**B10-Write a program using UDP sockets for wired network to implement**

**a. Peer to Peer Chat b. Multiuser Chat**

**a. Peer to Peer Chat**

**import random**

**from socket import \***

**from threading import Thread, Lock**

**def receiver(serverSocket):**

**global recv\_msg**

**global name**

**while flag:**

**recv\_msg,address=serverSocket.recvfrom(1024) #Receive data from using**

**recvfrom'''**

**if client\_addr == str(address[0]):**

**if recv\_msg == '\*quit\*':**

**print name+" is offline"**

**else:**

**print "\t\t"+recv\_msg**

**flag = True**

**serverSocket=socket(AF\_INET,SOCK\_DGRAM) #Declare a socket for UDP'''**

**serverSocket.bind(('', 12000))**

**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**

**recv\_thread = Thread(target=receiver, args=(serverSocket,))**

**recv\_thread.daemon = True**

**recv\_thread.start()**

**while flag:**

**send\_msg = raw\_input()**

**if send\_msg == '\*quit\*':**

**flag = False**

**serverSocket.sendto(send\_msg,address)**

**'''Send data to client using sendto'''**

**B10P2P CLIENT CODE:**

**from socket import \***

**import time**

**from threading import Thread, Lock**

**def receiver(clientSocket):**

**while flag:**

**recv\_msg,address=clientSocket.recvfrom(1024)**

**#Receive data from server using recvfrom'''**

**print "\t\t"+recv\_msg**

**flag = True**

**clientSocket = socket(AF\_INET, SOCK\_DGRAM)**

**addr = ('172.168.255.121', 12000)**

**name = str(raw\_input("Enter Your Name:"))**

**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\*':**

**clientScoket.sendto(send\_msg,addr) #Send data to client using sendto'''**

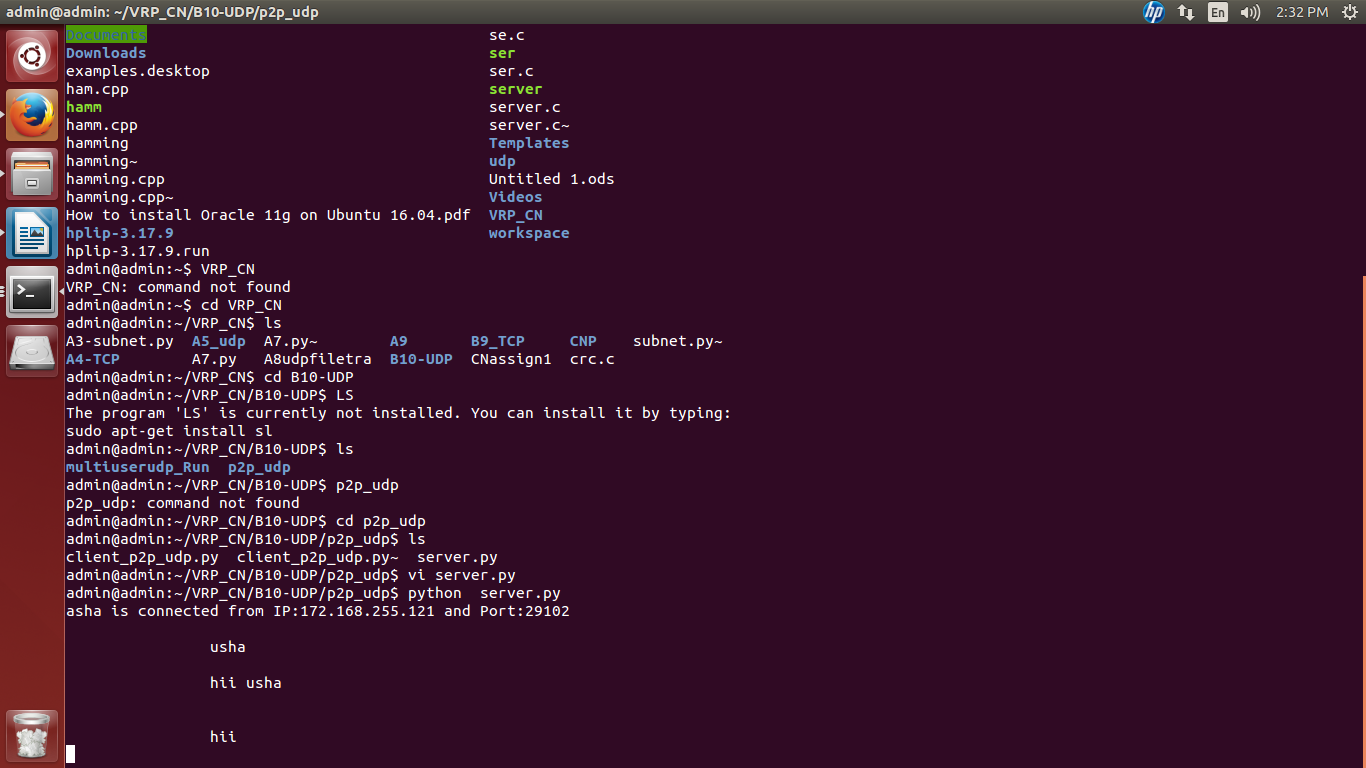
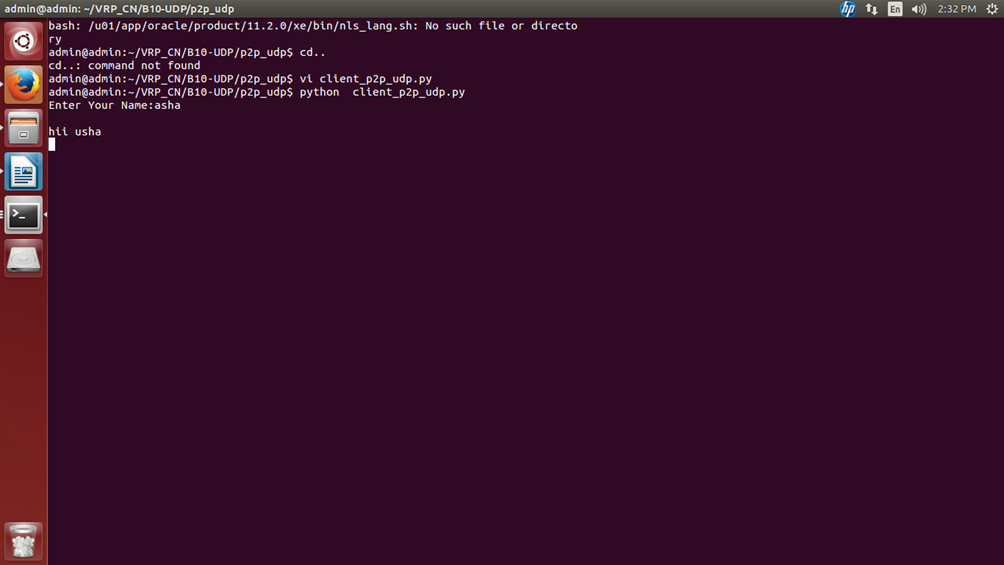
**flag = False**

**recv\_thread.stop()**

**else:**

**clientSocket.sendto(send\_msg,addr)#Send data to client using sendto'''**

**OUTPUT:**

****

**b. Multiuser Chat**

**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**

**multiclient udp:**

**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 = ('172.168.255.121', 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)**

**OUTPUT:**

