

**190905156**  
**Rhea Adhikari**  
**Roll No:23**  
**Lab 4**

### 1. Write a UDP time server to display the current time and day.

#### Client.py

```
import socket
import time
localIP = "127.0.0.1"
localPort = 20001
bufferSize = 1024
currentTime = time.ctime(time.time()) + "\r\n"
bytesToSend = str.encode(currentTime)
# Create a datagram socket
UDPServerSocket = socket.socket(family=socket.AF_INET, type=socket.SOCK_DGRAM)
# Bind to address and ip
UDPServerSocket.bind((localIP, localPort))
print("UDP server up and listening")
# Listen for incoming datagrams
while(True):
    bytesAddressPair = UDPServerSocket.recvfrom(bufferSize)
    message = bytesAddressPair[0]
    address = bytesAddressPair[1]
    clientMsg = "Message from Client:{}".format(message)
    clientIP = "Client IP Address:{}".format(address)
    print(clientMsg)
    print(clientIP)
    # Sending a reply to client
    UDPServerSocket.sendto(bytesToSend, address)
```

#### Server.py

```
import socket
msgFromClient = "Hello UDP Server"
bytesToSend = str.encode(msgFromClient)
serverAddressPort = ("127.0.0.1", 20001)
bufferSize = 1024
# Create a UDP socket at client side
UDPClientSocket = socket.socket(family=socket.AF_INET, type=socket.SOCK_DGRAM)
# Send to server using created UDP socket
UDPClientSocket.sendto(bytesToSend, serverAddressPort)
msgFromServer = UDPClientSocket.recvfrom(bufferSize)
msg = "Message from Server {}".format(msgFromServer[0])
print(msg)
```

```
student@dslab-12:~/Documents/190905156_DS/Lab4$ python3
ls.py
Message from Server b'Thu Mar 31 11:12:42 2022\r\n'
```

```
student@dslab-12:~/Documents/190905156_DS/Lab4$ █
```

```
student@dslab-12:~/Documents/190905156_DS/Lab4$ python3
lc.py
UDP server up and listening
Message from Client:b'Hello UDP Server'
Client IP Address:('127.0.0.1', 57284)
█
```

## 2. Write a UDP simple chat program for message send and receive.

### Client.py

```
import socket
sock = socket.socket(socket.AF_INET,socket.SOCK_DGRAM)
udp_host = socket.gethostname()
udp_port = 12345
name = input(str("\nEnter your name: "))
sock.sendto(name.encode(), (udp_host,udp_port));
s_name,addr = sock.recvfrom(1024)
s_name = s_name.decode()
print(s_name, "has joined the chat room\nEnter ctrl+c to exit chat room\n")
while True:
    message,addr = sock.recvfrom(1024)
    message = message.decode()
    print(s_name, ":", message)
    message = input(str("Me : "))
    sock.sendto(message.encode(),addr)
```

### Server.py

```
import socket
import datetime
sock = socket.socket(socket.AF_INET,socket.SOCK_DGRAM)
udp_host = socket.gethostname()
udp_port = 12345
listeningAddress = (udp_host, udp_port)
sock.bind((udp_host, udp_port))
s_name,addr = sock.recvfrom(1024)
s_name = s_name.decode()
print(s_name, "has connected to the chat room\nEnter ctrl+c to exit chat room\n")
name = input(str("Enter your name: "))
sock.sendto(name.encode(),addr)
while True:
    message = input(str("Me : "))
    sock.sendto(message.encode(),addr)
    message,addr = sock.recvfrom(1024)
    message = message.decode()
    print(s_name, ":", message)
```

```
student@dslab-12:~/Documents/190905156_DS/Lab4$ python3 2s.py
Rhea has connected to the chat room
Enter ctrl+c to exit chat room

Enter your name: XYZ
Me : Hi
Rhea : Hello
Me : How are you
Rhea : Im good
Me : This is so cool
█
```

```
student@dslab-12:~/Documents/190905156_DS/Lab4$ python3 2c.py

Enter your name: Rhea
XYZ has joined the chat room
Enter ctrl+c to exit chat room

XYZ : Hi
Me : Hello
XYZ : How are you
Me : Im good
XYZ : This is so cool
Me : █
```

### 3. Write a TCP/UDP peer to peer chat system between two different machines.

#### Client.py

```
import socket
sock = socket.socket(socket.AF_INET,socket.SOCK_DGRAM)
udp_host = '172.16.58.157'
udp_port = 12345
name = input(str("\nEnter your name: "))
sock.sendto(name.encode(), (udp_host,udp_port));
s_name,addr = sock.recvfrom(1024)
s_name = s_name.decode()
print(s_name, "has joined the chat room\nEnter ctrl+c to exit chat room\n")
while True:
    message,addr = sock.recvfrom(1024)
    message = message.decode()
    print(s_name, ":", message)
    message = input(str("Me : "))
    sock.sendto(message.encode(),addr)
```

#### Server.py

```
import socket
sock = socket.socket(socket.AF_INET,socket.SOCK_DGRAM)
udp_host = socket.gethostname()
udp_port = 12345
listeningAddress = (udp_host, udp_port)
sock.bind((udp_host, udp_port))
s_name,addr = sock.recvfrom(1024)
s_name = s_name.decode()
print(s_name, "has connected to the chat room\nEnter ctrl+c to exit chat room\n")
name = input(str("Enter your name: "))
sock.sendto(name.encode(),addr)
while True:
    message = input(str("Me : "))
    sock.sendto(message.encode(),addr)
    message,addr = sock.recvfrom(1024)
    message = message.decode()
    print(s_name, ":", message)
```

```
student@dslab-12:~/Documents/190905156_DS/Lab4$ python3 3c.py
```

```
Enter your name: Rhea
Mitul has joined the chat room
Enter ctrl+c to exit chat room

Mitul : Hello
Me : Hi!
Mitul : Successful execution!!!
Me : Yippee!!!
```

## 4. Debug Code

### Client.py

```
import socket

serverIP = 'localhost'
serverPort = 16008

clientSock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
clientSock.connect((serverIP, serverPort))

message = input("Input integers with space in between: ")
message2 = input("Enter the length of the set: ")

clientSock.send(message.encode())
clientSock.send(message2.encode())

data = clientSock.recv(1024)

temp = [float(x) for x in data.split()]

print("The total of all numbers is: " + str(temp[0]))
print("The lowest number is: " + str(temp[1]))
print("The highest number is: " + str(temp[2]))
print("The mean is: " + str(temp[3]))

clientSock.close()
```

### Server.py

```
import socket

serverIP = 'localhost'
serverPort = 16008

serverSock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
serverSock.bind((serverIP, serverPort))
serverSock.listen(1)

print("TCP server has started and is ready to receive")

while 1:
    connection, addr = serverSock.accept()
    data = connection.recv(1024)
    if not data: break

    temp = [float(x) for x in data.split()]
    print("Received data:", temp)

    length = len(temp)
    maximum = max(temp)
    minimum = min(temp)
    total = sum(temp)
    mean = total/length
```

```
msg = str(total) + " " + str(minimum) + " " + str(maximum) + " " + str(mean)
```

```
connection.send(str(msg).encode())
```

```
student@dslab-12:~/Documents/190905156_DS/Lab4$ python3 4s.py  
TCP server has started and is ready to receive  
Received data: [5.0, 4.0, 6.0, 8.0, 1.0, 96.0]  
█
```

```
student@dslab-12:~/Documents/190905156_DS/Lab4$ python3 4c.py  
Input integers with space in between: 5 4 6 8 1 9  
Enter the length of the set: 6  
The total of all numbers is: 120.0  
The lowest number is: 1.0  
The highest number is: 96.0  
The mean is: 20.0  
student@dslab-12:~/Documents/190905156_DS/Lab4$ █
```

Ln 14, Col 1 Spaces: 4 UTF-8