**三 shell 函数**

1. **函数定义**

**function 函数名()**

{

命令

...

return 数字值

}

**2. 函数调用**

第一种调用方法:

**函数名 参数1 参数2 ...**

**通过 $? (上一个函数或者命令返回的结果)，获得函数的返回值**。函数本身要在屏幕上输出结果。

(**适合返回的是一个整数，并且只能够返回0-255之间的整数**)。

**在函数内部通过位置变量($1, $2, $3 ...)来获得传递的参数**。

第二种调用方法:

**value=`函数名 参数1 参数2 ...`**

**将函数的标准输出返回给一个变量（使用反向单引号）**，**函数本身不再在屏幕上输出结果。**相当于是把函数的输出赋值给一个变量，而并没有使用函数本身的返回值。

**无论函数定义中有没有返回值，这种调用方式函数返回值始终是0**。

(**适合函数输出/返回的是一个字符串**)

注意:在shell中变量默认的作用域是整个文件的，即使是在函数内部定义的函数，其它函数也可以使用。如果需要变量只在函数内部有效，需要用local修饰变量。

函数的载入

如果函数在另外一个文件中，我们该怎么调用它呢？

比如 show 函数写在了function.sh里面了，我们可以用 source 命令：

source function.sh

这样就可以直接调用该函数了。

**Practice:**

**1. Design a function to receive the username and find the user in /etc/passwd, return its line number (if it exists: return line number, if it does not exist: return 0). (Tip: If the line number is an empty string, the user is not found)**

**2. Design a function to receive the line number, and outputs the corresponding information in the file /etc/passwd. The information must be output as follows**

**The user info: NAME:UID:GID**

**3. Read the username repeatedly, and then call the designed functions, output the user's information, if the user does not exist, report to the users that the username does not exist in the system. If the input is "quit", then the program ends.**

**You are required to complete all the above functions and operations in one script file.**

**1.设计一个函数，通过传递的用户名来寻找用户在/etc/passwd中的行号(存在:返回行号，不存在:返回0)。 （提示：如果行号为空字符串，则表明未找到该用户）**

**2.设计一个函数，通过传递的行号，输出这一行对应/etc/passwd文件中用户信息如下**

**The user info: NAME:UID:GID**

**3.循环读取用户输入的用户名，然后分别调用设计的函数，输出用户的信息，如果用户不存在，则告诉输入者这个用户在系统中不存在，如果输入的是"quit",则结束。**

**注：以上函数和操作都要在同一个脚本文件中完成。**