shell脚本基础

光驱永久挂载，默认mount命令临时有效，系统重启将不会再挂载

vim /etc/fstab 开机自动挂载文件，添加下面内容

/dev/cdrom /media/cdrom iso9660 ro 0 0 光驱自动挂载信息，iso9660光驱类型固定，ro只读

yum grouplist 列出安装软件包组，不是代表一个软件而是一堆软件构成某个功能

yum groupinstall -y "Chinese Support" 安装中文支持，支持中文环境支持中文输入法

如果希望将系统语言改为中文环境，可以在登录界面输入用户名后，回车，然后点击下面的语言选择中文后再输入密码确认即可，ctrl+空格切换输入法

脚本建议单独存放在一个文件夹中，便于管理和维护

vim welcome.sh 创建脚本文件，脚本文件的后缀一般建议使用.sh，脚本名尽量和脚本功能相接近

#!/bin/bash 声明shell

#这是一个报告系统状态的脚本，可以提示IP地址/可用内存/可用SWAP等 定义注释

IP=`ifconfig | head -2 | tail -1 | cut -d: -f2 | awk '{print $1}'` 定义IP变量

MEM=`free -m | grep Mem | awk '{print $4}'` 定义MEM变量

SWAP=`free -m | grep Swap | awk '{print $4}'` 定义SWAP变量

WHO=`who | wc -l` 定义WHO变量

echo "当前主机的IP地址是:$IP" 输出并应用IP变量

echo "当前主机的可用内存:$MEM(MB)"

echo "当前主机的可用SWAP:$SWAP(MB)"

echo "当前主机的用户数是:$WHO(个)"

脚本的执行，需要添加x权限

chmod +x welcome.sh 增加执行权限

第一种执行方式可以使用相对或绝对路径执行

./welcome.sh

/root/sh/welcome.sh

第二种方式，可以定义PATH变量将脚本所在的文件夹添加至PATH变量中，可以像系统命令一样执行

PATH=/root/sh:$PATH

welcome.sh 直接就能执行

echo /root/sh/welcome.sh >> ~/.bashrc 写入到用户打开终端执行的文件中，每次打开终端都能提示

有了条件判断可以使脚本智能化，之前脚本只是简单的命令堆砌，只是按照顺序依次执行所有的命令，条件判断可以使用test命令或者[]

[ -e /media/cdrom ] 判断/media/cdrom是否存在

echo $? 返回值是0代表存在，返回值非0代表不存在

[ -d /media/cdrom ] && echo Yes 前面条件判断/media/cdrom是不是文件夹，如果是文件夹将会执行后面的命令echo Yes，后面命令任意写

[ `who | wc -l` -gt 5 ] && echo "注意，当期系统登录的用户数量超过5个" 判断当前系统中登录用户数量是否超过了5个，如果超过了5个将会发送警告

[ $PWD = /etc ] && echo Yes 判断用户当前所在的目录是否为/etc，如果是/etc直接在屏幕输出Yes

逻辑测试的与或非

&& 代表多个条件做逻辑与，每个条件都必须成立才是正确的

|| 代表多个条件做逻辑或，有条件成立即可

! 代表条件做逻辑否，不成立才可以

[ -e /media/cdrom ] && [ -d /media/cdrom ] && echo Yes 判断/media/cdrom是否存在，是否是文件夹，两个都成立才可以

[ -e /media/cdrom ] && [ -f /media/cdrom ] && echo Yes 这个不会输出Yes

[ -e /media/cdrom ] || [ -f /media/cdrom ] && echo Yes 会输出Yes，逻辑或，只要一个成立即可

[ -e /media/cdrom ] || [ -d /media/cdrom ] && echo Yes 会输出Yes

vim httpd.sh 创建一个脚本

#!/bin/bash

#这是一个判断httpd服务启动的脚本，如果服务启动直接提示用户，如果服务没有启动尝试启动该服务

service httpd status &> /dev/null

if [ $? -eq 0 ]

then

echo "Httpd服务正在运行中..."

else

service httpd start &> /dev/null

echo "正在尝试启动httpd服务..."

service httpd status &> /dev/null

if [ $? -eq 0 ]

then

echo "Httpd服务成功启动..."

else

echo "Httpd服务启动失败..."

fi

fi

猜数字小游戏

vim caishuzi.sh 目前属于半成品，一次必须猜对，每次随机的数字都是不同的，只能玩一次，可以借助于后面的循环让他不同的做判断，后面再完善

#!/bin/bash

#这是一个猜数字的脚本，提示用户输入数字，根据用户输入的数字来判断大小进行输出

NM=`echo $RANDOM | cut -c-2`

read -p "请输入一个数字（1--100）:" nm

if [ $nm -lt $NM ]

then

echo "数字太小了..."

elif [ $nm -gt $NM ]

then

echo "数字太大了..."

else

echo "恭喜你，答对了..."

fi

for循环语句，按照取值挨个执行，执行完毕后结束循环

计算从1加到100的数值

vim num.sh

#!/bin/bash

#计算1加到100的数值总和

SUM=0

for i in {1..100}

do

SUM=`expr $SUM + $i`

done

echo $SUM

vim ls.sh

#!/bin/bash

#这是一个显示目录中文件信息的脚本

a=1

read -p "请输入需要查询的目录的绝对路径:" PA

for i in `ls $PA`

do

echo "第$a个文件是:$i"

a=`expr $a + 1`

done

for循环的作业：

删除用户的脚本，提示需要删除用户的前缀，然后会删除以输入的前缀开头的所有用户，不要删除管理员用户和程序用户（UID小于500），删除后会在屏幕打印删除了哪些账户

/etc/passwd文件 ‘cut’ ‘grep’ ‘userdel -r’ ‘id’ ‘echo’ ‘for’ ‘if’