Practical No.4

AIM: - Write a program to implement the echo client.

SOFTWARE REQUIRED:- Operating System: - Ubuntu Python 3

THEORY:-

A **TCP Echo Client** is a network client that sends a message to a server and waits for the same message to be returned. Similarly, a **TCP Echo Server** is a network server that listens for incoming client connections, reads messages from clients, and sends back the exact message that was received (i.e., echoes the message). The communication happens over the **TCP (Transmission Control Protocol)**, which ensures reliable data transmission.

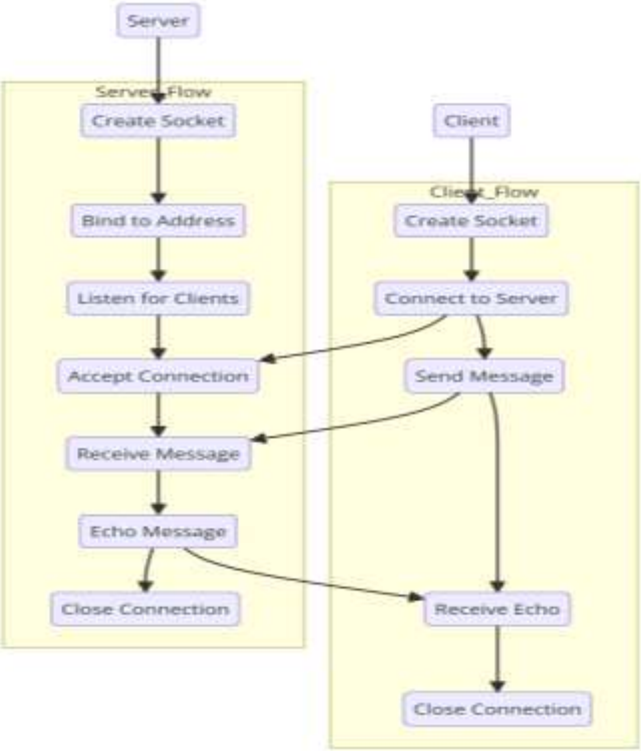
TCP Echo Client-Server Architecture:

TCP Echo Server:

1. **Create Socket**: The server creates a TCP socket to listen for client connections.
2. **Bind to Address**: The server binds the socket to an IP address and port number.
3. **Listen for Clients**: The server listens for incoming client connections.
4. **Accept Connection**: The server accepts the connection when a client requests it.
5. **Receive Message**: The server receives a message from the connected client.
6. **Echo Message**: The server sends the exact message back to the client.
7. **Close Connection**: The server closes the connection with the client and continues to listen for new connections.

TCP Echo Client:

1. **Create Socket**: The client creates a TCP socket to establish communication.
2. **Connect to Server**: The client connects to the server using the server's IP address and port.
3. **Send Message**: The client sends a message to the server (e.g., "Hello, Server!").
4. **Receive Echo**: The client waits for the server to send the same message back.
5. **Close Connection**: The client closes the connection once the echoed message is received



ALGORITHM

Server

1. Create a server socket and bind it to port.
2. Listen for new connection and when a connection arrives, accept it.
3. Read the data from client.
4. Echo the data back to the client.
5. Repeat steps 4-5 until „bye‟ or „null‟ is read.
6. Close all streams.
7. Close the server socket.
8. Stop.

Client

1. Create a client socket and connect it to the server‟s port number.
2. Get input from user.
3. If equal to bye or null, then go to step 7.
4. Send user data to the server.
5. Display the data echoed by the server.
6. Repeat steps 2-4.
7. Close the input and output streams.
8. Close the client socket.
9. Stop.

CONCLUSION: -

Thus the program for TCP echo client server was executed and the output was verified.

Course Teacher
Ms. Prajakta Sawle