

# Lab 0 Writeup

My name: 李易

My Student number : 1918300079

This lab took me about 8 hours to do. I did attend the lab session.

My secret code from section 2.1 was: ALOHA,NJUser!

Welcome to computer networking!

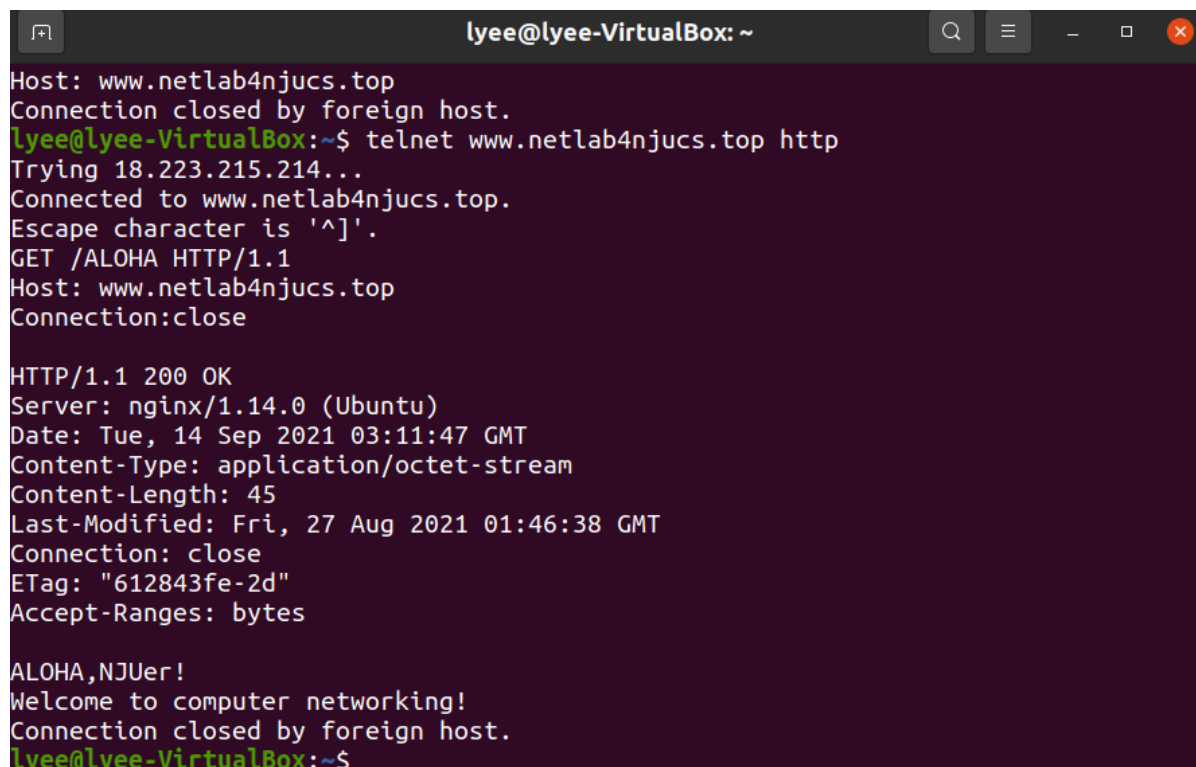
## 1. Program Structure and Design:

### (1)环境的搭建

相关虚拟机的安装，编译工具的安装等。

### (2)获取网页内容

利用telnet程序与其他服务器建立字节流，从而获取网页的内容。



```
lyee@lyee-VirtualBox: ~  
Host: www.netlab4njucs.top  
Connection closed by foreign host.  
lyee@lyee-VirtualBox:~$ telnet www.netlab4njucs.top http  
Trying 18.223.215.214...  
Connected to www.netlab4njucs.top.  
Escape character is '^]'.  
GET /ALOHA HTTP/1.1  
Host: www.netlab4njucs.top  
Connection:close  
  
HTTP/1.1 200 OK  
Server: nginx/1.14.0 (Ubuntu)  
Date: Tue, 14 Sep 2021 03:11:47 GMT  
Content-Type: application/octet-stream  
Content-Length: 45  
Last-Modified: Fri, 27 Aug 2021 01:46:38 GMT  
Connection: close  
ETag: "612843fe-2d"  
Accept-Ranges: bytes  
  
ALOHA,NJUser!  
Welcome to computer networking!  
Connection closed by foreign host.  
lyee@lyee-VirtualBox:~$
```

### (3)监听与连接

通过netcat建立一个简单的服务器，并在另一终端中利用telnet与之连接。现在两终端可以以一种“回声”的方式相互传递内容。

```
lyee@lyee-VirtualBox: ~  
lyee@lyee-VirtualBox:~$ telnet localhost 9090  
Trying 127.0.0.1...  
Connected to localhost.  
Escape character is '^]'.  
Connection Established  
NJU Networking Lab  
Echo  
lyee@lyee-VirtualBox:~$  
lyee@lyee-VirtualBox:~$ netcat -v -l -p 9090  
netcat: getnameinfo: Temporary failure in name resolution  
lyee@lyee-VirtualBox:~$ netcat -v -l -p 9090  
netcat: getnameinfo: Temporary failure in name resolution  
lyee@lyee-VirtualBox:~$ netcat: getnameinfo: Temporary failure in name resolution  
Command 'netcat:' not found, did you mean:  
  command 'netcat' from deb netcat-openbsd (1.206-1ubuntu1)  
  command 'netcat' from deb ncat (7.80+dfsg1-2build1)  
  command 'netcat' from deb netcat-traditional (1.10-41.1ubuntu1)  
Try: sudo apt install <deb name>  
lyee@lyee-VirtualBox:~$ netcat -v -ln -p 9090  
Listening on 0.0.0.0 9090  
Connection received on 127.0.0.1 52960  
Connection Established  
NJU Networking Lab  
Echo  
^
```

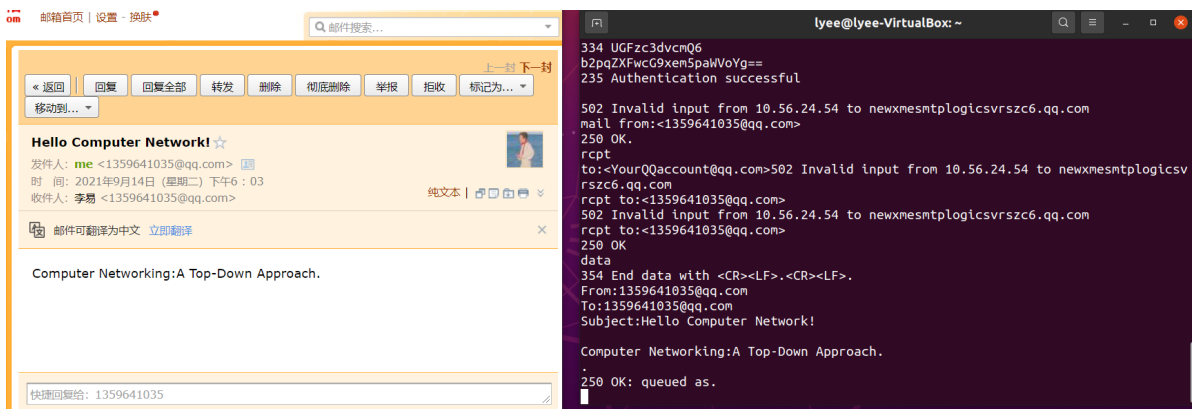
#### (4)尝试一些简单的命令

Ping指令记录了信息在源和目的地之间的往返时间。而traceroute指令则更为详细一些：它能追踪记录连接过程中的每个跃点，显示出它们的时延。因此，Ping指令常用于测试连通性，而traceroute或tracert指令可用于检测路径中存在的问题，排查故障。（如多次出现同一地址，可能是由于出现了环路）

```
lyee@lyee-VirtualBox: ~  
ms 202.97.41.202 (202.97.41.202) 154.220 ms  
10 202.97.92.45 (202.97.92.45) 173.794 ms 202.97.50.26 (202.97.50.26) 192.163 ms  
202.97.92.45 (202.97.92.45) 173.741 ms  
11 218.30.53.2 (218.30.53.2) 172.209 ms 186.216 ms *  
12 * * *  
13 * * *  
14 * * 52.93.129.129 (52.93.129.129) 239.650 ms  
15 * * *  
16 * * *  
17 * * *  
18 * * *  
19 52.93.128.128 (52.93.128.128) 229.346 ms 150.222.244.188 (150.222.244.188) 21  
.545 ms 150.222.244.192 (150.222.244.192) 234.250 ms  
20 54.239.47.184 (54.239.47.184) 243.457 ms 54.239.47.186 (54.239.47.186) 240.99  
ms 54.239.47.182 (54.239.47.182) 246.739 ms  
21 * * *  
22 * * *  
23 * * *  
24 * * *  
25 * * *  
26 52.93.239.83 (52.93.239.83) 247.663 ms 52.95.2.98 (52.95.2.98) 230.307 ms 52.  
3.239.115 (52.93.239.115) 231.365 ms  
27 52.95.2.191 (52.95.2.191) 218.584 ms 52.95.2.223 (52.95.2.223) 236.964 ms 52.  
5.3.25 (52.95.3.25) 243.287 ms  
28 52.95.2.140 (52.95.2.140) 262.625 ms 52.95.3.20 (52.95.3.20) 214.676 ms 52.95  
3.22 (52.95.3.22) 244.720 ms  
29 52.95.2.173 (52.95.2.173) 218.715 ms 15.230.140.122 (15.230.140.122) 232.689  
s 52.95.2.229 (52.95.2.229) 236.537 ms  
30 15.230.134.77 (15.230.134.77) 202.999 ms 15.230.39.196 (15.230.39.196) 221.55  
ms *  
lyee@lyee-VirtualBox:~$
```

#### (5) 给自己发送邮件

利用telnet和SMTP（Simple Mail Transfer Protocol）给自己发送一封邮件。



## (6)完成Webget

利用提供的socket接口完成webget，实现get\_URL函数。

```
void get_URL(const string &host, const string &path) {
    TCPSocket soc;
    soc.connect(Address(host, "http"));
    soc.write("GET " + path + " HTTP/1.1\r\n" + "Host: " + host + "\r\n" +
"Connection: close\r\n\r\n");
    while(!soc.eof())
    {
        cout << soc.read();
    }
    soc.shutdown(SHUT_WR);
}
```

## 2. Implementation Challenges:

(1) 代码部分实现的难点主要在于需要了解TCPSocket类中为我们提供的相关函数接口，这方面只需要阅读好html文件中对这些函数的介绍即可。我们使用eof函数来控制输出内容，shutdown函数需要我们传入参数，当我们传入SHUT\_WR时，仅关闭写入流。比起close函数来说，shutdown函数的调控更细一些。

write函数中，我们需要构造出http的GET请求：

```
GET /docs/index.html HTTP/1.1
Host: www.nowhere123.com
Accept: image/gif, image/jpeg, */*
Accept-Language: en-us
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
(blank line)
```

另外，需要注意文档中提及的\r\n和Connection: close等细节。

## 3. Remaining Bugs:

```
lyee@lyee-VirtualBox: ~/Documents/lab0-Lyee-Git
[ 33%] Built target sponge
Consolidate compiler generated dependencies of target webget
[ 40%] Built target webget
[ 46%] Built target spongechecks
[ 53%] Built target byte_stream_many_writes
[ 60%] Built target byte_stream_capacity
[ 66%] Built target byte_stream_two_writes
[ 73%] Built target byte_stream_one_write
[ 80%] Built target byte_stream_construction
[ 86%] Built target parser_dt
[ 93%] Built target socket_dt
[100%] Built target address_dt
lyee@lyee-VirtualBox:~/Documents/lab0-Lyee-Git/sponge/build$ make check_webget
[100%] Testing webget...
Test project /home/lyee/Documents/lab0-Lyee-Git/sponge/build
  Start 28: t_webget
1/1 Test #28: t_webget ..... Passed    0.57 sec

100% tests passed, 0 tests failed out of 1

Total Test time (real) =  0.76 sec
[100%] Built target check_webget
lyee@lyee-VirtualBox:~/Documents/lab0-Lyee-Git/sponge/build$ cd ../../
lyee@lyee-VirtualBox:~/Documents/lab0-Lyee-Git$ git status
```

第一次作业代码量较小，以上为通过测试用例截图。