Project : Centralized Multi-User Concurrent Bank Account Manager

Overview:

The project is implemented to handle requests from multiple clients by a server to perform basic operations such as deposit and withdrawal. Along with basic operations server also adds periodic interest in each account. Communication between client and server is established using UNIX-based TCP sockets. Threads are used to perform parallel operations. Race conditions are dealt by using the *'pthread_mutex_t '* locks.

Execution:

Run the *make* file.

Server will start first. Run *./server* to start server.

To start the client, use *./client*.

(Note: If you get 'Binding Failed' error please change the port number in both files.)

Structure:

Project consists of two files, server.cpp and client.cpp.

server.cpp

It is made of three functions, *main(), calculate interest(), operations().*

calculate_interest(), operations() functions are handled with threads for achieving parallel processing. *'pthread_mutex_t '* is used to deal with the race condition which occurs in case of modifying the amount by both thread functions mentioned above.

First step carried about by server is reading the account.txt file and store the details into respective arrays. Then it *creates, binds*, and *listens* the listening socket and *accept* the connection from client. Once the connection is established thread function *operations()* is called with file descriptor of newly connected client. Buffer sent by client is read with *read()* function and string broken into pieces with *strtok()* function. Then account number is compared with the account numbers which are held in array and respective transaction is performed.

Assumption: Server will be always running.

client.cpp

It consists of the *main()* function in which transactions.txt file is read and then with help of a client socket the data from the file is sent to server with *write()* function. Data is sent line by line.

Output

Screenshot 1:

Successful execution with 1 client

Screenshot 2:

Successful execution with 5 clients

Performance analysis:

(Number of accounts: 10, Number of transactions: 10)