Systems Programming Project 4 Readme (Daniel Schley and Ryan Dunn)

For this project, we were assigned to create a banking system that allows any number of clients to connect to it and allows them to credit and debit accounts as well as open new ones. Our initial server loop simply waits on a blocking accept call. When a client tries to connect, it spawns off a thread, gives that thread the client's socket, and continues accepting.

The client thread, when connected, spawns off a thread to take user input and send it, putting 2 second pauses between messages to simulate load. The main body of the client is a reader that waits on messages from the server thread and writes them to standard output.

Once the thread is spawned, the user is in the "outer loop". Here, the only commands accepted are 'open', 'start', and 'exit'. Exit kills the thread and client. Start begins a session with the already-created account specified (if one exists). Open opens a new account with the given name. Calls to open lock the openlock mutex, create the account, and then unlock, making sure that only one client at a time can open an account. Once an account is opened, a user may then start a session with it.

Once a user starts a session with an account, they are in the "inner loop". Once in, they lock the mutex for that account so that only one client at a time can access one account. If the account is in use, the thread sends the user a message that it is waiting and tries again in 2 seconds (Extra Credit!). Here, accepted commands are 'credit', 'debit', 'balance', 'finish', and 'exit'. Exit does the same as it does in the outer loop. Finish exits the session and brings the user back to the outer loop. Balance prints out the account's balance. Credit adds that amount to the balance of the account in use. Debit subtracts the amount, if that amount is less than the account's current balance. Both credit and debit protest if the amount given is either 0 or not a

number. The mutex is unlocked when the user either exits or finishes, allowing another client to access that account.