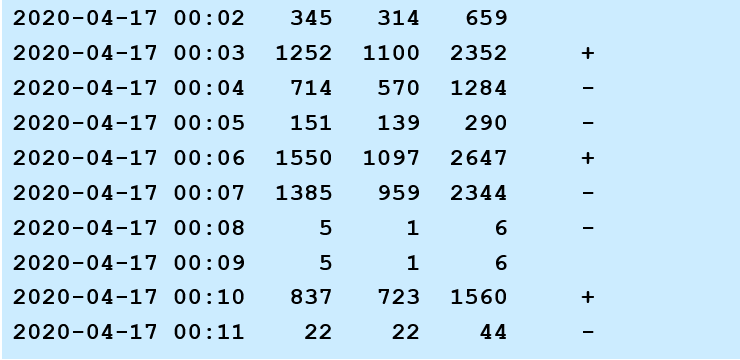
**实验三：生成TCP活动状况报告**

**一、实验目的**

* 掌握shell脚本编程技术，掌握bash脚本相关知识点
* 熟练结合课程所学其他内容完成实验，例如vi编辑器的使用、文件权限的修改、字符处理指令的使用等
* 提高Linux系统的基本操作能力

**二、实验要求**

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-）。运行示例如下：



即每一行信息包含：

* 当前时间
* 一分钟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

**三、实验环境**

* Ubuntu 20.04.2 LTS(GNU/Linux 5.4.0-58-generic x86\_64)
* GNU bash, version 5.0.17(1)-release (x86\_64-pc-linux-gnu)
* xshell

**四、实验过程**

**1.程序设计**

本程序有多个要解决的点，包含当前时间获取、TCP收发流量数据的获取、流浪变化符号的判断、数据的存储与处理、格式化输出、周期循环实现等。

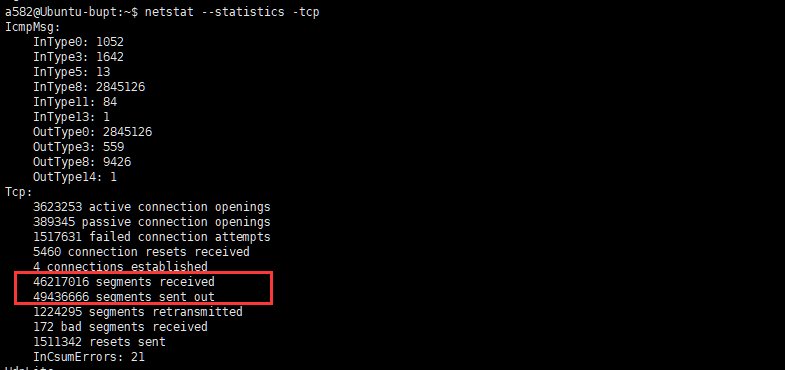
* **当前时间获取**

本次实验中需要输出的时间包括有“年”、“月”、“日”、“小时”、“分钟”，可以通过date命令获取当前时间，具体的指令为：



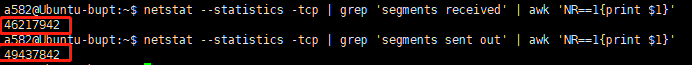
* **TCP收发流量数据的获取**

使用netstat –statistics -tcp可以获取到当前系统的流浪数据，如图：



我们需要的是“segments received”和“segments sent out”前面的数字，因此使用grep匹配这两行数据，再使用awk脚本选择第一列即数据列作为结果输出：

|  |
| --- |
| **netstat --statistics -tcp | grep 'segments received' | awk 'NR==1{print $1}'**  **netstat --statistics -tcp | grep 'segments sent out' | awk 'NR==1{print $1}'** |



通过两次不同收发报文总量的差，可以计算出一分钟内的总收发报文量。

* **流量变化符号的判断**

行尾要给出符号+或-或空格，+代表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shell控制中的条件判断语句来实现，首先我们使用这一分钟的总报文数减去上一分钟的总报文数，得到流量的变化大小，将其存储在临时变量num中，通过条件判断语句判断num的值，大于10则将符号置为‘+’，小于-10将符号置于‘-’，否则将符号置于‘ ’。

同时，由于第一次输出时并没有上一分钟的总报文数，因此需要对第一次输出进行热哦按，将第一次输出的符号置为‘ ’，同样使用条件判断语句实现。

|  |
| --- |
| **num=$(expr $total - $totalLast)**  **if [ $num -gt 10 ]; then**  **sign="+"**  **elif [ $num -lt -10 ]; then**  **sign="-"**  **else**  **sign=" "**  **fi**    **if [ $cnt -eq 1 ]; then**  **sign=" "**  **fi** |

**2.数据结构设计**

* **time**
* 当前时间，使用date命令获取
* **nowRecv,nowSend**
* 当前时间获取到的收发报文数据数量，使用netstat命令并处理后得到
* **lastRecv,lastSend**
* 上一时间获取到的收发报文数据数量，每次循环时保存上一时间数据
* **recv，send**
* 当前一分钟收到的报文数量和发出的报文数量
* **total**
* 当前收发报文总量
* **lastTotal**
  + 上一时间收发报文总量
* **num**
* 当前时间与上一时间收发报文差值，通过total减去lastTotal得到
* **sign**
  + 流量变化符号‘+’、‘-’‘ ’，通过num值判断得到
* **cnt**
  + 记录输出行数，主要用于判断是否为第一行输出

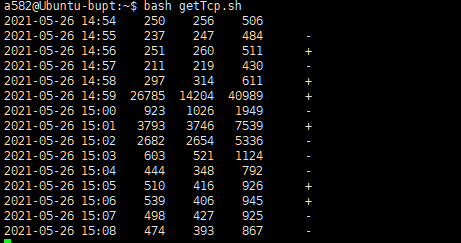
**3.输出格式化**

利用 printf 来进行格式化输出， %s 表示输出为字符串、使用 %-s 、 %s 来控制左右对齐、使用%6s 来控制占位数、使用 \n 来进行换行。 由于time中间有空格存在，因此输出时需要使用两个“%-s %-s”来进行输出

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

**五、实验运行结果**

编写**getTcp.sh**并使用**bash getTcp.sh**命令运行，得到如下的结果：



**六、实验心得**

本次实验,使我初步掌握了shell脚本的编写方法，通过对于getTcp.sh的编写与运行，加深了我对于shell脚本语言的理解和编写熟练度，在本次实验中我也遇到了一些问题。主要有以下两个问题：

* 使用netstat –statistics -tcp命令展示的结果出错

一开始使用该命令时，得到的结果并没有显示出当前总收发的数据包数

量，而是一行行的tcp数据，检查输入命令后发现，在**statistics的命令前是 - -**，修改错误后得到想要的数据

* time输出异常

使用 printf 进行格式化输出的时候始终格式有误。多次尝试总结得出，由于保存time时中间有空格存在，因此在输出过程中，“年月日”与“时间分钟”被分别分配到两个 %s 格式符中进行处理，因此导致格式错乱。在输出过程中需要使用 %-s %-s 来对时间部分time进行输出。

本次实验也让我复习了之前学过的知识，包括使用vi编辑器来编写脚本、查看权限以及使用chmod对其进行修改、使用grep与awk结合管道机制进行字符串的处理等等。通过实验，让我对Linux更加的熟悉并掌握了基础的一些操作。

**附 录**

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}'**)**

total=0

cnt=0

*#Ñ­»·*

**while** **true**

**do**

*#ÖÜÆÚÊ±¼ä*

**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** cnt+=1

recv=$**(expr** $nowRecv - $lastRecv**)**

send=$**(expr** $nowSend - $lastSend**)**

total=$**(expr** $recv + $send**)**

*#ÅÐ¶ÏÁ÷Á¿±ä»¯*

num=$**(expr** $total - $lastTotal**)**

**if** **[** $num -gt 10 **]**; **then**

sign="+"

**elif** **[** $num -lt -10 **]**; **then**

sign="-"

**else**

sign=" "

**fi**

**if** **[** $cnt -eq 1 **]**; **then**

sign=" "

**fi**

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

**done**