

编程作业 3：Shell 脚本程序设计

姓名：张清戌

学号：2020211259

生成 TCP 活动报告

1. 实验内容

`netstat --statistics` 命令可以列出 `tcp` 等协议的统计信息。编写 shell 脚本程序，每隔 1 分钟生成 1 行信息：当前时间；这一分钟内 TCP 发送了多少报文；接收了多少报文；收发报文总数；行尾给出符号 `+` 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`-`）。

2. 实验步骤

`netstat` 命令输出的 TCP 部分如图所示。

```
qing@PC-Zhang:~$ netstat --statistics --tcp
Tcp:
    3351 active connection openings
    2 passive connection openings
    0 failed connection attempts
    21 connection resets received
    4 connections established
  276742 segments received
  280970 segments sent out
    154 segments retransmitted
    26 bad segments received
    312 resets sent
```

我们只需要获取 `segments received` 和 `segments sent out` 这两行的数据，并且和上次结果进行一些算术运算即可。另外需要记录一下每两次之间的差值，与上次的差值进行比较以决定输出行末的符号。

源代码：

```
#!/bin/bash

# 获取第一个时刻的计数数据，带 PREV 的变量将用于后续时刻参照计算前一时刻数据的变化量
SEND_PREV=$(netstat --statistics --tcp | grep -o '[0-9]* segments received' | grep -oP '[0-9]*')
RCV_PREV=$(netstat --statistics --tcp | grep -o '[0-9]* segments sent out' | grep -oP '[0-9]*')
```

```

TOTAL_PREV=0

# 打印第一个时刻初始为 0 的统计数据
CURRENT_DATETIME=$(date "+%Y-%m-%d %H:%M")
echo "${CURRENT_DATETIME} 0 0 0"

while true
do
    # 休眠一分钟
    sleep 60

    # 获取当前时间 netstat 中 TCP 相关统计数据
    SEND=$(netstat --statistics --tcp | grep -o '[0-9]* segments received'
| grep -oP '[0-9]*')
    RECV=$(netstat --statistics --tcp | grep -o '[0-9]* segments sent out'
| grep -oP '[0-9]*')
    CURRENT_DATETIME=$(date "+%Y-%m-%d %H:%M")

    # 计算此一分钟内的发送接收量，并计算 I/O 总和
    SEND_RECORD=$(expr $SEND - $SEND_PREV)
    RECV_RECORD=$(expr $RECV - $RECV_PREV)

    TOTAL=$(expr $SEND_RECORD + $RECV_RECORD)

    # 计算差值
    SUB_VALUE=$(expr $TOTAL - $TOTAL_PREV)

    OUTPUT="${CURRENT_DATETIME} ${SEND_RECORD} ${RECV_RECORD} ${TOTAL}"

    # 判断差值落在(10, +∞) [-10, 10] (-∞, -10)中的哪个区间内
    if [ $SUB_VALUE -gt 10 ]; then
        OUTPUT=${OUTPUT}" +"
    elif [ $SUB_VALUE -ge -10 ]; then
        OUTPUT=${OUTPUT}" "
    else
        OUTPUT=${OUTPUT}" -"
    fi

    echo $OUTPUT

    # 更新参照数据
    SEND_PREV=${SEND}
    RECV_PREV=${RECV}
    TOTAL_PREV=${TOTAL}

```

done

运行结果如下：

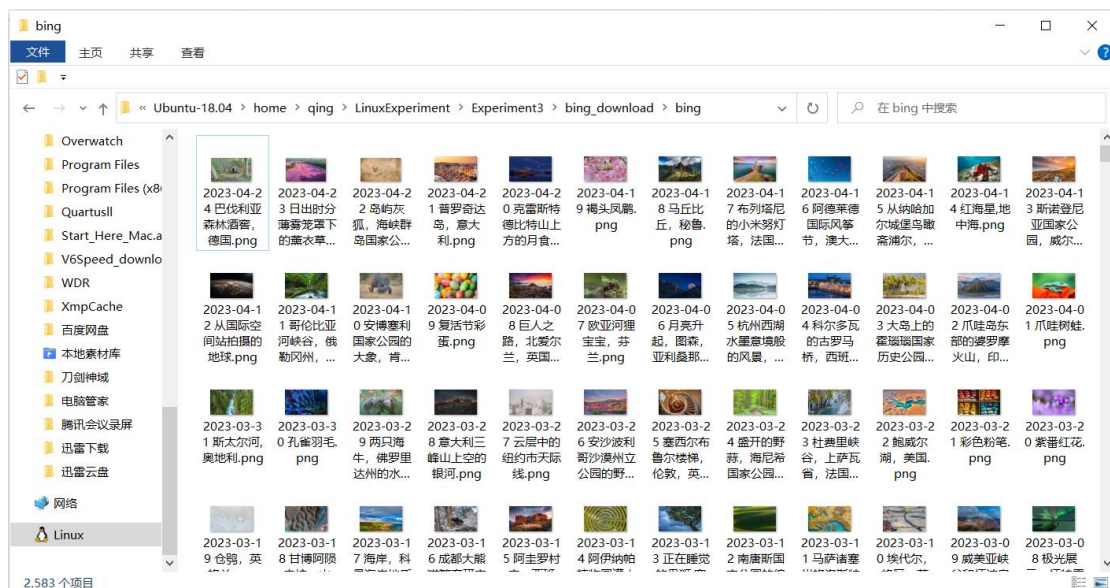
```
qing@PC-Zhang:~/LinuxExperiment/Experiment3$ bash ./tcp_stat.sh
2023-04-23 17:43 0 0 0
2023-04-23 17:44 638 638 1276 +
2023-04-23 17:45 483 483 966 -
2023-04-23 17:46 438 438 876 -
2023-04-23 17:47 550 550 1100 +
2023-04-23 17:48 438 438 876 -
2023-04-23 17:49 483 483 966 +
2023-04-23 17:50 438 438 876 -
2023-04-23 17:51 438 438 876
2023-04-23 17:52 438 438 876
2023-04-23 17:53 438 438 876
2023-04-23 17:54 452 452 904 +
2023-04-23 17:55 432 432 864 -
2023-04-23 17:56 438 438 876 +
2023-04-23 17:57 421 421 842 -
2023-04-23 17:58 438 438 876 +
2023-04-23 17:59 454 454 908 +
2023-04-23 18:00 438 438 876 -
2023-04-23 18:01 438 438 876
2023-04-23 18:02 438 438 876
2023-04-23 18:03 438 438 876
2023-04-23 18:04 454 454 908 +
2023-04-23 18:05 438 438 876 -
2023-04-23 18:06 438 438 876
2023-04-23 18:07 438 438 876
2023-04-23 18:08 438 438 876
2023-04-23 18:09 454 454 908 +
2023-04-23 18:10 438 438 876 -
2023-04-23 18:11 438 438 876
2023-04-23 18:12 438 438 876
2023-04-23 18:13 438 438 876
2023-04-23 18:14 448 448 896 +
```

下载 bing 图库中图片

访问 <https://bing.ioliu.cn/> 网站，观察网页的源代码，用 `sed` 和 `grep` 命令筛选出有用的信息，存入多个 `txt` 文件中。下载时从 `txt` 文件中依次读取每一行，通过开启后台子进程的方式下载。

下载结果：

```
qing@PC-Zhang:~/LinuxExperiment/Experiment3/bing_download/bing$ ls -lr
'2023-04-24 巴伐利亚森林酒窖，德国.png'
'2023-04-23 日出时分薄雾笼罩下的薰衣草田，印度.png'
'2023-04-22 岛屿灰狐，海峡群岛国家公园，美国加利福尼亚州.png'
'2023-04-21 普罗奇达岛，意大利.png'
'2023-04-20 克雷斯特德比特山上方的月食，科罗拉多州，美国.png'
'2023-04-19 褐头凤鹏.png'
'2023-04-18 马丘比丘，秘鲁.png'
'2023-04-17 布列塔尼的小米努灯塔，法国.png'
'2023-04-16 阿德莱德国际风筝节，澳大利亚.png'
'2023-04-15 从纳哈加尔城堡鸟瞰斋浦尔，印度.png'
'2023-04-14 红海星，地中海.png'
'2023-04-13 斯诺登尼亚国家公园，威尔士，英国.png'
'2023-04-12 从国际空间站拍摄的地球.png'
'2023-04-11 哥伦比亚河峡谷，俄勒冈州，美国.png'
'2023-04-10 安博塞利国家公园的大象，肯尼亚.png'
'2023-04-09 复活节彩蛋.png'
'2023-04-08 巨人之路，北爱尔兰，英国.png'
'2023-04-07 欧亚河狸宝宝，芬兰.png'
'2023-04-06 月亮升起，图森，亚利桑那州，美国.png'
'2023-04-05 杭州西湖水墨意境般的风景，浙江省，中国.png'
'2023-04-04 科尔多瓦的古罗马桥，西班牙.png'
'2023-04-03 大岛上的霍瑞瑞国家历史公园，夏威夷.png'
'2023-04-02 爪哇岛东部的婆罗摩火山，印度尼西亚.png'
'2023-04-01 爪哇树蛙.png'
'2023-03-31 斯太尔河，奥地利.png'
```



```
qing@PC-Zhang:~/LinuxExperiment/Experiment3/bing_download/bing$ ll | grep ^- | wc -l
2583
```

源代码:

```
#!/bin/bash

page_num1=$1
page_num2=$2

#未设定参数，默认下载全部
if [ -z $1 ];then
    page_num1=1
fi
if [ -z $2 ];then
    page_num2=217
fi

#处理 url，获取图片信息
for i in `seq $page_num1 $page_num2`
do

echo "当前处理第$i 个 url"

#获取图片连接
curl https://bing.ioliu.cn/?p=$i | grep -Eo
'<img[^>]*src="https://[^\"]*"[^>]*>' | grep -Eo 'src="[^\"]*"' | sed
's/src="//g' | sed 's/"//g' >> image.txt
#获取图片名称
curl https://bing.ioliu.cn/?p=$i | grep -Eo '<h3>(.\|\\n)*?</h3>' | sed
's/<[^>]*>/ /g' | sed 's/([^\s])*/ /g' | sed 's/分享/\\n/g' | sed
's/[0-9a-zA-Z]*/ /g' | sed 's/- / /g' | sed 's/\\ / /g' >> name.txt
#获取图片日期
curl https://bing.ioliu.cn/?p=$i | grep -o
'20[0-9][0-9]-[0-9][0-9]-[0-9][0-9]' >> time.txt

done

path="bing"
if [ ! -d "$path" ]; then
    mkdir "$path"
fi

a=0
for line in `cat name.txt`
do
    name=$line
    names[$a]=$name
```

```
    let a++
done

b=0
for line in `sort -r time.txt | uniq` #去重
do
    time=$line
    times[$b]=$time
    let b++
done

i=0;

for url in `cat imgage.txt`
do
    echo "当前下载第`expr $i + 1`个图片"
    curl -o ./path/${times[$i]} "${names[$i]}.png" $url
    i=`expr $i + 1`
done
```

实验总结

本次实验中我主要应用了 `expr`, `echo`, `sed`, `grep` 等 shell 指令和 `if`, `for`, `while` 编写 shell 脚本, 使我对 shell 文本编辑和脚本编写更加熟悉。