README - BANK SIMULATOR

Hunter Betz (hhb8) and Thomas Allen (tja66)

This program is intended to simulate an online bank account manager by allowing users to connect to a server and either open accounts, or credit/debit accounts. Our program is multithreaded which allows several clients to connect to the server at the same time. However, we implement mutexes to protect the information stored in the bank. As a result, only one account can be opened at a time and an account can be accessed only by a single client at any time.

Server.c:

Our server file contains all of the code needed to listen and handle connecting clients. Whenever a client connects, the server spawns a new thread to handle the commands coming from that client. Additionally, our server has two other threads running in the background. One thread is to simulate a server backup every seconds. When the backup occurs, no new accounts are allowed to be opened. The second thread is just a safety precaution to shutdown the server after 5000 seconds. Again, this is only a safety feature for the purpose of this assignment.

When the server adds a new account, it first begins locking a mutex before adding the new account information to memory. After the account has been added, the mutex is unlocked which allows another client to open an account or for the server to backup the data. When the server backs up the data, it locks that same mutex, prints the bank information, then unlocks that mutex and continues waiting for 20 seconds before repeating.

Client.c:

The client program is very simple. It merely connects to the server, and if the client fails to find a connection immediately, then it waits 3 seconds before trying again. It repeats this until it successfully connects to the server. The client program receives the server host name and the port number as command line arguments. Upon connection, the client program spawns two threads. One thread is to handle the commands coming from the user and send it to the server. It waits 2 seconds between commands to simulate thousands of commands being sent to the server. The second thread listens for any messages being sent to the client from the server and prints it to the client window. This thread also handles the shutdown of the client program if it receives a signal that the server shutdown.

ClientHandle.c:

This file is intended to assist the server by tokenizing the client's inputs. The clientHandle.c is used to figure out what type of command the client sent, as well as account names, credit values, or debit values.

Bank.c:
This file is the bank part of the entire program. It contains the information necessary to create accounts as well as debit or credit accounts and return the balance.