

编程作业 3--Shell 脚本程序设计

一. 必做题---生成 TCP 活动状况报告

1. 获取时间:

通过 `date` 命令获取当前时间, 具体的指令为:

```
root@lhfh1-virtual-machine:/home/lhfh1/linux_3# date +"%Y-%m-%d %H:%M"
2023-04-21 22:02
```

2. 获取 tcp 报文信息:

通过 `netstat -s -tcp` 获取 tcp 的报文信息, 如图:

```
root@lhfh1-virtual-machine:/home/lhfh1/linux_3# netstat -s -tcp
IcmpMsg:
  InType8: 1
  OutType0: 1
Tcp:
  354 active connection openings
  0 passive connection openings
  185 failed connection attempts
  25 connection resets received
  2 connections established
  54571 segments received
  31789 segments sent out
  754 segments retransmitted
  0 bad segments received
  34 resets sent
UdpLite:
TcpExt:
```

3. 从报文中得到我们需要的数据:

我们需要的是 TCP 发送了多少报文; 接收了多少报文。也就是“segments received”和“segments sent out”前面的数据, 所以可以通过如下命令, 即先用 `grep` 匹配这两行数据再用 `awk` 选出第一列, 即可得到所需数据:

```
root@lhfh1-virtual-machine:/home/lhfh1/linux_3# netstat --statistics -tcp | grep 'segments received' | awk 'NR==1{print $1}'
54599
```

```
root@lhfh1-virtual-machine:/home/lhfh1/linux_3# netstat --statistics -tcp | grep 'segments sent out' | awk 'NR==1{print $1}'
31821
```

然后做和, 即可得到收发报文总数。

4. 判断行尾符号:

行尾要求给出符号+或-或空格 (+表示这分钟收发报文数比上分钟多 10 包以上, 差别在 10 包或以内用空格, 否则用符号-), 可以通过如下图的代码来进行判断, 其中 `num` 代表收发报文数之差, 对其进行判断来决定行尾符号, 但对于第一次输出, 由于没有上一次的作为参考, 故直接在行尾输出空格, 其中 `line_num` 代表行数, `total` 和 `lastTotal` 分别存放一分钟后的收发报文总数和一分钟前的收发报文总数:

```

nowRecv=$(netstat -s -tcp | grep 'segments received' | awk 'NR==1{print $1}')
nowSend=$(netstat -s -tcp | grep 'segments sent out' | awk 'NR==1{print $1}')
total=0
line_num=0
while true
do
    sleep 60
    time=$(date +"%Y-%m-%d %H:%M")
    lastRecv=$nowRecv
    lastSend=$nowSend
    lastTotal=$total
    nowRecv=$(netstat --statistics -tcp | grep 'segments received' | awk 'NR==1{print $1}')
    nowSend=$(netstat --statistics -tcp | grep 'segments sent out' | awk 'NR==1{print $1}')

    let line_num+=1
    recv=$(expr $nowRecv - $lastRecv)
    send=$(expr $nowSend - $lastSend)
    total=$(expr $recv + $send)

    num=$(expr $total - $lastTotal)
    if [ $line_num -eq 1 ]; then
        sign=" "
    fi

    if [ $num -gt 10 ]; then
        sign="+"
    elif [ $num -lt -10 ]; then
        sign="-"
    else
        sign=" "
    fi

    printf "%-s %-s %6s %6s %6s %6s\n" $time $recv $send $total $sign
done

```

5.运行结果

通过 bash tcp.sh 来运行，可得到如下结果：

```

root@lhfh1-virtual-machine:/home/lhfh1/linux_3# bash tcp.sh
2023-04-21 22:24      4      4      8
2023-04-21 22:25   2679   1913   4592      +
2023-04-21 22:26     54     54    108      -
2023-04-21 22:27     68     67    135      +
2023-04-21 22:28     34     34     68      -
2023-04-21 22:29      6      6     12      -
2023-04-21 22:30      8      8     16
2023-04-21 22:31      8      8     16
2023-04-21 22:32      6      5     11

```

二．选做---下载 bing 图库中图片

1.设置参数传递如下：

```

#!/bin/bash
#获取参数作为页数
page_num1=$1
page_num2=$2
#判断是否有输入，没有则用默认的
if [ -z $1 ];then
    page_num1=1
fi
if [ -z $2 ];then
    page_num2=5
fi

```


6.执行脚本下载第一页的图片：

```
root@lhfh1-virtual-machine:/home/lhfh1/linux_3# ./bing.sh 1 1
当前处理第1个url
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload  Total   Spent    Left     Speed
100 27579    0 27579    0     0    12507      0 --:--:--  0:00:02 --:--:-- 12507
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload  Total   Spent    Left     Speed
100 27579    0 27579    0     0    13132      0 --:--:--  0:00:02 --:--:-- 13132
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload  Total   Spent    Left     Speed
100 27579    0 27579    0     0    14356      0 --:--:--  0:00:01 --:--:-- 14349
当前下载第1个图片
curl: (3) <url> malformed
当前下载第2个图片
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload  Total   Spent    Left     Speed
100 25059  100 25059    0     0    70193      0 --:--:--  --:--:--  --:--:-- 70193
当前下载第3个图片
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           Dload  Upload  Total   Spent    Left     Speed
100 49178  100 49178    0     0    50751      0 --:--:--  --:--:--  --:--:-- 50698
当前下载第4个图片
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
```

查看如下：

```
root@lhfh1-virtual-machine:/home/lhfh1/linux_3# cd bing
root@lhfh1-virtual-machine:/home/lhfh1/linux_3/bing# ls
'2023-04-15 ' '2023-04-17 从国际空间站拍摄的地球' '2023-04-18 红海星,地中海' '2023-04-20 褐头凤鹛'
'2023-04-16 安博塞利国家公园的大象, 肯尼亚' '2023-04-17 斯诺登尼亚国家公园, 威尔士, 英国' '2023-04-19 阿德莱德国际风筝节, 澳大利亚' '2023-04-20 马丘比丘, 秘鲁'
'2023-04-16 哥伦比亚河谷, 俄勒冈州, 美国-' '2023-04-18 从纳哈加尔城堡鸟瞰斋浦尔, 印度' '2023-04-19 布列塔尼的小米努灯塔, 法国'
```

下载成功。

代码整体如下：

```
#!/bin/bash
#获取参数作为页数
page_num1=$1
page_num2=$2
#判断是否有输入，没有则用默认的
if [ -z $1 ];then
    page_num1=1
fi
if [ -z $2 ];then
    page_num2=5
fi
#处理url
for i in `seq $page_num1 $page_num2`
do
    echo "当前处理第$i个url"
    #获取图片的url
    curl https://bing.loliu.cn/?p=$i | grep -Eo '<img[>]*src="[a-zA-Z0-9_*/]*">' | grep -Eo 'src="[a-zA-Z0-9_*/]*" | sed 's/src="//g' | sed 's/"//g' > image.txt
    #获取图片名
    curl https://bing.loliu.cn/?p=$i | grep -Eo '<h3>([a-zA-Z0-9_*/]*)</h3>' | sed 's/[<]*[a-zA-Z0-9_*/]*//g' | sed 's/[>]*[a-zA-Z0-9_*/]*//g' | sed 's/[<]*[a-zA-Z0-9_*/]*//g' | sed 's/[>]*[a-zA-Z0-9_*/]*//g' | sed 's/[<]*[a-zA-Z0-9_*/]*//g' | sed 's/[>]*[a-zA-Z0-9_*/]*//g' > name.txt
    #获取时间
    curl https://bing.loliu.cn/?p=$i | grep -o '20[0-9][0-9]-[0-9][0-9]-[0-9][0-9]' > date.txt
done
#创建文件夹bing
path="bing"
if [ ! -d "$path" ]; then
    mkdir "$path"
fi

#将图片名加入到数组中
a=0
for line in `cat name.txt`
do
    name=$line
    names[$a]=$name
    let a++
done
#将时间加入到数组中
b=0
for line in `cat date.txt`
do
    time=$line
    times[$b]=$time
    let b++
done
#下载图片
i=0;
for url in `sort image.txt | uniq`
do
    i=`expr $i + 1`
    echo "当前下载第$i个图片"
    curl -o ./path/${times[$i]} "${names[$i]} $url"
done
```

三.心得总结

通过本次实验，我进一步掌握了 **shell** 脚本编程技术和 **bash** 脚本相关知识，提高了 **linux** 编程实验能力，对各种命令的使用更加熟练，收获很多。