CSE3461 Mini-Lab: Build Your Echo Client-Server Communication using BSD Sockets

Due Date for Electronic submission: Tue Jan 31, 2017 (23:59:59 PM)

1 Goal

In this project, we are going to develop a TCP client and a TCP server in C/C++. You need to build your own socket programming.

2 Lab Working Environment

Students should use Linux OS to finish the project (and all the labs in this course). You are free to use your own Linux OS (suggestion: Ubuntu 14.04+). It is recommended that doing the projects or at least testing them in this provided environment, because:

- C/C++ compiler (gcc, g++) and BSD socket library on those workstations have been verified.
- Your submitted code will be tested and evaluated on them.

Note: in most cases, your Linux OS should be able to finish this project since C/C++ complier is widely installed. If you have environment issues, please contact the grader right away.

You need a plain-text editor to create the source code. You can use vi/pico/emacs in Unix/Linux systems. Since C/C++ is a cross-platform language, you should be able to compile and run your code on any machine with C/C++ compiler and BSD socket library installed.

Windows workstations are **NOT** recommended. Also, the grader **WILL NOT** be able to help you on any of these platforms.

3 Instructions

This is an **individual** project. You must work alone, not with your team members or anyone else.

1. Learn C/C++ socket programming carefully. Examples of socket programming Client/Server code in C are available on the course website (ClientServer_Example.zip). They are also put in the folder `~/cse3461_fall15/mini-lab/`.
2. Instead of using port 80 or 8080 or 6789 for the listening socket, you should pick your own to avoid conflicts. It is suggested not to use port numbers 0-1024.
3. The project consists of Task 1, Task 2 and Task 3, which are specified as following:
   - Task 1: Implement an Echo client-server. On the client, you will be prompted to input a message, and the message will be sent to the server after clicking "Enter". Upon receiving the message, the server will send the message back, which will be displayed on the client terminal. The sever will record and display the message on its terminal, too. be allowed to input Whatever sent from the
   - Task 2: Based on the code you have done in Task 1, add one more function to process a special message. The message is "HELLO" and it is not case sensitive. Upon receiving this special message, the server will send the following message "Welcome!", which will be displayed on the client side.

- Task 3: Based on the code you have done in Tasks 1 and 2, add one more function to process a special message. The server will not tear down the connection unless the client sends the message is ”Bye” (which is not case sensitive). Upon receiving this special message, the server will tear down the connection. In other words, the server will the client to send any number of messages from the client and echo each message (if not ”welcome”).
- After you're done with all tasks, you need to test your server and client. You need to open two terminals: one for client and the other for server. Run your server like this:
  > yourserver *PortNumber*
  and run your client like this:
  >yourclient *serverIPAddress PortNumber* Note: if the server and the client runs on the same host, *serverIPAddress =127.0.0.1*.

4 Project Submission

You should submit the following materials.

1. Your source code files (e.g. server.c, client.c)
2. A Makefile. The grader will only type “make” to compile your code, make sure your Makefile works at stdlinux.cse.ohio-state.edu.
3. A readme.txt file. It should not exceed 1 page (except from source codes). In this file, you should include a brief manual to explain how to compile and run your source code (if the grader can't compile and run your source code by reading your manual, the project is considered not to be finished). You need to include your name of the course and project, your name and student ID at the top.

   You do not have to write a lot of details about the code, but just adequately comment your source code.

To submit,

1. Put all your files into one directory.
2. Zip or tar all the files in this directory and get one submission file, which is named “minilab_name_SID.zip”, where name and SID are your name and student ID.
3. Submit this zipped file to Carmen.