

## 09.Shell练习题目

1.按照时间生成文件 2018-05-22.log 将每天的磁盘使用状态写入到对应的文件

分析

- 1.时间打印 date +%F
- 2.磁盘状态 df -h

```
df -h >$(date +%F).log
[root@Shell timujia]# cat time_df.sh
#!/usr/bin/bash
Time=$(date +%F)
df -h> $Time.log
```

2.统计Nginx日志中每个IP的访问量有多少,日志如下:

```
192.168.56.1 - - [21/May/2018:20:44:06 -0400] "GET /index.html HTTP/1.0" 404 169 "-"
"ApacheBench/2.3" "-" /code/index.html
```

分析

- 1.筛选所有的IP地址
- 2.排序、去重、统计

```
awk '{print $1}' access.log|sort|uniq -c|sort -n|tail
awk '{print $1}' access.log|sort|uniq -c|sort -rn|head
```

3.写一个脚本计算一下Linux系统所有进程占用内容大小的和。

分析

- 1.如何获取内存的大小 top、ps
- 2.如何统计大小之和

```
[root@Shell timujia]# cat total.sh
#!/usr/bin/bash
Mem=$(ps aux|awk '{print $6}'|grep -v "RSS")
num=0
for i in $Mem
do
    num=$((i+$num))
done
num=$((num/1024))
```

```
echo ${num}M
```

4.找到/backup目录下所有后缀名为.txt的文件

- 1.批量修改txt为txt.bak
- 2.把所有的.bak文件打包压缩为123.tar.gz
- 3.批量还原文件的名字，及把增加的.bak再删除

分析

- 1.使用find命令进行查找
- 2.使用mv命令进行移动改名
- 3.使用tar命令进行打包

```
[root@Shell timujia]# cat for_name.sh
#!/usr/bin/bash

File_txt=$(find /backup/ -type f -iname "*.txt"|xargs >/backup/txt.tt)
for i in $(cat /backup/txt.tt)
do
    mv  $i $i.bak
done

cd /backup/
tar czf 123.tar.gz ./*.bak

File_bak=$(find /backup/ -type f -iname "*.bak" >/backup/tar.tt)
for j in $(cat /backup/tar.tt)
do
    mv $j ${j%.*}
done
```

5.写一个脚本，判断本机80端口(假如服务为httpd)是否开启,如果开启什么都不干,如果发现端口不存在，那么重启一下http服务,并发邮件通知自己,脚本写好后可以每分钟执行一次，也可以写一个死循环

分析

- 1.检测80端口是否正常 netstat -lntp|grep ":80"
- 2.如果不正常则重启Nginx
- 3.如果进程是启动的则重启，否则直接启动

```
[root@Shell timujia]# cat port.sh
#!/usr/bin/bash

Process=$(ps aux|grep [n]ginx|wc -l)
```

```
#Port_close=$(nmap -p80 127.0.0.1|grep closed)
Port_open=$(nmap -p80 127.0.0.1|grep open|wc -l)

if [ $Port_open -ne 1 ];then
    if [ $Process -ne 2 ];then
        /usr/sbin/nginx
    else
        /usr/sbin/nginx -s reload
    fi
fi
```

6.现在有一个lnmp环境, 经常出现502错误, 只要一重启php-fpm即可解决, 如果不重启则会持续非常长的一段时间, 所有有必要写一个监控脚本, 监控访问日志的状态码, 一旦发生502, 则自动重启一下php-fpm

分析

- 1.通过日志尾部最后300行, 统计502出现的次数
- 2.精准判断是否是502错误, 不是则不处理, 是则重启php-fpm

```
[root@Shell timujia]# cat http_status.sh
#!/usr/bin/bash

while true
do
    sleep 3
    Log=$(tail -n300 /var/log/nginx/access.log|grep "502"|wc -l)
    if [ $Log -ge 100 ];then
        systemctl restart php-fpm
        sleep 5
    fi
done
```

7.用 shell 打印下面这句话中字母数小于6个的单词 Bash also interprets a number of multi-user options

分析

- 1.使用循环遍历
- 2.怎么统计单词数值 wc -c

```
[root@Shell timujia]# cat for_danci.sh
#!/usr/bin/bash
for i in Bash also interprets a number of multi-user options
do
    #统计单词的数量
```

```

number=$(echo $i|wc -c)
#判断单词是否小于7
if [ $number -lt 7 ];then
#输出当前循环的单词是什么
    echo $i
fi
done

```

8.添加 user\_00->user\_09 10个用户, 并且给他们设置一个随机密码, 密码要求10位包含大小写字母以及数字, 注意需要把每个用户的密码记录到一个日志文件中

分析

- 1.怎么实现00-09思路 `echo user_{00..10} seq -w 0 10`
- 2.随机密码 `mkpasswd( yum install expect)`

```

cat user_passwd.sh
#!/usr/bin/bash
if [ $UID -ne 0 ];then
    echo "请使用root账户执行创建用户"
    exit 1
fi

for i in $(seq -w 00 10)
do
    username=user_${i}
    pass=$(mkpasswd -l 10 -C 3 -c 3 -d 2 -s 2)
    id $username &>/dev/null
    if [ $? -eq 1 ];then
        useradd $username && \
        echo "$pass"|passwd --stdin $username &>/dev/null
        id $username &>/dev/null
        if [ $? -eq 0 ];then
            echo "Useradd $username is OK"
        else
            echo "Useradd $username is Err"
        fi
    else
        echo "Useradd $username is already exists"
    fi
    echo -e "$username \t $pass" >> /tmp/user.txt
done

```

9.写个 shell 看看你的 linux 系统中是否有自定义用户(普通用户),若有一共有多少个?

分析

1.查看/etc/passwd文件

2.只有UID大于1000的都是普通

```
cat /etc/passwd|awk -F ":" '{print "$3}'
```

```
cat user.sh
#!/usr/bin/bash

user_id=$(cat /etc/passwd|awk -F ":" '{print $3}')
total=$(cat /etc/passwd|awk -F ":" '$3>1000'|wc -l)
for i in $user_id
do
    if [ $i -ge 1000 ];then
        user_uid_undef=$(cat /etc/passwd|awk -F ':' '{print $1":"$3}'|grep "$i")
        echo $user_uid_undef
    fi
done

echo "总共普通用户有: $total"
```

10.写一个 shell 脚本，检测所有磁盘分区使用率和 inode 使用率，并记录到以当天时间命名的日志文件中，当发现某个分区容量或 inode 使用量大于85%时，发邮件通知自己

分析

1.打印当前磁盘inode和使用率至文件中

2.取值判断是否使用率超过85%

3.发邮件通知自己(/tmp/error.txt)

```
cat disk_info.sh
#!/usr/bin/bash

Disk_Use=$(df -h|grep "/"|awk '{print $(NF-1)}'|awk -F '%' '{print $1}')
Disk_Inode=$(df -i|grep "/"|awk '{print $(NF-1)}'|awk -F '%' '{print $1}')
Time=$(date +%F)

#1. 将信息记录至$Time.Log
echo "磁盘已使用百分比: ${Disk_Use}%" > ${Time}_disk.log
echo "磁盘已使用inode百分比: ${Disk_Inode}%" >> ${Time}_disk.log

#2. 数值比较
if [ $Disk_Use -ge 20 ];then
    echo "磁盘已使用超过容量${Disk_Use}%" > /tmp/disk.txt
fi

if [ $Disk_Inode -ge 0 ];then
    echo "磁盘已使用超过容量${Disk_Inode}%" >> /tmp/disk.txt
```

11.写一个 shell 脚本来看看你使用最多的命令是那些，列出常用的命令top10

分析

1.拿到想要的值，排序、去重、统计

```
history |awk '{print $2}'|sort|uniq -c|sort -rn|head
hash |sort -rn|head #缓存的命令
cat ~/.bash_history |sort|uniq -c|sort -rn|head
```

12.写一个脚本判断你的 Linux 服务器里是否开启 web 服务, 如果开启了请判断跑的是什么服务, 是 httpd 还是 nginx 又或是其他?

分析

1.使用 netstat 查看是否存在80

2.筛选80端口对应的是 nginx 还是 httpd

```
cat nginx_http_pd.sh
#!/usr/bin/bash

Nginx_status=$(ss -lntup|grep ":80"|grep nginx|wc -l)
Httpd_status=$(ss -lntup|grep ":80"|grep httpd|wc -l)

if [[ $Nginx_status -eq 0 && $Httpd_status -eq 0 ]];then
    echo "没有启动80端口"
    exit 1
fi

if [ $Nginx_status -eq 1 ];then
    echo "启动的是Nginx服务"
elif [ $Httpd_status -eq 1 ];then
    echo "启动的是Httpd服务"
else
    echo "启动的是未知服务"
    exit 2
fi
```

13.当前 mysql 服务的 root 密码为 123456 写脚本检测 mysql 服务是否正常(比如，可以正常进入 mysql 执行 show processlist)，并检查一下当前的MySQL服务是主还是从，如果是从，请判断他的主从服务是否异常，如果是主，则不需要做什么

分析

- 1.使用非交互的方式登录mysql进行取值 `mysql -h -u -p -e`
- 2.检查主从, 如果是主则返回空, 如果是从则返回数值(`show slave status\G`)
- 3.检查从的IO线程和SQL线程是否正常。

```
//配置邮箱
cat /etc/mail.rc
set from=552408925@qq.com
set smtp=smtps://smtp.qq.com:465
set smtp-auth-user=552408925@qq.com
set smtp-auth-password=
set smtp-auth=login
set ssl-verify=ignore
set nss-config-dir=/etc/pki/nssdb/

cat mysql_status.sh
#!/usr/bin/bash
IP=192.168.70
User=remote
Pass=Bgx123.com
Sync_Date=$(date +%F)

for i in {160..161}
do
    IP_MySQL=$IP.$i
    echo "##正在操作$IP_MySQL"
    /usr/bin/mysql -h$IP_MySQL -u$User -p$Pass -e "show slave status\G" 2>/dev/null
    |egrep -i 'Slave_IO_Running:|Slave_SQL_Running:' &>/tmp/status.txt
    if [ ! -s /tmp/status.txt ];then
        echo "当前MySQL是主库"
        break
    fi
    IO=$(grep "IO" /tmp/status.txt |awk '{print $2}')
    SQL=$(grep "SQL" /tmp/status.txt |awk '{print $2}')

    if [[ $IO == "Yes" && $SQL == "Yes" ]];then
        echo "当前MySQL主从状态OK"
    else
        #mail -s "主从同步失败$IP_MySQL" </tmp/status.txt
        /usr/bin/mail -s "${Sync_Date}-${IP_MySQL}主从同步失败" 552408925@qq.com <"/tmp/status.txt"
        echo "当前Slave的IO状态是: $IO"
        echo "当前Slave的SQL状态是: $SQL"
    fi
done
```

14.写一个脚本, 计算100以内能被3整除的正整数之和

## 分析

- 1.先找出1-100能除以3等于0的数值
- 2.让数值进行相加

```
[root@Shell timujia]# cat zs.sh
#!/usr/bin/bash

sum=0
for i in {1..100}
do
    ys=$((i%3))
    if [ $ys -eq 0 ];then
        sum=$((i+$sum))
    fi
done
echo $sum
```

15.提示用户输入网卡名字,然后用脚本输出网卡的IP

## 分析

- 1.需要读入用户输入的网卡名称
- 2.判断输入的网卡名称是否正确
- 3.判断网卡是否有IP地址, 有则输出, 没有则提示

```
cat network.sh
#!/usr/bin/bash

read -p "请输入网卡名称: " net

ErrNet=$(ifconfig $net &>/dev/null;echo $?)
NetWork=$(ifconfig $net 2>/dev/null|awk 'NR==2 {print $2}')
if [ $ErrNet -eq 1 ];then
    echo "你输入的网卡名称错误"
    exit 1
else
    echo "你输入的网卡名称是:$net"
    echo "对应的IP地址是: $NetWork"
fi
```

16.写一个猜数字脚本, 当用户输入的数字和预设数字(随机生成一个0-100的数字)一样时, 直接退出, 否则让用户一直输入, 并且提示用户的数字比预设数字大或者小  
猜苹果是多少钱一斤



## 分析

- 1.随机数字如何生成
- 2.死循环，直到猜对才退出
- 3.判断大小

```
cat caipingguo.sh
#!/usr/bin/bash
while true
do
    n=$(echo $RANDOM|cut -c 1)
    read -p "猜猜我的苹果多少钱一斤：" num

    if [[ ! $num =~ ^[0-9]+$ ]];then
        echo "你tmd拿的是人民币吗！滚蛋！"
        exit 1
    fi

    if [ $num -lt $n ];then
        echo "苹果卖太便宜了，不买"
    elif [ $num -gt $n ];then
        echo "苹果卖的太贵了，不买"
    else
        echo "苹果的价位合适，买了"
        exit 2
    fi
done
```

17.写一个脚本判断输入的用户是否登入，如果未登陆提示没有登陆，如果登陆，显示登陆的终端以及通过那个IP登陆过来的

## 分析

- 1.使用w获取当前所有登陆系统的用户
- 2.判断

```
cat who.sh
#!/usr/bin/bash

read -p "请输入你想查看的用户：" user
login_total=$(w|grep "$user"|awk '{print $1,$2,$3}')
login_in=$(w|grep "$user" &>/dev/null;echo $?)

if [ $login_in -eq 0 ];then
    echo $login_total|xargs -n3
```

```
else
    echo "用户没有登陆"
fi
```

18.写一个 shell 先判断是否安装 http 和 mysql 没有安装进行安装，安装了检查是否启动服务，若没有启动则需要启动服务。

分析

- 1.使用rpm命令判断是否安装对应的软件
- 2.检查是否启动(ssystemctl|ps)

```
[root@Shell timujia]# cat http_mysql.sh
#!/usr/bin/bash

#setup1.判断是否安装了httpd或mysqld
Httpd=$(rpm -qa httpd|wc -l)
MySQL=$(rpm -qa mysql-community-server|wc -l)

if [ $Httpd -eq 0 ];then
    echo "开始安装Httpd服务"
    yum install httpd -y
else
    echo "Httpd服务已经安装成功"
    #setup2.是否已启动服务
    MySQL_Status=$(ps aux|grep [h]ttpd|wc -l)
    if [ $MySQL_Status -eq 0 ];then
        systemctl start httpd
        if [ $? -eq 0 ];then
            echo "Httpd启动成功"
        else
            echo "Httpd启动失败"
        fi
    fi
fi

if [ $MySQL -eq 0 ];then
    echo "开始安装MySQL服务"
    yum install mysql-community-server -y
else
    echo "MySQL服务已经安装成功"
    #setup2.是否已启动服务
    MySQL_Status=$(ps aux|grep [m]ysqld|wc -l)
    if [ $MySQL_Status -eq 0 ];then
        systemctl start mysqld
        if [ $? -eq 0 ];then
            echo "MySQL启动成功"
```

```

        else
            echo "MySQL启动失败"
        fi
    fi
fi

```

19.写一个 shell 脚本，通过 curl -I 返回的状态码来判断所访问的网站是否正常，比如:当状态码为 200|301|302 时，才算正常

分析

- 1.获取状态码的关键值
- 2.根据状态码进行判断即可

```

cat http_code.sh
#!/usr/bin/bash

[ $# -ne 1 ] && echo "请输入url" && exit 1

CURL=$(curl -sI http://$1|grep "HTTP/1.1"|awk '{print $2}')
if [ -z $CURL ];then
    echo "你输入的url地址无法正常访问"
    exit 1
fi

if [ $CURL -eq 200 -o $CURL -eq 301 -o $CURL -eq 302 ];then
    echo "$1 的网站访问正常"
else
    echo "error"
fi

```

20.已知 Nginx 访问的日志文件在 /var/log/nginx/access.log 请统计下早上10点到早上12点来访IP最多的是哪个？

```

grep "22/May/2018:1[0-2]:[0-5][0-9]" 20180523_log.xuliangwei.log|awk '{print $1}'|s
ort|uniq -c|sort -rn|head

```