**课堂练习：**

例题1：根据用户需求生成n位的随机密码。

pass\_all='abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ123456789'

#计算密码库的总个数

len\_pass=${#pass\_all}

#根据用户参数执行n次循环，默认值为8

while [ ${i:-0} -lt ${1:-8} ]

do

#获取随机数

num=$[RANDOM%len\_pass]

#将每一次循环的随机密码进行追加

pass=${pass\_all:num:1}$pass

let i++

done

echo $pass

例题2：循环随机获取姓名，如果获取到本组同学的姓名，输出xx好靓仔!

xing='高王陈'

mid='锦永慧'

ming='全伟熙'

#获取姓名库的总字数

len\_xing=${#xing}

len\_mid=${#mid}

len\_ming=${#ming}

while :

do

ran1=$[RANDOM%len\_xing]

ran2=$[RANDOM%len\_mid]

ran3=$[RANDOM%len\_ming]

xm=${xing:ran1:1}${mid:ran2:1}${ming:ran3:1}

echo -e "$xm\n"

sleep 1

case $xm in

'高锦全'|'王永伟'|'陈慧熙')

echo "$xm最靓仔！" && break

esac

done

例题3：批量修改文件夹中文件的拓展名，允许带参数$1，默认改为.txt

cd /root/test

for name in `ls \*.${1:-doc}`

do

mv $name ${name%%.\*}.${2:-txt}

done

案例4：

过滤下载文件中包含 the 关键字

grep the 1.txt

过滤下载文件中丌包含 the 关键字

grep -v the 1.txt

过滤下载文件中丌论大小写 the 关键字

grep -i the 1.txt

过滤 test 或 taste 这两个单字

grep -E "test|taste" 1.txt

grep "t[ae]ste\{0,1\}" 1.txt

过滤有 oo 的字节

grep oo 1.txt

过滤不想要 oo 前面有 g 的

grep "[^g]oo" 1.txt

过滤 oo 前面不想有小写字节

grep "[^a-z]oo" 1.txt

过滤有数字的那一行

grep "[0-9]" 1.txt

过滤以 the 开头的

grep "^the" 1.txt

过滤以小写字母开头的

grep "^[a-z]" 1.txt

过滤开头不是英文字母

grep -v "^[a-Z]" 1.txt

grep "^[^a-Z]" 1.txt

过滤行尾结束为小数点.那一行

grep "\.$" 1.txt

过滤空白行

grep ^$ 1.txt

过滤出 g??d 的字串

grep "g..d" 1.txt

过滤至少两个 o 以上的字串

grep "ooo\*" 1.txt

grep "oo\{2,\}" 1.txt

过滤 g 开头和 g 结尾但是两个 g 之间仅存在至少一个 o

grep "goo\*g" 1.txt

过滤任意数字的行

grep "[0-9]" 1.txt

过滤两个 o 的字串

grep oo 1.txt

过滤 g 后面接 2 到 5 个 o,然后在接一个 g 的字串

grep "go\{2,5\}g" 1.txt

过滤 g 后面接 2 个以上 o 的

grep "go\{2,\}"

**Homework：**

**1.任意虚拟机编写脚本:**

自动安装软件包 vsftpd;

自动创建一个系统账户 tomcat;

在tomcat 家目录下创建一个测试文件/home/tomcat/test.txt(内容为 hello);

生成 1 个 8 位随机密码,将 tomcat 账户的密码修改为该随机密码;

将该随机密码通过邮件的形式通知 root(给 root 发邮件,内容是随机密码);

脚本自动启动 vsftpd,设置开机自启。

真实机测试,是否可以使用 tomcat 账户登陆虚拟机的 ftp,并成功读写数据。

yum -y install vsftpd

systemctl restart vsftpd

systemctl enable vsftpd

useradd tomcat

echo 'hello' > /home/tomcat/test.txt

#8位随机密码生成

pass\_all='abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ123456789'

len\_pass=${#pass\_all}

while [ ${i:-0} -le 7 ]

do

num=$[RANDOM%len\_pass]

tmp=${pass\_all:num:1}

pass=$pass$tmp

let i++

done

#将tomcat用户的密码设为随机密码

echo $pass | passwd --stdin tomcat

#邮件把密码发送给root

mail -s 'Tomcat passwd' root << EOF

Tomcat password is $pass

EOF

**2.编写脚本:**

循环生成 20 个 4 位随机字符串,以这 20 个随机字符串为文件名,创建 20 个文件,扩展名为 jpg。

arr\_all='abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ123456789'

len\_arr=${#arr\_all}

#创建20个文件并改名为随机字符串

for i in `seq 20`

do

#获取4位随机字符串

for j in `seq 4`

do

rng=$[RANDOM%len\_arr]

tmp=${arr\_all:rng:1}

arr=$arr$tmp

done

touch "/root/test/$arr.jpg"

#清空字符串

arr=''

done

**3.编写脚本:**

修改前面创建的 20 个随机文件名的文件,新的文件名为 1.jpg,2.jpg,3.jpg 以此类推(修改文件名)。

cd /root/test

num=1

for i in `ls \*`

do

mv $i $num.${i##\*.}

let num++

done

**4.编写脚本:**

将/etc/passwd 中所有的账户名称提取出来。

for i in `cat /etc/passwd`

do

echo ${i%%:\*}

done

1. **给任意一台虚拟机添加 2 块 10G 磁盘,编写脚本使用 fdisk 自动分区(通过 expect 自动交互完成分区)、格式化、挂载。**

#!/bin/bash

expect << EOF

spawn fdisk /dev/vdb

expect "命令" { send "n\n"}

expect "Select" { send "\n"}

expect "分区" { send "\n"}

expect "起始" { send "\n"}

expect "Last" { send "+1G\n"}

expect "命令" { send "w\n"}

expect

EOF

partprobe

lsblk

mkfs.ext4 /dev/vdb1

mkdir /mnt/b1

echo '/dev/vdb1 /mnt/b1 ext4 defaults 0 0' >> /etc/fstab

mount -a

df -h

**6.真实主机安装 expect 软件,编写脚本:**

使用 ftp 命令自动连接虚拟机的 FTP 服务器(题目 1 搭建的 ftp);

结合 expect 自动输入登陆的帐户名 tomcat 及对应的密码;

脚本自动下载 test.txt 文件到真机。

备注:ftp 是需要通过 yum 安装的一个基于命令行的 ftp 客户端软件

语法格式:# ftp 服务器IP

**7.在真机通过 grep 命令过滤下列数据:**

从/etc/hosts 中过滤以 127 或 172 开头的数据;

grep -E "^127|^172" /etc/hosts

从/etc/passwd 中过滤所有解释器为 bash 的数据

grep bash$ /etc/passwd

从/etc/passwd 中过滤所有三位以及以上的数字

grep [0-9][0-9][0-9] /etc/passwd

从/etc/chrony.conf 中过滤所有以 server 开头的内容

grep ^server /etc/chrony.conf

从/etc/ssh/sshd\_config 中过滤包含 Port 的内容

grep Port /etc/ssh/sshd\_config

从/etc/passwd 中过滤以 root 或者 adm 开头的内容

grep -E ^root|^adm /etc/passwd

从/var/log/httpd/access\_log 中过滤所有包含时间的内容

egrep "[0-9]{4}:[0-9]{2}:[0-9]{2}:[0-9]{2}" /var/log/httpd/access\_log

从/var/log/secure 中过滤包含 Failed 的内容

grep Failed /var/log/secure

通过管道将 uptime 命令结果输出给 grep 命令,使用 grep 过滤所有包含小数点.的内容

uptime | grep "\."

1. **判断系统中是否已经安装了 httpd,如果已经安装则提示:已经安装,否则直接安装该软件,并提示完成完成。**

if rpm -q httpd > /dev/null ;then

echo "已经安装"

else

yum -y install httpd && echo '完成完成。'

Fi

**9. 编写脚本:编写函数,函数的功能是给网卡配置 IP**

检测某个网卡是否有 IP(通过 read 读取网卡名称,如果用户未输入则默认为 eth0)

如果网卡没有 IP,调用前面的函数给网卡配置 IP

如果网卡已经有 IP,则提示该网卡已经有 IP 地址。

get\_ip(){

echo "

DEVICE=$1

IPADDR=$2

BOOTPROTO=static

NETMASK=255.255.255.0

ONBOOT=yes" > /etc/sysconfig/network-scripts/ifcfg-$1

}

read -p "请输入网卡名：" ifname

ifname=${ifname:-eth0}

if ifconfig $ifname | grep netmask > /dev/null ;then

#截取网卡的IP值

tmp\_ip=`ifconfig $ifname | grep netmask`

n\_ip1=${tmp\_ip#\*net}

n\_ip2=${n\_ip1%net\*}

#显示网卡的IP值

echo "网卡$ifname的IP地址为：$n\_ip2"

else

read -p "请输入IP：" ifip

get\_ip $ifname $ifip

systemctl restart network

ifconfig $ifname

fi