Exit 谁知推出状态

默认 exit 0 表示成功

exit 1 表示失败

和return的关系 return用户函数中，如果没有函数，则两者是相同的

If的参数 （之一空格 if [ -d $1 ] ）：

（1）文件测试：

-d 测试目标是否存在，并且是个目录

-f 测试目标是否存咋，并且是个普通文件

-L 测试其是否是个连接

-b 测试是否是个块设备

-c 测试是否是个字符设备

-e 测试指定的文件或目录是否存在

-w 判断文件是否存在且拥有写入权限

-r 判断文件是否存在且拥有可读权限

-x 判断文件是否存在且拥有可执行权限

-u 判断文件是否存在且具有SUID权限

（2）字符串及数值测试：

= 测试字符串是否相等

！= 测试字符串是否不等

-n 测试字符串是否为空 [ -n “$1”]

数值测试： [ $1 –eq $2 ]

-eq 如果两个数相等返回真

-ne 如果两个数不等返回真

-lt 如果1<2返回真

-le 如果1<=2返回真

-gt 如果1>2返回真

-ge 如果1 >=2返回真

（3）逻辑操作符：

-a 逻辑与

-o 逻辑或

！ 逻辑非

If语句的结构：

（1）最简单的形式

If [ $# = 0 ]

Then

命令/函数

Fi

（2）if –else- 结构

If [ $# = 0 ]

Then

命令/函数

Else

命令/函数

Fi

（3）if –elseif-else—结构

If [ $# = 0 ]

Then

命令/函数

Elif [ $# = 0 ]

Then

命令/函数

Else

命令/函数

Fi

Switch语句的格式：

Casee 变量 in

变量可能以）

命令/函数

；；

变量可能以）

命令/函数

；；

变量可能以）

命令/函数

；；

变量可能以）

命令/函数

；；

Esac

For语句结构（要给出递进的范围）：

例子：

J=”1,2,3,4,5,6”

For LOOP in $j

Do

命令/函数

Done

While循环语句（常用的，相比于until语句）：

格式：

While [ $# -lt $1 ]

Do

Done

捕获系统信号（当在执行重要任务时，如果不捕获中断信号，那脚本直接推出运行）：

可以捕获的信号：

1（signup)： 来自终端的挂起或来自控制进程的死亡信号

2（sigint）： 来自键盘的中断信号

3（sigquit）： 来自键盘的推出线号

15（sigterm）：终止信号

捕获的方法：trap “命令（或者是函数）” 信号

Break和continue的作用：

Break的作用是跳出循环语句

Continue的作用是控制循环，跳过本次循环，进入下一个