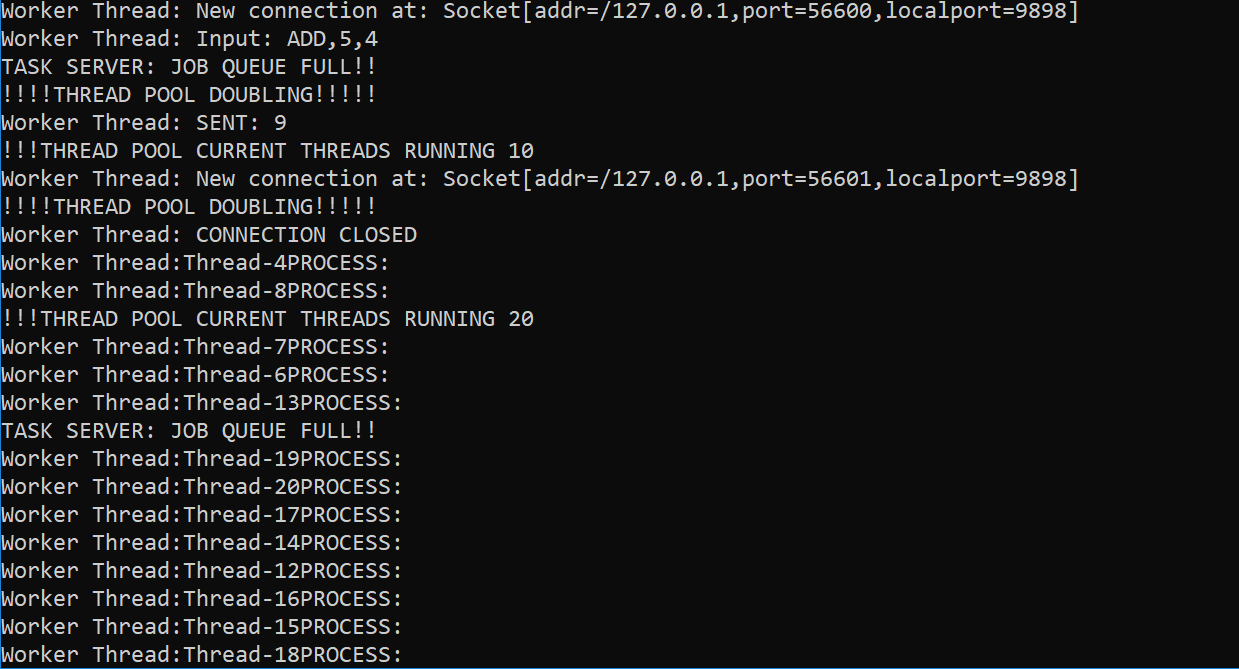
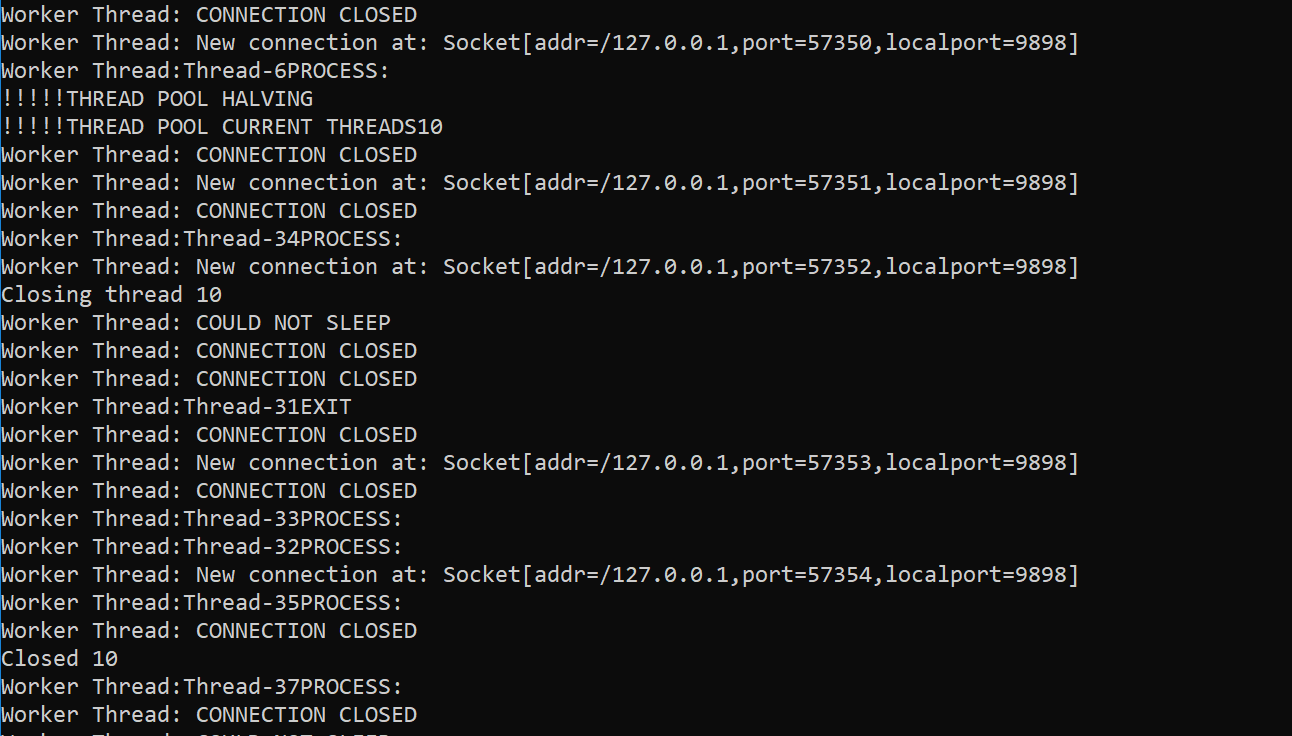
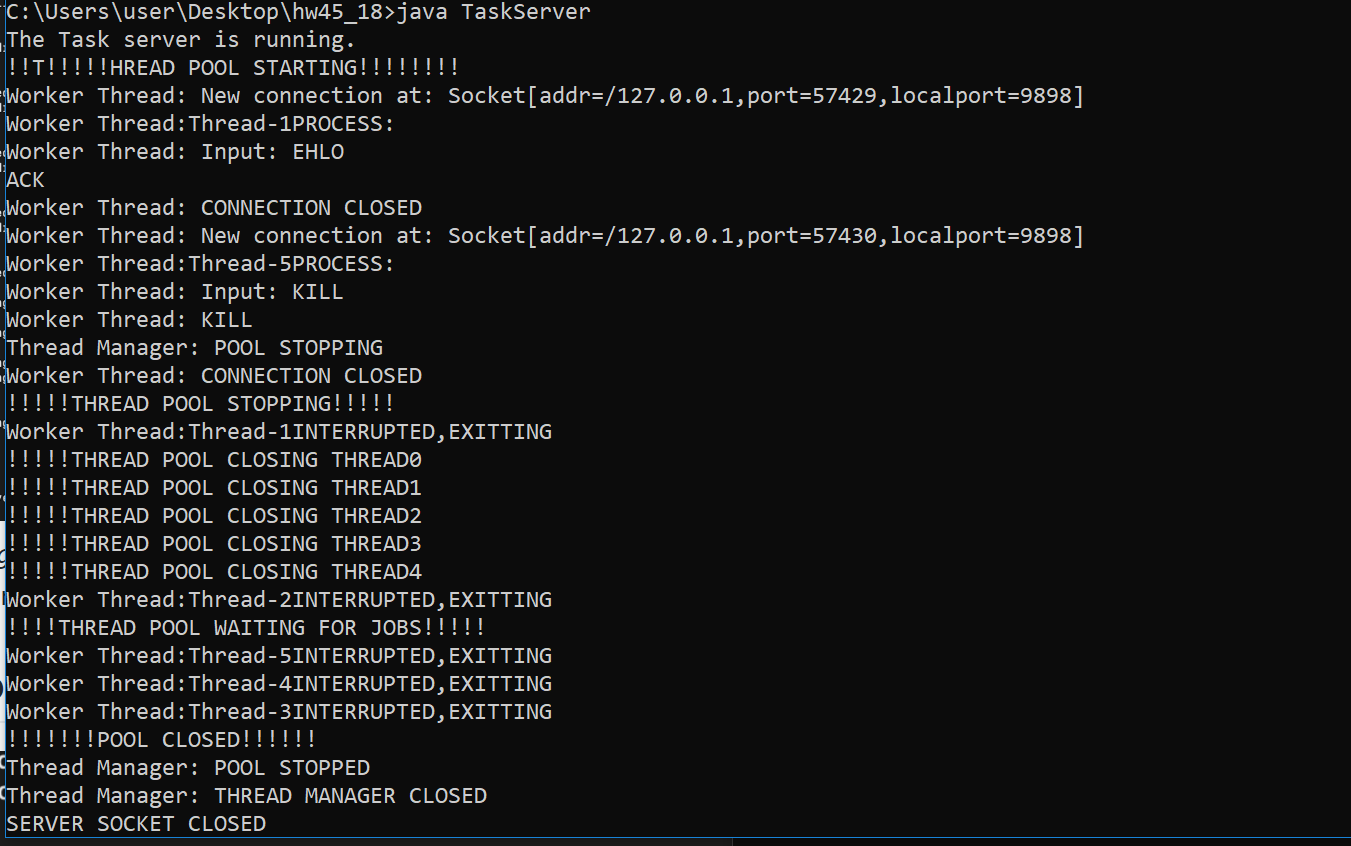
1)Doubling



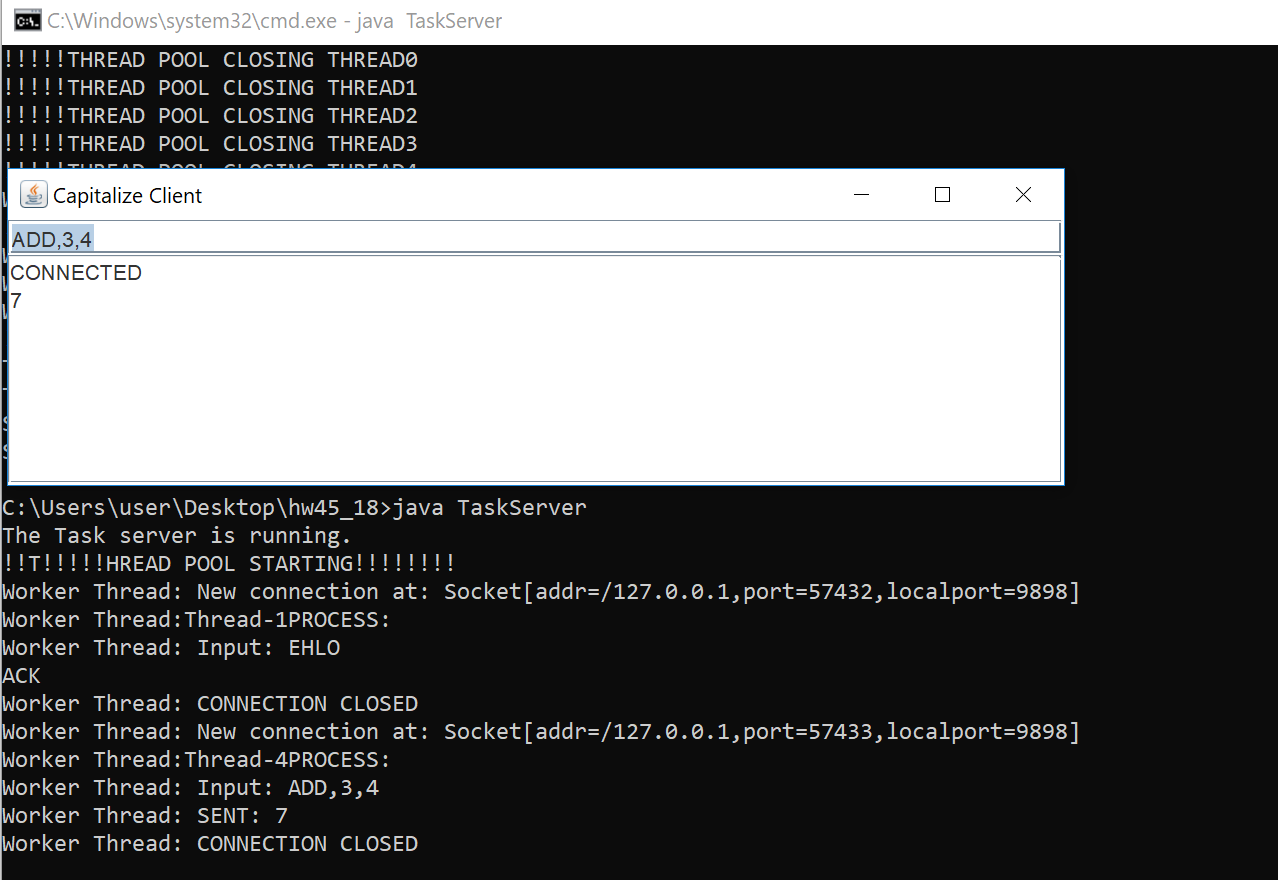
2)Halving



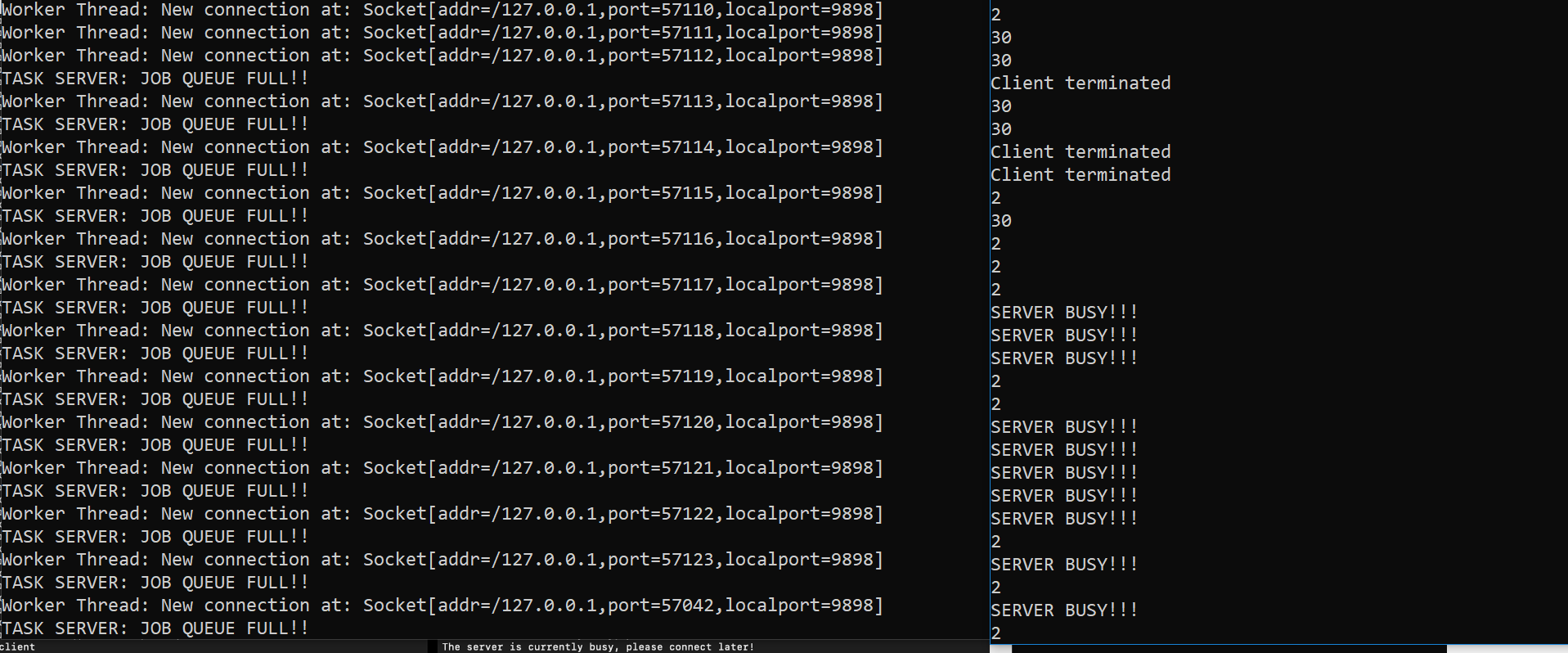
3) Handle Kill



4) Respond Commands



5)Full



Design:

WorkDistributor: Responsible for main server, starting ThreadPool and ThreadManager. Also responsible for accepting connections and job allocation.

Job: Responsible for receiving connection and command from TaskServer. Also does input validation. Will display error otherwise.

ThreadPool: Responsible for starting array of threads. Runs jobs. Scalable.

WorkerThread: As described.

ThreadManager: As described

TesterClient: A non-GUI client that tests a bunch of commands, including ones that don’t work. Automated.

ParallelClient: GUI client that allows for connection to the server and sending of user-input commands.

Issues:

Implementing KILL: I ran into an endless error where whenever KILL was sent, I’d get an exception. Problem was the method would wait for all the threads to die before terminating, causing exception. Fixed it by making it kill the entire server.

General Coding Mishaps: With this project I realized how much more about Java I have to learn. I struggled a lot with regex and general pattern recognition. However I felt okay with the concept of threads, but I think there are still some errors in the program. However as you can see, the program is far from perfect. If I had more time, I would clean up the way the program prints.