**CS5341.501/502**

**Summer I, 2020**

**Advanced Network Programming**

**Assignment 2**

Please write a short program that will disable Nagle’s algorithm. The program can be adapted from the client server code in the textbook (such as from Chapter 5, but be sure not to use the fgets and fputs functions as they belong to C standard library and buffer input and output). Describe the behaviors of the program after Nagle’s algorithm is disabled and after Nagle’s algorithm is enabled.

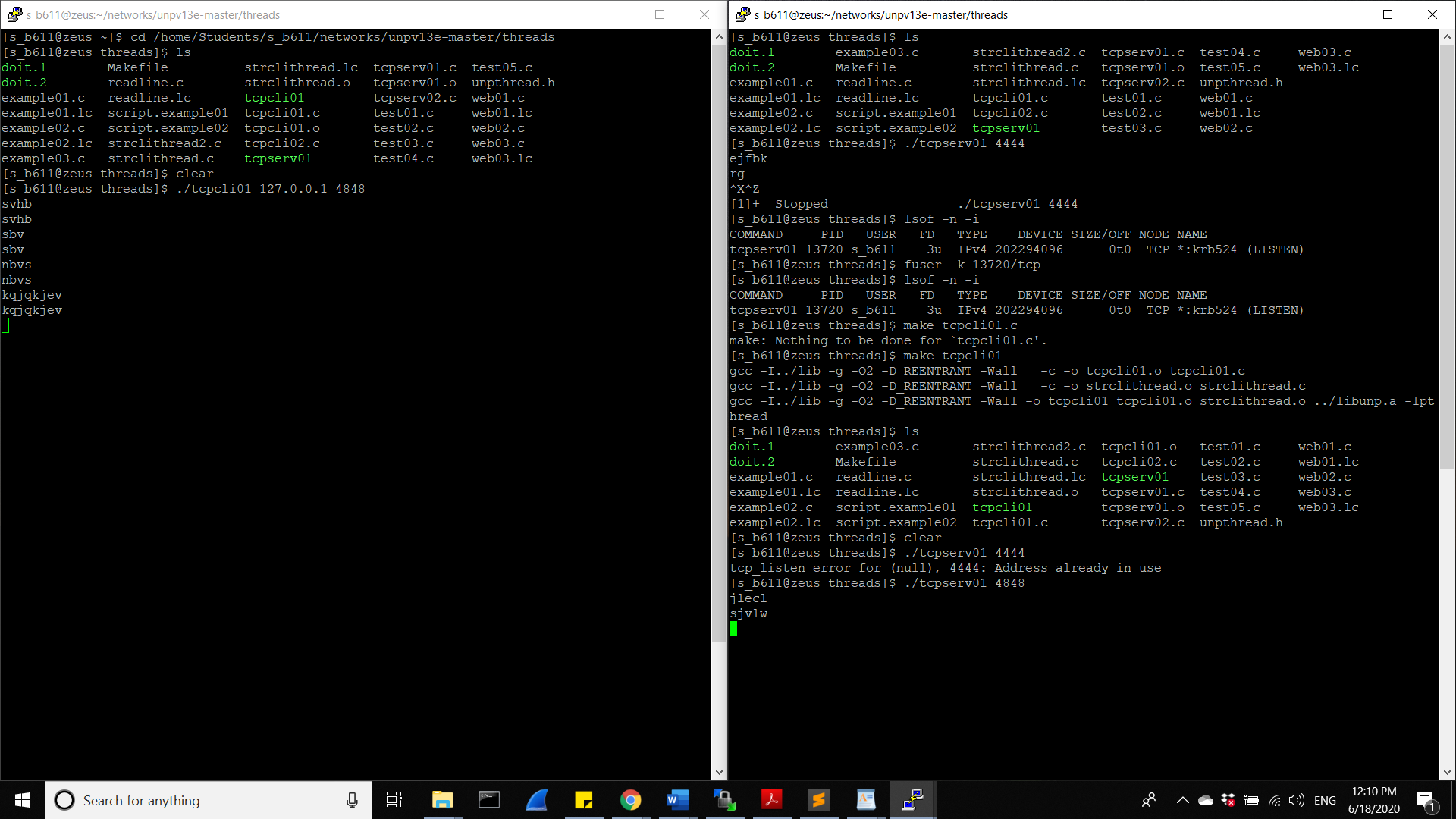
Ans: The purpose of the Nagle algorithm is to reduce the number of small packets on a

WAN and prevent a connection from having multiple small packets outstanding at any time.

These clients need a way to disable the Nagle algorithm, TCP\_NODELAY option is used in this case. The TCP\_NODELAY option is by default set to 0, I used setpocketopts function to set flag to 1 and send characters with write function.

To compile simple gcc command is used and it is exicutes with ./a.out <ip address>

we can notice a delay in the character echoing, and this delay is often exaggerated by the Nagle algorithm.



1. ..

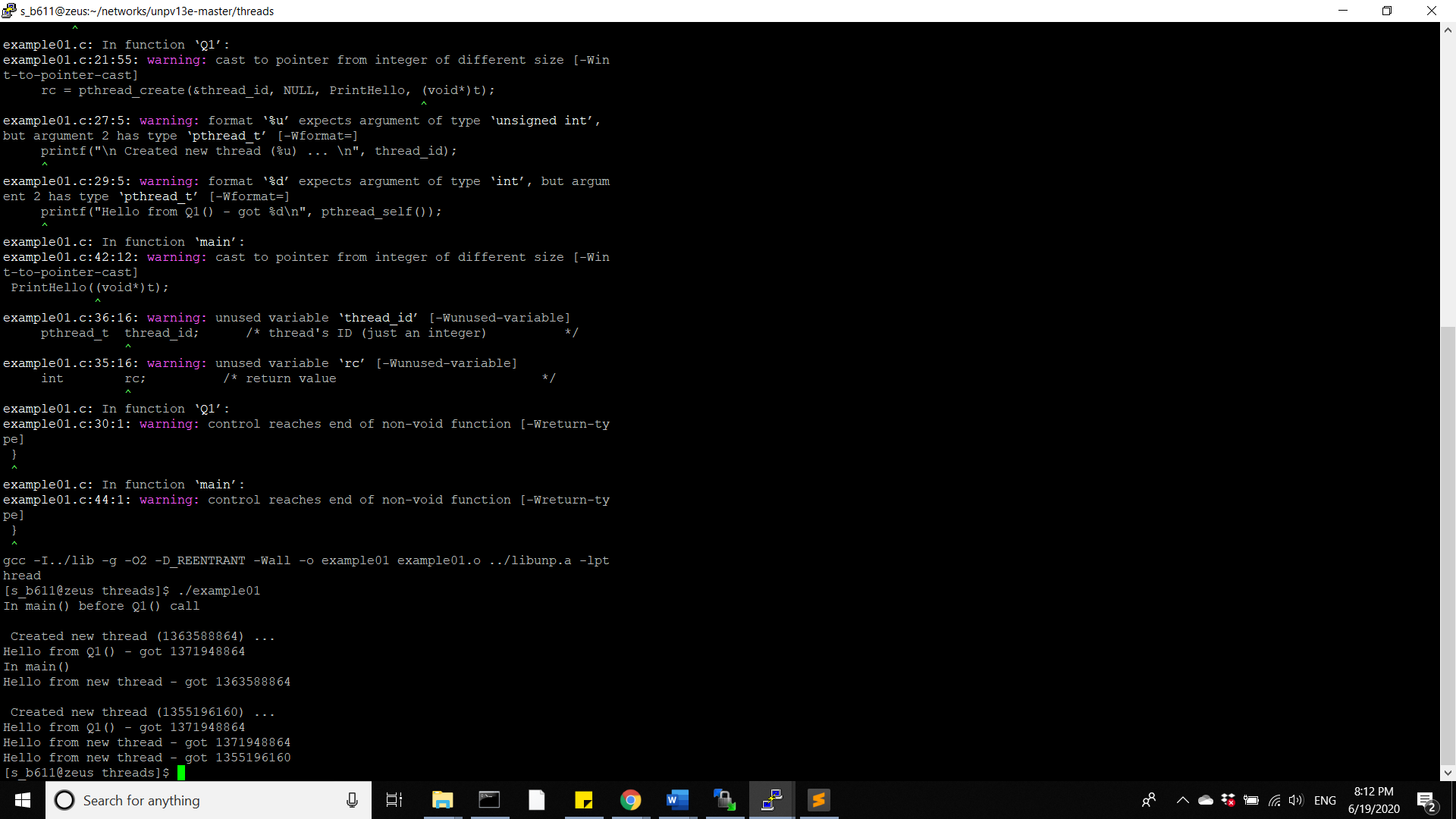
This problem is pertaining to pthreads. You need to write and submit a program to test and support your answer.

ANS: To compile the code copy all the file made for each section into unpv13e -> thread-> folder. Compile them with make <filename> and execute with ./<file name>

1. Can a thread still exist if the thread that creates it terminates by calling pthread exit()?

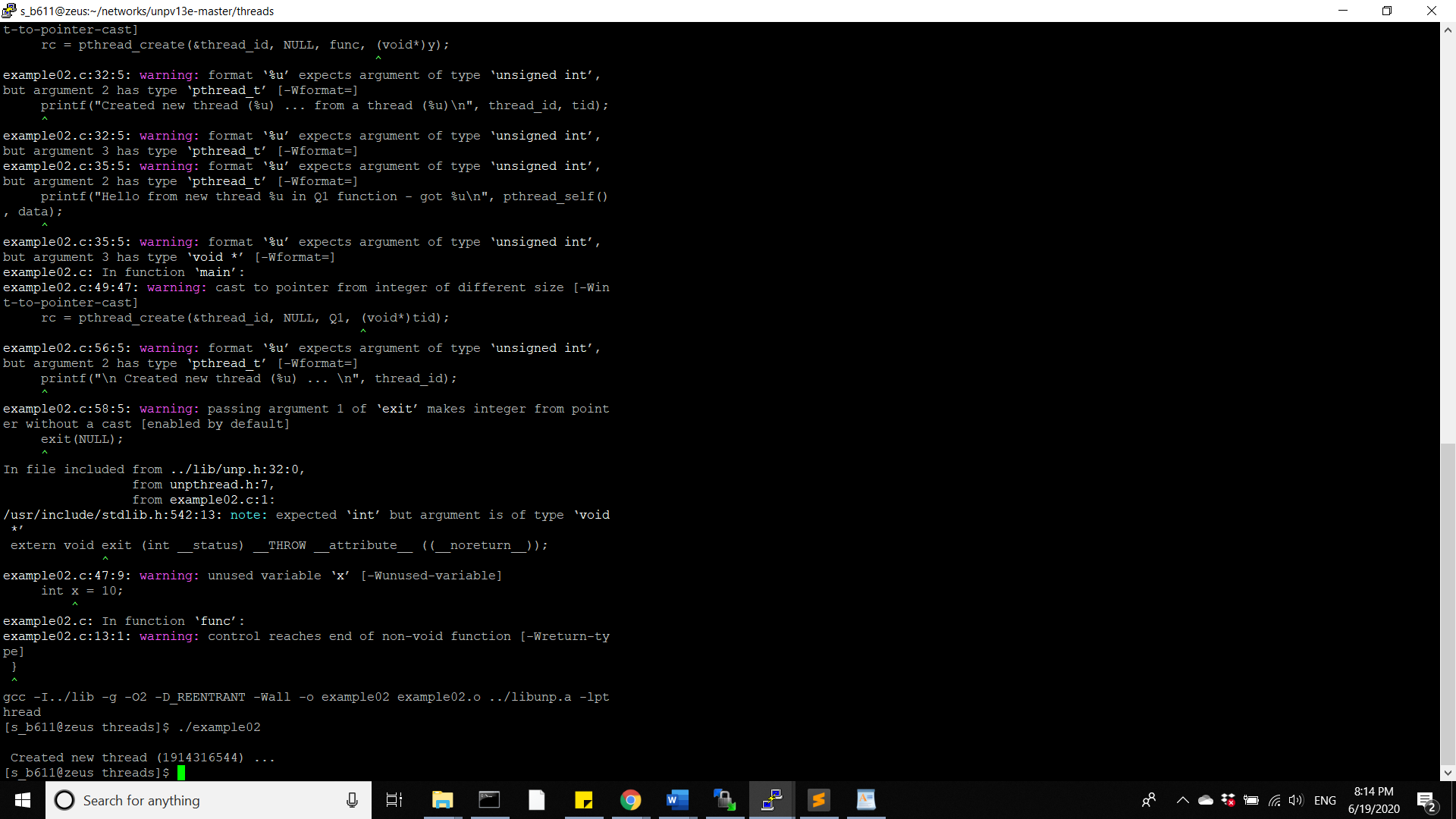
ANS: If main() function finishes before the threads it has created, and exits with pthread\_exit(), the other threads will continue to execute.

Output screenshot:



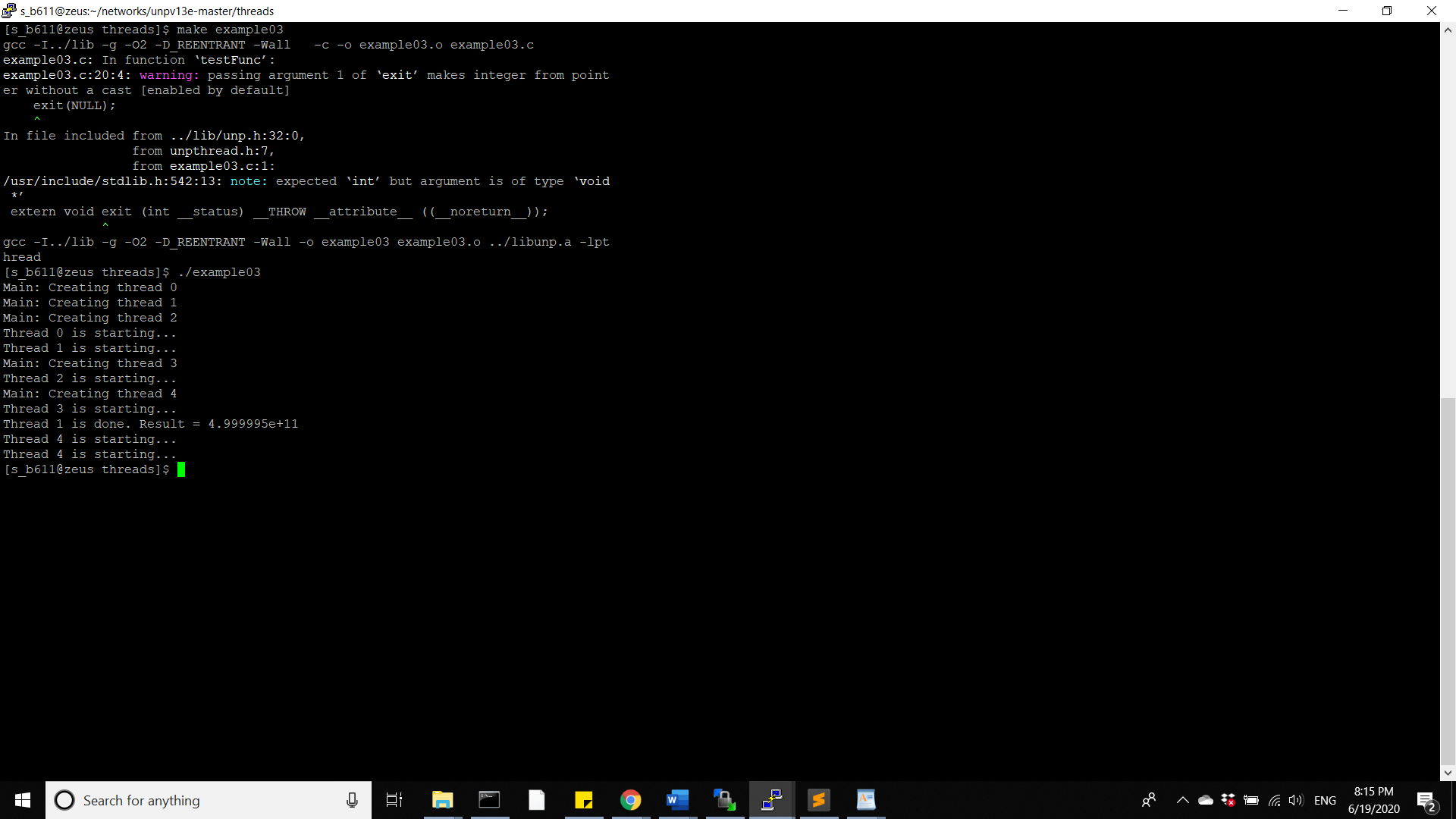
(2) Can a detached thread still exist if the main thread of the whole process terminates by calling exit function?

ANS: When the program terminates, any detached threads that are still running are silently terminated.



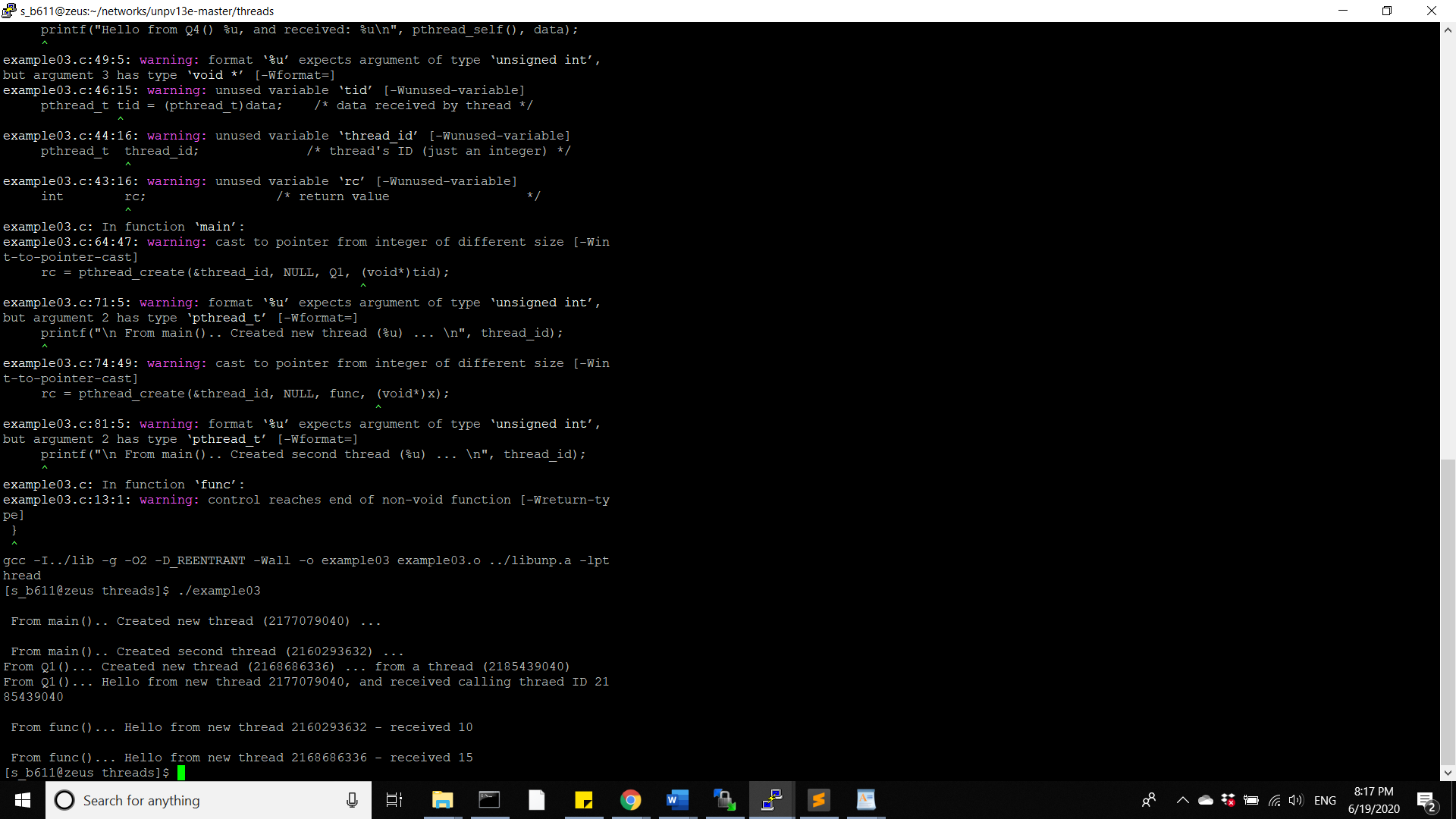
(3) Will the process still continue if one of the normal (not detached) and non-main threads within it calls exit function?

ANS: If any thread calls exit() directly, the process and all its threads and LWPs will exit immediately.



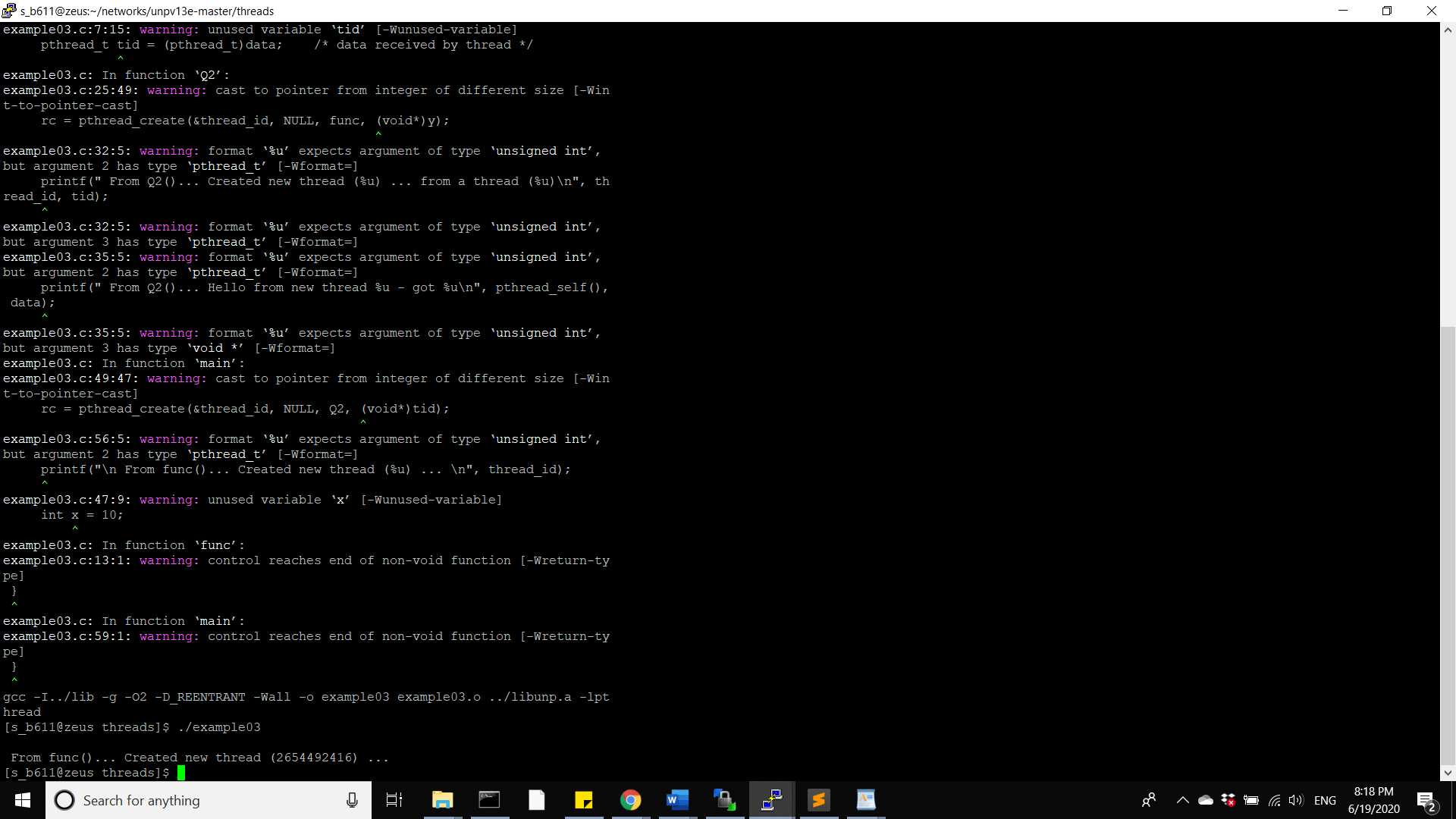
(4) Can the original process still continue (hence all other threads within the process) if a detached thread calls exit function to terminate?

ANS: If any thread calls exit() directly, the process and all its threads exit immediately.



(5) Can the process still continue if the main thread terminates normally (ie the control of the main thread falls off the last statement of the main function)? Can the main thread detach itself?

ANS: If the main thread terminates normally, any other threads within the process continues to execute, No, main thread cannot detach itself.



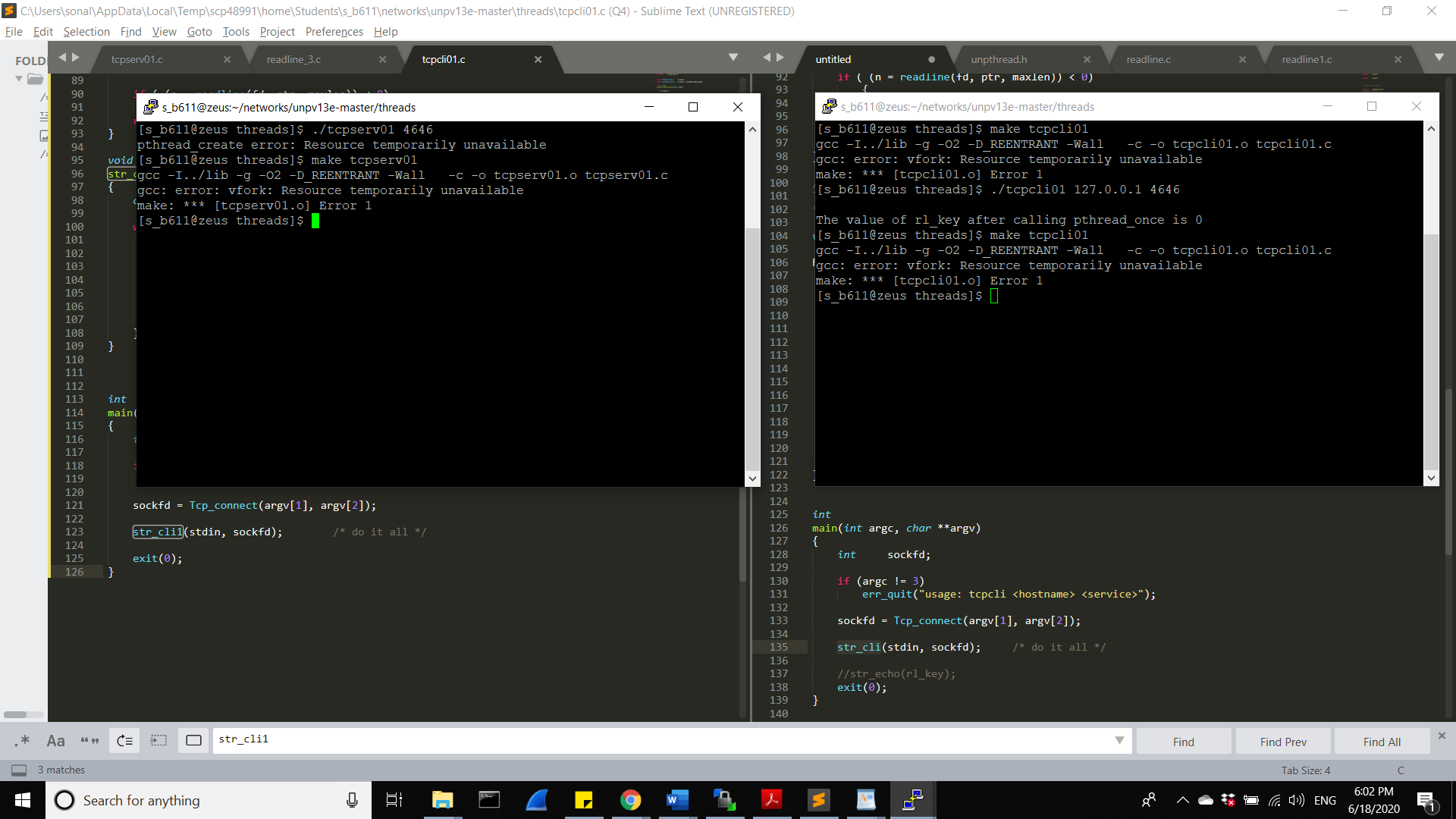
1. ..

ANS: a. Compile and run the thread version of TCP client-server program that uses the thread-safe readline function in Fig.26.11 and 26.12 (p.692 & p.693). Verify (print if possible) the value returned in the variable rl key. Does each thread have the same rl key value?

b. Yes, threads can use more than one thread-specific data item.

The function Print\_Info() is used to explain this concept. It is used to print out the value of the rl\_key. It checks if the value of rl\_key is zero or non-zero and prints accordingly. Thus, we get different values each time we create a data item.

No each thread will use only one thread specific data because our function does not allow you to create rl\_key for each thread.



1. ..

ANS: a) *Pthread\_once* function is used to guarantee that *pthread\_key\_create* is called only by the first thread to call *readline*. This function is normally called every time a function is called that uses thread-specific data, but this function uses the value in the variable pointed to by *onceptr* to guarantee that the *init* function is called only one time per process. Without using pthread\_once, the rl\_key value will be increased.

b) Static type has a broader scope than other types. If it is made no-static, their values may not be set at run-time or may change subtly during program execution.