LAB 1 – Socket Programming: Simple Chat

**Overview**

Objective is to implement a chat server using socket programming.

**Steps**

Client file:

1. The first part of any socket programming is to create the socket itself.
2. The client socket is created with a socket() call. The socket() function returns an integer. In the socket call, we specify the following parameters
   1. Domain: IPv4 (AF\_INET)
   2. Type of socket: TCP/UDP (SOCK\_STREAM)
   3. Protocol: IP (0)
3. Connection to a remote address is created with connect() call. Here, we specify the IP address and the port that we are going to connect with. If the connection is successful, a value is returned.
4. The data is retrieved with a recv() call. The received data can be stored in a file, or into a string.
5. Connection to a remote address is created with connect() call. Here, we specify the IP address and the port that we are going to connect with. If the connection is successful, a value is returned.
6. The data is retrieved with a recv() call. The received data can be stored in a file, or into a string.

Server file:

1. First, a socket will be created (similar to the client program).
2. Next, the IP and port of the socket will be bound using the bind() function (client used to connect() function to connect to the server; the server will use the bind() function to listen for the connections).
3. Then listen() function is called to listen to the connections (to see if any client is trying to connect to the server socket).
4. The function accept() is called. The function returns a client socket (the client that has connected).

**Key Commands**

socket – create unbound socket in communications domain

server.sin\_addr – IP address in the socket

connect - used by the client application to establish a connection to a server

send – sends data

recv - receives data

listen - marks the socket referred to by sockfd as a passive socket,

**References**

https://www.geeksforgeeks.org/simple-client-server-application-in-c/