Erick Galvez 189005643
Eric Zhang 172005801
Professor Francisco

Asst3 Documentation/ReadMe

How to use:

PORT NUMBER: 169

After running the program, the program will prompt the user to enter a valid server connection every 3 seconds. After inputting the server, the user will now be prompted to enter commands, the first being to create the account. From there, the user will be prompted to enter other valid commands to manipulate their account. Every 15 seconds, the program is locked and all account information is printed.

Methods Per File:

AccountLog.h

Function prototypes of functions in AccountLog.c

BankServer.h

Function prototypes of functions in BankServer.c

AccountLog.c:

Create_Logger:

Creates Account Log

Create_Acc:

Creates New Account, with name, balance, and session flag

Add_To_Log:

Creates log of account names

If_Acc_Exist:

Checks if Account Exists within the log

Find_Account:

Finds index of account within the log

Print_Log:

Prints all accounts in the log

Query:

Sees if session is offline or in session

Deposit:

Adds deposit to account balance

Withdraw:

Removes amount from account balance

BankingClient.c

Main:

This is used to handle all given information, such as address, error checking variables, ip address, user given port number, user given host name, and the actual port number. First, the program checks to see if the client gives both hostname and port number, which following that checks to see if the port number given matches the one on the server itself. Afterwards, a struct is initialized to 0 to make sure everything is cleared, and AF is unspec'd to allow both versions, and a sock_stream is used to accept a tcp port. The address information is set for a given name and the port of specified type by hints and put all of this into the result, connecting to the first available socket given. If unable to do so, the socket is closed instead. Now, the ip address is convert from binary to text and stored. Two threads are created to handle responses from the server and sending commands to the server after checking for sigints, and then both wait until done sending messages before exiting the main method and finally closing the socket.

Get_in_addr

This method gets the socket address of IPv4 or IPv6.

Sigint_handler

Closes socket gracefully in the case of ctrl+c command.

Command_thrd

Allows communication with threads. Takes the file descriptor that allows communication through the socket, looping continuously sent messages to the server until it is given a certain command.

Response_thrd:

This method allows a thread to receive responses from the server by first opening a file descriptor socket'same socket for command' that allows responses to reach the client. Then, it makes sure the receive buffer is cleared before a message is received from the server.

Is_empty

Checks to see if parameter string is empty

BankServer.c

Main:

Creates initial socket, then creates a new thread to check for connection. THe thread will listen and accept clients, then print the account information. From there it will wait for when SIGNIT to shutdown all clients. Then, a new thread is created to find other clients and accept their information. Again, the program will wait until all clients are done before server closes.

Listen_connect:

Accepts arguments to open a socket and listen for sockets attempting to connect to the server. Creates a holder for the ip address, as well as sent messages in a buffer. From there, address information as well as client information are held within structs. Listeners are initialized, and the server address is set to be given information, then moved into servinfo for the final result. Afterwards, we check for results and bind to connection, listening for sockets if found retrieve all finrmation, setting up socket operations to be done, an then to reuse the socket if closed. Sockets are then bound by getting the address and address length. Also made to allow up to 5 clients connected to the socket at once. The connection is continuously listened for, and if a connection is found communication between the client server is begun, and a new thread is created for a specific client.

Client_thread:

Handles client exiting the server, as well as other communication.

Client_communication:

Provides user/client with prompts, and stores, as well as interprets for errors, entered data to each account.

Serve_Acc:

Following the command "end", this method will return back to the communication method

Print_all_accounts:

Prints all account information in the log by calling the Print_Log function in AccountLog.c

Get_in_addr:

Obtains socket address of IPv4 or IPv6

Sigint_handler:

Signal handler

Error:

Prints out error message on when code encounters an error

Is_empty:

Checks to see if parameter string is empty

Fdnode methods:

1. Create_node
    a. Creates a new fdnode
2. Add
    a. Adds a new fdnode to the list
3. Delete
    a. Deletes an fdnode