

Aim:

To implement client client server using TCP socket

Algorithm:

- * Start server: Create a socket, bind to host & port, and wait for client message.
- * Receiver message: Server receives data from client.
- * Server sender back the same message to the client.
- * Client prints echoed message on screen.

Code:

```

import socket
mode = new_input("Run as (s)erver or (c)lient?")
if mode == 's':
    s = socket.socket()
    s.bind(('localhost', 12345))
    s.listen(1)
    print("Server waiting for condition...")
    conn, addb = s.accept()
    print("connected", addb)
    data = conn.recv(1024)
    print("Received", data)
    conn.send("Echo:" + data)
    conn.close()

```

client mode == "c":

s = socket.socket()

s.connect(("localhost", 12345))

msg = raw_input("Type a message")

s.send(msg)

print(s.recv(1024))

s.close()

Python TCP.py
Run on (s)erver or (c)lient?

s

Serve waiting for connection

(server) connected(12345) from client = client

c

Type a message: Hello

Received: Hello

Code:

Implement chat client serve using UDP sockets

import socket

mode = raw_input("Run on (s)erver or (c)lient?")

strip().lower()

if mode == 's':

s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

s.bind(("localhost", 12345))

print("UDP serve waiting for messages..")

61

```
while True:  
    data, addr = s.recvfrom(1024)  
    print("Received:", data, "from:", addr)  
    s.sendto(("Echo: " + data).encode(), addr)  
  
elif mode == 'c':  
    s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)  
    server = ('localhost', 12345)  
    msg = raw_input("Type a msg: ")  
    s.sendto(msg.encode(), server)  
    data = s.recvfrom(1024)  
    print("Received from server:", data)  
    s.close()
```

Input:

Python tcp.py

Run on (s)erver or (c)lient?

s

UDP server waiting for messages...

c

Output:

Type a message: "Hello!"

waiting

Received: "Hello!"

Q ⑨

Result:

Implemented chat client server using
TCP / UDP sockets.