**shell day 03**

================================================

**while循环，可以根据条件测试决定循环次数，可以实现无限循环**

**基本语法结构：**

**while 条件测试 //如果条件测试得到成功的结果就执行下面的指令，然后再回来继续看条件测试能否成功，如果成功就继续执行指令，且可以实现无限循环，一旦发现条件测试失败了，就立刻终止循环**

**do**

**指令**

**done**

-------------------------------

**另外如果想故意创造一个无限循环可以在while后面写冒号即可**

while : //冒号代表测试永远正确

do

echo abc //循环任务

sleep 0.1 //如果系统执行任务消耗cpu比较多，可以每次稍微休息一下

done

#!/bin/bash

x=1

while [ $x -le 5 ] //通过条件测试还可以决定执行任务的次数

do

echo abc

sleep 0.1

let x++

done

#!/bin/bash

for i in 1 2 3

do

for j in a b c //使用for循环的嵌套可以实现复杂需求

do

echo -n '\* ' //-n选项可以实现输出但是不回车换行

done

echo //空执行一个echo可以实现回车换行效果

done

--------------------------------------------------------------------------

**case分支，功能类似if，不如if强大，语句比if精简**

**基本语法格式：**

case 调用变量名 in //如果调用的变量内容与下面某个模式一致，就执行模式下面的指令

模式1) //这里的模式可以有很多

执行指令;; //指令需要用双分号结尾，如果一个模式有多个指令，那只需在该模式的最后一条指令后加双分号即可

模式2)

执行指令;;

…

\*) //如果上述模式都没有被匹配，那就匹配这个ZZ

执行指令

esac

#!/bin/bash

case $1 in

a)

echo abc;;

b)

echo xyz;;

\*)

echo "请输入a或者b"

esac

**练习：**

**测试case分支的实际应用**

#!/bin/bash

case $1 in //使用执行脚本后的第1个位置变量作为匹配对象

t|T|tt) //如果$1是t或者T或者tt，都可以算匹配

touch $2;; //此处是创建文件的命令，后面是第二个位置变量的参数

m)

mkdir $2;;

r)

rm -rf $2;;

\*)

echo "t|m|r"

esac