Elijah Joshua Mamon
CS-447 Computer Networking
Project 1
20 February 2019

Introduction:

The gist of the assignment was to design a client-server application using a socket interface. This application functions as a basic "scientific calculator" with additional network protocols for added interaction. The main purpose of this assignment is to implement and understand socket programming for UDP clients.

Architecture:

Why I choose Java is because it's relatively user-friendly for socket programming. I used a simple data input and output stream for data communication between client and server. To be very careful, I just went through the interaction step by step and only one communication (both input and output) between client and server were present per while loop cycle. This is to just make it easier for me to keep information in sync between the two.

Multithreading was used to handle multiple clients connecting the server. Each client is binded to their own socket. Anything that had to with data manipulation, any algorithms, and any management was done in the server to make everything consistent and safe.

For the server itself, I relatively stayed away from any big recursions to make everything linear and relatively easy to follow. This is also for me to make sure there aren't any anomalies happening to the connection between the client and server.

For both the client and server, the server port and client information are gathered from the input arguments themselves. This just provides versatility for anyone setting up the servers.

Server Reply Codes:

- 200: Successful Command. Interaction happened in correct succession and no errors occurred
- 250: Correct answer response to a calculator command
- 500: Syntax Error (Unrecognized Character)
- 501: Syntax Error (Invalid Parameter or Arguments)
- 503: Sequencing Error (Command not executed in the right manner)

Calculation Functions:

- POWER <x> <e>: Finds x^e
- CUBE <x>: Finds the cubed root of x
- FACT <x>: Finds the factorial of x

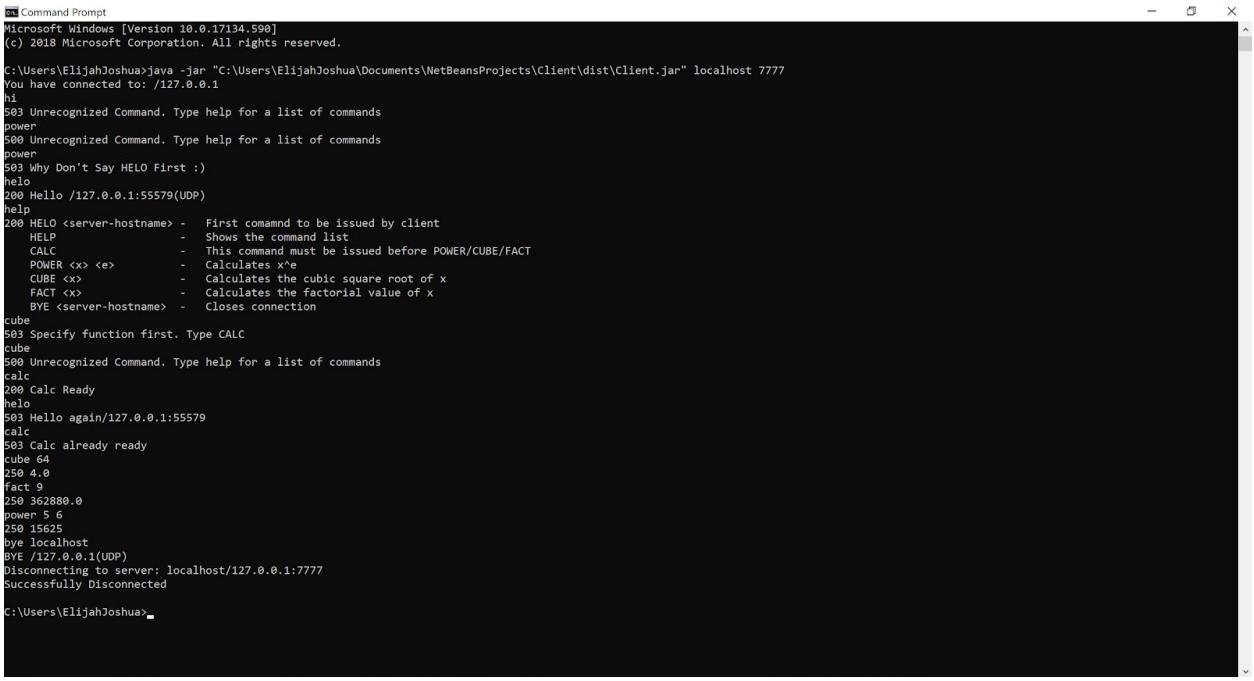
Auxiliary Commands:

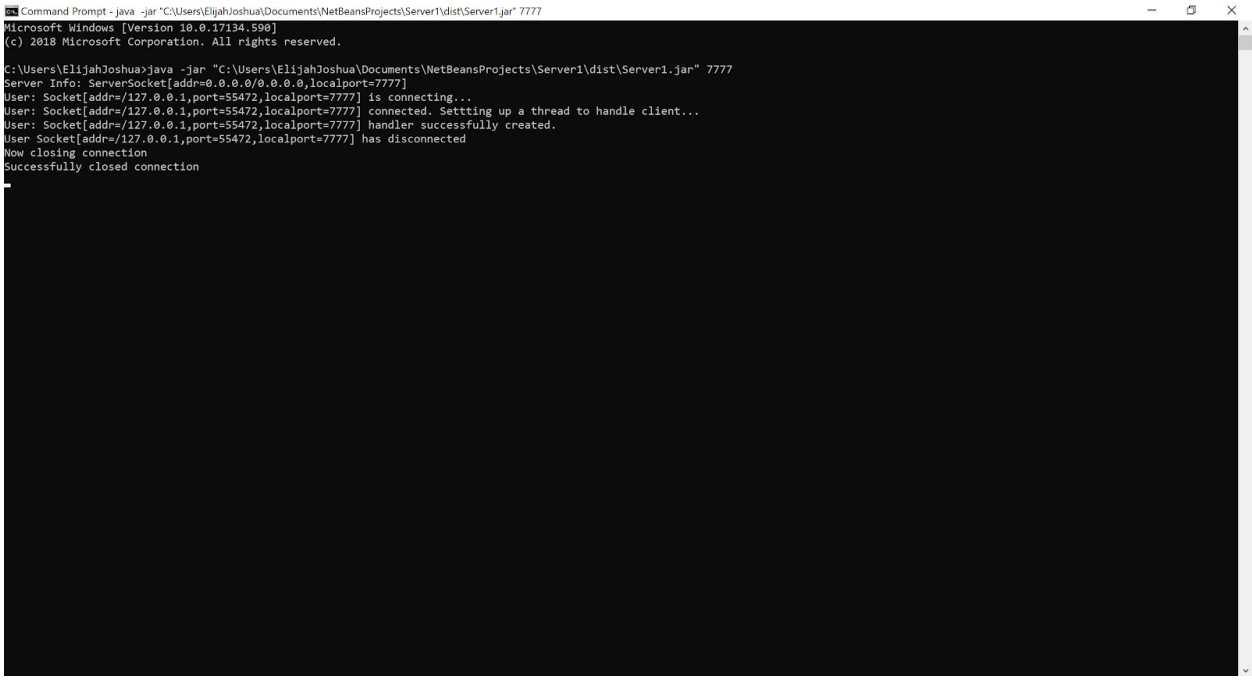
- HELP: Prints the syntax and uses of all commands
- HELO: The initial command to start every interaction
- CALC: "Runs" the calculator function
- BYE: closes the connection

Summary and Difficulties:

Making sure the client was properly bound to a socket became fickle at times. Overall figuring out the design of the server itself took some time. There were troubles in figuring out the storyboard of the interaction of how the users will use the application. By having the program linearly traverse through instructions, it became easier to catch bugs and other problems. Also, since some commands were not four characters long and/or required parameters, it somewhat made it harder because the application would need to know what part of the string is the command and which part of the string are the parameters. This was alleviated by just having a strict syntax rule in which the user needs to follow. In these cases a diagram and/or a graph showing which paths the users will take is very useful.

Demo:




Command Prompt - java -jar "C:\Users\ElijahJoshua\Documents\NetBeansProjects\Server1\dist\Server1.jar" 7777
Microsoft Windows [Version 10.0.17134.590]
(c) 2018 Microsoft Corporation. All rights reserved.
C:\Users\ElijahJoshua>java -jar "C:\Users\ElijahJoshua\Documents\NetBeansProjects\Server1\dist\Server1.jar" 7777
Server Info: ServerSocket[addr=0.0.0.0/0.0.0.0,localport=7777]
User: Socket[addr=/127.0.0.1,port=55472,localport=7777] is connecting...
User: Socket[addr=/127.0.0.1,port=55472,localport=7777] connected. Settting up a thread to handle client...
User: Socket[addr=/127.0.0.1,port=55472,localport=7777] handler successfully created.
User Socket[addr=/127.0.0.1,port=55472,localport=7777] has disconnected
Now closing connection
Successfully closed connection