

# 上机作业 3：Shell 脚本程序设计

088 于海鑫

2017211240

*name1e5s@bupt.edu.cn*

版本：8

更新：May 1, 2020

## 1 第一题：生成 TCP 活动状况报告

### 1.1 题目要求

`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 -）。

### 1.2 解法

#### 1.2.1 获取信息

执行 `netstat -statistics`，获得信息如下：

```
1 ...  
2 Tcp:  
3     3634892 active connection openings  
4     4228 passive connection openings  
5     3506040 failed connection attempts  
6     5947 connection resets received
```

```

7      15 connections established
8      9911704 segments received
9      10086150 segments sent out
10     13700 segments retransmitted
11     8 bad segments received
12     3514876 resets sent
13     ...

```

我们需要的就是 `9911704 segments received` 以及 `10086150 segments sent out` 这两部分的数据。

使用 `直接获取之`。

```

1 $ COMMAND=$(netstat --statistics)
2 $ RECV=$(expr "$COMMAND" : '^.* \([0-9]*\) segments received')
3 $ SENT=$(expr "$COMMAND" : '^.* \([0-9]*\) segments sent out')
4 $ echo $RECV $SENT
5 9914989 10090134

```

### 1.3 编写完整程序

在上一节的基础上，我们加上一些循环的代码，即可得到结果。

```

1  #!/bin/bash
2
3  FIRST_FLAG=1
4
5  COMMAND=$(netstat --statistics)
6  RECV_PREV=$(expr "$COMMAND" : '^.* \([0-9]*\) segments received')
7  SENT_PREV=$(expr "$COMMAND" : '^.* \([0-9]*\) segments sent out')
8  TOTAL_PREV=0
9
10 while true
11 do
12     sleep 60
13     DATE=$(date + "%Y-%m-%d %H:%M")
14     COMMAND=$(netstat --statistics)
15     RECV=$(expr "$COMMAND" : '^.* \([0-9]*\) segments received')
16     SENT=$(expr "$COMMAND" : '^.* \([0-9]*\) segments sent out')

```

```

17     R_DELTA=$(expr $RECV - $RECV_PREV)
18     S_DELTA=$(expr $SENT - $SENT_PREV)
19     TOTAL=$(expr $R_DELTA + $S_DELTA)
20     TOTAL_DELTA=$(expr $TOTAL - $TOTAL_PREV)
21
22     if [ $FIRST_FLAG = 1 ]
23     then
24         echo $DATE $S_DELTA $R_DELTA $TOTAL
25         FIRST_FLAG=0
26     else
27         if [ $TOTAL_DELTA -gt 10 ]
28         then
29             echo $DATE $S_DELTA $R_DELTA $TOTAL +
30         elif [ $TOTAL_DELTA -lt 0 ]
31         then
32             echo $DATE $S_DELTA $R_DELTA $TOTAL -
33         else
34             echo $DATE $S_DELTA $R_DELTA $TOTAL
35         fi
36     fi
37     RECV_PREV=$RECV
38     SENT_PREV=$SENT
39     TOTAL_PREV=$TOTAL
40 done

```

结果如下:

```

1 name1e5s@sumeru:~$ bash test.sh
2 2020-05-01 19:46 365 253 618
3 2020-05-01 19:47 335 239 574 -
4 2020-05-01 19:48 362 246 608 +
5 2020-05-01 19:49 373 250 623 +
6 2020-05-01 19:50 350 245 595 -
7 2020-05-01 19:51 376 260 636 +
8 2020-05-01 19:52 400 275 675 +
9 2020-05-01 19:53 368 250 618 -
10 2020-05-01 19:54 389 265 654 +
11 2020-05-01 19:55 391 274 665 +
12 2020-05-01 19:56 416 279 695 +
13 2020-05-01 19:57 380 259 639 -

```

```
14 2020-05-01 19:58 451 293 744 +
15 2020-05-01 19:59 381 267 648 -
16 2020-05-01 20:00 411 280 691 +
17 2020-05-01 20:01 427 291 718 +
18 2020-05-01 20:02 399 270 669 -
```

符合要求。

## 2 第二题：下载 bing 图库中图片

### 2.1 题目要求

访问 <https://bing.ioliu.cn/?p=23> 可以看到 bing 图库第 23 页的内容，这个 Web 页有多个图片小样，将鼠标放到某个小样上，如右上角，可见中文说明信息“野花草甸上的一只欧亚雕鸮，德国莱茵兰-普法尔茨”和日期信息 2019-08-03，点击一下，此图片就可以下载。

编写脚本程序 `bing.sh`，将这个图库中照片全部下载下来存放到本地 `bing` 目录中，上面 URL 中 `p=23` 可以换成 `p=1` 到 `p=126` 可访问 126 个页，每页有 12 个图，每个图的日期，中文说明信息和下载地址及文件名 `html` 文件中可提取。要求下载后的文件命名为“日期中文说明.jpg”

例如：2019-08-03 野花草甸上的一只欧亚雕鸮，德国莱茵兰-普法尔茨.jpg

### 2.2 获取结果

首先使用 Chrome 的调试功能，确定出来单个小样的表示形式如下：

```
1 <div class="item">
2   <div class="card_progressive">
3     
8     <a class="mark" href="/photo/BurgAltdahn_ZH-CN8281669977?force=
9       home_1"></a>
10    <div class="description">
```

```

8      <h3>达恩附近普法尔茨森林中的Altdahn城堡，德国莱茵兰-普法尔茨(
      Dahn Rockland), Palatinate Forest, Rhineland-Palatinate,
      Germany (© Reinhard Schmid/Huber/eStock Photo)</h3>
9      <p class="calendar">
10         <i class="icon icon-calendar"></i>
11         <em class="t">2020-05-01</em></p>
12         <p class="view">
13             <i class="icon icon-eye"></i>
14             <em class="t">453</em></p>
15     </div>
16     <div class="options">
17         ...
18     </div>
19 </div>
20 </div>

```

显然我们需要的只有 `mark` 下面的 `href` 以及 `description` 下面的描述。

显然我们使用如下指令即可匹配需要的数据：

```

1 cat index.html | sed -e 's/<div class="item">/\n/g' | \
2 grep "<div class=\"card progressive\" | \"
3 sed -e 's#.*\"mark\"_href=\"\"(.*)\".*\"description\"><h3>\"(.*)\"
   (@.*\"([0-9]{4})-([0-9]{2})-([0-9]{2})\"<.*#1$2$3#g' \
4     -e 's#home_\"([0-9]*)\"#download#g' \
5     -e 's#_\"$2$2#g'

```

输出如下：

```

1 /photo/BurgAltdahn_ZH-CN8281669977?force=download$达恩附近普法尔茨森
   林中的Altdahn城堡，德国莱茵兰-普法尔茨(Dahn Rockland), Palatinate
   Forest, Rhineland-Palatinate, Germany$2020-05-01
2 /photo/ArcticRedpoll_ZH-CN7968973967?force=download$极北朱顶雀的巢，
   芬兰拉普兰$2020-04-30
3 /photo/PalouseSpring_ZH-CN6803103328?force=download$普尔曼附近的帕卢
   斯一辆拖拉机在耕作时扬起尘土，华盛顿州$2020-04-29
4 /photo/SalisburyCathedral_ZH-CN6366350896?force=download$索尔兹伯里大
   教堂与放牧的羊群，英格兰$2020-04-28
5 /photo/SouthAmericanTapir_ZH-CN6151058361?force=download$一只南美獭幼
   崽小跑着穿过草地$2020-04-27

```

```

6 /photo/RubySunset_ZH-CN5544596519?force=download$红宝石海滩的日落，华盛顿州奥林匹克国家公园$2020-04-26
7 /photo/FalklandRockhoppers_ZH-CN5370686595?force=download$福克兰群岛上的南跳岩企鹅$2020-04-25
8 /photo/MegellanicCloud_ZH-CN5132305226?force=download$由哈勃太空望远镜拍摄的大麦哲伦星云$2020-04-24
9 /photo/KingEider_ZH-CN3559595357?force=download$正在游水的雄性王绒鸭，挪威特罗姆斯-芬马克郡$2020-04-23
10 /photo/KauriTree_ZH-CN3695568740?force=download$怀波瓦森林中一棵名为Te Matua Ngahere的巨型贝壳杉树，新西兰北地$2020-04-22
11 /photo/GPS_ZH-CN5160095061?force=download$黄石国家公园的大棱镜泉，怀俄明州$2020-04-21
12 /photo/BluebellWood_ZH-CN8128422960?force=download$Micheldever Wood的蓝铃花，英国汉普郡$2020-04-20

```

之后就是简单的多线程下载等步骤，全部代码如下：

```

1  #!/bin/bash
2
3  if [ $# -gt 2 ]
4  then
5      echo "Usage: _$0_ <begin> _<end>"
6      exit 1
7  fi
8
9  if [ $# = 0 ]
10 then
11     START=1
12     END=126
13 else
14     START=$1
15     END=$2
16 fi
17
18 if [ ! -d bing ]
19 then
20     mkdir bing
21 fi
22
23 if [ ! -d html ]

```

```

24 then
25     mkdir html
26 fi
27
28 if [ ! -d temp ]
29 then
30     mkdir temp
31 fi
32
33 for i in $(seq $START $END)
34 do
35     if [ ! -f html/$i.html ]
36     then
37         wget --no-verbose -O html/$i.html https://bing.ioliu.cn/?p=$i
38     fi
39
40     if [ ! -f temp/$i.dat ]
41     then
42         cat html/$i.html | \
43         sed -e 's/<div_class="item">/\n/g' | \
44         grep "<div_class=\"card progressive\"" | \
45         sed -e 's#.*\"mark\"_href=\"\" (.*)\".*\"description\"><h3
46             >\" (.*)\" (@.*\" ([0-9]\{4\}\|- [0-9]\{2\}\|- [0-9]\{2\}\)\<.*#\1$ \2$
47             \3#g' \
48             -e 's#home_\" ([0-9]*\" )#download#g' \
49             -e 's#_\"$2#$2#g' > temp/$i.dat
50     fi
51 done
52
53 for i in $(seq $START $END)
54 do
55     while read line
56     do
57         eval $(echo "$line" | awk -F "[" '{printf("url=https://bing
58             .ioliu.cn%s;name=\"%s\";date=%s",$1,$2,$3)}')
59         img="bing/$date_$name.jpg"
60         if [ ! -f "$img" ]
61         then
62             tmp="$img.tmp$$"

```

```

60         wget --no-verbose -O "$tmp" $url
61         if [ $? = 0 -a ! -f "$img" ]
62         then
63             mv "$tmp" "$img"
64         else
65             rm "$tmp"
66         fi
67     fi
68     done < temp/$i.dat
69 done

```

运行结果如下（部分）：

```

1  name1e5s@sumeru:~/t$ bash a.sh 1 1
2  2020-05-01 22:21:29 URL:https://bing.ioliu.cn/?p=1 [23648/23648] -> "
   html/1.html" [1]
3  2020-05-01 22:21:33 URL:https://bing.ioliu.cn//photo/BurgAltdahn_ZH-
   CN8281669977?force=download [290069] -> "bing/2020-05-01 达恩附近
   普法尔茨森林中的Altdahn城堡，德国莱茵兰-普法尔茨(Dahn Rockland),
   Palatinate Forest, Rhineland-Palatinate, Germany.jpg.tmp5340" [1]
4  2020-05-01 22:21:40 URL:https://bing.ioliu.cn//photo/ArcticRedpoll_ZH
   -CN7968973967?force=download [327709] -> "bing/2020-04-30 极北朱顶
   雀的巢，芬兰拉普兰.jpg.tmp5340" [1]
5  2020-05-01 22:21:44 URL:https://bing.ioliu.cn//photo/PalouseSpring_ZH
   -CN6803103328?force=download [181230] -> "bing/2020-04-29 普尔曼附
   近的帕卢斯一辆拖拉机在耕作时扬起尘土，华盛顿州.jpg.tmp5340" [1]
6  2020-05-01 22:21:49 URL:https://bing.ioliu.cn//photo/
   SalisburyCathedral_ZH-CN6366350896?force=download [284849] -> "
   bing/2020-04-28 索尔兹伯里大教堂与放牧的羊群，英格兰.jpg.tmp5340"
   [1]
7  2020-05-01 22:21:52 URL:https://bing.ioliu.cn//photo/
   SouthAmericanTapir_ZH-CN6151058361?force=download [297595] -> "
   bing/2020-04-27 一只南美獭幼崽小跑着穿过草地.jpg.tmp5340" [1]
8  2020-05-01 22:21:55 URL:https://bing.ioliu.cn//photo/RubySunset_ZH-CN
   5544596519?force=download [310129] -> "bing/2020-04-26 红宝石海滩
   的日落，华盛顿州奥林匹克国家公园.jpg.tmp5340" [1]
9  2020-05-01 22:21:57 URL:https://bing.ioliu.cn//photo/
   FalklandRockhoppers_ZH-CN5370686595?force=download [142025] -> "
   bing/2020-04-25 福克兰群岛上的南跳岩企鹅.jpg.tmp5340" [1]
10 2020-05-01 22:22:01 URL:https://bing.ioliu.cn//photo/MegellanicCloud_

```



```

    ZH-CN5132305226?force=download [338962] -> "bing/2020-04-24 由哈勃
    太空望远镜拍摄的大麦哲伦星云.jpg.tmp5340" [1]
11 2020-05-01 22:22:03 URL:https://bing.ioliu.cn//photo/KingEider_ZH-CN
    3559595357?force=download [140302] -> "bing/2020-04-23 正在游水的
    雄性王绒鸭，挪威特罗姆斯-芬马克郡.jpg.tmp5340" [1]
12 2020-05-01 22:22:05 URL:https://bing.ioliu.cn//photo/KauriTree_ZH-CN
    3695568740?force=download [343549] -> "bing/2020-04-22 怀波瓦森林
    中一棵名为Te Matua Ngahere的巨型贝壳杉树，新西兰北地.jpg.tmp5340"
    [1]
13 2020-05-01 22:22:08 URL:https://bing.ioliu.cn//photo/GPS_ZH-CN
    5160095061?force=download [292443] -> "bing/2020-04-21 黄石国家公
    园的大棱镜泉，怀俄明州.jpg.tmp5340" [1]
14 2020-05-01 22:22:10 URL:https://bing.ioliu.cn//photo/BluebellWood_ZH-
    CN8128422960?force=download [269926] -> "bing/2020-04-20
    Micheldever Wood的蓝铃花，英国汉普郡.jpg.tmp5340" [1]

```

完成后打开 **bing** 文件夹，即可看到文件。

```

1 name1e5s@sumeru:~/t/bing$ ls
2 '2020-04-20 Micheldever Wood的蓝铃花，英国汉普郡.jpg'
3 '2020-04-21 黄石国家公园的大棱镜泉，怀俄明州.jpg'
4 '2020-04-22 怀波瓦森林中一棵名为Te Matua Ngahere的巨型贝壳杉树，新西
    兰北地.jpg'
5 '2020-04-23 正在游水的雄性王绒鸭，挪威特罗姆斯-芬马克郡.jpg'
6 '2020-04-24 由哈勃太空望远镜拍摄的大麦哲伦星云.jpg'
7 '2020-04-25 福克兰群岛上的南跳岩企鹅.jpg'
8 '2020-04-26 红宝石海滩的日落，华盛顿州奥林匹克国家公园.jpg'
9 '2020-04-27 一只南美貊幼崽小跑着穿过草地.jpg'
10 '2020-04-28 索尔兹伯里大教堂与放牧的羊群，英格兰.jpg'
11 '2020-04-29 普尔曼附近的帕卢斯一辆拖拉机在耕作时扬起尘土，华盛顿州.
    jpg'
12 '2020-04-30 极北朱顶雀的巢，芬兰拉普兰.jpg'
13 '2020-05-01 达恩附近普法尔茨森林中的Altdahn城堡，德国莱茵兰-普法尔茨(
    Dahn Rockland), Palatinate Forest, Rhineland-Palatinate, Germany.
    jpg'

```

### 3 作业总结

本次上机作业比较耗费时间，主要是要回顾之前学过的内容。

通过本次作业，我对 `sed` 的使用有了更加深刻的了解。