



RTE Socket Programming

Calculator, Chat and Multiuser Chat

Submitted by Aleena Anna Plessey

Reg no: 21BCE5767

Course Program: B.Tech

Course code: BCSE308L

Course Title: Computer Networks

Batch:2021-25

Submitted to: Dr. Swaminathan A

Submitted on 07/06/2023

Problem Statement:

5.1 Write a TCP Client Server program to perform Mathematical Calculations.

5.2 Write a UDP Client Server program to perform a simple chat.

5.3 Using the given components and devices, such as routers, switches and cat6 cables, create a of any topology. Configure the router and configure the IP for all n nodes connected in the network using dynamic DHCP. Write a UDP Client Server program to perform a multiuser chat program.

Aim and Objective:

- a) To create a TCP Client Server program
- b) To create a UDP client server program
- c) To write and complete a UDP client server program to make a multiuser chat program using all the computers connected in a network.

5.1 TCP Client Server program to perform Mathematical Calculations.

Code:

Client side:

```
1  import socket
2  def start_client():
3      host="127.0.0.1"
4      port=12345
5
6      client_socket=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
7      client_socket.connect((host,port))
8
9      print("Connected to server.")
10
11  while True:
12      expression=input("Enter an expression to calculate (q to leave):")
13      if expression=="q":
14          break
15      client_socket.send(expression.encode())
16      result=client_socket.recv(1024).decode()
17      print("Result:",result)
18  client_socket.close()
19  print("Connect closed")
20  if __name__=="__main__":
21      start_client()
22
23
```

Server side:

```

1  import socket
2  def calculate(expression):
3      try:
4          result=eval(expression)
5          return str(result)
6      except:
7          return "ERROR"
8  def start_server():
9      host="127.0.0.1"
10     port=12345
11
12     server_socket=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
13     server_socket.bind((host,port))
14     server_socket.listen(1)
15     print("Server Started.Waiting...")
16     while True:
17         client_socket,addr=server_socket.accept()
18         print("Client connected:",addr)
19
20         while True:
21             data=client_socket.recv(1024).decode()
22             if not data:
23                 break
24             result=calculate(data)
25             client_socket.send(result.encode())
26
27         client_socket.close()
28         print("Disconnected")
29
30 if __name__=="__main__":
31     start_server()

```

- 1)Run the server-side code first to start the server.
- 2)Run the client-side server and input the expressions the output will be given.

Output:

```
Command Prompt
Microsoft Windows [Version 10.0.19045.2965]
(c) Microsoft Corporation. All rights reserved.

C:\Users\popun>cd Desktop\cn
C:\Users\popun\Desktop\cn>cd python cn
C:\Users\popun\Desktop\cn\python cn>python clientside.py
Connected to server.
Enter an expression to calculate (q to leave):1+4
Result: 5
Enter an expression to calculate (q to leave):496/7+3
Result: 73.85714285714286
Enter an expression to calculate (q to leave):q
Connect closed
C:\Users\popun\Desktop\cn\python cn>

Command Prompt - python serverside.py
Microsoft Windows [Version 10.0.19045.2965]
(c) Microsoft Corporation. All rights reserved.

C:\Users\popun>cd Desktop
C:\Users\popun\Desktop>cd cn
C:\Users\popun\Desktop\cn>cd python cn
C:\Users\popun\Desktop\cn\python cn>python serverside.py
Server Started.Waiting...
Client connected: ('127.0.0.1', 50226)
Disconnected
```

5.2 UDP Client Server program to perform a simple chat.

Code:

Server:

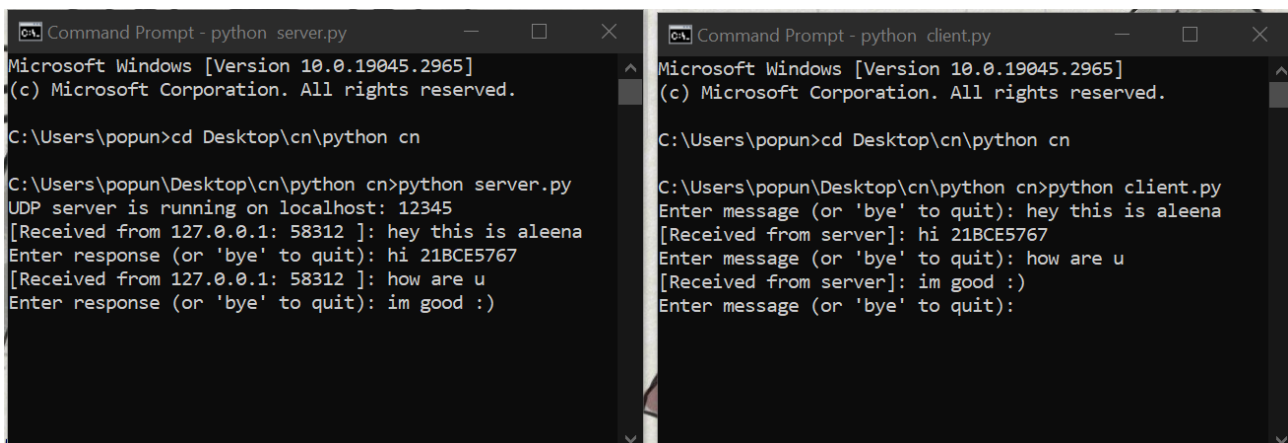
```
1  import socket
2  SERVER_IP = 'localhost'
3  SERVER_PORT = 12345
4
5  server_socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
6  server_socket.bind((SERVER_IP, SERVER_PORT))
7
8  print("UDP server is running on", SERVER_IP + ":", SERVER_PORT)
9
10 while True:
11     data, client_address = server_socket.recvfrom(1024)
12     message = data.decode()
13     print("[Received from", client_address[0] + ":", client_address[1], "]:", message)
14     response = input("Enter response (or 'bye' to quit): ")
15     server_socket.sendto(response.encode(), client_address)
16
17     if response.lower() == "bye":
18         break
19
20 server_socket.close()
```

Client:

```
1  import socket
2  SERVER_IP = 'localhost'
3  SERVER_PORT = 12345
4  client_socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
5  while True:
6      message = input("Enter message (or 'bye' to quit): ")
7      client_socket.sendto(message.encode(), (SERVER_IP, SERVER_PORT))
8
9      if message.lower() == "bye":
10         break
11
12     data, _ = client_socket.recvfrom(1024)
13     response = data.decode()
14     print("[Received from server]:", response)
15
16 client_socket.close()
17
```

- 1) Run the server-side code first to start the server.
- 2) Run the client-side server and send the first message. It will be reflected and the server side will get the message.
- 3) Then the server side can send the message which will be received in the client side.

Output:



```
Command Prompt - python server.py
Microsoft Windows [Version 10.0.19045.2965]
(c) Microsoft Corporation. All rights reserved.

C:\Users\popun>cd Desktop\cn\python cn

C:\Users\popun\Desktop\cn\python cn>python server.py
UDP server is running on localhost: 12345
[Received from 127.0.0.1: 58312 ]: hey this is aleena
Enter response (or 'bye' to quit): hi 21BCE5767
[Received from 127.0.0.1: 58312 ]: how are u
Enter response (or 'bye' to quit): im good :)

Command Prompt - python client.py
Microsoft Windows [Version 10.0.19045.2965]
(c) Microsoft Corporation. All rights reserved.

C:\Users\popun>cd Desktop\cn\python cn

C:\Users\popun\Desktop\cn\python cn>python client.py
Enter message (or 'bye' to quit): hey this is aleena
[Received from server]: hi 21BCE5767
Enter message (or 'bye' to quit): how are u
[Received from server]: im good :)
Enter message (or 'bye' to quit):
```

5.3 UDP Client Server program to perform a multiuser chat program.

Code:

Server:

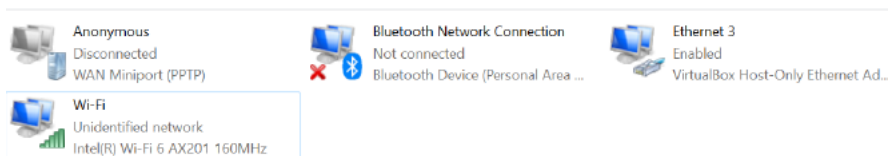
```
1 import socket
2 server_socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
3 server_address = ('192.168.1.6', 8888)
4 server_socket.bind(server_address)
5 print('UDP server listening on {}:{}'.format(*server_address))
6 while True:
7     message, client_address = server_socket.recvfrom(1024)
8     print('Received from {}: {}'.format(client_address, message.decode()))
9     response = 'Message received: {}'.format(message.decode())
10    server_socket.sendto(response.encode(), client_address)
```

Client:

```
1 import socket
2 client_socket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
3 server_address = ('192.168.1.6', 8888)
4 while True:
5     message = input("Enter a message: ")
6     client_socket.sendto(message.encode(), server_address)
7     response, server = client_socket.recvfrom(1024)
8     print('Received from {}: {}'.format(server, response.decode()))
9     client_socket.close()
```

Output:

1) Connect to the router and connect the system to the router



2) Verify that the network is connected using ping ipaddress

3) Then assign unique IP addresses to each of the computers manually

4) Run the code.

```
Python Shell 1.10.7
File Edit Shell Debug Options Window Help
>>>
===== RESTART: C:/Users/win10/Desktop/socket/server.py =====
UDP server listening on 192.168.1.6:8888
Received from ('192.168.1.6', 61204): conda activate base
Received from ('192.168.1.6', 61204): hi
Received from ('192.168.1.6', 61204): g
Received from ('192.168.1.7', 51838): Jai Mata Di
Received from ('192.168.1.1', 61352): ankush 21BPS1052
Received from ('192.168.1.7', 51838): Hello , too much heat in Chennai
Received from ('192.168.1.6', 61204): hi i am ankush
Received from ('192.168.1.10', 50901): This Is Srujan
Received from ('192.168.1.1', 61352): 21BPS1052
Received from ('192.168.1.6', 61204): 21BRS1039
Received from ('192.168.1.10', 50901): 21BCE5993
Received from ('192.168.1.10', 50901): 21BCE1979
Received from ('192.168.1.8', 61880): HAR HAR MAHADEV
Received from ('192.168.1.8', 61880): running on PC IP 1.911.68.1.8
Received from ('192.168.1.8', 61880): connected
Received from ('192.168.1.8', 61880): 192.168.1.8
Received from ('192.168.1.12', 51875): 21BPS1615 VIKRAM SINGH
Received from ('192.168.1.106', 54957): Hello Ji
Received from ('192.168.1.13', 58527): harini
Received from ('192.168.1.13', 58527): 21bai1856
```

Result:

We were able to successfully create a TCP client server program, a UDP client server single user chat and a multiuser chat as well.