Due Date: 10/19/2020

Upload to Canvas

CS5133: Data Networks

Project-1

TCP Client Server Programming

Fall - 2020

100 pts

1. Objective

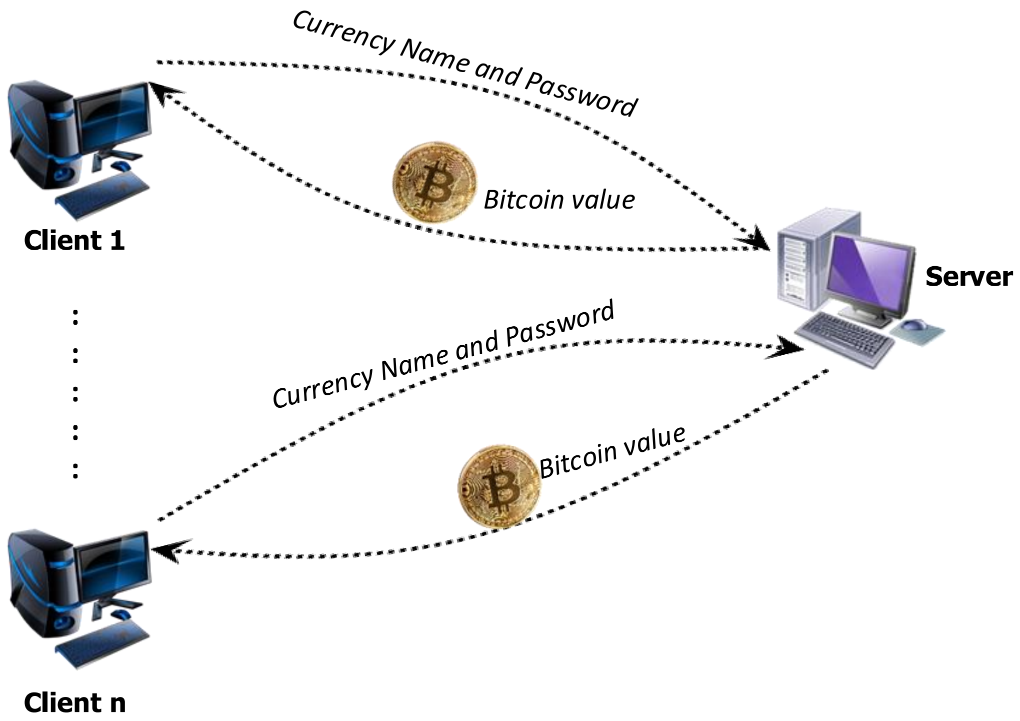
The objective of this first programming project is to learn TCP iterative client-server interaction using socket interface in C programming language. After completing the project, you will have a basic understanding of the steps required to develop a networking application.

2. Project Specification

In this project, you are required to do socket programming in C programming to implement a pair of client and server that can achieve simple password verification to indicate the Bitcoin value. Your client will send a pair of *currency name* and *password* to your server and your server will verify whether the pair of currency name and password is legitimate or not for retrieving the Bitcoin value. *Allow up to 3 clients to simultaneously connect to the server.* Assume the only legitimate pairs of currency names and passwords that will be accepted by your server are as follows:

Currency Name	Password	Bitcoin value
US Dollar	uCh781fY	11081.00
Canadian Dollar	Cfw61RqV	14632.87
Euro	Pd82bG57	9359.20
British Pound	Crc51RqV	8578.96
Japanese Yen	wD82bV67	1158748.55
Swiss Franc	G6M7p8az	10100.44

For example. If the Client sends a pair of currency and password values as: "**US Dollar**" and "**uCh781fY**", the server retrieves the Bitcoin value as *11081.00*.



Your client and server programs entail to achieve the following requirements:

1. Your client program needs to take two arguments that specify the name of server and the port that it is trying to connect to. Your program for server needs to take an argument that specifies the port that it is listening to. You can use (*5000+last 4 digits* of your student-id number) to avoid requesting same port by multiple students.
2. The server program will start first and keep listening to the specified port. Your client will connect to the port that your server is listening to, and a socket between your client and server is constructed.
3. After successful startup, the server program will ask the clients to join the server and maximum number of clients (i.e., 3) can join the server for retrieving the Bitcoin value.
4. Your client program will first prompt a welcome message that asks the user to enter a *currency name* using the keyboard. This currency name will then be sent to the server. Then, your server, after receiving the currency name from your client, will send an acknowledgment message to the client.
5. Your client, after receiving the acknowledgment message from your server, will prompt a message that asks the user to enter the corresponding *password*. This password will then be sent to the server. Then, your server, after receiving the password from your client, will verify the received pair of currency name and password against the list of legitimate pairs. If the result is positive, the server will send a success message along with the current Bitcoin value for the currency to the client. If the result is negative, the server will send a failure message to the client.
6. Your client, after receiving the result message (i.e., Bitcoin value), will print out the result. Then, the client sends "**QUIT**" message to close the socket. Your server will close the socket following the client and keep listening for the next client request.
7. Your server will close the socket after waiting for the *currency name* or the *password* for 30 seconds.

3. Programming Notes

I suggest you start modifying the TCP server program and TCP client program given in the *Project 0*. Your server program needs to handle currency name and password for authentication. I would recommend you to store the currency name, password, and Bitcoin values in the TCP server program as an *Array of Strings*. However, other forms of key-value pairs, such as Dictionary, etc. are also accepted.

File names:

Make sure you follow the file name guideline given below for your project:

lastNameP1Client.c, and lastNameP1Server.c

4. Points Distribution

Components	Points
Client Program	35
Server Program	35
Program Style (Coding style, comments etc.)	10
Documentation	20

5. Submission Instructions:

This project requires the submission of a *soft copy.*

Soft Copy (Due October 19, 2020, 11:59 pm)

The soft copy should consist of:

- source code of the Client program,
- source code of the Server program,
- any header file(s), and
- detailed documentation should consist of:
    - discussion of your problem-solving approach
    - detailed analysis of data structures, algorithms, and user define functions
    - any legitimate assumption(s) with justification, and
    - screen shots of outputs

These must be submitted through Canvas (http://canvas.ou.edu).

6. Late Penalty:

You have to submit your project on or before the due date to avoid any late penalty. **A late penalty of *15% per day* will be imposed after the *due date***. After one week from the due date, you will not be allowed to submit the project under any circumstances.

Good Luck!!