

**3. 判断语句**

判断语句也就是对各种功能进行判断，可以包括文件判断，字符串判断，数值判断，逻辑判断：格式为 test condition 或 [ condition ]。其实还是[ condution ]用的比较多。

判断语句非常重要，一定要熟练掌握。**判断语句主要是语法，下一节就有应用了，所以如果不理解，就死记住。**

1. 文件判断：判断文件的各种属性，主要有以下参数：
   * 1. -e ：（exist）判断文件是否存在
     2. -d ：（directory）判断是否为目录
     3. -f ：（file）判断是否为普通文件
     4. -r ： （read）判断可读
     5. -w ： （write）判断可写
     6. -x ： （execute）判断可执行

其实参数还有很多，比较常用的就这些了。

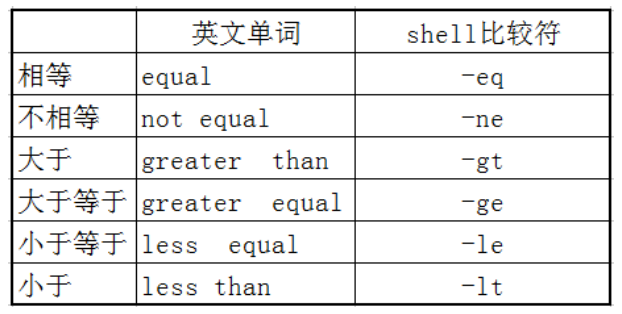
1. 字符串判断

**注意：在使用字符串进行判断时，一定要加上””，否则容易出现莫名其妙的错误。**

* + 1. -z : （zero）表示空串
    2. -n: （not zero）表示不是空串
    3. == ：表示两个字符串相等，其实这里使用=也可以，**但写=就必须留空格，否则就变成赋值了。**
    4. != ：表示两个字符串不相等

**在判断相等或不相等时，推荐在变量后加“x”，在某些版本的bash中，如果一个字符串为空，会直接报错的。**

1. 数值判断：因为“>”，“<”等已经被重定向占用了，所以比较数值也只能采取这种比较麻烦的方式了。



1. 逻辑判断
   * 1. [ ! expr ] 逻辑非。
     2. [ expr1 -a expr2 ] （a为and）逻辑与，只有两个表达式均为真，结果为真。
     3. [ expr1 -o expr2 ] （o为or）逻辑或，expr1或expr2为真，则结果为真。
     4. **[] || [] 用或来合并表达式**
     5. **[] && [] 用且来合并表达**

**4. 控制语句**

**注意：学习shell脚本的控制语句可以类比c语言的控制语句，作用是一样的。**

1. if语句：
   1. 格式一：if [条件1]; then

执行第一段程序

else

执行第二段程序

fi

* 1. 格式二：if [条件1]; then

执行第一段程序

elif [条件2]; then

执行第二段程序

else

执行第三段程序

fi

1. case语句：
   1. 格式：case $变量名 in

“第一个变量的内容”）

程序段一

;;

“第二个变量的内容”)

程序段二

;;

\*)

其它程序段

;;

esac

1. for语句：
   1. 格式1 for (( 初始值；限制值；执行阶段))

do

程序阶段

done

**注意：这种方法支持c语言的循环，也就是说可以使用“<，>”等符号来表示范围了。但是不能在for内部定义变量**

* 1. 格式2 for var int num1 num2 num 3

do

程序阶段

done

1. whie语句：可以理解为，但满足什么条件时，就会执行
   1. 格式1 while []

do

程序阶段

done

* 1. 格式2 while 语句

do

程序阶段

done

1. until语句：until语句和while刚好相反，until为条件为假时执行接下来的语句。**可以理解为：当满足什么条件时，就不执行了**

这些没什么讲的必要，都是语法，记住就可以了。多用一用就熟练了。

**5. 函数**

shell脚本的函数和C语言的函数类似。大家可以类比的学。

1. 格式 function 函数() 。其中function和()可以省略一个。
2. 函数的返回值。函数的返回值有两种格式。
   1. return的形式：

注意：shell的函数和c语言的return关键字意义并不相同。在shell脚本中，return关键字主要用来返回函数的执行状态，如果为0，则函数正确，否则视为出现错误，return可以返回我们想要的错误码。

注意：return只能返回0-255的数字，超过这个返回就会出错。

* 1. echo的形式：

echo更接近c语言的返回值，我们一般用echo的形式来接收代码的返回值。

1. 在函数内部：$#，$\*等内容表示函数的参数情况。

总结：shell脚本就这样了。开发只要写一些简单的shell脚本就可以了，这些也就够用了，如果想要扩展，可以去网站上查看。

**shell脚本擅长运营系统，执行shell脚本提供的各种命令。如果是进行其他操作，那还是C++方便的多。对于开发来说，shell脚本终究只是个辅助。**