CPSC 3600 Homework1
Spring 2023 (50 points)

Communication ring with TCP sockets

Instructions:

This assignment is done individually. You *cannot discuss the solution to this programming exercise* with anyone except your teacher or TA. Please review TCP code in Donahoo.

Exercise

You can use the TCPEchoClient.c and TCPEchoServer.c (along with all the file needed to run the program) from Donahoo as a starter code. (NOT the TCPEchoClient4.c version) You can also use code examples with posix threads.

You will create a communication ring with 3 computers that will work as follows:

Computer A will send a file specified on the command line to computer B and later receive a thank you note from B and print it to screen.

Computer B will print the contents of the file to the screen, send a thank you note to A and send the received file to computer C. It will later receive and print thank you note from C.

Computer C will print the received file to the screen, send a thank you note to B and send that file to computer A. Will later receive and display a thank you note from A.

When the file is received by computer A, A will print the file to the screen and send a thank you note to C.

At this point communication is finished.

Please keep in mind that your program should work with any text file, not just the one that was provided to you as an example, so hard-coding is not an option here. The name of the file will be passed as a CLA to A only (not B or C), along with other CLAs: IP address of the party you are communicating with, and port number. (Example of output below)

It is up to you to figure out the details. There are several different solutions to this problem, and everyone's solution may be different. I do recommend that you review how to read from files in C and go over the code line by line following the Chapter 2 in Donahoo.

Before you run your executable, login to three different SoC machines and figure out their IP addresses (ifconfig will work). Tile the three terminals on your screen so you can see all three at the same time.

When two computers communicate in a client/server architecture, their roles are clear (a host is either a client or a server). How would you make three computers communicate with each other, as in this case each can play both roles?

What to submit:

Write a makefile that will compile your code to produce executable named *ring*. Include all C files needed to run your program into the archive named *asg1.tar.gz* and submit to Canvas before the deadline. Your archive should contain no subdirectories or executables, only C files.

Example output of the communication

Computer A:

$ ./ring <B's IP address> <port> filename

My IP address is: <A>
Sending data to <B>
Received a thank you note from <B>
Received data from <C>:

//File contents are printed here

Sending thank you note to <C>
Goodbye!

Computer B:

$ ./ring <C's IP address> <port>

My IP address is: <B>
Received data from <A>:

//File contents are printed here

Sending thank you note to <A>
Sending data to <C>
Received a thank you note from <C>
Goodbye!

Computer C:

$ ./ring <A's IP address> <port>

My IP address is: <C>
Received data from <B>:

//File contents are printed here

Sending thank you note to <B>
Sending data to <A>:

Received a thank you note from <A>
Goodbye!