Jason Byrd

Journal

First, I've got some experience with server/client models, so to begin with I used the code provided to us to create a basic multithreaded server with 2 way communication with a client. After talking with my team, we decided we would split the assignment into 3 parts: the client, the server, and the report. I volunteered to do the server.

In the new thread function, the first thing I did was create the basic loop and switch to handle the code. The first problem I ran into was on the client side, the example given to us used scanf to get the input from user, but scanf only reads one element. I changed it to fgets so it will take in the whole line of input and not just the first word.

Then I started implementing each of the protocols one by one. I decided to handle the input from the server by creating a 2D char array. Each row or the array represented an element sent by the server. I stored account information locally and the biggest problem I ran into was converting char arrays to ints and back. So I made a couple of functions of my own to do it. It also required the making of a power function that worked with int (the basic one in C was giving me problems).

After hearing that we had to have some way of storing data for future use and that SQLite had been installed on the cse machines, I started working on creating a database for the server. I followed a tutorial online and it all went fairly smoothly.

Finally, I began implementing the SQLite in the server. The biggest problem I had that I spent hours on was also the simplest. For the rest of the server I was working in char arrays but to send a command to execute a SQL statement requires a char pointer. Simple enough. I could just make the pointer point to the first array element. So I did the account creation and verification and everything went fine. Then I did the deposit and because it used much the same code, copy and pasted most of it to the withdraw protocol.

I tested them and it looked like everything went fine (the error check I was using reported the SQL statement ran correctly). I moved on to the transactions protocol and after implementing it pulled up the latest transactions and all the deposit/withdraws I had done were missing. I spent a few hours trying to figure out why the SQL statement was reporting it ran correctly but there was no record of it. I finally noticed that there was one crucial line missing. I hadn't set the sql pointer to the first element in the char array that had the SQL statement in it. Apparently the check was right, the SQL statement was operating correctly, it just wasn't doing anything. I finished up the rest of the server with no problems.