CSC 4200/CSC5200
Computer Networks, Fall 2018
Programming Assignment 1 (Ping Pong)
Assign Date: September 11, Due September 27 1:30 pm

For this assignment you will write a library for a server and client to communicate using socket. The client will send null terminated ASCII strings to the server using TCP and the server will reply with the same null terminated string unless the string “PING” is sent, in which case the server will reply with the null terminated string “PONG”.

The library will contain the following:

Data structures

- A structure or class of type ConnectionInfo that contains the connection data. The structure of ConnectInfo is left up to the implementer and is not specified.

Functions

- **int sendMessage(ConnectionInfo *con, char *msg)**
  - This function will send the message to the server.
  - ***msg*** is a null terminated string to be sent to the server and it's length may be arbitrarily long and is **NOT** limited to a few kilobytes.
  - ***con*** contains the information required to send to the server. Note that this does mean that if there are multiple ConnectionInfo data structures this may not be the same server between calls.
  - If the send was successful return 0. If it was unsuccessful for any reason then a non 0 value is returned. **msg** or **con** being null is a sufficient condition for failure.

- **char * recieveMessage(ConnectionInfo* con)**
  - **con** contains the information required to receive from the server. Note that this does mean that if there are multiple ConnectionInfo data structures this may not be the same server between calls.
  - This return value is null if an error has occurred. Otherwise the returned value is a null terminated string that contains the entire message and thus must be able to hold a large amount of data (more than a few kilobytes) and is most easily accomplished by using the heap to allocate the memory. This memory should not be modified by any further actions in the client library before dealocate_message is called on the returned value.

- **void dealocate_message(char * mem)**
  - This function will deallocate the memory returned by recieveMessage. This may be as simple as a wrapper to c's free function or c++'s delete. If the value passed in **mem** did not come from recieveMessage then the results are undefined.

- **int connect_to_server(char * who, int port, ConnectionInfo* con)**
  - This function will connect to a server and initilize the con data structure so that it can be used in future calls to sendMessage.
  - **who** is the name of the host as a null terminated ASCII string.
  - **port** is the port the server is running on.

- ↘ **con** will contain after this call, if it is successful, the required information to be used in the sendMessage and recieveMessage functions.
- ↘ If the connection attempt was successful 0 is returned. If the connection attempt was unsuccessful then a non 0 value is returned. **who** or **con** being null is a sufficient condition for failure. In the event of a failure the state of the system and the arguments is such that another attempt can be made, for instance with a new port number.

- **int run_server(int port)**
  - ↘ **port** is a port number to bind to.
  - ↘ If the bind was successful then the server is now running on a new thread and 0 is returned. If it was unsuccessful a nonzero value is returned. After an unsuccessful it must be possible to reattempt the bind and have it succeed if the operating system will allow the program to bind to the requested port number.
  - ↘

A sample server and client code which will use your library is uploaded on ilearn. You can test your library with this code.

*** Make sure that you do not have int main() in your library.

Deliverables
The library must have it's interfaces defined in a header called "pingPong.h". The name of the library source file should be pingpong.cpp/pingpong.c. The programing language used must be C/C++ and the networking code must be using linux system calls such as connect and accept and be written by you. The library and any dependencies used must compile and be usable on shell.csc.tntech.edu.