**案例1：编写一键部署软件脚本**

本案例要求编写脚本实现一键部署Nginx软件（Web服务器）：

-一键源码安装Nginx软件

-脚本自动安装相关软件的依赖包gcc,openssl-devel,pcre-devel

-脚本自动判断yum是否可用

#!/bin/bash

#Describe:编写一键部署软件脚本

#Author:JQ\_Gao

#判断yum是否可用

yum clean all

repo\_num=$(yum repolist | awk '/repolist/{print $2}')

if [ $repo\_num == 0 ];then

echo 'yum不可用'

exit

else

echo "repolist: $repo\_num"

fi

#安装Nginx源码包编译相关工具

yum -y install gcc openssl-devel pcre-devel 2>> /root/yum.log

#Nginx源码编译安装

tar -xf /root/nginx-1.12.2.tar.gz 2>>/root/yum.log

cd /root/nginx-1.12.2

./configure 2>>/root/yum.log

make 2>>/root/yum.log

make install 2>>/root/yum.log

#检查Nginx软件信息

/usr/local/nginx/sbin/nginx -V

**案例2：启动脚本**

本案例要求编写Nginx启动脚本，要求如下：

-脚本支持start、stop、restart、status

-脚本支持报错提示

-脚本具有判断是否已经开启或关闭的功能

#!/bin/bash

#Describe:Nginx启动脚本

#Author:JQ\_Gao

case $1 in

"start")

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

"stop")

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

"restart")

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

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

"status")

netstat -ntulp | grep nginx

[ $? -eq 0 ] && echo "nginx已启动" || echo "nginx未启动";;

\*)

echo "Input start|stop|restart|status."

esac

**案例3：编写监控脚本**

本案例要求编写脚本，实现计算机各个性能数据监控的功能，具体监控项目要求如下：

-CPU负载 uptime

-网卡流量 ifconfig eth0

-内存剩余容量 free

-磁盘剩余容量 df

-计算机账户数量 cat /etc/passwd | wc -l

-当前登录账户数量 who |wc -l

-计算机当前开启的进程数量 ps aux

-本机已安装的软件包数量 rpm -qa | wc -l

#!/bin/bash

#Describe:计算机各个性能数据监控脚本

#Author：JQ\_Gao

echo '【CPU负载】'

cpu\_rl=`uptime | awk '{print $NF}'`

echo "CPU最近15分钟的平均负载为：$cpu\_rl"

echo '【网卡流量】'

netcard=`ifconfig -a | awk '/flag/{print $1}'`

for netc in $netcard

do

echo $netc

net\_rx=`ifconfig ${netc%:\*} | awk '/RX packets/{print $5}'`

net\_tx=`ifconfig ${netc%:\*} | awk '/TX packets/{print $5}'`

echo "接收流量：$net\_rx"

echo "发送流量：$net\_tx"

done

mem\_free=`free | awk '/Mem/{print $4}'`

echo -e "【内存剩余容量】\n$mem\_free"

disk\_free=`df | awk '/\/$/{print $4}'`

echo -e "【磁盘剩余容量】\n$disk\_free"

num\_user=`cat /etc/passwd | wc -l`

echo -e "【计算机账户数量】\n$num\_user"

num\_login=`who | wc -l`

echo -e "【当前登陆账户数量】\n$[num\_login-1]"

num\_process=`ps aux | wc -l`

echo -e "【计算机当前开启的进程数量】\n$num\_process"

num\_software=`rpm -qa | wc -l`

echo -e "【本机已安装的软件包数量】\n$num\_software"

**案例4：编写安全检测脚本**

本案例要求编写脚本，防止远程ssh暴力破解密码，具体监控项目要求如下：

-检测ssh登录日志，如果远程登陆账号名错误3次，则屏蔽远程主机的IP

-检测ssh登录日志，如果远程登陆密码错误3次，则屏蔽远程主机的IP

#!/bin/bash

#Describe:安全检测脚本

#Author:JQ\_Gao

#提取密码错误3次以上的IP

ip\_errpass=`awk '/Failed password/{arr[$(NF-3)]++}

END{

for(i in arr)

{ip\_pass[m++]=i;

num\_pass[n++]=arr[i];

print ip\_pass[m-1],num\_pass[n-1]}

}' /var/log/secure | awk '$2>=3{print $1}'`

#提取用户名错误3次以上的IP

ip\_erruser=`awk '/Invalid user/{arr[$NF]++}

END{

for(i in arr)

{ip\_user[m++]=i;

num\_user[n++]=arr[i]

print ip\_user[m-1],num\_user[n-1]}

}' /var/log/secure | awk '$2>=3{print $1}'`

ip\_errall=`echo "$ip\_errpass $ip\_erruser"`

for ip in $ip\_errall

do

echo "$ip暴力破解3次，即将被屏蔽..."

firewall-cmd --permanent --zone=block --add-source=$ip

done

firewall-cmd --reload &>/dev/null

echo

firewall-cmd --zone=block --list-all

**案例5：编写进度显示脚本**

本案例要求编写脚本，实现带进程显示的复制脚本，具体要求如下：

-默认Linux的cp命令不具有进度显示

-我们需要自己编写脚本实现进度显示

-可以使用进度条的方式，或者显示百分比的方式

trap "exit" INT

#!/bin/bash

#Describe：编写进度显示脚本

#Author：JQ\_Gao

echo -n "拷贝中："

process(){

while :

do

echo -ne "\033[7m#\033[0m"

sleep 0.5

done

}

process &

cp -r $1 $2

kill $! &> /dev/null

echo -e "\n拷贝完成！"

ls $2