David Pham: dhp62
Sam Pak: sp1301

Assignment 4 readme

Our program is built into two main files, server.c and client.c. Using the makefile both of them are turned into executables, server and client. The server is compiled with bank.c which holds the bank structure and bank functions.

Client.c:

Our client connects to the server in the main. Upon successfully connecting the client initializes two threads, one to send input to the server and one to receive output from the server. They begin functions SendInput and ReceiveOutput respectively.

SendInput prompts the user and accepts user input from stdin and writes it to the server, it then sleeps for 2 second to bottleneck the input stream. If the user inputs “exit”, after writing to the server, the thread begins shutting down all processes.

ReceiveOutput reads from the Server and prints out any messages received.

Server.c:

Our server creates a client acceptor thread in the mean that starts at the function ConnectionHandler. This thread will continuously wait/look for incoming connections. Upon finding a client that wants to connect it connects the sockets.

Once a client is successfully connected, the ConnectionHandler makes a new thread, starting at ClientHandler. This receives all the inputs from the client and handles them appropriately, being sure to send a message back regarding the success or failure of the process.

These processes include: Creating a bank, Starting a bank, exiting the program, crediting to a bank, debiting from a bank, checking the balance of a bank, and finish a customer session. The banks are held in a bank array of size 20. They are created linearly from left to right.

Each bank structure has name, balance, and inSession flag variables. All of these 3 items are changed appropriately when properly called by a function.

The ClientHandler threads are held in a linkedlist for access to all the threads.

In the main the server also starts a thread on a 20 second delay to display the current status of all the banks. This is mutex protected and hence stops all the other processes to print the server status. After the processes finish the mutex unlocks.

Opening a bank is also mutex protected and unlocks when the process is finished.

NOTE: we had a problem with disconnecting out server using CTRL-C. The clients would stay connected and the port would remain in use for an extended period of time. We apologize for the inconvenience but for testing purposes please change the port number located in our port.h (line 4) to a new port number and make (make command using our makefile).