Sakshi Umesh Reddy

YO19530

* Project Design:

In this project I have implemented a Centralized Multi-User Concurrent Bank Account Manager. The system has two important components.

1. Bank Server: The server program implements the withdrawal and deposit transactions requested by client using threads and lock. It also adds 2% interest in all bank accounts after every 60 seconds.
2. Clients: It sends request to Server to perform withdrawal and deposit transaction on bank account.

* Steps to execute and run the project:

1. Open two windows Terminal-1 and Terminal-2 for Server and Client respectively
2. Compile the code on Terminal-1 by using command: make
3. Run the server on Terminal-1 by using: ./server
4. On Terminal-2 execute the shell script: bash script.sh

* Working:

The server first reads the records from “Record.txt” file and add those records in accnumber[], accname[] and balance[] respectively. After that both server and client create sockets. At server side, I have used bind() and listen() functions and at client side I have used connect() function to establish a connection between them. Once the connection is accepted at server side, a thread is assigned to that client to implement the transaction by calling the handle\_transaction() method and “Transaction.txt” file is opened on client side to send the transaction request to server one by one.

In handle\_transaction(), the transaction request is received from client and then split into accno, operation and amount using strtok() function. After that if the accno is found in the accnumber[] array, the requested operation is performed and an acknowledgement is send to client for the performed transaction.

Along with this server adds interest in all account after every 60 seconds by implementing add\_interest() method.

* Assumptions and Limitations:

Here I have hard coded the port number in server code and used that in client code to connect, because of this if that port is not available at given time the application will not work. I consider this as one of the limitations of my application.

Second, I had to open and close the Records.txt file 2 times, once to calculate the number of records and second time to fetch the records from the file. For this I tried to use seekg() but it was not working properly. I’ll try to improve this in next version.

* Referred links:

1. <https://stackoverflow.com/questions/14287942/cannot-convert-void-networkvoid-to-void-void> - to check how to use pthread but instead I found C++11 std::thread in this link.
2. <https://www.javatpoint.com/how-to-split-strings-in-cpp> - to split the string.