一：Shell是什么

Shell是内核之上的一个外壳，就像鸡蛋，鸡蛋有一个鸡蛋外壳一样，他是保护内核的。另外一个角度，我们可以通过Shell可以与内核进行交换，内核本身是Linux的心脏，从开机自检就驻留在计算机内存中，直到计算机关闭为止，而用户的应用程序他会存储在计算机的硬盘上，当我们运行这个应用程序的时候，他会被调入内存。Shell本身也是一个应用程序，用户登录进Linux系统的时候，shell就会被调进内存中去执行。例如bash（终端）

你可以任务Shell是开发者与内核进行交换的一个桥梁，他有输入设备的读取指令，例如ls指令，它读取这个指令之后，然后就是解释执行。

Shell有许多的种类：我学习的是Bash。

Shell的存在基本有两种方式。一种是直接基于Shell（终端）直接书写交互式命令，第二种在shell文件中，他不过是把许多指令封装起来了而已。

二：实战

1，”#!”代表shell脚本的开始符号，通过他来指定具体文本类型的特殊标记。在其之后指定Linux系统中解释执行当前Shell文件的具体的解释器，一般是：#!/bin/bash或者#!/bin/sh

Vim who.sh

#!/bin/bash //说明脚本的解释器

data

who

ls –l /usr/local

2，当执行shell文件的时候需要有执行权限，使用chmod来改变执行权限，例如

chmod u+x 文件名

3，在Shell文件中可以通过”#”后面直接加上注释的内容来增加Shell的可读性和可维护性；

4，shell脚本中其实是基本命令的组合，在交互终端中可以写入什么内容，那么在Shell脚本中就可以写入什么内容；

5，同一行如果有多条指令的话，我们使用“；”来分隔开不同的指令；

#!/bin/bash //说明脚本的解释器

Data;who;ls –l /usr/local

三：Shell中变量及引用

1. 变量可以分为三种类型
   1. 本地变量：只有当前Shell声明周期内有效；
   2. 环境变量：在用户登录后到注销前对于所有的编辑器、脚本、进程等都有效。
   3. 位置参数：主要使用于向Shell脚本传递参数；例如

spark-shell –master spark：//master:7077

1. ${variablename}会具体提取出变量的内容，当然可以个{}去掉。或者“$变量名”，加不加引号都没有关系

#! /bin/bash

x1="hello word!!!"

echo "$x1"

echo ${x1}

echo $x1

echo "${x1}"

1. 如果要打印变量的内容，可以通过echo $variablename来打印，例如echo $Usage
2. 在Shell脚本中编写关于变量的基本使用

#!/bin/bash

a=100

let "a+=10" #a plus 10

echo "a=$a"

declare -i b

b=10

echo "b=$b

1. 环境变量

使用export关键字来把变量变成环境变量。例如

export SCALA=....

可以通过使用“env”命令来查看系统中的环境变量的内容

1. 系统中核心的配置文件

常见的有 .bashrc bash\_profile,修改了系统文件后一般要通过source命令使其立即生效

1. 位置参数：主要用于从命令行向shell脚本传递参数

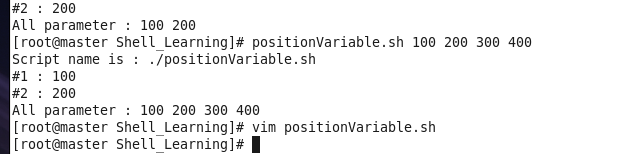
具体而言，$0代表脚本的名称，$1代表传入shell的第一个参数，$2代表传入shell的第er个参数.....，如果要代表全部参数，则需要使用$\*或者$@.

echo "Script name is : $0"

echo "#1 : $1"

echo "#2 : $2"

echo "All parameter : $@"

~ 

~

常用命令：

$# ： 代表传到脚本中的参数的数量

$$ ：脚本运行的进程号

$? ：代表命令退出状态，0表示没有错误，非0表示有错误；

1. 命令替代

export SPARK\_HOME="$(cd "`dirname "$0"`"/..**;** pwd)"

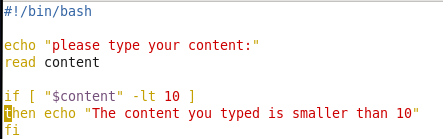
命令替代有两种方式 第一种是` ` 另一种是$()

"$(cd "`dirname "$0"`"/..; pwd)"包含了两条命令，第一条命令是(cd "`dirname "$0"`"/..

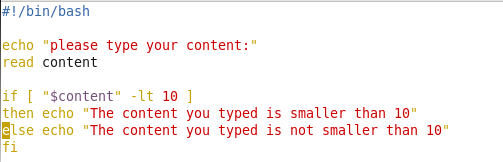
执行dirname $0 然后cd进入，然后/..表示返回上一级目录。

Pwd显示当前的目录并返回，第一条指令和第二条指令的结果是一样的。

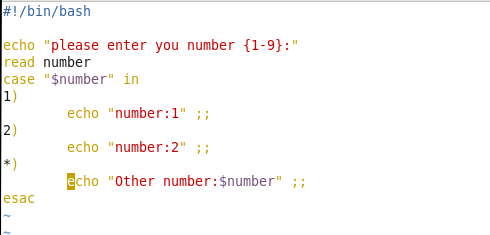
1. 控制语句—条件判断
   1. 使用if，基本语法：if...then...fi，注意，if条件判断后如果没有“；”则需要把then换行，若果if和then处于同一行的话，则需要在then前面加上“；”；要终止if语句使用“fi”即可



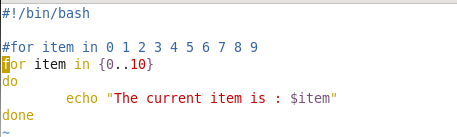
* 1. 复杂的if...then...else...fi



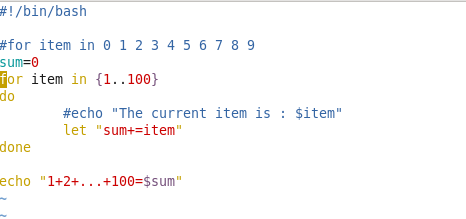
补充说明：case表达式用于多分支选择语句，会根据表达式的值来选择要执行的语句，如果遇到匹配的值，就执行该值后面的语句，直到遇到“；；”为止，你可以使用\*）来匹配所有未匹配到具体值得内容，esac作为case代码块的结束，实例：



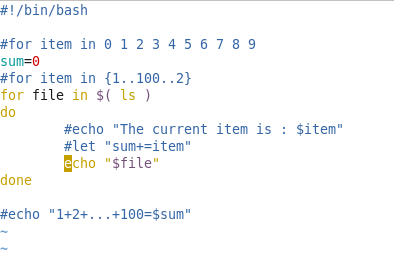
1. 循环
   1. 循环的类型：for、while、until；
   2. 循环的控制语句break和coutinue；
   3. for类型循环实例：



求1+2+3.。。+100的和



循环打印出所有的当前目录下的文件（夹）

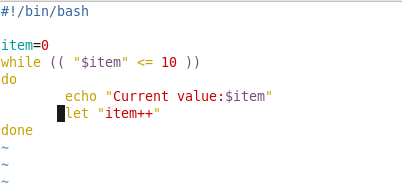


1. while循环实战

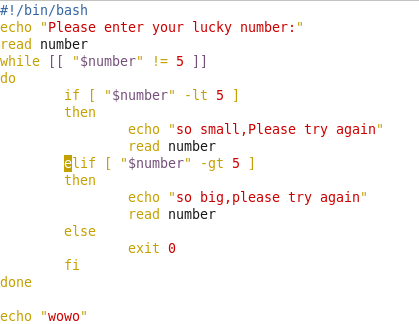
while循环会根据测试语句决定是否循环及循环的次数

while expression;do...done

1. 最简单的while循环：



1. 有停止循环标记的while



1. 使用shift命令，shift本身是使得变量的位置下移动一位,$5代替$4



1. Exec

一般exec和ssh配合会启动分布式系统！

Exec命令可以通过文件标识符打开或者运行具体的文件，也可以将文件重定向到标准输入或者输出文件中