

Ex: 12 (a)

Implementation of echo client server
using TCP/UDP sockets

To implement an echo client
(server) by using TCP/UDP sockets.

(Client side) Algorithm

import socket and client

server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

server_socket.bind(('localhost', 12345))

server_socket.listen()

print("server is waiting for connection")

conn, addr = server_socket.accept()

print(f"connected to {addr}")

while True:

data = conn.recv(1024), decode()

if not data or data.lower() == 'exit':

print("connection closed")

break

print(f"Received from client {data}")

conn.send(data.encode())

conn.close()

Client side Algorithm

import socket

client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

client.connect(('localhost', 12346))

while True:

message = input("you: ")

client.send(message.encode())

if message.lower() == "bye":
break

reply = client.recv(1024).decode()

print(f"server: {reply}")

if reply.lower() == "bye":
break

client.close()

Sample Input and Output:

server side:-

server waiting for connection

connected to ('127.0.0.1', 58944)

Received from client: Hello server

~~you? Hello client!~~

Received from client: How are you?

Connection closed.

Client side:-

Enter message: Hello server

Echo from server: Hello server

Enter message: How are you?

Echo from server: How are you?

Enter message: Bye

Bye

(('localhost', 12346))

(1) total server

of address in server

(('localhost', 12346))

(('localhost', 12346))

(('localhost', 12346))

(('localhost', 12346))

(('localhost', 12346))

(('localhost', 12346))

(('localhost', 12346))

(('localhost', 12346))

(('localhost', 12346))

(('localhost', 12346))

(('localhost', 12346))

(('localhost', 12346))

(('localhost', 12346))

(('localhost', 12346))

(('localhost', 12346))

(('localhost', 12346))

(('localhost', 12346))

(('localhost', 12346))

(('localhost', 12346))

(('localhost', 12346))

(('localhost', 12346))