Thomas Zimmardi

Assignment 4 Write up Partner: Jason Eggleston

10/27/2017

1. You need to new classes so that one class can run the main thread, and another class to handle the sending and receiving of the data. The two classes need to run concurrently.

2.The reason there are two separate classes, and one to handle receiving data and printing it, and the client only receives data from the users and sends it to the server, is so that these things can happen simultaneously, and not be caught up on a command due to it waiting, since the code works sequentially. The class is called a listener because it waits for the client to say something and then handles that command.

3. Each client needs a separate thread in order to be able to communicate, and work with multiple clients. Each client needs a start method and everything included in run() for each instance. The main client method should only be concerned with the main commands and not with the threads. The listener handles input from all clients on a server, the ServerSideClientIO class is only attached and accessed for a single user.

4. broadCast() and remove() are synchronized so we do not broadcast to a user who has been removed. This ensures we don’t have redundancy in our data and does not perform commands out of sync.

5. New Methods that were implemented for ListUsers

To make listusers work, the following methods were added/implemented. userList() is the command that lists the users.

getDataToRecieveFromClient in ServerSideClientIO

This returns the data that is currently being stored, so we can use it across classes.

userList in ClypeServer

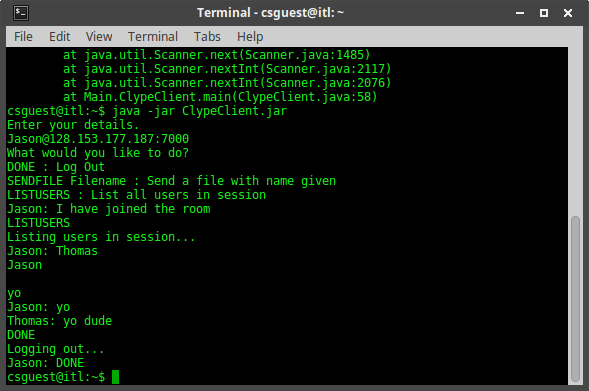
This method prints all users when a user types “listusers”. It uses a for loop to iterate through all elements in the ServerSideClientIOList ArrayList. It returns a string value with all the usernames of clients currently connected to the session.

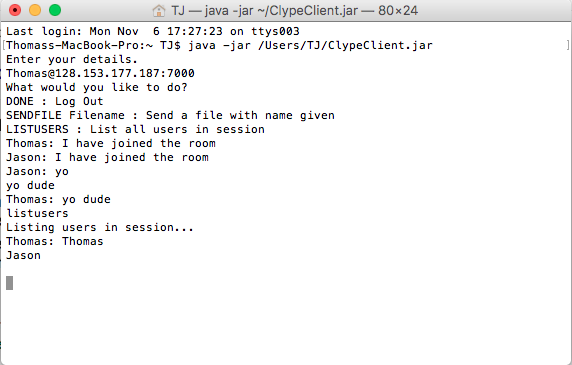
setDataToSendToServer in ClypeClient

This mutates the data that is being sent to server, so we can send over the data that we want when this method is invoked.

**Test 1:**

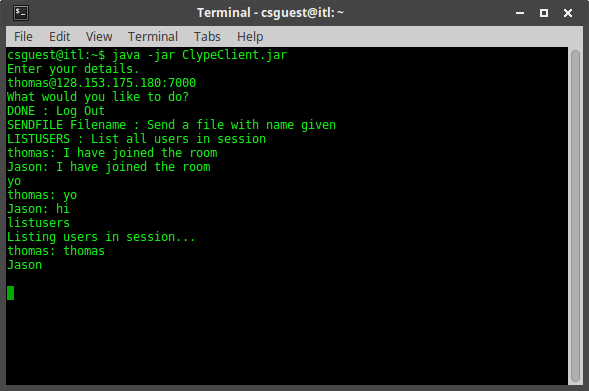
**My ClypeServer on Partners Machine, ITL computer ClypeClient, My Machine ClypeClient**

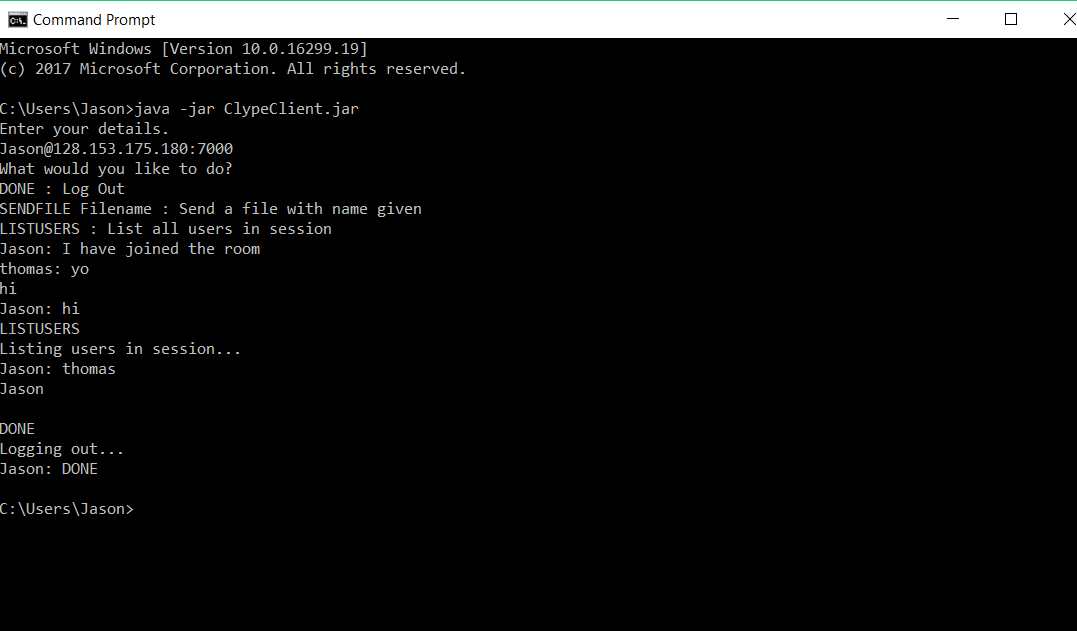




**Test 2:**

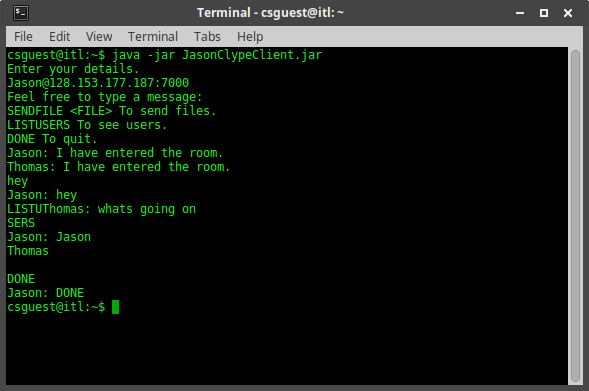
**My ClypeServer on My Machine, ITL Computer ClypeClient, Partners machine ClypeClient**

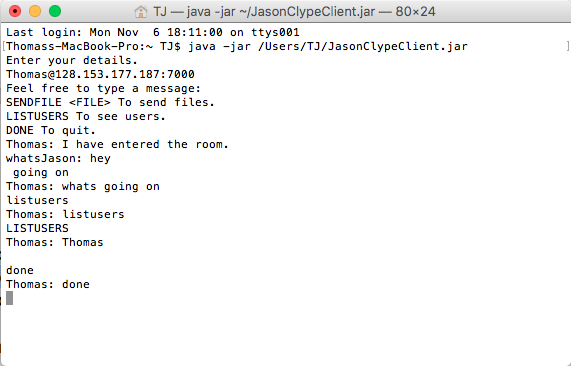




**Test 3:**

**Jason’s ClypeServer on Partners Machine, ITL Computer ClypeClient, My machine ClypeClient**





**Test 4:**

**Jason’s ClypeServer on my machine, ITL Computer ClypeClient, Partners machine ClypeClient**

