条件测试

1,字符串

== != -z ! -z -n

2,数字

-eq -ne -gt -ge -lt -le

3,文件

-e -f -d -r -w -x

&& || ;

单分支

if 条件测试 ;then

命令序列

fi

双分支

if 条件测试 ;then

命令序列1

else

命令序列2

fi

多分支

if 条件测试 ;then

命令序列1

elif 条件测试 ;then

命令序列2

elif 条件测试 ;then

命令序列3

else

命令序列X

fi

循环

for 变量名 in {1..10} 或 `seq 10`

do

命令序列

done

while :

do

命令序列

done

case分支, 是简化版本的if,代码编写比if精简,但功能不如if强大

case 变量 in

模式1)

命令序列;;

模式2)

命令序列2;;

\*)

命令序列3

esac

case $1 in

aa|a|A)

echo aaaaa;;

bb)

echo bbbbb;;

\*)

echo "aa|bb"

esac

1,在真机找到/linux-soft/02,释放tar包

tar -xf lnmp\_soft.tar.gz -C /home/student

2,进入释放的目录找到nginx-1.12.2.tar.gz文件,拷贝到虚拟机

cd /home/student

scp lnmp\_soft/nginx-1.12.2.tar.gz root@172.25.0.10:/opt

3,回到虚拟机编写一键源代码安装nginx服务的脚本

#!/bin/bash

yum -y install gcc openssl-devel pcre-devel

tar -xf /opt/nginx-1.12.2.tar.gz

cd nginx-1.12.2

./configure

make

make install

运行该脚本,自动安装nginx

4,运行nginx服务

cd /usr/local/nginx/sbin

./nginx

关闭防火墙后,打开火狐浏览器测试

#!/bin/bash

case $1 in

st|start)

/usr/local/nginx/sbin/nginx;;

stop)

/usr/local/nginx/sbin/nginx -s stop;;

rest)

/usr/local/nginx/sbin/nginx -s stop

/usr/local/nginx/sbin/nginx;;

stat)

netstat -ntulp | grep -q nginx

[ $? -eq 0 ] && echo "服务已启动" || echo "服务未启动";;

\*)

echo "st|stop|rest|stat"

esac

函数,可以定义公共的语句块,在脚本中反复调用,达到精简脚本的目的

函数名 () {

}

#!/bin/bash

abc () { //定义函数,名字是abc

mkdir $1 //函数中的任务是创建并进入目录

cd $1

}

abc abc //调用函数(函数名是刚刚创建的abc),创建并进入abc目录

脚本写完后使用 source test4.sh 测试

function 函数名称 {

}

#!/bin/bash

abc () {

echo -e "\033[$1m$2\033[0m"

}

abc 31 ABCDEFG

abc 32 ABCDEFG

abc 33 ABCDEFG

abc 34 ABCDEFG

abc 35 ABCDEFG

abc 36 ABCDEFG

abc 37 ABCDEFG

exit 终止脚本程序

break 跳出循环,执行循环后任务

continue 结束本次循环,继续下一次循环

#!/bin/bash

x=0

while :

do

read -p "请输入数字(0是结束)" n

[ -z $n ] && break //没给任何数字就退出循环

[ $n -eq 0 ] && break //给0也退出循环

let x+=n

done

echo "所有数字之和是$x"

改良版

#!/bin/bash

x=0

while :

do

read -p "请输入数字(0是结束)" n

[ -z $n ] || [ $n -eq 0 ] && break //没给数字或者给的是0都退出循环

let x+=n

done

echo "所有数字之和是$x"

1~20 找出6的倍数并分别加10

#!/bin/bash

for i in {1..20}

do

x=$[i%6] //x是余数

[ $x -ne 0 ] && continue //如果余数不是0,结束本次循环,进入下一次循环

echo $[i+10]

done

字符串处理

1,字符串截取

RANDOM

#!/bin/bash

a=abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789

x=$[RANDOM%62] //获取随机数0~61

echo ${a:x:1} //随机截取1位字符

思考 ,如何得到8位随机字符?

#!/bin/bash

a=abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789

for i in {1..8}

do

x=$[RANDOM%62]

pass=${a:x:1}

pass2=$pass2$pass

done

echo $pass2

2,字穿替换

echo ${变量名/old/new} //替换一个

echo ${变量名//old/new} //替换所有

a=aabbcc

echo ${a/aa/88}

echo ${a/a/8}

echo ${a/b/8}

echo ${a/bb/88}

echo ${a//b/8}

3,字串删除

echo ${a#} 掐头

echo ${a%} 去尾

a=`head -1 /etc/passwd` //定义变量

[root@desktop0 opt]# echo ${a#root} //从左往右删除,删除第一个root

[root@desktop0 opt]# echo ${a##\*root} //从左往右删除,删除到最后一个root

使用字串删除功能,编写批量修改扩展名的脚本

#!/bin/bash

for i in `ls \*.txt` //找到要改的所有文件,有多少个文件,就循环执行下面的命令多少次

do

x=${i%.\*} //去掉扩展名,得到纯文件名

mv $i $x.doc //将原本的文件修改成纯文件名.doc

done

升级版

#!/bin/bash

for i in `ls \*.$1`

do

x=${i%.\*}

mv $i $x.$2

done

定义变量初值(备用值)

echo ${变量名:-初值}

echo ${p:-123} //定义备用值,如果变量p有值则不使用备用值