回顾：

1）数值运算（整数、小数）

expr 整数值1 运算符 整数值2

$[整数值1 运算符 整数值2]

let 变量名++、let 变量名--、let 变量名+=2

echo "数值1 运算符 数值2" | bc

2）条件测试操作

判断依据：$? ==》0 成功，非0 失败

测试操作的形式：任何一条命令、test或[ ]

test操作常用选项：

文件状态测试：-e、-d、-f、-x、-w、-r

整数值比较：-gt、-ge、-eq、-ne、-lt、-le

!!!! greater、lesser、equal、not、than

字符串比较：== 或 = 、!=

逻辑分隔：&&、||

多个条件的组合：

[[ 条件1 && 条件2 ]] || [[ 条件3 ]]

[ 条件1 -a 条件2 ] || [[ 条件3 ]]

3）if判断结构

简单的判断结构（只需要做一种处理） ——

如果 XXXX 那么 YYYY

方式1：

XXXX操作 && YYYY操作

方式2：

if XXXX操作

then

YYYY操作

fi

常见的判断结构（需要根据条件分别做两种不同处理） ——

如果 XXXX 那么 YYYY 否则 ZZZZ

方式1：

XXXX操作 && YYYY操作 || ZZZZ操作

方式2：

if XXXX操作

then

YYYY操作

else

ZZZZ操作

fi

复杂的判断结构（需要根据多个条件分别做多种不同处理） ——

如果 XX1 那么 YYYY 否则再如果 XX2 再那么 ZZZZ 否则 AAAA

if XX1操作

then

YYYY操作

elif XX2操作

then

ZZZZ操作

else

AAAA操作

fi

分号的作用：分隔前后两条命令

任务需求：编写一个脚本chweb.sh，检查目标网址的Web服务可用性

1）服务器地址从命令行提供（$1）

2）期望的网页/.tz.html，内容为"TEDU.CN"

3）如果检测成功，显示"passed"

4）如果检测失败，显示"failed"

#!/bin/bash

A=$(wget http://$1/.tz.html -O - 2>/dev/null)

if [ "$A" == "TEDU.CN" ]

then

echo "passed"

else

echo "failed"

fi

针对业务系统（比如多台Web服务器）的巡检：

提前准备一个探针网页

比如，检测 http://server0.example.com 是否可用

提前在网页目录下准备一个网页 .tz.html

如果能下载 http://server0.example.com/.tz.html 网页，而且内容是对的

Shell输出颜色控制：

文字颜色30-37：

echo -e "\033[30m 黑色字 \033[0m"

echo -e "\033[31m 红色字 \033[0m"

echo -e "\033[32m 绿色字 \033[0m"

echo -e "\033[33m 黄色字 \033[0m"

echo -e "\033[34m 蓝色字 \033[0m"

echo -e "\033[35m 紫色字 \033[0m"

echo -e "\033[36m 天蓝字 \033[0m"

echo -e "\033[37m 白色字 \033[0m"

背景颜色40-47

echo -e "\033[40;37m 黑底白字 \033[0m"

echo -e "\033[41;37m 红底白字 \033[0m"

echo -e "\033[42;37m 绿底白字 \033[0m"

echo -e "\033[43;37m 黄底白字 \033[0m"

echo -e "\033[44;37m 蓝底白字 \033[0m"

echo -e "\033[45;37m 紫底白字 \033[0m"

echo -e "\033[46;37m 天蓝底白字 \033[0m"

echo -e "\033[47;30m 白底黑字 \033[0m"

末尾控制选项：

\33[0m 关闭所有属性

\33[1m 设置高亮度

\33[4m 下划线

\33[5m 闪烁

\33[7m 反显

\33[8m 消隐

\33[30m — \33[37m 设置前景色

\33[40m — \33[47m 设置背景色

\33[nA 光标上移n行

\33[nB 光标下移n行

\33[nC 光标右移n行

\33[nD 光标左移n行

\33[y;xH设置光标位置

\33[2J 清屏

\33[K 清除从光标到行尾的内容

\33[s 保存光标位置

\33[u 恢复光标位置

\33[?25l 隐藏光标

\33[?25h 显示光标

for列表式循环

==》 针对一个列表中的多个值，重复执行类似的处理操作

类似场景：

去银行办业务、火车站买票、义务献血车

for 第一个献血人 in 张三 李四 王五 .. ..

do

为 $第一个献血人 执行 采血操作

done

任务需求：

1）编写一个脚本chkhosts.sh，用于机房巡检

2）巡检项为检查XX机房内的一批服务器IP地址是否能ping通

比如：172.25.0.10、172.25.0.11、172.25.0.12

for IP地址 in IP1 IP2 .. ..

do

ping命令检查并输出结果

done

#!/bin/bash

for ip in 10 11 12

do

ping -c3 -i0.2 -W1 172.25.0.$ip &> /dev/null

if [ $? -eq 0 ]

then

echo "Host $ip is up"

else

echo "Host $ip is down"

fi

done

如何为for循环提供值的队列 ——

1）直接给出，以空格分隔

2）提取命令行的显示结果 $(命令行)

用户提供的参数指定的一个文件 $(cat $1)

3）提取通配符匹配的结果 /home/\*

4）通过命令获取一组整数 {1..20} 或者 $(seq 20)

1 2 3 4 。。。 20

任务需求：使用for循环

1）统计 /home/ 目录下每个用户家目录占用的空间大小

2）分别备份每个用户的家目录，存储到 /opt/目录名.tar.gz

#!/bin/bash

for mulu in /home/\*

do

du -sh $mulu

tar -zcPf /opt/$(basename $mulu).tar.gz $mulu

done

!!!! 补充：获取一个路径的基本名称

# basename 文档路径

任务需求：编写模拟双色球摇号脚本 yaohao.sh

1）依次显示6个随机的红球号码

2）最后显示1个随机的蓝球/特别号码

!!!! 随机整数 $[RANDOM%33+1]

!!!! 次数控制 for i in {1..6} .. ..

!!!! 等待时间控制 sleep 3

#!/bin/bash

#1）依次显示6个随机的红球号码

echo -n "红色号码："

for i in {1..6}

do

sleep 3

echo -n "$[RANDOM%33+1] "

done

echo

#2）最后显示1个随机的蓝球/特别号码

echo "特别号码: $[RANDOM%33+1]"

现有的问题：有重复的号码

while条件式循环

当 XXXX 的时候，反复执行YYYY操作

while XXXX条件测试操作

do

YYYY操作

done

#!/bin/bash

while [ 1 -gt 0 ]

do

echo Hello

done

永久成立的条件：

[ 1 -gt 0 ]

或

true

或

:

比如：使用while语句模拟摇号

依次显示6个红色球的号码

i=1 //记录第几次摇号

while [ $i -le 6 ]

do

sleep 3

echo -n "$[RANDOM%33+1] "

let i++

done

任务需求：编写一个添加用户的脚本 uad.sh

1）添加10个用户账号，用户的名称分别是 tuser1、tuser2、……

2）这10个用户账号的密码都设置为 1234567

i=1

while 已经添加的用户数不超过10个时

do

添加用户 tuser$i 并设置密码

i的值增加1

done

#!/bin/bash

i=1

while [ $i -le 10 ]

do

useradd tuser$i

echo 1234567 | passwd --stdin tuser$i &>/dev/null

let i++

done

任务需求：猜商品价格的游戏脚本 price.sh

1）随机生成一个1～500商品价格，并且提示用户

2）用户有机会反复猜测价格

3）如果用户猜的价格比商品价格低，显示"太低了"，继续让用户猜

4）如果用户猜的价格比商品价格高，显示"太高了"，继续让用户猜

5）当用户猜对价格的时候，显示"恭喜你答对了"，并提示用户一共猜了多少次，并退出脚本

!!!! 商品价格 jiage1=$[RANDOM%500+1]

!!!! 使用 while read -p "请给出你猜测的价格：" jiage2

!!!! 记录猜测次数 num，每猜完一次 let num++

!!!! 三分支if结构

猜高了，。。。

猜低了，。。。

猜对了（退出条件） $jiage1 == $jaige2

#!/bin/bash

jiage1=$[RANDOM%500+1]

num=0

while read -p "请给出你猜测的价格(1~500)：" jiage2

do

let num++

if [ "$jiage2" -gt "$jiage1" ] ; then

echo "高了"

elif [ "$jiage2" -lt "$jiage1" ] ; then

echo "低了"

else

echo "恭喜你猜对了，商品价格是 $jiage2"

echo "你一共猜了 $num 次."

exit 0

fi

done

常见问题：

当脚本/root/a.sh中有 exit 0

通过 . a.sh 执行脚本，结果退出了当前的命令行中断

原因：

/root/a.sh执行时，在单独的Bash环境运行

. a.sh执行时，在当前Bash环境运行

脚本的中断和退出 ——

break ==》彩票中大奖了，辞职（放弃所在的整个循环）

continue ==》老婆生孩子，临时请假（跳出本次循环，继续下一次循环）

exit ==》退出整个脚本

while ture

do

命令1（摇号）

命令2（检查是否重复）

continue

命令3

break

命令4

done

命令5（退出脚本前显示XX信息）

exit 1

命令6

任务需求：解决双色球摇号脚本重号的问题

实现思路：

1）存储已经摇出的号码

每次摇出的号码 RED=$[RANDOM%33+1]

已经摇出的全部号码 var1="$var1 $RED"

2）对比新摇出的号码是否已经出现过

echo $var1 | grep -qw $RED

3）如果重复，跳出本次循环，重新摇号

if 第2）步成立，应该 continue

#!/bin/bash

#1）依次显示6个随机的红球号码

echo -n "红色号码："

i=1

while [ $i -le 6 ]

do

sleep 3

# 摇出新的号码

RED=$[RANDOM%33+1]

# 检查号码是否重复，如果重复，重新摇号

echo "$var1" | grep -qw "$RED" && continue

# 显示出不重复的号码

echo -n "$RED "

# 添加不重复的号码到 var1

var1="$var1 $RED"

let i++

done

echo

#2）最后显示1个随机的蓝球/特别号码

echo "特别号码: $[RANDOM%33+1]"

Shell函数，相当于脚本内的脚本/内部命令

—— 为需要频繁调用的一批代码定义一个简短的名称（函数）

任务需求：

1）编写一个脚本 color.sh

2）在脚本内定义一个函数 cecho

3）在脚本内执行 cecho 32m "passwd" 可以显示绿色字串

3）在脚本内执行 cecho 31m "failed" 可以显示绿色字串

#!/bin/bash

cecho() {

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

}

cecho 32m "passed"

cecho 31m "failed"

case分支，情况

针对 XX 参数等于不同的结果时候，分别做不同处理

case "$交通信号灯颜色" in

红色)

停

;;

绿色)

行

;;

黄色)

慢行

;;

\*)

听交警指挥

esac

任务需求：

1）改写chuquan.sh脚本，随机生成1～3之间的随机数，根据结果分别显示 石头、剪刀、布

2）改写 foo.sh 脚本，给 redhat显示fedora，给fedora显示redhat，如果都不是显示用法

3）编写一个 /sbin/webctl脚本，能支持参数 start|stop|status|restart，如果给的参数不是这些，提示正确用法

!!!! 提示

start)

systemctl start httpd

;;

restart)

$0 stop

$0 start

;;

#!/bin/bash

quantou=$[RANDOM%3+1]

case "$quantou" in

1)

echo "石头"

;;

2)

echo "剪刀"

;;

3)

echo "布"

;;

\*)

echo "犯规"

esac

#!/bin/bash

case "$1" in

redhat)

echo fedora

;;

fedora)

echo redhat

;;

\*)

echo "Usage:$0 redhat|fedora"

esac