Multithreaded Bank System

Authors: Alexander Rozenblit & Ryan Mulhall

**bankingServer.c**

Description:

This is the server file .It spawns a session-acceptor thread to catch clients attempting to connect to the server. Once a client connects the session-acceptor spawns a new thread that maintains communication with the client.

The server maintains all the accounts in the bank using a linked list that holds the name of the account, the balance, and whether it is in session. The server receives messages from the client and figures out what changes need to be made to the linked list according that command.

The server also prints out diagnostic data, which consists of the full list of account names, their balances, and if they are in session. To do this, the server uses a SIGALRM which is sent every 15 seconds and a handler which waits until the data is accessible and then prints it out.

Lastly, the server gracefully closes when a SIGINT signal is sent. This is done using a signal handler as well as keeping track of the file descriptor of every client. We keep track of the file descriptors by putting them in a linked list and having the signal handler traverse the linked list and close every file descriptor.

**bankingClient.c**

Description:

This is the client file. Each client spawns two threads. One for communication of the user, which prompts the user for a command and then sends it to the server. The other thread receives messages from the server and prints them to the user. The user entering commands is throttled by 2 seconds in order to simulate server load. The client also makes sure that the commands being sent to the server are valid.

**How to use:**

Use the make file to by typing in "make" to compile both the bankingServer.c to bankingServer and bankingClient.c to bankingClient.

Start the server by running ./bankingServer <PORT NUMBER>

Start clients by running ./bankingClient <MACHINE NAME> <PORT NUMBER>

The client can:

create <accountName> : this creates an account called accountName.

serve <accountName> : this allows the user to deposit and withdraw from <accountName>.

deposit <money> : this deposits the amount <money> into the account in service.

withdraw <money> : this withdraws the amount <money> from the account in service.

query : this returns the balance in the current account in service.

end : ends the current session from the account in service.

quit : closes the client.