**CS5341.501/502**

**Summer I, 2020**

**Advanced Network Programming**

**Assignment 3**

**Submitted by: Sonali Bante**

**A04911126**

All Source code is stored Q1, Q2, Q3 folder as per questions. Result output is given in screenshot folder and commands needs to execute are as given in respective questions below:

1. (30 pts) Modify TCP client-server example in Chapter 5 by using the recvmsg and sendmsg functions, instead of read and write functions.

ANS:

I used two codes from chapter 5 for performing this question which are tcpcli01 and tcpserv01. Source code for this is given in Q1 folder.

The str\_cli and str\_echo functions are declared in the same file independently. Compiling them with library was resulting in segmentation errors so to avoid that it is done this way.

Exicution commands are as follows:

**On eros:**

cd …/Q1

cc -o tcpserv01 tcpserv01.c

./tcpserv01

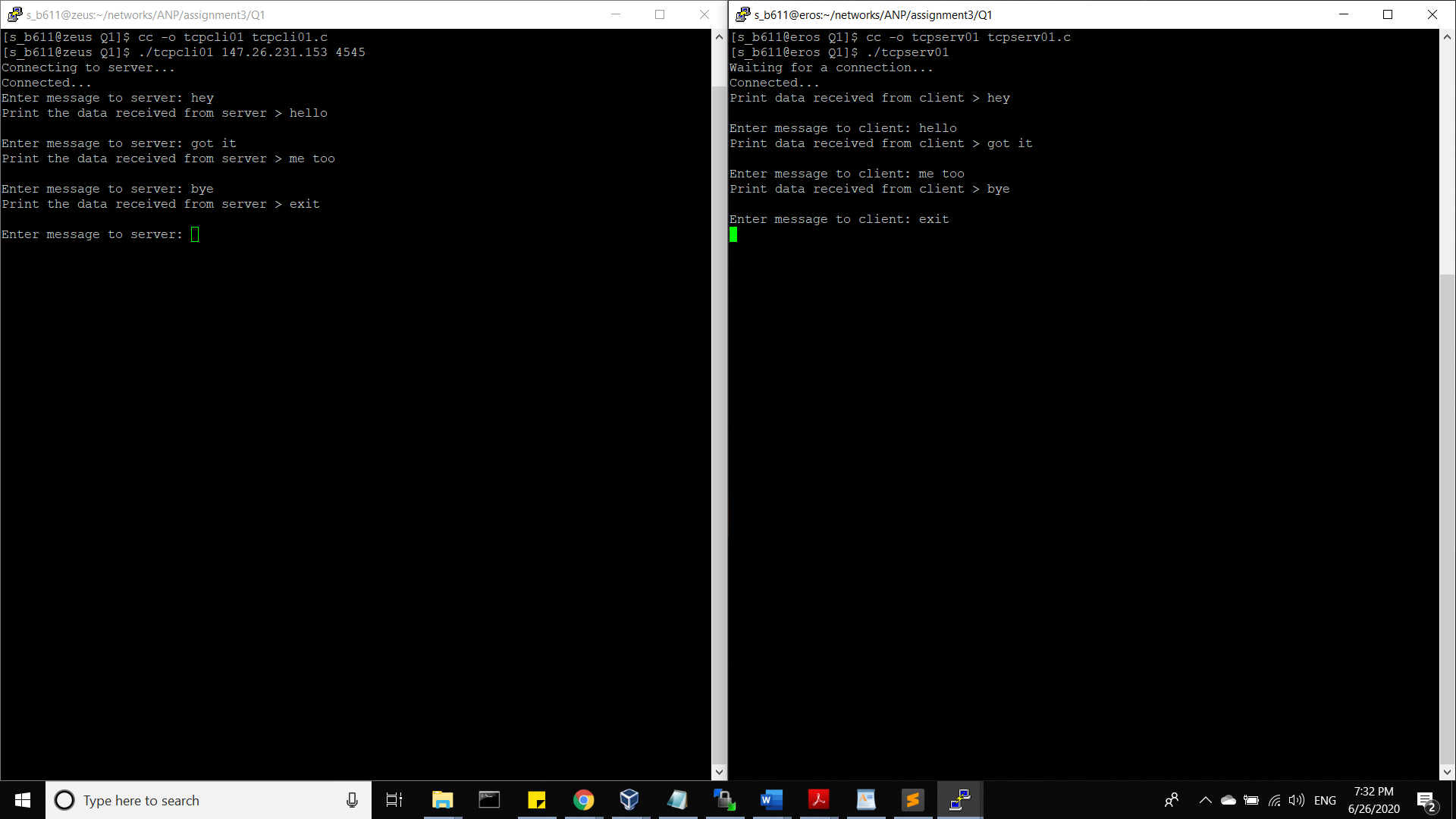
**on zeus:**

cd …/Q1

cc -o tcpcli01 tcpcli01.c

./tcpcli01 147.26.231.153 4545

The client and server communication is shown below:



2. (30 pts) Modify UDP client-server example by using the recvmsg and sendmsg functions, instead of recvfrom and sendto functions.

ANS: I used two codes from chapter 5 for performing this question which are udpcli01 and udpserv01. Source code for this is given in Q2 folder.

The str\_cli and str\_echo functions are declared in the same file independently. Compiling them with library was resulting in segmentation errors so to avoid that it is done this way.

Exicution commands are as follows:

**On eros:**

cd …/Q2

cc -o udpserv01 udpserv01.c

./udpserv01

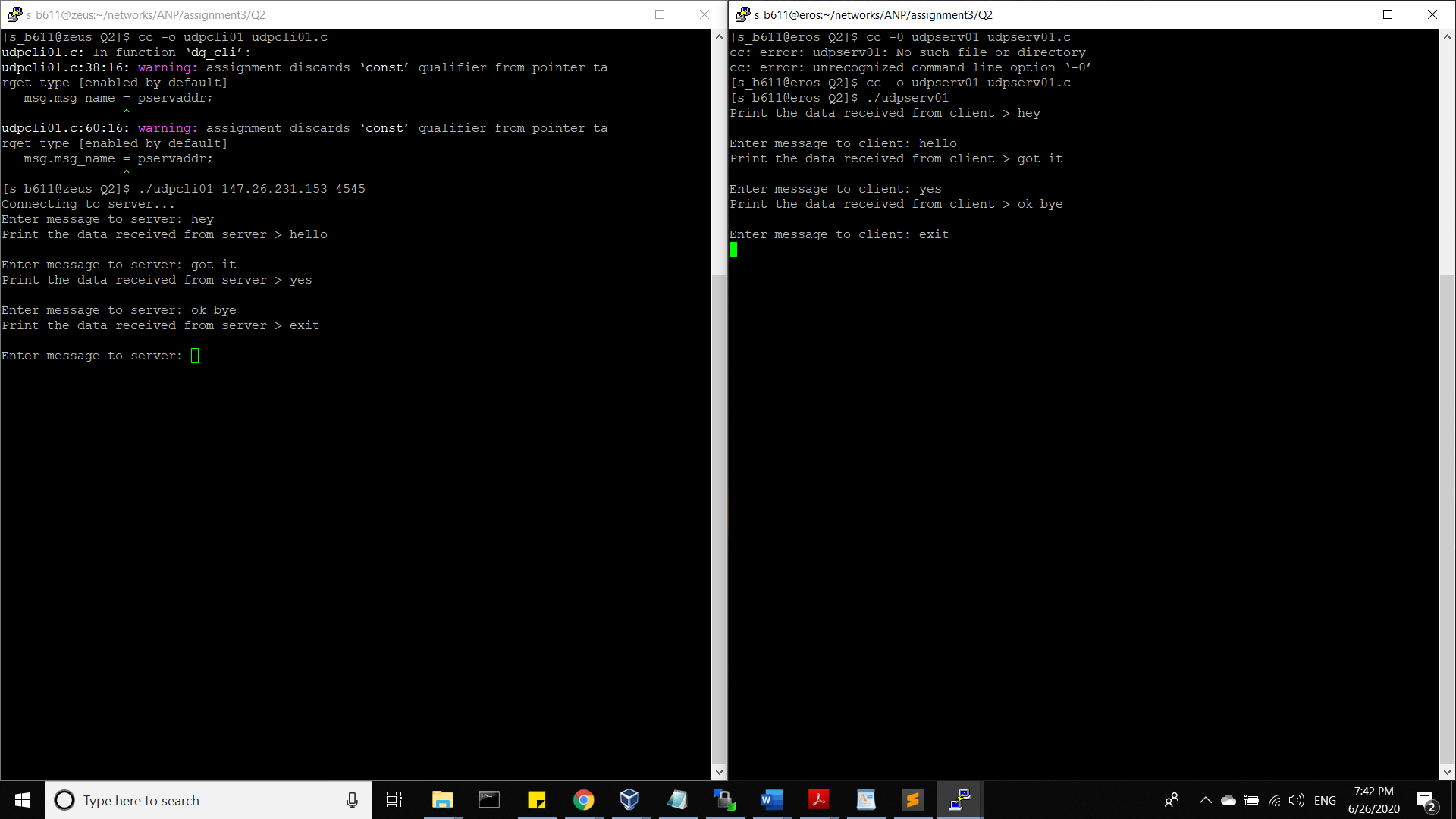
**on zeus:**

cd …/Q2

cc -o udpcli01 udpcli01.c

./udpcli01 147.26.231.153 4545

The client and server communication is shown below:



3. (40 pts) Modify the non-blocking IP str cli function in Lecture 6 by using two different buffer sizes (MAXLINE variable values). Run the program on a large file, say over 10,000 lines (average length of each line shouldn’t be too small. Otherwise the total number of bytes in the file is still not high). Compare the total time used in the two cases and make some observations.

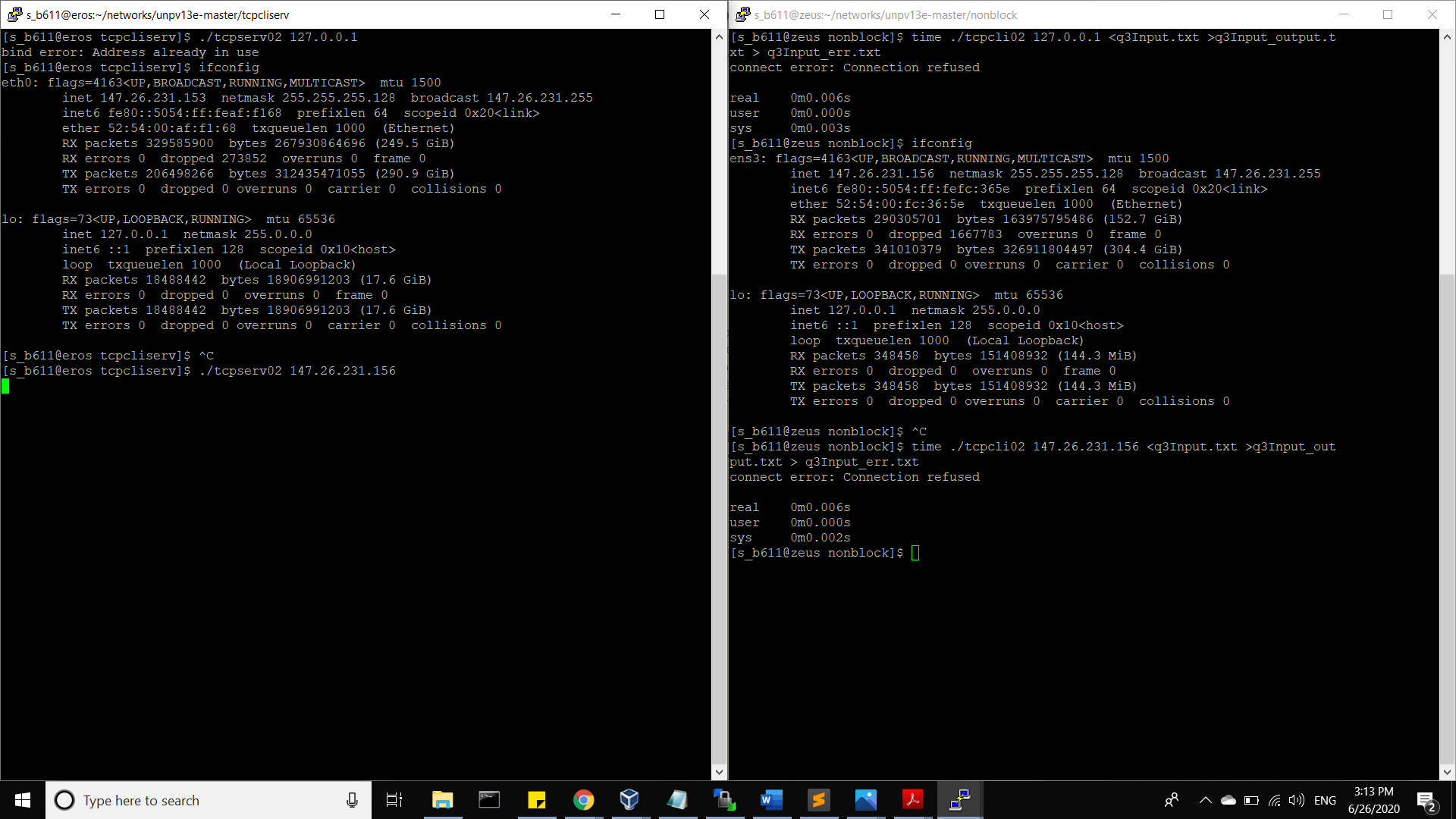
ANS:

For this question I used tcpcli02.c code from unp library/ nonblock folder as CLIENT and tcpserv02.c from unp library/ tcpservcli folder as SERVER.

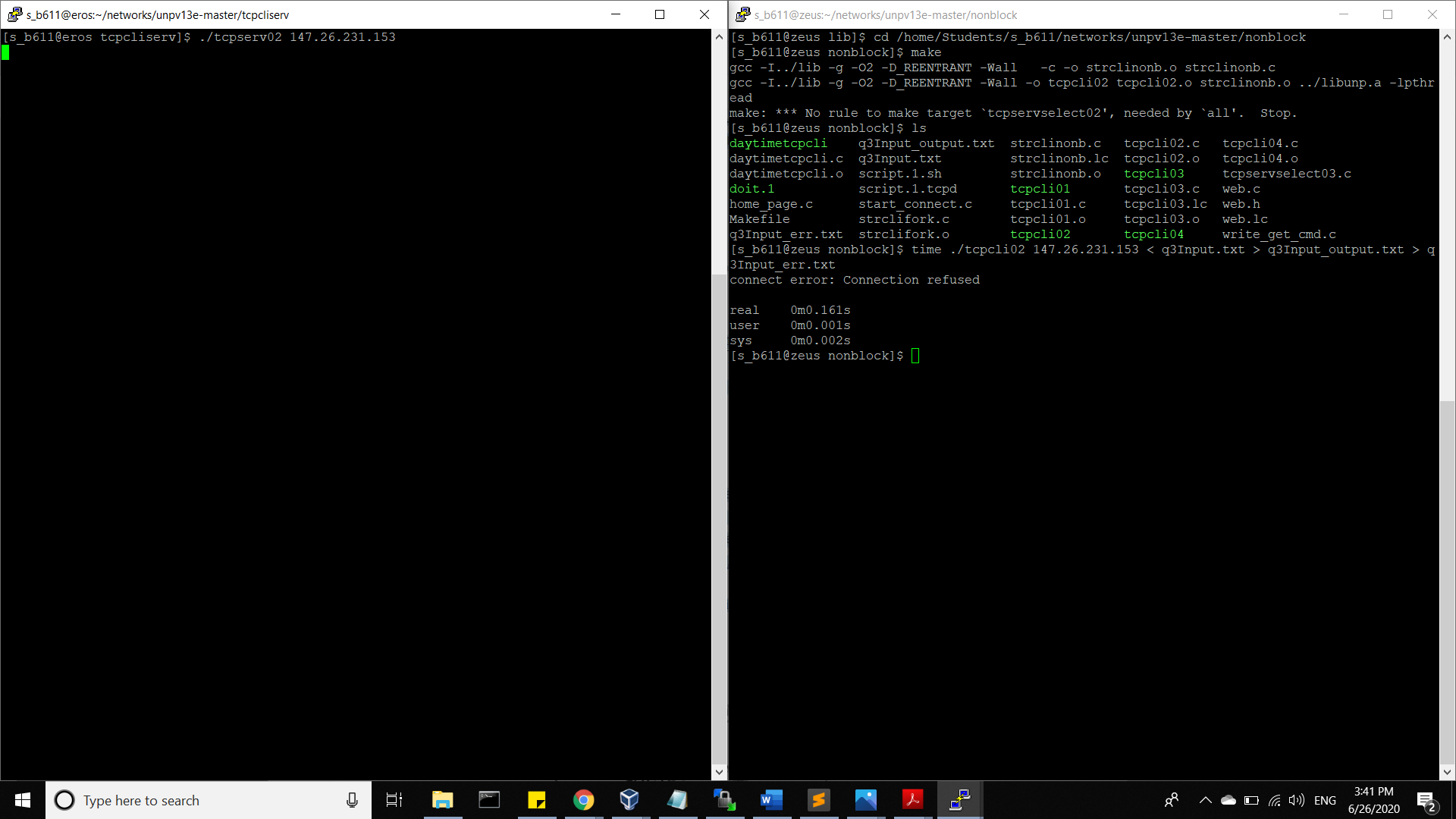
The non blocking str\_cli function is declared in strclinonb.c.

To observe the changes I executed the server client code with smaller sized file and MAXLINE which is the macro used to define buffer size..

Result output is as below:



After changing value of MAXLINE to 10000000 and nearly 2000kb sized input file The changed output observed is as followed:



I used time command to see the execution time.

Size of text file was incleared by last amount of text from UNP book.

Explanation on real, user and sys (from man time):

* real: Elapsed real (wall clock) time used by the process, in seconds.
* user: Total number of CPU-seconds that the process used directly (in user mode), in seconds.
* sys: Total number of CPU-seconds used by the system on behalf of the process (in kernel mode), in seconds.

The changed observed was that real execution time was that in increased from 6 sec to 161 sec. While system time for execution was constant.

It is observed that with increase of buffer size elapsed real time is increase while CPU process time in changed by 1 seconds and processor time is constant.