Writeup - Assignment 2 Multi Threaded HTTP Server:

Scott Fischer

Early Design

The design for this assignment was particularly tricky for me when I first started working on it. I found it tricky to keep track of all the data that flowed in and out of every thread in my server. At the start I knew I was going to have a server thread(s) that would do the process that ASGN1 did, but I did not realize how much data needed to be passed in and out of this thread in order to function properly. Then when it came time to logging I wanted to have a seperate thread do this as to speed up response time of the server threads. I decided to create a Linked List that would store all the data that the server threads processed, then the writer thread could iterate throught this linked list and process each byte of data. I think my design is a solid choice for something that can be built upon in the future, but for something that needs to be running within a week it was quite tricky.

When I run the first experiment from the spec there is a definite gain in speed that I see from the multithreaded server. The response is almost simultaneously as they all finish, however with the single threaded one I see that it has to wait for itself to finish the first request then the next and so on. The real bottleneck in my system is the logging of data. Because my algorithm loops through each char of data that is read in the writer thread is easily the slowest part of my system. However becuase this is modularized from the server threads there is no speed reduction to actual data writing, getting, and response time.

The concurrency in my system is a lot. I tried to reduce the critical areas to just writing to a file, and mostly achieved that. The only other critical region is the log queue that I didn't want to be corrupted when one thread loops through and adds a log to it. I don't really see a way to increase concurrency in my program, the only increase in efficiency would be in the small algorithms and the writer thread.

In real life we wouldn't want to log all the data that passes through a server since that would be a huge waste of storage because its basically doubling where you store the data. It would also be a secutiry concern, because if someone got the logfile then they would have the whole server's information.