SOCKET PROGRAMMING (TCP):

TCP SERVER CODE:

import socket

def server\_program():

# get the hostname

host = socket.gethostname()

port = 5000 # initiate port no above 1024

server\_socket = socket.socket() # get instance

# look closely. The bind() function takes tuple as argument

server\_socket.bind((host, port)) # bind host address and port together

# configure how many client the server can listen simultaneously

server\_socket.listen(2)

conn, address = server\_socket.accept() # accept new connection

print("Connection from: " + str(address))

while True:

# receive data stream. it won't accept data packet greater than 1024 bytes

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

if not data:

# if data is not received break

break

print("from connected user: " + str(data))

data = input(' -> ')

conn.send(data.encode()) # send data to the client

conn.close() # close the connection

if \_\_name\_\_ == '\_\_main\_\_':

server\_program()

TCP CLIENT CODE:

import socket

def client\_program():

host = socket.gethostname() # as both code is running on same pc

port = 5000 # socket server port number

client\_socket = socket.socket() # instantiate

client\_socket.connect((host, port)) # connect to the server

message = input(" -> ") # take input

while message.lower().strip() != 'bye':

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

data = client\_socket.recv(1024).decode() # receive response

print('Received from server: ' + data) # show in terminal

message = input(" -> ") # again take input

client\_socket.close() # close the connection

if \_\_name\_\_ == '\_\_main\_\_':

client\_program()

TCP SERVER:

comp@comp:~/A-36$ python3 server.py

Connection from: ('127.0.0.1', 36218)

from connected user: Helloooo

-> Hi

from connected user: This is UNIX system

-> This is also UNIX but Ubuntu

from connected user: Do you have VS Code?

-> No. What about you?

from connected user: yes

->

TCP CLIENT:

comp@comp:~/A-36$ python3 client.py

-> Helloooo

Received from server: Hi

-> This is UNIX system

Received from server: This is also UNIX but Ubuntu

-> Do you have VS Code?

Received from server: No. What about you?

-> yes

SOCKET PROGRAMMING (UDP):

UDP SERVER CODE:

import socket

# Define the server IP address and port

server\_ip = '127.0.0.1' # Replace with your server's IP address or use 'localhost' for local testing

server\_port = 12345 # Choose a port number

# Create a UDP socket

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

# Bind the socket to the server address and port

server\_socket.bind((server\_ip, server\_port))

print(f"Server listening on {server\_ip}:{server\_port}")

while True:

try:

# Receive data from the client

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

print(f"Received message from {client\_address}: {data.decode('utf-8')}")

# Process the received data (You can add your own logic here)

# Send a response back to the client

response = "Message received by the server."

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

except KeyboardInterrupt:

print("Server terminated by the user.")

break

# Close the server socket

server\_socket.close()

UDP CLIENT CODE:

import socket

# Define the server IP address and port

server\_ip = '127.0.0.1' # Replace with the server's IP address or use 'localhost' for local testing

server\_port = 12345 # Match the server's port number

# Create a UDP socket

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

while True:

try:

# Get user input

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

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

break

# Send the message to the server

client\_socket.sendto(message.encode('utf-8'), (server\_ip, server\_port))

# Receive a response from the server

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

print(f"Response from server ({server\_address}): {response.decode('utf-8')}")

except KeyboardInterrupt:

print("Client terminated by the user.")

break

# Close the client socket

client\_socket.close()

UDP SERVER:

comp@comp:~/A-36$ python3 socket\_udp\_server.py

Server listening on 127.0.0.1:12345

Received message from ('127.0.0.1', 55249): Hi

Received message from ('127.0.0.1', 55249): I am back

Received message from ('127.0.0.1', 55249): I am Ubuntu

UDP CLIENT:

comp@comp:~/A-36$ python3 socket\_udp\_client.py

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

Response from server (('127.0.0.1', 12345)): Message received by the server.

Enter a message (or 'exit' to quit): I am back

Response from server (('127.0.0.1', 12345)): Message received by the server.

Enter a message (or 'exit' to quit): I am Ubuntu

Response from server (('127.0.0.1', 12345)): Message received by the server.

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