Introduction:

For this assignment we attempted to create a network layer, with both a server and clients using sockets and IPs. The server was decided to have an IP of 127.0.0.1, although running the server executable with a different IP would change the behavior, and variable client IPs. The client was decided to send their IP to the server, and when the server receives a new connection, it sends all IPs to all clients.

Design

The server was decided to contain a list of strings of all IPs, a constant port, a socket, and a lock. The server was given a lock because it was decided there would be a subclass using a thread to wait for connecting sockets. This lock is used when accessing the array of strings so that the array isn't being mutated while the messages are being sent to all clients. When a server is created, it binds the socket at the port / ip combination given. The server can broadcast to all sockets currently connected to it, or send a message to a specific IP address given.

A Client contains an int which is the socket descriptor, constant which is the server IP, and a constant PORT. When a client it initialized with an ip, it connects to the server and sends a message to the server with the clients IP. The client also was decided to have a thread that reads, so that it can wait for messages from the server and write messages at the same time.

Challenges and Open Issues

Currently, our server displays each IP address as the clients name. We think it would be better for a client to be initiated with an IP and an alias, and to replace the string list in server with a string to string map which would map names to ips. We had trouble with the threads, as they produced a segfault for some reason if we didn't join before leaving the function. We also had trouble figuring out how to test the code. We will most likely be adding more threads in the future, most likely having them in the client.cpp file so that a client can send actual messages to the server.