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 22nd, 11:55pm EST
Points: 100

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 four named accounts: mySavings, myChecking, myRetirement and myCollege. 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 two message types, one called BAL used as in the first part to ask the server for the balance on a named account, and the other called WITHDRAW 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!". Your client and server will need to be modified to allow two message types. Your client should take two command line arguments in addition to the server IP address and port number: one that specifies the type of request (BAL or WITHDRAW) and the other that gives the named account.

Finally you will protect your accounts by adding a timeout. Your server will need to track the number of WITHDRAW requests to each account every minute. If the number of requests for an account exceeds 3 per minute respond with "Error: Account Timed Out!" 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.

Submission Instructions

Students are to turn in all their files as a single tarfile, submitted directly to T-Square. 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 saunders-proj1.tar.gz client.c server.c Makefile saunders-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 and WITHDRAW 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.