

**SOFE 4790U: Distributed Systems (Fall 2023)**

**Instructor: Dr. Ahmed Badr**

**Assignment #1**

**Honour code**: By submitting this assignment, I (name and banner id# below) affirm this is my own work, and I have not asked any of my fellow students or others for their source code or solutions to complete this assignment, and I have not offered my source code or solutions for this assignment to any of my fellow students.

**Name**: Matthew Gardiner

**Banner ID#**: 100768198

1. **Application idea**

For this assignment we were tasked with creating an open-ended, multithreaded Server/Client application. Therefore, I created an application that has a calculator, a server local date function, as well as a server sleep function.

When the client first connects to the server, they are presented with four options, 1. Calculator, 2. Server Local Date, 3. Server Sleep Timer, and 4. Exit. Therefore, the Client experience is seamless.

1. **Describe the two core functionalities**

**-Multithreading**

My application uses multithreading to perform several functions at the same time, and accept multiple client requests at once. I did this by first implementing a Socket, connecting a client to that socket, and opening a thread with the socket on a separate class called MainThread. The MainThread class then chooses the correct user choice and opens subthreads to perform the relevant computations. This way, each client obtains its own MainThread and can create its own threads from within, keeping all connections independent from each other.

**-MultiClient**

I won’t go into as much detail here, but the way that I coded the threads in the application allows the application to be multiclient as well without sacrificing too much performance.

1. **Describe the two novel features**

I included two novel features in my application, the calculator and the server localdate/sleep timer. The user must select which function they would like to use at runtime. The calculator takes two numbers as an input from the client and produces the Sum, Difference, Product, and Quotient. Each output is calculated on its own subthread in its own class to achieve faster calculation speeds, but also to separate and clean up the code.

The server localdate/sleep timer is implemented as two separate functions in the application. Like the calculator, the client will be asked to provide their choice first, they can choose 2 for the server local date, or 3 for the server sleep timer. Each function has its own class and is computing on its own thread as a subthread of a MainThread.

1. **Challenges and solutions**

The most challenging problem I encountered was with the DataInput/OutputStreams. I found that many of their functions had been deprecated, causing issues in my application when grabbing/sending user data. I originally tried to switch over to BufferedReader/Writer’s, but I encountered difficulties with the BufferedWriter, so I switched back to Data Streams. Ultimately, the issue was ignorable as any data loss that occurred didn’t seem to affect the results at all.

Another challenging issue I faced was when a Client would close their end of the socket. I found that if the Client terminated the connection on the exit option, the server would begin to spam text in the terminal since it was still trying to write to the output buffer. Therefore, I restructured the while loops in my main server class and discovered a way to return from the loop while closing the socket from the server side, resolving the spamming issue.

1. **Testing**

Describe each test and include screenshots

I have included a video showing the application running and some tests to show its functionality. One test I performed in the video is a divide by zero test for the calculator app. As can be seen in the video, dividing by zero does not produce an error and instead gives a value of -1. Another test I performed was for the multiclient functionality. I opened a third terminal and connected it as a second simultaneous client.