Name: G.R.Nithishkumar (20ucs088)

Ex6: Implementing subnetting concepts

**Classful addressing**

**Client:**

import socket as skt

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

server = ('localhost', 7777)

active = True

while active:

ip = input('Enter the ip: ').encode()

client.sendto(ip, server)

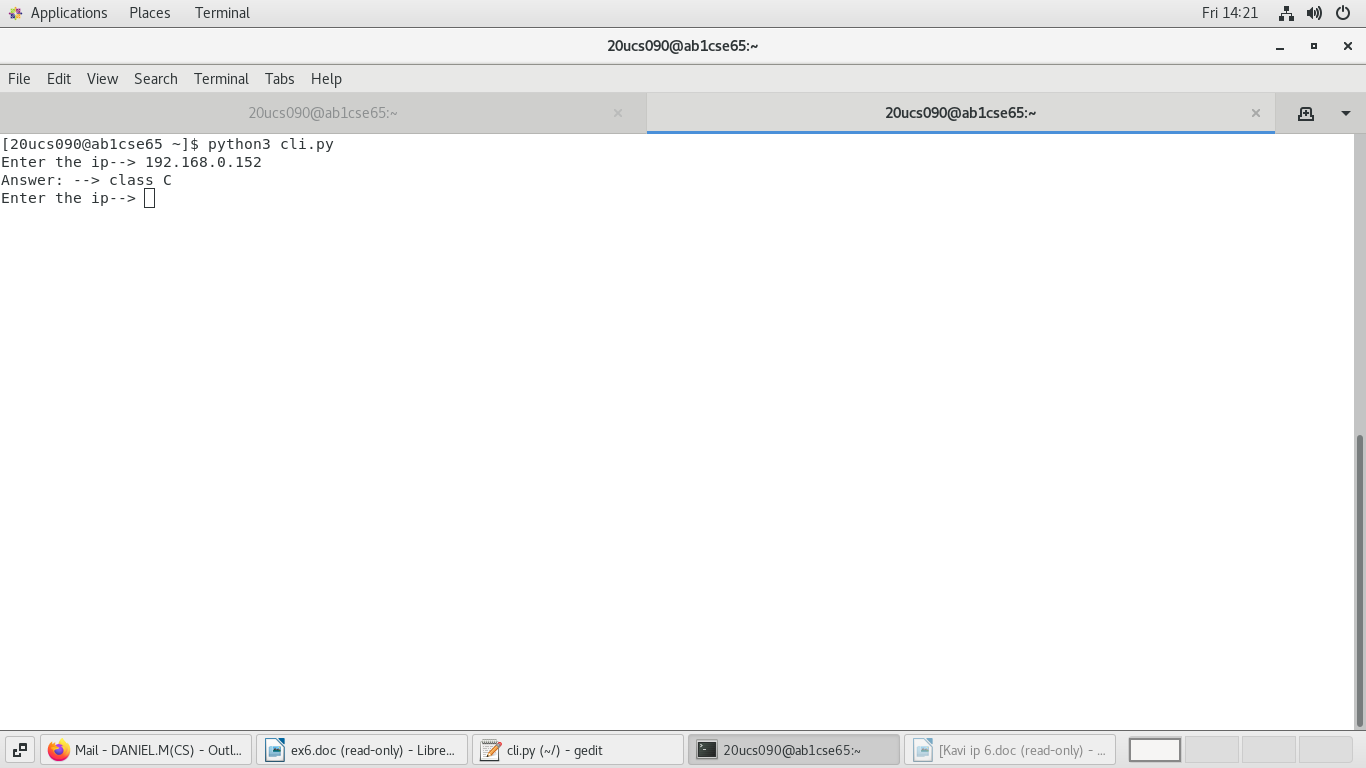
data, addr = client.recvfrom(1024)

data = data.decode()

print(data)

client.close()

**output:**



**server:**

import socket as skt

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

server.bind(('localhost', 7777))

active = True

while active:

data, addr = server.recvfrom(1024)

print(addr)

ip = data.decode()

db = { "A":("1.0.0.0 - 126.255.255.255", "255.0.0.0"),

"B":("128.0.0.0 - 191.255,255,255", "255.255.0.0"),

"C":("192.0.0.0 - 223.255.255.255", "255.255.255.0"),

"D":("224.0.0.0 - 239.255.255.255", "multicasting Address"),

"E":("240.0.0.0 - 254.255.255.255", "Experimental")

}

first\_octet=int(ip.split(".")[0])

if(first\_octet<=126):

server.sendto(str(db["A"]).encode(), addr)

elif(first\_octet==127):

server.sendto("Loop back addressing".encode(), addr)

elif (first\_octet<192):

server.sendto(str(db["B"]).encode(), addr)

elif (first\_octet<224):

server.sendto(str(db["C"]).encode(), addr)

elif(first\_octet<240):

server.sendto(str(db["D"]).encode(), addr)

elif(first\_octet<256):

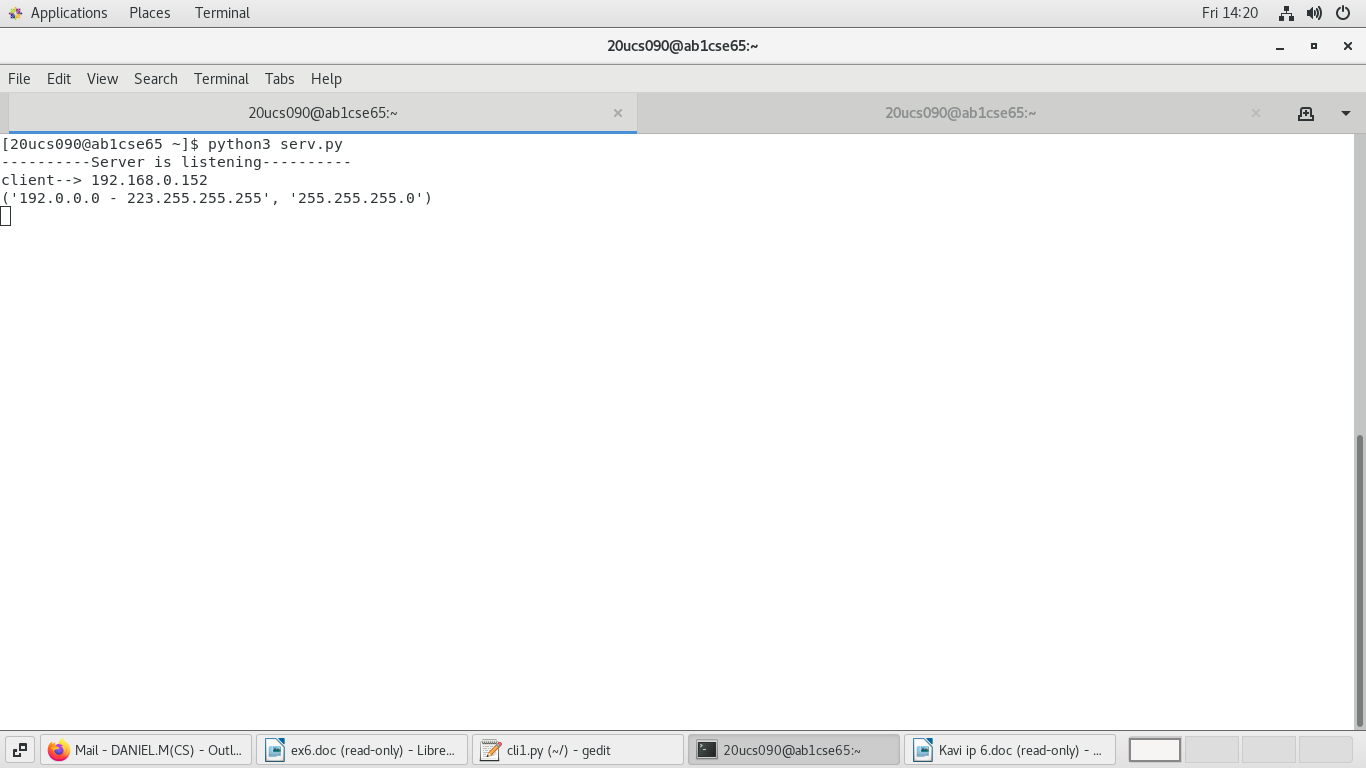
server.sendto(str(db["E"]).encode(), addr)

else:

server.sendto("Invalid ip".encode(), addr)

server.close()

**output:**



**Classless addressing**

**Client:**

import socket

s=socket.socket(socket.AF\_INET,socket.SOCK\_DGRAM)

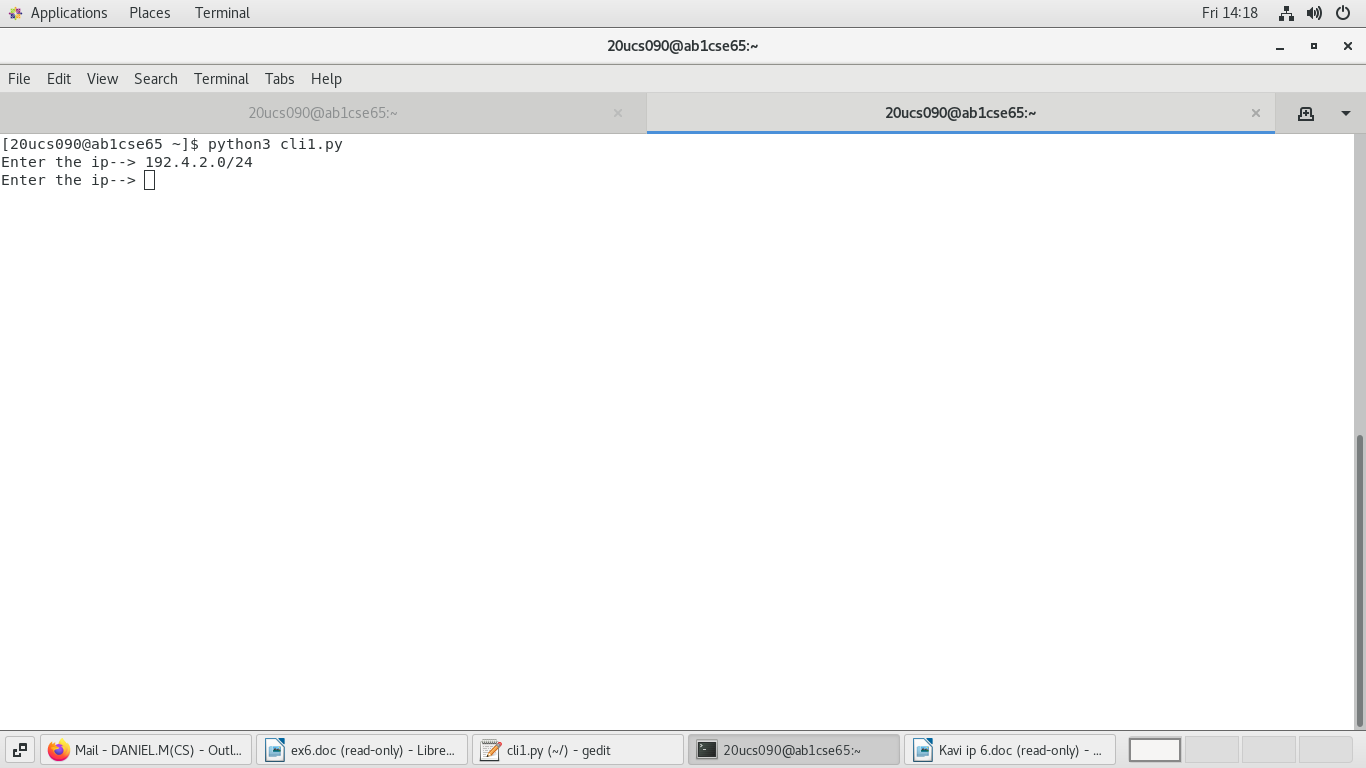
while True:

msg=input("Enter the ip--> ");

s.sendto(msg.encode(),('localhost',5000))

s.close()

**output:**



**server:**

import socket

s=socket.socket(socket.AF\_INET,socket.SOCK\_DGRAM)

s.bind(("localhost",5000))

print("----------Server is listening----------")

while True:

data,address=s.recvfrom(1024)

print("client--> "+data.decode())

ip=data.decode()

from ipaddress import IPv4Network

from ipaddress import IPv4Address

net = IPv4Network(ip)

print(net.num\_addresses)

print(net.prefixlen)

print(net.netmask)

print(net.network\_address)

print(net.broadcast\_address)

print(IPv4Address("192.4.2.12") in net)

for addr in net:

print(addr)

for sn in net.subnets(new\_prefix=25):

print(sn)

s.close()

**output:**

