**Name:Omkar Deshpande**

**Roll number-33213**

**Batch-K10**

* Aim: To implement socket Programming in C/C++ on Linux.

a) TCP Client , TCP Server

b) UDP Client , UDP Server

* **What is socket programming?**  
  Socket programming is a way of connecting two nodes on a network to communicate with each other. One socket(node) listens on a particular port at an IP, while other socket reaches out to the other to form a connection. Server forms the listener socket while client reaches out to the server.
* TCP Server-Client implementation in C:

If we are creating a connection between client and server using TCP then it has few functionalities like, TCP is suited for applications that require high reliability, and transmission time is relatively less critical. It is used by other protocols like HTTP, HTTPs, FTP, SMTP, Telnet. TCP rearranges data packets in the order specified. There is absolute guarantee that the data transferred remains intact and arrives in the same order in which it was sent. TCP does Flow Control and requires three packets to set up a socket connection, before any user data can be sent. TCP handles reliability and congestion control. It also does error checking and error recovery. Erroneous packets are retransmitted from the source to the destination.

* The entire process can be broken down into following steps:
* TCP Server –
  1. using create(), Create TCP socket.
  2. using bind(), Bind the socket to server address.
  3. using listen(), put the server socket in a passive mode, where it waits for the client to approach the server to make a connection
  4. using accept(), At this point, connection is established between client and server, and they are ready to transfer data.
  5. Go back to Step 3.
* TCP Client –

1. Create TCP socket.
2. connect newly created client socket to server.

# UDP Server-Client implementation in C

* **Theory:**

In UDP, the client does not form a connection with the server like in TCP and instead just sends a datagram. Similarly, the server need not accept a connection and just waits for datagrams to arrive. Datagrams upon arrival contain the address of sender which the server uses to send data to the correct client.

* The entire process can be broken down into following steps:
* UDP Server:

1. Create UDP socket.
2. Bind the socket to server address.
3. Wait until datagram packet arrives from client.
4. Process the datagram packet and send a reply to client.
5. Go back to Step 3.

* UDP Client:

1. Create UDP socket.
2. Send message to server.
3. Wait until response from server is received.
4. Process reply and go back to step 2, if necessary.
5. Close socket descriptor and exit.

* Conclusion:

Hence, we have successfully implemented socket programming through both methods TCP client-server and UDP client-server in ‘C’