Design Documentation

In terms of the file structure for this project, the files used were the provided agent.c file (unmodified) and my HunterNoeyServer.cpp file. In my server’s main function, the backbone of the program is a do-while loop used for handling the agent’s job requests that terminates when an unrecognized command is entered. Before we even access this do-while loop, we open up the socket and bind the sock\_add instance, as if these two functions fail the program will not run. Given these two functions pass, we access the do-while loop and accept the argument sent from the agents, and according to the command sent the program will perform that desired operation.

For each operation, I created a function that performed the necessary functionality with two vectors holding the agent’s data; the first vector contained the IP of the agent connected (a string), while the second vector corresponds to the first vector and serves as a time storage holder (time\_t) that was used for calculating the agent’s time connected when using the LIST function. For the JOIN operation, the vector checks if the agent has already connected (in our case stored in the vector) and stores both an IP address and a timestamp for the agent given it is not connected. If the agent is already in the vector, the agent is not stored. For the LEAVE operation, the vector checks if the agent being removed is in the vector, and if it is in the vector the function removes the IP address from the first vector as well as the timestamp in the same position in the second vector in respect to the IP Address. Once again, if the vector does not contain the agent nothing is removed. For both the LOG and LIST functions, we once again check if the agent is connected to the server, and if the agent is connected the contents are sent to the agent in the form of a char array.