Shell编程

shell :弱类型、 解释型语言

解释器：bash

## 一、变量

bash的变量类型：

环境变量

本地变量（局部变量）

位置变量

特殊变量：bash内置的用来保存某些特殊数据的变量。（也叫系统变量）

本地变量：只属于某一个bash的变量。

var\_name=值

作用域：整个bash进程

局部变量：

local var\_name =值，

作用域：当前代码段。

环境变量：

export 名字=值

作用域：当前的shell和其子shell。

注意：脚本在执行时都会启动一个子shell进程：

命令行中启动的脚本会继承当前shell环境变量。

系统自动启动脚本（非命令行启动）：则需要自我定义环境变量。

位置变量： 用于 脚本执行的参数，$1 表示第一个参数，以此类推

$1,$2….

特殊变量：

$? :上一个命令的执行状态返回值。

$# 传递到脚本的参数个数

$\* 传递到脚本的参数，与位置变量不同，此选项参数可超过9个

$$ 脚本运行时当前进程的ID号，常用作临时变量的后缀，如 haison.$$

$! 后台运行的（&）最后一个进程的ID号

$@ 与$#相同，使用时加引号，并在引号中返回参数个数

$- 上一个命令的最后一个参数

$? 最后命令的退出状态，0表示没有错误，其他任何值表明有错误

程序有两类返回值：

1. 执行结果
2. 执行状态，$? : 0:表示正确，1-255：错误

输出重定向：

>覆盖重定向

>> 追加重定向

2> 错误覆盖重定向

2>>错误追加重定向

&> 全部重定向

撤销变量：

unset 变量名

查看shell中变量：

set 命令

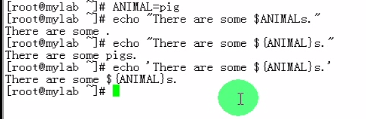
查看shell中的环境变量

printenv

env

export

引用变量：${变量名}，一般可以省略{}



单引号：强引用，不作变量替换

双引号：弱引用，做变量替换

反引号：``命令替换

## 二、脚本

脚本：命令的堆砌。

练习：写一个脚本，完成以下任务。

1. 添加5个用户，user1，，，，user5
2. 每个用户的密码同用户名，要求：添加密码完成后不显示passwd执行结果。
3. 显示添加成功信息

练习2：写一个脚本，完成以下任务。

1. 使用一个变量保存一个用户名
2. 删除此变量中的用户，且一并删除其家目录
3. 显示“用户删除成功”信息。

条件判断：

条件表达式：

1. [ expression ]
2. test expression

整数比较：

-eq ： 比如：[ $A –eq $B ]

-ne ， -gt ,-lt,-ge,-le

命令的逻辑关系：

在linux 中 命令执行状态：0 为真，其他为假

逻辑与： &&

第一个条件为假时，第二条件不用再判断，最终结果已经有；

第一个条件为真时，第二条件必须得判断；

逻辑或： ||

逻辑非： ！

命令执行的状态 的逻辑关系

1、如果用户user6不存在则添加用户6

！ id user6 && useradd user6

Id user6 || useradd user6

2、如果用户不存在，添加用户并显示添加成功，否则显示其已存在

3、如果/etc/inittab文件的行数大于50，就显示好大的文件；

练习：

1. 添加3个用户user1.user3,并且设置密码和用户名相同，如果用户已经存在，显示：已存在
2. 添加完成之后计算系统一共有多个用户。

练习：给定一个用户，

1. 如果其UID=0，就显示其为管理员，否则显示其为普通用户

备注：

passwd默认是要用终端作为标准输入,加上--stdin表示可以用任意文件做标准输入  
于是这里用管道作为标准输入

条件判断，控制结构：

If 条件 ；then

语句

elif 条件 ； then

语句

else

语句

fi

-a ： 逻辑与，并且 ： if [ $# -gt 1 –a $# -lt 3 –o $# -eq 2 ] ; then

-o ：或者 比如：

练习：判断命令历史中历史命令的总条目大于500，如果大于，则显示“Some command is done.”,否则显示：“OR”。

练习：给定三个整数，判断其中的最大值和最小数。并显示出来

bash -n shell文件 ：检查文件是否有语法错误。

bash –x shell 文件 ：debug 执行文件

Shell 中如何算术运算

1. let 算术运算表达式

let C=$A + $B

2、$[算术表达式]

C = $[$A+$B]

3、$((算术表达式))

C=$(($A+$B))

1. expr 算术表达式 ，注意：表达式中各操作数及运算符之间要有空格。而且要使用命令引用

C=`expr $A + $B`

练习：给定一个用户，获取其密码警告期限，然后判断用户密码使用期限是否已经小于警告期限，如果小于，则是显示“WARN” ，否则显示密码还有多少天到期。

提示：date +%s :今天的秒数

Cat /etc/shadow 密码时间。

exit : 退出脚本

退出脚本可以指定脚本执行的状态：exit 0 。

复习：

测试方法：

[ 表达式 ]

[[ 表达式 ]]

test 表达式

INT1=33

INT2=32

[ $INT1 -eq $INT2 ]

[[ $INT1 -eq $INT2 ]]

test $INT1 -eq $INT2

文件测试： [ ] 需要中括号

-e FILE :测试文件是否存在

-f FILE :测试文件是否为普通文件

-d FILE ：测试文件是否为目录

-r 权限

-w

-x

特殊变量:

$# $@

字符串测试:

== 等号两端需要空格

!=

-n string : 判断字符串是否为空

-s string : 判断字符串是否不空

练习:指定一个用户名,判断此用户的用户名和它的基本组 组名是否相同.

#!/bin/bash

if ! id $1 &>/dev/null ; then

echo “No such user.”

exit 12

fi

if [ $1 == `id –n –g $1` ] ;then

echo “xiangtong”

else

echo “bu xiangtong”

fi

练习: 判断当前主机的CPU生产商，（其信息保存在/proc/cupinfo文件中）。

如果是：AuthemticAMD ,就显示其为AMD公司

GenuineIntel ，就显示其为 Intel公司

否则，就显示其为非主流公司。

练习：将那些可以登录的用户查询出来，并且将用户的帐号信息提取出来，后放入/tmp/test.txt文件中，并给定行号。在行首。

循环：进入条件，退出条件

for 变量 in 列表 ； do

语句

done

比如： for I in 1 2 3 4 5 ；do

语句

done、

如何生成列表：

1、{1..100}

2、seq [起始数] [跨度数] 结束数

3、ls /etc 文件列表

练习：依次向/etc/passwd中的每个用户问好：hello 用户名，并显示用户的shell：

Hello ，root ，your shell ：/bin/bash。

1. 只向默认shell为bash的用户问好。

While 循环

格式一

while 条件;do

语句

[break]

done

格式二 死循环

while true

do

语句

done

格式三 死循环

while :

do

语句

done

格式四 死循环

while [ 1 ]

do

语句

done

格式五 死循环

while [ 0 ]

do

语句

done

练习：计算100以内所有能被3整除的整数的和

练习：使用echo输出10个随机数，并且一行显示。提示：$RANDOM

练习：传给脚本一个参数：目录，输出该目录中文件最大的，文件名和文件大小：

比如：1.txt 100KB

ls -l | awk '{print $5,$9}' | sort -nr

2、查看该目录下是否有大小为0的文件，如果有则删除。同时显示删除信息。

练习：查询当前192.168.1.x网段内，那些IP被使用了，输出这些IP到一个文件中。

练习：请根据一个关键字，杀掉系统进程中包含此关键字的进程。

echo –n $RANDOM

case 语句

case 变量 in

value1）

语句

；；

value2）

语句

；；

\*)

语句

；；

esac