**CONCURRENT MULTIPROCESSING TCP CLIENT PROGRAM**

**OVERVIEW:**  In this program we have designed a client. The function of the client is to fetch the file that is present in the server. First step of the connection oriented TCP service is that it has to bind to the port number given by the client. Then by calling CONNECT() function the connection is established between the client and the server. First the server would display list of files that are present in the sever. Then client would choose the file that has to be transferred from the server to the client. All the data is put in the send buffer and is sent to the client side. Then in the client side it would store the file in the local machine.

**MAIN**

The main function of the program receives 2 arguments from the users. First argument would take the ip address in our case as it is a local system, we can enter it as local host. The Second one would be the port number through which the server is configured to listen for the new connection. Then creating the new socket by taking its family name, type of the of the data being send like SOCK\_STREAM or SOCK\_DGRAM. Then after that socket is created successfully. Input all the necessary corresponding values. In this procedure, client would call connect() function in order to establish a communication between the client and the server. After a successful establishment connection with the server. Client would request for a file based on the available files in the server. Then the corresponding file is opened using the fopen() function by the sever and transfer all the contents from the file to the buffer and transfer the contents from buffer to the file named "clientfile.txt", which is stored in the client local system.

--------------------------------------------------------------------------------------------------------------------------------------

--------------------------------------------------------------------------------------------------------------------------------------

Execution in Command line:

On Client1:

gcc -o tcp\_client1 tcp\_client1.c

./tcp\_client1 localhost 10000

On Client2:

gcc -o tcp\_client2 tcp\_client.c

./tcp\_client2 localhost 10000