回顾：

1）Shell概述（种类、历史命令、输入输出、重定向、管道）

更改用户历史命令的数量：

# vim /etc/profile

.. ..

HISTSIZE=1000

export 变量名=值

重定向操作：

< 、>、2>、&>、>>

屏蔽命令行输出/结果：

&> /dev/null

管道操作：

命令1 | 命令2 | 命令3

请思考：统计当前系统中使用/bin/bash的用户数量？

# grep '/bin/bash' /etc/passwd | wc -l

或者

# grep -c '/bin/bash' /etc/passwd

为什么要用脚本？

—— 偷懒（更简单、速度更快）

评估一条命令的执行时间：

# time 命令行

2）编写和执行Shell脚本

编写一个新文件

==》 整理脚本语句

方式1：按顺序编写任务实现命令

方式2：使用变量、循环/判断等控制接口来简化脚本

==》 添加x权限

3）Shell变量

自定义变量：

变量名=变量值

$变量名

echo $变量名

环境变量（env）：

PATH、PWD、USER、UID、HOME、SHELL、PS1、……

位置变量/参数：

$1、$2、……、${10}、……

预定义变量：

$0、$#、$\*、$?

任务需求：

1. 识别当前主机中是否存在用户 ldapuser0

# id ldapuser0

# echo $?

2. 识别当前系统是否是64位的

# lscpu | grep x86\_64

# echo $?

export 变量1名=值 变量2名=值 ......

export 变量1名 变量2名 .. ..

read -p '提示信息' 变量1名 变量2名

双引号、单引号、反撇号或$()

脚本创建思路：

==》 整理脚本语句

按顺序编写任务实现命令

==》 添加x权限

任务需求：

在server0上编写脚本 rhce\_demo.sh，执行脚本后能完成以下任务

1. 自动部署好 Web 服务器

网页来自 http://classroom/pub/materials/station.html

2. 自动调整好防火墙策略

禁止网段172.34.0.0/24访问

允许其他任何客户机访问

实现端口转发 5423 ==> 80

验证脚本运行结果：

浏览器访问 http://server0.example.com:5423/

#!/bin/bash

#1. 自动部署好 Web 服务器

# 网页来自 http://classroom/pub/materials/station.html

yum -y install httpd

wget http://classroom/pub/materials/station.html -O /var/www/html/index.html

systemctl restart httpd

systemctl enable httpd

#2. 自动调整好防火墙策略

# 禁止网段172.34.0.0/24访问

# 允许其他任何客户机访问

# 实现端口转发 5423 ==> 80

systemctl restart firewalld

systemctl enable firewalld

firewall-cmd --set-default-zone=trusted

firewall-cmd --permanent --add-source=172.34.0.0/24 --zone=block

firewall-cmd --permanent --zone=trusted --add-forward-port=port=5423:proto=tcp:toport=80

firewall-cmd --reload

今日目标：

1）学会使用各种测试命令识别系统指标

2）通过if语句使脚本具备基本的判断能力

3）数值运算

Bash环境默认不支持小数运算

运算操作 ——

expr 数学表达式 ==》计算表达式并且显示结果

$[数学表达式] ==》计算表达式并调用结果

取余数 ——

expr 整数1 % 整数2

特殊变量RANDOM，随机获得一个整数值 0~32767

任务需求：

1. 设置一个变量 X=12

2. 使用 expr 计算变量X的值 与 整数 5 的加减乘除余的结果

3. 使用 $[] 计算变量X的值 与 整数 5 的加减乘除余的结果

4. 利用随机整数变量RANDOM，通过运算获取一个1～33之间的整数

# echo $[RANDOM%33+1]

福利彩票摇号

CISHU=1

修改变量的值：

let 变量名++

小数运算

echo "表达式" | bc

任务需求：

1. 定义两个变量 X=1234，Y=56.78

# X=1234

# Y=56.78

2. 使用let命令使变量X的值减小100，

# let X-=100

3. 使用let命令使变量X的值增加1

# let X++

4. 使用bc计算器计算出变量X减去Y的结果

# echo "$X-$Y" | bc

检查用户ldapuser0是否存在？

# id ldapuser0 &> /dev/null

# ehco $?

&& 而且

|| 或者

代码简化 ==》

# id ldapuser0 &> /dev/null && ehco yes || echo no

任务需求：

1）识别目标主机 172.25.0.254 的连通性，能通 yes，否则 no

# ping -c 4 172.25.0.254 &> /dev/null && echo yes || echo no

2）识别目标网站 http://server0.example.com/ 可用性，能访问 yes，否则 no

# wget http://server0.example.com/ &> /dev/null && echo yes || echo no

逻辑测试 -- 逻辑与（而且）：

命令1 && 命令2

比方：要吃就吃好

有酒 && 有菜 && 有主食

逻辑测试 -- 逻辑或（或者）：

命令1 || 命令2

比方：只要吃饱

米饭 || 馒头

检查XX指标的操作 && 命令A ||命令B

==》如果检查操作成功，那么执行命令A

==》否则，执行命令B

任务需求：

检查 httpd 进程是否在运行，是则 running，否则 halted

# pgrep -x httpd &> /dev/null && echo running || echo halted

如何进行条件测试 —— 识别系统环境中的一些参数

方式1：任何一条命令行

方式2：

test -选项 参数

test 参数1 -选项 参数2

[ -选项 参数 ]

[ 参数1 -选项 参数2 ]

任务需求：

1）检查当前用户是否是root，是则显示"大王万岁"，否则显示"小朋友你好"

# [ $USER == "root" ] && echo "大王万岁" || echo "小朋友你好"

2）检查当前登录的用户数量，如果不超过5个，那么显示OK

# [ $(who | wc -l) -le 5 ] && echo OK

3）检查当前运行的进程数量，如果超过100，发一封邮件给root

# [ $(pgrep . |wc -l) -gt 100 ] && echo "进程数太多" | mail -s "服务监控警告" root

4）检查 /media/cdrom 目录是否存在，报告结果

# [ -d /media/cdrom ] && echo yes || echo no

检查XX指标的操作 && 命令A || 命令B

面临的问题：

当需要执行的命令不止一条命令时，命令A、AA、AAA、AAAA

解决办法：

检查XX指标的操作 && 命令A && 命令AA && 命令AAA && 命令AAAA || 命令B

推荐使用 if 判断结构

如果 车子、房子、票子

那么

下手

结束

if单分支结构，只需要一种处理

if 条件检测----车子、房子、票子

then

处理XX事情的命令行操作....

fi

任务需求：使用if判断结构编写脚本，完成下列操作

1）检查当前运行的进程数量，如果超过100，发一封邮件给root

#!/bin/bash

if [ $(pgrep . |wc -l) -gt 100 ]

then

echo "进程数太多" | mail -s "服务监控警告" root

fi

2）检查 /media/cdrom 目录是否存在，若不存在，则创建此目录

#!/bin/bash

if [ ! -d /media/cdrom ]

then

mkdir -p /media/cdrom

fi

if双分支结构，只需要两种不同处理

if 条件检测

then

做第一种处理操作....

else

做第二种处理

fi

任务需求：编写一个脚本，用来检查目标主机（由$1提供）的连通性

1）如果目标主机能ping通，显示 Host XXX is up

2）如果目标主机不能ping通，显示 Host XXX is down

3）如果执行脚本时未指定目标地址，那么显示正确的用法，退出脚本

#!/bin/bash

if [ $# -eq 0 ]

then

echo "用法: $0 目标主机地址"

exit 1

fi

if ping -c3 -i0.2 -W1 $1 &> /dev/null

then

echo "Host $1 is up"

else

echo "Host $1 is down"

fi

if多分支结构，只需要三种或三种不同处理

if 条件检测1

then

做第一种处理操作....

elif 条件检测2

then

做第二种处理

else

做第三种处理

fi

区分用户的论坛京豆等级（钻石plus、钻石、金牌、……）

成绩分档（超过90 优秀，70～90 良好，70以下 不合格）

任务需求：石头剪刀布

1）编写一个出拳脚本 chuquan.sh

2）执行此脚本以后，随机显示 石头、剪刀、布 当中某一个

#!/bin/bash

N=$[RANDOM%3+1]

if [ $N -eq 1 ]

then

echo "石头"

elif [ $N -eq 2 ]

then

echo "剪刀"

else

echo "布"

fi

任务需求：石头剪刀布

1）编写一个猜拳脚本 chuquan.sh

2）对比用户、计算机的出拳结果，并显示输赢情况

实现思路：

#1）取得用户出拳结果（$1）

#2）取得计算机出拳结果

#3）对比两个结果，并且显示输、赢、平情况

如何判断多个条件？

if [[ 条件a && 条件b ]] || [[ 条件测试2 ]] || [[ 条件测试2 ]]

if [[ 用户==石头 && 计算机==剪刀 ]] || [[ 用户==剪刀 && 计算机==布 ]] || [[ 用户==布 && 计算机==石头 ]]

#!/bin/bash

#1）取得用户出拳结果（$1）

yonghu="$1"

#2）取得计算机出拳结果

N=$[RANDOM%3+1]

if [ $N -eq 1 ]

then

jisuanji="石头"

elif [ $N -eq 2 ]

then

jisuanji="剪刀"

else

jisuanji="布"

fi

#3）对比两个结果，并且显示输赢情况

if [ $yonghu == $jisuanji ]

then

echo 平局

elif [[ $yonghu == "石头" && $jisuanji == "剪刀" ]] || [[ $yonghu == "剪刀" && $jisuanji == "布" ]] || [[ $yonghu == "布" && $jisuanji == "石头" ]]

then

echo 你赢了

else

echo 你输了

fi

总结：

1. 数值运算（加减乘除余的$[]、let控制变量值）

2. 条件测试（字符串、整数值、文件状态）

3.if选择结构（单分支、双分支 > 多分支）

参考案例里"论坛积分分档"