**Computer Networks**

**Project 2**

I originally had no idea what a c2c server was or even how to look up how to make one. I later came across a “geeks for geeks” webpage that had a chat program that ran via a c2c server. I played with their example for a few days trying to better understand the concepts to this project. Eventually I ripped out the main functionality of their example to have what was basically a c2c server that simply only received commands. I defined a few variables to make coding this much easier to read and faster to implement. I ended up using a 2D array of ‘char’ to handle the address of the person connecting to the device, this made it very easy to scrub through what clients had connected and been saved. I set the first character of each array to an end of string character so that if you were to try and print a client’s address it wouldn’t show anything even if you were missing all but the first character of the address string.

I decided to use a time\_t structure to be able to properly track time on the C99 compiler that the school servers run on. I went above and beyond to handle misspelling by stating “$INVALID ACTION” and having a “$SHUTDOWN” action that would properly kill the server process instead of having to use ctrl+c.

I really struggled with getting the time functions working on this project, thus I never managed to get milliseconds up on the log time stamps. Another issue I ran into was that it seemed that both zeus and eros had firewalls enabled for me to where it was impossible to test more than one client at a time. I know for a fact that the code would support multiple agents in the event that firewall could be disabled, in my case this was impossible since opening ports in the firewall for this project would of required me to have root access.

I have attached one of my log files that shows every action in use with and without the the client being “active”.

Command line for server run:

gcc \*file to compile\* -o \*output name\*

\*filepath/filename\* \*port number\*