Jordan University of Science & Technology
Department of Network Engineering and Security
NES416- Network Programming
Programming Assignment 1

Due Date: see course E-Learning site

Description:

Use TCP socket programming in C to implement a pair of **iterative** client and server programs. The server waits for client's request which represents an arithmetic expression, evaluates the expression, and sends the results back to the client. The client asks the user to input an expression and sends that expression to the server for evaluation. Once the reply from the servers arrives, the client displays the result to the user and asks for another expression, and so on. Note that the connection between the client and server should stay open, so that the client can repeat the operation again until the user ask for termination by entering the word "**exit**".

At the very minimum, the server should be able to handle addition, multiplication, subtraction, and division operations on two integer operands. You may also assume that any input (operands) might be separated by and number of spaces. So, some sample test cases are:

- User types: 1 + 2, server replies 3
- User types: 2 * 3, server replies 6
- User types: 4 - 7, server replies -3
- User types: 30 / 10, server replies 3

The client program should use command line arguments to read the IP and Port numbers of the server. In addition, the client must use bind() system call to assign the local port ***55abc*** for its connection, where ***abc*** is the least significant 3 digits of your students ID. For example, a student whose ID is 12345 will have the clients source port = 55345.

Furthermore, the server should use command line arguments to indicate the port it will listen to. Once the client prints the received replyfrom the server, it also prints the IP and port number of the server. Also, once the server receives the clients requests, it is locally displayed together with the client's IP and port number

Output Sample:

For example, running the code should produce something similar to this output:
Client_side> please enter your expression:
1 + 2 (input from a user)
Client_side > 1+2 = 3 (***server IP , Port#***), where 3 is the answer received from the server
Client_side > please enter your expression:
exit (input from a user)
Client_side > exiting (→then the client side exits, and also the server side)

Server_side > waiting for client messages.
Sever_side > received "1 + 2" from the client (**IP address, port #**)
Server_side > Sending " 3 " to the client

Server_side > waiting for client messages
Server_side > received "exit" from the client (**IP address, port #**)
Server_side > exiting (→ then the server exits)

Hints:

- ☐ DO NOT use the header file “unp.h” from the book
- ☐ Each student should use a port number for the server such that is larger than 45000
- ☐ Your program for client needs to take two arguments that specify the IP address of the 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 (the same one provided to the client)
- ☐ You must use the bind() function on the client, as specified in the HW description
- ☐ Ask questions as early as possible.
- ☐ Submit your source code for both the client and the server, and some running sample of your code as one zipped file whose name is your student ID number
- ☐ Your programs should be compiled and run without any single error or warning.
- ☐ **Comment and error-check you code**