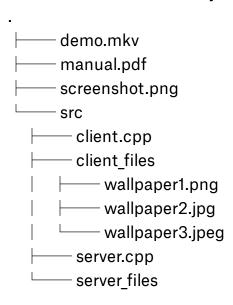
Socket Programming - Practical 1

Akshansh Bhanjana - 11805

Anveshan Lal - 11808

Given below is the directory structure:



3 directories, 8 files

Source

server.cpp

Contains the following functions:

- CreateSocket() → creates a socket which uses the TCP/IP protocol suite's IPv4
 protocol
- BindSocketToAddress() → binds the given socket to a specified address in order to be able to listen to incoming connection requests to the address
- **GetConnectionFromQueue()** → listens for incoming connections at the given address; accepts in case of a connection request
- ListenForMessages() → continuously checks for changes in the buffer so as to monitor incoming messages, and sends back response codes

client.cpp

Contains the following functions:

- CreateSocket() → creates a socket which uses the TCP/IP protocol suite's IPv4
 protocol
- ConnectSocketToAddress() → connects the socket to the given address and port combination
- SendMessages() → waits for user input, and sends the entered messages / files to the server

Screenshot

The left side shows the server receiving messages / files, while the right side is the client sending messages / files, and getting back response codes.

Client Files / Server Files

The **client_files** folder contains some sample files for sending to the server. As soon as one sends a file, it should reflect in the **server_files** folder.

Usage

- compile the server.cpp and client.cpp files (preferably using g++)
- run the server compiled file
 (if everything goes well, the server should be listening for connections now)
- run the client compiled file
 (if everything goes well, the server and client should both be connected, and the
 client should be asking for user input)

The **client** accepts 3 kinds of input:

- plain text → write any plain text message from the client side, and the server should receive it, and send back a response code
- 2. file → write file:: followed by the file name (with extension, sample file::wallpaper1.png) you want to send (the file should be present in the client_files directory)
- 3. "end connection" → write this message as plain text (without inverted commas) to end the connection between the server and client, and exit safely

NOTE: It might take some time to reinitialize the server after exiting, as it takes some time for the system to verify that the port is no longer in use.