Centralized Multi-User Concurrent Bank Account Manager

Abhishek Chintalapati EC82333

- As part of this project, I implemented Bank client-server communication. Here I used TCP protocols for connections.
- The connection between the server and client is through a socket. This is a multiple threaded in a process that is multiple users can connect to the server through the port and their operations in concurrency.
- In My Implementation each thread would be a file with a list of operations, account number, and a name.
- Once client is connected to the server it would communicate by sending the customer's required operation and data. This would be received by the server through the socket and read () from that.
- Once the data is received by the server, it would check for the account number and if its matches it would do the required operation and update as required.
- In this case another server concurrently can connect to the server through the same port and perform its required operations.
- Customer can perform deposit, withdrawal from his bank account. Whenever the account number matched the server's data the server would lock that line in the file and perform its operations and then release the lock.
- To run this, user should the server script and then the client script. Server script expects a port and records file as argument.
- The Client script expects a port number and an operations file that is given by the customer.

Working

- When the records file is given as argument to the server file it would split the strings with Account number, Name, Balance.
- Whenever data is received from the client it would create a thread and then pass it to the test function in the server script.
- In this test function we would check the account number of the user with the records data file. If it matches the data, then lock would be applied till the transaction is completed and then lock is released.
- This is done as a protecting mechanism when the account is accessed simultaneously from multiple clients.
- After the transaction is done, acknowledgement is sent to the client that it is successful with the transaction.
- Similarly in Client the customer’s required operations file, the lines are split and passed to the sever line by line.
- In this way data is maintained consistently securing from concurrent transactions by the client. \\