

Exp : 129

Date : 18/10/24

Echo Client Server

Aim :-

To implement echo client server using TCP/UDP socket

Algorithm:

Server.py

- Create a UDP socket
- Bind the socket to specific IP address (127.0.0.1) & port (12345)
- Continuous listen for incoming message
- When message received - decode it
- Display message along with sender address
- Repeat infinitely

Client.py

- Create UDP socket
- Set a timeout for socket to avoid waiting
- Send a predefined message hello for Server IP address & port 12345
- If no response received in timeout period, print timeout message
- Close socket after sending message.

Code:

Server.py

```
import socket

def start_server(host = '127.0.0.1', port = 12345):
    with socket.socket(socket.AF_INET, socket.SOCK_DGRAM) as s:
        s.bind((host, port))
        print(f"UDP server running on {host}:{port}")
        while True:
            data, addr = s.recvfrom(1024)
            print(f"Received message from {addr}: {data}")

if __name__ == "__main__":
    start_server()
```

Client.py

```
def ping_server(host = "127.0.0.1", port = 12345):
    with socket.socket(socket.AF_INET, socket.SOCK_DGRAM) as s:
        s.connect((s, port))
        try:
            s.sendto(b"Hello", (host, port))
            print("message sent to server")
        except socket.timeout:
            print("Request timeout")

if __name__ == "__main__":
    ping_server()
```

Output:-

Server.py

Terminal

> Python Server.py

>>

UDP Server running on 127.0.0.1 : 12345

Client.py

Terminal

> Python client.py

>>

message sent to server

Server terminal:-

Received message from ('127.0.0.1', 56003): Hello

Result:-

Thus, the program of echo client server using UDP socket has been implemented & executed successfully.

Signature