

## Multi User-UDP

**Input :**

**server\_multi\_udp.py**

```
import random
from socket import *
from threading import Thread, Lock
client_info = {}
def listener(serverSocket):
    global client_info
    global recv_msg
    global name
    while flag:
        #print "started"
        recv_msg, address = serverSocket.recvfrom(1024)
        #print bool(client_info)
        if bool(client_info):
            if address in client_info.keys():
                msg = "\t\t"+client_info[address]+":"+recv_msg
                print msg
                for addr in client_info:
                    "Send data using sendto to addr"
                    serverSocket.sendto(msg,address)

            else:
                client_info[address] = recv_msg
                msg = "\n"+str(client_info[address])+" is online"
                print msg
                for addr in client_info:
                    "Send data using sendto to addr"
                    serverSocket.sendto(msg,address)

        else:
            client_info[address] = recv_msg
            msg = "\n"+str(client_info[address])+" is online"
            print msg
            for addr in client_info:
                "Send data using sendto to addr"
                serverSocket.sendto(msg,address)

flag = True
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(("", 5500))
#recv_msg, address = serverSocket.recvfrom(1024)
#client_addr = str(address[0])
```

```

#print recv_msg + "is connected from IP:"+str(address[0])+" and Port:"+str(address[1])+"\n"
#name = recv_msg
# Start channel receiver thread
listener=Thread(target=listener,args=(serverSocket,))
listener.daemon=True
listener.start()
"Create a thread and call listener"
print "Server Started Listening.."
while flag:
    send_msg = raw_input()

    if send_msg == '*quit*':
        flag = False
        serverSocket.sendto(send_msg, address)
    else:
        serverSocket.sendto(send_msg, address)

```

### **client\_multi\_udp.py**

```

from socket import *
import time
from threading import Thread, Lock
def receiver(clientSocket):
    global recv_msg
    while flag:
        "receive data from server"
        recv_msg,address=clientSocket.recvfrom(1024)
        print "\t\t"+recv_msg

flag = True
clientSocket = socket(AF_INET, SOCK_DGRAM)
addr = ('172.19.229.227', 5500)
name = str(raw_input("Enter Your Name:"))
"Send name to client"
clientSocket.sendto(name,addr)
#try:
# Start channel receiver thread
recv_thread = Thread(target=receiver, args=(clientSocket,))
recv_thread.daemon = True
recv_thread.start()

recv_msg = 'NUll'

while recv_msg != '*quit*':

```

```

send_msg = raw_input()
if send_msg == '*quit*':
    "Send the data"
    clientSocket.sendto(send_msg,addr)
    flag = False
    recv_thread.stop()
else:
    "Send the data"
    clientSocket.sendto(send_msg,addr)

```

## Output :

The screenshot shows a Linux desktop environment with three terminal windows open. The top-left window is the server terminal, and the other two are client terminals. The server terminal shows the command `python server_multi_udp.py` and the output `Server Started Listening..`. It also shows the names of the connected clients: `snehal is online` and `rutuja is online`. The client terminals show the command `python client_multi_udp.py` and the prompt `Enter Your Name:`. The client `snehal` has entered `hello` and received `snehal:hello` from the server. The client `rutuja` has entered `hi` and received `rutuja:hi` from the server.

```

admin@ACA8FC68:~$ python server_multi_udp.py
Server Started Listening..

snehal is online
rutuja is online
rutuja:hi
snehal:hello

admin@ACA8FC68:~$ python client_multi_udp.py
Enter Your Name:snehal

snehal is online
hello
snehal:hello
snehal:hello

admin@ACA8FC68:~$ python client_multi_udp.py
Enter Your Name:rutuja

rutuja is online
rutuja is online
hi
rutuja:hi
rutuja:hi

```