

Project 1

Goal

The goal of the first project is to familiarize you with sockets programming in C. You will create a client that sends a string to a server, and a server that responds with a value. After you get this basic operation working, you will enhance the client and server so that two message types are supported as well as adding a timeout function.

Description

Due: Friday, September 20, 11:55pm EST
Points: 10

You are to develop a networked application that takes a text string specifying the name of a bank account on the client side, transmits that string to the server, and then receives a return value from the server that is the account balance (in integer dollars) for the named account. You should support five named accounts: mySavings, myChecking, myCD, my401k, and my529. You should pre-configure your server with balances for each. Make them all different (and positive integers!). Both client and server must use TCP to exchange messages. The client must take the account name as a command line argument as well as the server IP address and port number. Be sure to include error checking in case any of these arguments are missing or incorrectly formatted.

After getting this working, you are to develop a more complex version that supports three message types. The first one, called BAL with usage "BAL [account-name]", is used as in the first part to ask the server for the balance on a named account. The second, called WITHDRAW with usage "WITHDRAW [account-name] [amount]", is used to withdraw funds from a named account. If the named account does not have sufficient funds to complete the transaction your client should print "Error: Insufficient Funds!". The third, called TRANSFER with usage "TRANSFER [from-account] [to-account] [amount]", is used to transfer funds between two distinct named accounts. The insufficient funds error message should also be displayed when the balance in the "[from-account]" is smaller than the "[amount]".

Your client and server will need to be modified to allow these three message types. Your client should take 2 or 4 (in the case of TRANSFER) command line arguments in addition to the server IP address and port number.

Finally you will protect your accounts by adding the following constraint. Your server should not allow 4 or more withdrawals **during any sliding window of one minute**. If a new withdrawal request violates this limit, respond with "Error: too many withdrawals in a minute!" instead of completing the withdraw on that account.

To get you started, you are provided with a code framework. These files include client.c,

server.c and a Makefile to get things compiled. Note that this work must be done on a Linux/UNIX machine.

The first part of the project (simple request/response) should not be difficult. The second will be slightly more difficult. Do not wait until the last minute to work on it. Note that students should use the "Shuttle" servers here at the College of Computing. More information can be found [here](#).

```
ssh <gt_username>@cc-shuttle1.cc.gatech.edu
```

Submission Instructions

Students are to turn in all their files as a single tarfile, submitted directly to Canvas. The tarfile should be named as follows: your_last_name-proj1.tar.gz. To make a tarfile, I would do the following on the command line:

```
tar zcvf liu-proj1.tar.gz client.c server.c Makefile liu-output.txt
```

As shown above, students must also turn in a file containing sample output from their program (last_name-output.txt). Specifically, students must enter a sequence of BAL, WITHDRAW, and TRANSFER requests to different accounts and show that you may not withdraw more than your account balance or more than 3 times in a minute for an account.

Remember that the Academic Integrity policies outlined both in the Student Handbook, the course syllabus and the first lecture are in effect. This is an individual assignment. **Students detected cheating or collaborating will immediately be referred to the Office of Student Integrity.**

Credit: this assignment is adapted from Professor Traynor