

This is the specifications for the second CSCI-340 project. You are to design a program that acts as an echo server. Your echo server will receive a character string terminated by the return key sent to the server using network communications from a client using a telnet or ssh session. The server will send back the character string to the client and the client will display the string on the screen. What the client sees is the string typed on the client terminal followed by an echo of the string being displayed after it is received from the server.

Implement your server so that it will receive and respond to strings from multiple clients. Have the server create a thread to respond to each client requests. You will receive more credit if you dynamically create threads for each client than if you arbitrarily create a fixed number of threads. In any case, you should be able to respond to at least three clients, possibly on the same machine as the server, or all on different machines than the other clients and the server.

You will choose a port number for the clients and server to communicate over such as 6001. Since all clients share that port number you must use synchronization techniques to ensure that one client does not interfere with another. You will use POSIX semaphores to provide this synchronization. Implement the buffer to store the string at the server for relay as a single string variable instead of an array of strings or a link list of strings. The string for only one client should be in this buffer at a time. After two minutes of execution, have the server raise a signal that sends the message "MN terminated" to all clients, and also sends the message "MN terminated" to the file /user/faculty/hilzer/csci340/proj2\_output\_MN, where MN is your team magic number.