1. **Write an echo program with client and iterative server using TCP.**

**TCP Server**

import socket

def start\_server(host='127.0.0.1', port=65432):

with socket.socket(socket.AF\_INET, socket.SOCK\_STREAM) as s:

s.bind((host, port))

s.listen()

print(f'Server started. Listening on {host}:{port}')

while True:

conn, addr = s.accept()

with conn:

print(f'Connected by {addr}')

while True:

data = conn.recv(1024)

if not data:

break

conn.sendall(data)

if \_\_name\_\_ == "\_\_main\_\_":

start\_server()

**\*\*\*\*\* OUTPUT \*\*\*\*\***

PS C:\Users\pruch\OneDrive\Documents> python TCPserver.py

Server started. Listening on 127.0.0.1:65432

Connected by ('127.0.0.1', 61258)

**TCP Client**

import socket

def start\_client(host='127.0.0.1', port=65432):

    with socket.socket(socket.AF\_INET, socket.SOCK\_STREAM) as s:

        try:

            s.connect((host, port))

            print(f'Connected to server {host}:{port}')

            while True:

                message = input("Enter message to send (or 'exit' to quit): ")

                if message.lower() == 'exit':

                    break

                s.sendall(message.encode())

                data = s.recv(1024)

                print(f'Received from server: {data.decode()}')

        except Exception as e:

            print(f'An error occurred: {e}')

if \_\_name\_\_ == "\_\_main\_\_":

    start\_client()

**\*\*\*\*\* OUTPUT \*\*\*\*\***

PS C:\Users\pruch\OneDrive\Documents> python TCPclient.py

Connected to server 127.0.0.1:65432

Enter message to send (or 'exit' to quit): Hii

Received from server: Hii

Enter message to send (or 'exit' to quit): Hello TCP Iterative Server.py

Received from server: Hello TCP Iterative Server.py

Enter message to send (or 'exit' to quit):

**Assi No 2. - Write an echo program with client and iterative server using UDP.**

**UDP SERVER**

import socket

HOST = 'localhost' # You can change this to your desired hostname

PORT = 65535

server\_socket = socket.socket(socket.AF\_INET, socket.SOCK\_DGRAM)

server\_socket.bind((HOST, PORT))

print(f'Server listening on {HOST}:{PORT}')

while True:

print('Waiting for a connection...')

data, address = server\_socket.recvfrom(1024)

print(f'Received {len(data)} bytes from {address}')

print(f'Message: {data.decode()}')

server\_socket.sendto(data, address)

print('Sent back to client')

**\*\*\*\*\* OUTPUT \*\*\*\*\***

C:\Users\pruch>cd "C:\Users\pruch\OneDrive\Documents

C:\Users\pruch\OneDrive\Documents>python UDPserver.py

Server listening on localhost:12345

Waiting for a connection...

Received 3 bytes from ('127.0.0.1', 55605)

Message: Hii

Sent back to client

Waiting for a connection...

Received 17 bytes from ('127.0.0.1', 55605)

Message: UDPclient is here

Sent back to client

Waiting for a connection...

**UDP CLIENT**

import socket

HOST = 'localhost' # Must match the server's hostname

PORT = 12345

client\_socket = socket.socket(socket.AF\_INET, socket.SOCK\_DGRAM)

try:

while True:

message = input("Enter a message (or 'exit' to quit): ")

if message.lower() == 'exit':

break

client\_socket.sendto(message.encode(), (HOST, PORT))

data, server\_address = client\_socket.recvfrom(1024)

print(f'Received {len(data)} bytes from {server\_address}')

print(f'Message: {data.decode()}')

except Exception as e:

print(f"An error occurred: {e}")

finally:

client\_socket.close()

**\*\*\*\*\* OUTPUT \*\*\*\*\***

C:\Users\pruch>cd "C:\Users\pruch\OneDrive\Documents

C:\Users\pruch\OneDrive\Documents>python UDPclient.py

Enter a message (or 'exit' to quit): Hii

Received 3 bytes from ('127.0.0.1', 12345)

Message: Hii

Enter a message (or 'exit' to quit): UDPclient is here

Received 17 bytes from ('127.0.0.1', 12345)

Message: UDPclient is here

Enter a message (or 'exit' to quit):

**3. Write an echo program with client and concurrent server using TCP.**

**TCP Server**

import socket

def server\_program():

host = socket.gethostname()

port = 5000

server\_socket = socket.socket()

server\_socket.bind((host, port))

server\_socket.listen(2) # Maximum 2 clients can connect

print(f"Server listening on {host}:{port}")

while True:

conn, address = server\_socket.accept()

print(f"Connection from {address}")

while True:

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

if not data:

break

print(f"Received: {data}")

conn.send(data.encode())

conn.close()

if \_\_name\_\_ == "\_\_main\_\_":

server\_program()

**\*\*\*\*\* OUTPUT \*\*\*\*\***

PS F:\2024-25\MSc II\MSC II Practicals> python concurrentserver.py

Server listening on Ruchita:5000

Connection from ('192.168.91.61', 61545)

Received: Hello

Received: Hii there

**TCP Client**

import socket

def client\_program():

host = socket.gethostname()

port = 5000

client\_socket = socket.socket()

client\_socket.connect((host, port))

while True:

message = input("Enter a message (or 'bye' to exit): ")

client\_socket.send(message.encode())

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

break

data = client\_socket.recv(1024).decode()

print(f"Received from server: {data}")

client\_socket.close()

if \_\_name\_\_ == "\_\_main\_\_":

client\_program()

**\*\*\*\*\* OUTPUT \*\*\*\*\***

PS F:\2024-25\MSc II\MSC II Practicals> python concurrentclient.py

Enter a message (or 'bye' to exit): Hello

Received from server: Hello

Enter a message (or 'bye' to exit): Hii there

Received from server: Hii there

Enter a message (or 'bye' to exit):

**5. Write a client and server program for chatting.**

**Server**

import socket

import threading

def handle\_client(client\_socket, client\_address):

print(f"New connection: {client\_address}")

while True:

try:

# Receive message from the client

message = client\_socket.recv(1024).decode('utf-8')

if not message:

break

print(f"{client\_address} says: {message}")

# Broadcast the message to all connected clients

broadcast\_message(f"{client\_address} says: {message}", client\_socket)

except:

break

client\_socket.close()

def broadcast\_message(message, sender\_socket):

for client in clients:

if client != sender\_socket:

try:

client.send(message.encode('utf-8'))

except:

client.close()

clients.remove(client)

def start\_server(host='127.0.0.1', port=65432):

server\_socket = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

server\_socket.bind((host, port))

server\_socket.listen()

print(f"Server started on {host}:{port}")

while True:

client\_socket, client\_address = server\_socket.accept()

clients.append(client\_socket)

# Start a new thread for each client

client\_thread = threading.Thread(target=handle\_client, args=(client\_socket, client\_address))

client\_thread.start()

if \_\_name\_\_ == "\_\_main\_\_":

clients = []

start\_server()

**\*\*\*\*\* OUTPUT \*\*\*\*\***

PS F:\2024-25\MSc II\MSC II Practicals> python chatbotserver.py

Server started on 127.0.0.1:65432

New connection: ('127.0.0.1', 55430)

('127.0.0.1', 55430) says: Hii

('127.0.0.1', 55430) says: I am Chatbot

**Client**

import socket

import threading

def receive\_messages(client\_socket):

while True:

try:

message = client\_socket.recv(1024).decode('utf-8')

if not message:

break

print(message)

except:

print("An error occurred. Exiting...")

client\_socket.close()

break

def send\_messages(client\_socket):

while True:

message = input("")

client\_socket.send(message.encode('utf-8'))

def start\_client(host='127.0.0.1', port=65432):

client\_socket = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)

client\_socket.connect((host, port))

receive\_thread = threading.Thread(target=receive\_messages, args=(client\_socket,))

receive\_thread.start()

send\_thread = threading.Thread(target=send\_messages, args=(client\_socket,))

send\_thread.start()

if \_\_name\_\_ == "\_\_main\_\_":

start\_client()

**\*\*\*\*\* OUTPUT \*\*\*\*\***

PS F:\2024-25\MSc II\MSC II Practicals> python chatbotclient.py

Hii

I am Chatbot

**Assi No 6. - Write a program to retrieve date and time using TCP.**

**TCP SERVER**

import socket

import datetime

def start\_server(host='127.0.0.1', port=65432):

with socket.socket(socket.AF\_INET, socket.SOCK\_STREAM) as s:

s.bind((host, port))

s.listen()

print(f"Server listening on {host}:{port}")

while True:

client\_socket, client\_address = s.accept()

with client\_socket:

print(f"Connected by {client\_address}")

now = datetime.datetime.now().strftime('%Y-%m-%d %H:%M:%S')

client\_socket.sendall(now.encode('utf-8'))

if \_\_name\_\_ == "\_\_main\_\_":

start\_server()

**\*\*\*\*\* OUTPUT \*\*\*\*\***

C:\Users\admin>cd documents

C:\Users\admin\Documents>python datetimeserverusingtcp.py

Server listening on 127.0.0.1:65432

Connected by ('127.0.0.1', 53523)

----------------------------------------------------------------

**TCP CLIENT**

import socket

def get\_datetime\_from\_server(host='127.0.0.1', port=65432):

with socket.socket(socket.AF\_INET, socket.SOCK\_STREAM) as client\_socket:

client\_socket.connect((host, port))

data = client\_socket.recv(1024)

print("Received date and time from server:", data.decode('utf-8'))

if \_\_name\_\_ == "\_\_main\_\_":

get\_datetime\_from\_server()

**\*\*\*\*\* OUTPUT \*\*\*\*\***

C:\Users\admin>cd documents

C:\Users\admin\Documents>python datetimeclientusingtcp.py

Received date and time from server: 2024-10-09 14:01:09

**Assi. No. - 7. Write a program to retrieve date and time using UDP.**

**Server Program**

import socket

import datetime

def start\_server(host='127.0.0.1', port=65432):

# Create a UDP socket

with socket.socket(socket.AF\_INET, socket.SOCK\_DGRAM) as server\_socket:

# Bind the socket to the address and port

server\_socket.bind((host, port))

print(f"UDP server up and listening on {host}:{port}")

while True:

# Wait for a connection

message, client\_address = server\_socket.recvfrom(1024)

print(f"Received request from {client\_address}")

# Retrieve current date and time

now = datetime.datetime.now().strftime('%Y-%m-%d %H:%M:%S')

# Send date and time to the client

server\_socket.sendto(now.encode('utf-8'), client\_address)

if \_\_name\_\_ == "\_\_main\_\_":

start\_server()

**\*\*\*\*\* Output \*\*\*\*\***

C:\Users\admin>cd documents

C:\Users\admin\Documents> python udpdatetimeserver.py

**UDP server up and listening on 127.0.0.1:65432**

**Received request from ('127.0.0.1', 52891)**

**Client Program**

import socket

def get\_datetime\_from\_server(host='127.0.0.1', port=65432):

# Create a UDP socket

with socket.socket(socket.AF\_INET, socket.SOCK\_DGRAM) as client\_socket:

# Send a request to the server

client\_socket.sendto(b"Requesting date and time", (host, port))

# Receive the response from the server

data, server\_address = client\_socket.recvfrom(1024)

print("Received date and time from server:", data.decode('utf-8'))

if \_\_name\_\_ == "\_\_main\_\_":

get\_datetime\_from\_server()

**\*\*\*\*\* Output \*\*\*\*\***

C:\Users\admin>cd documents

C:\Users\admin\Documents>python udpdatetimeclient.py

**Received date and time from server: 2024-09-14 12:24:26**

**Assi 8. Write a client and server program to implement file transfer.**

**Server Program**

import socket

def write\_and\_send\_file(filename, conn):

    # Write some data to the file

    with open(filename, 'w') as file:

        file.write("Hello, this is the data written by the server.\n")

        file.write("This file will be sent to the client.\n")

    # Now, send the file

    with open(filename, 'rb') as file:

        data = file.read()

        conn.sendall(data)

def main():

    host = '127.0.0.1'  # Localhost

    port = 65432  # Arbitrary non-privileged port

    with socket.socket(socket.AF\_INET, socket.SOCK\_STREAM) as s:

        s.setsockopt(socket.SOL\_SOCKET, socket.SO\_REUSEADDR, 1)

        s.bind((host, port))

        s.listen()

        print(f"Server listening on {host}:{port}")

        conn, addr = s.accept()

        with conn:

            print(f"Connected by {addr}")

            filename = "C:\\Users\\pruch\\OneDrive\\Documents\\Desktop\\myFile.txt"  # File to create and send

            write\_and\_send\_file(filename, conn)

            print("File written and sent successfully")

if \_\_name\_\_ == "\_\_main\_\_":

    main()

**\*\*\*\*\* Output \*\*\*\*\***

PS F:\2024-25\MSc II\MSC II Practicals> python fileserver.py

Server listening on 127.0.0.1:65432

Connected by ('127.0.0.1', 52306)

File written and sent successfully

PS F:\2024-25\MSc II\MSC II Practicals>

**Client Program**

import socket

def receive\_file(filename, conn):

    # Receive and write the file content

    with open(filename, 'wb') as file:

        while True:

            data = conn.recv(1024)

            if not data:

                break

            file.write(data)

def main():

    host = '127.0.0.1'  # The server's hostname or IP address

    port = 65432  # The port used by the server

    with socket.socket(socket.AF\_INET, socket.SOCK\_STREAM) as s:

        s.connect((host, port))

        print(f"Connected to server at {host}:{port}")

        filename = "C:\\Users\\pruch\\OneDrive\\Documents\\Desktop\\myFile.txt"   # File to save as

        receive\_file(filename, s)

        print("File received successfully")

if \_\_name\_\_ == "\_\_main\_\_":

    main()

**\*\*\*\*\* Output \*\*\*\*\***

PS F:\2024-25\MSc II\MSC II Practicals> python fileclient.py

Connected to server at 127.0.0.1:65432

File received successfully

PS F:\2024-25\MSc II\MSC II Practicals>