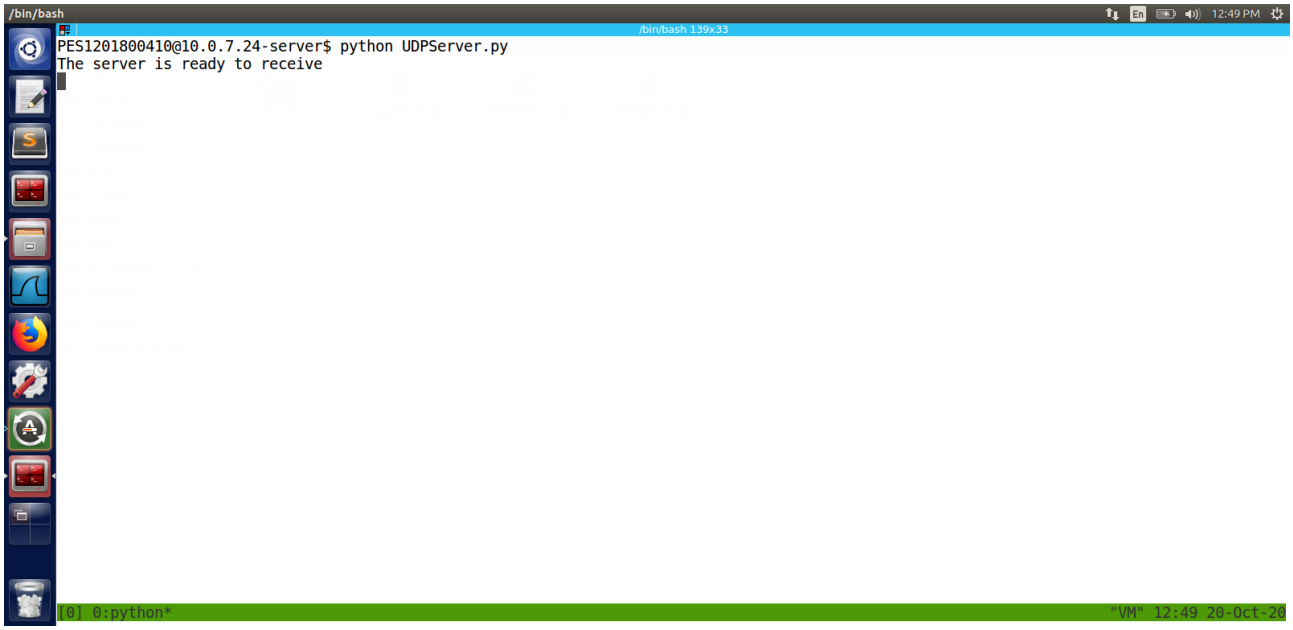
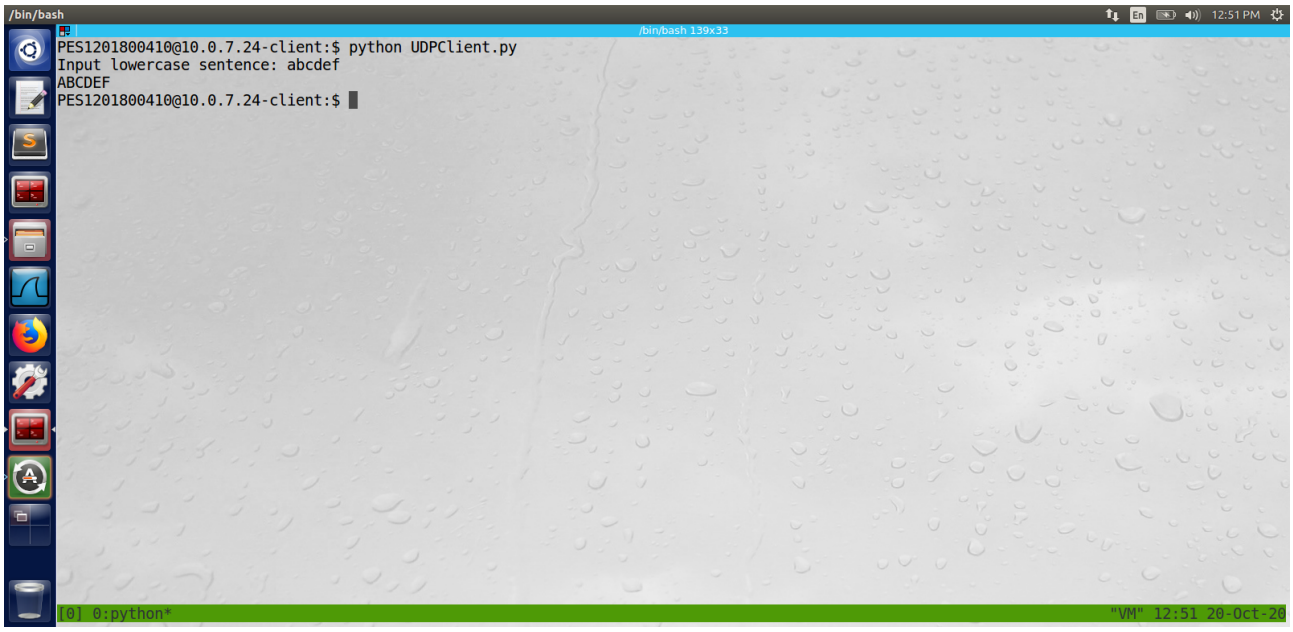
PES1201800410 sem V roll no 24 Prashanth A R

Task 1

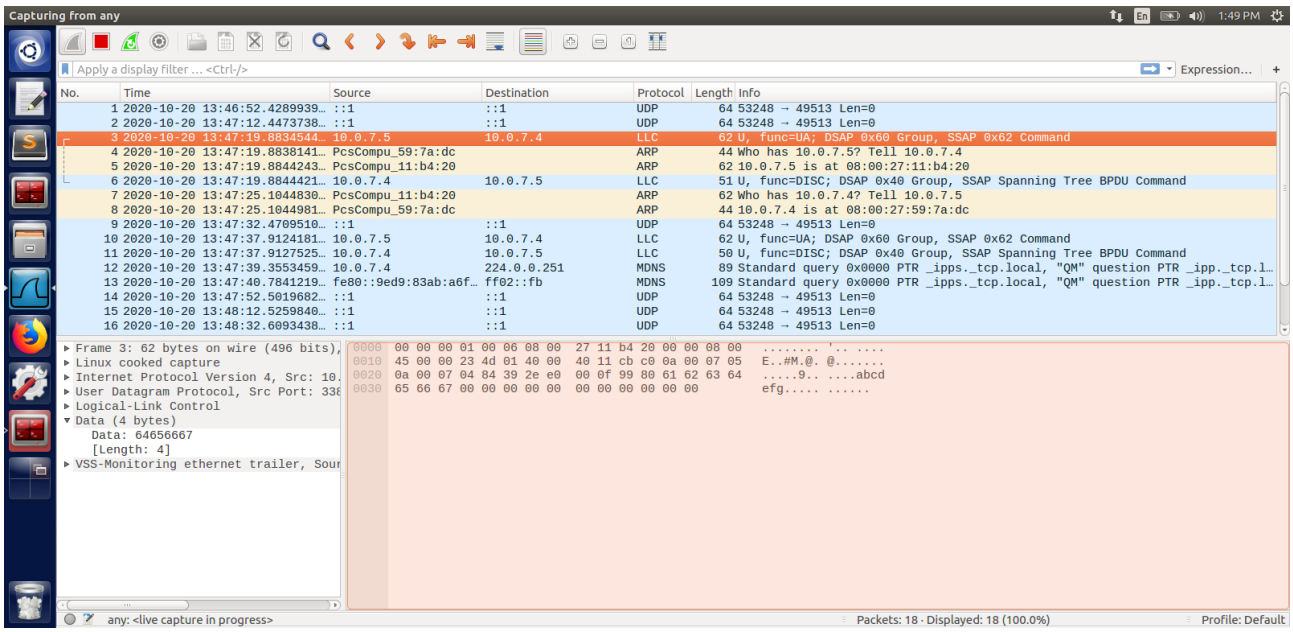
UDP Server



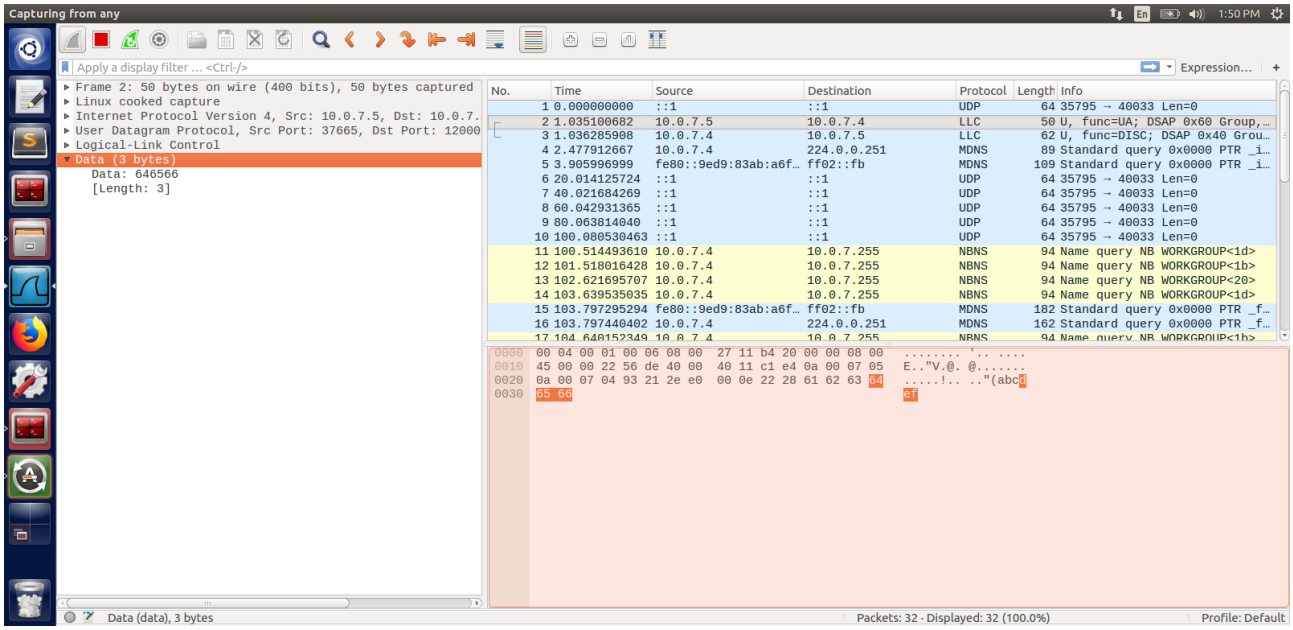
UDP Client



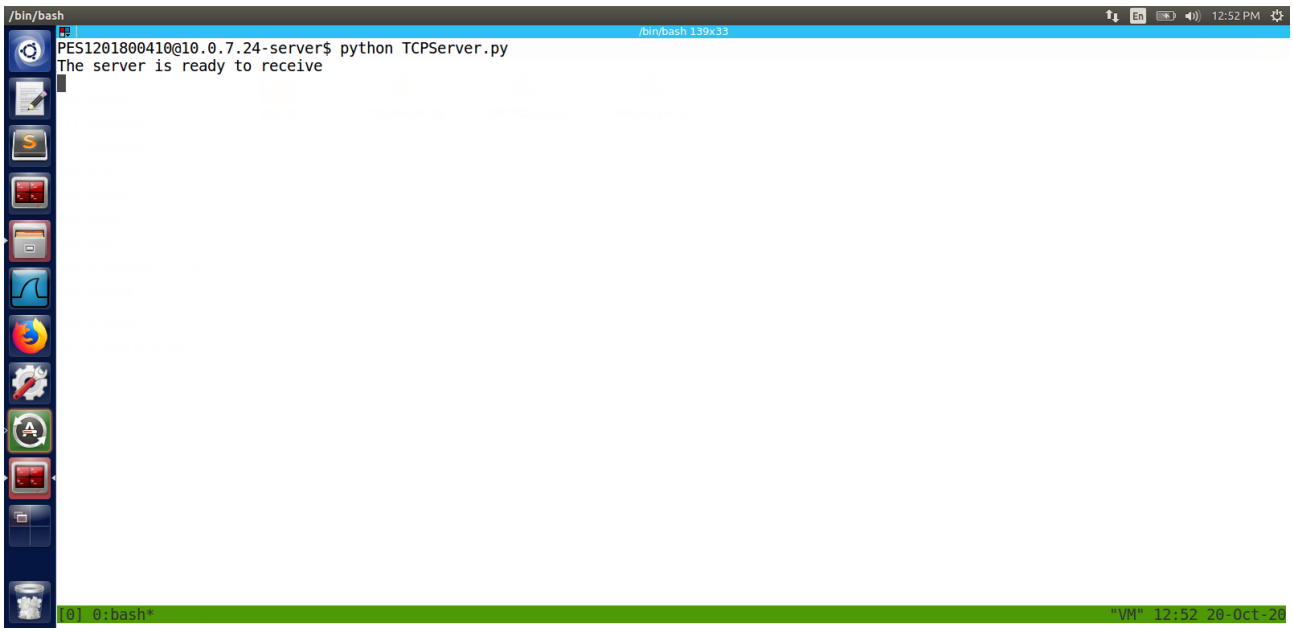
wireshark udp server output



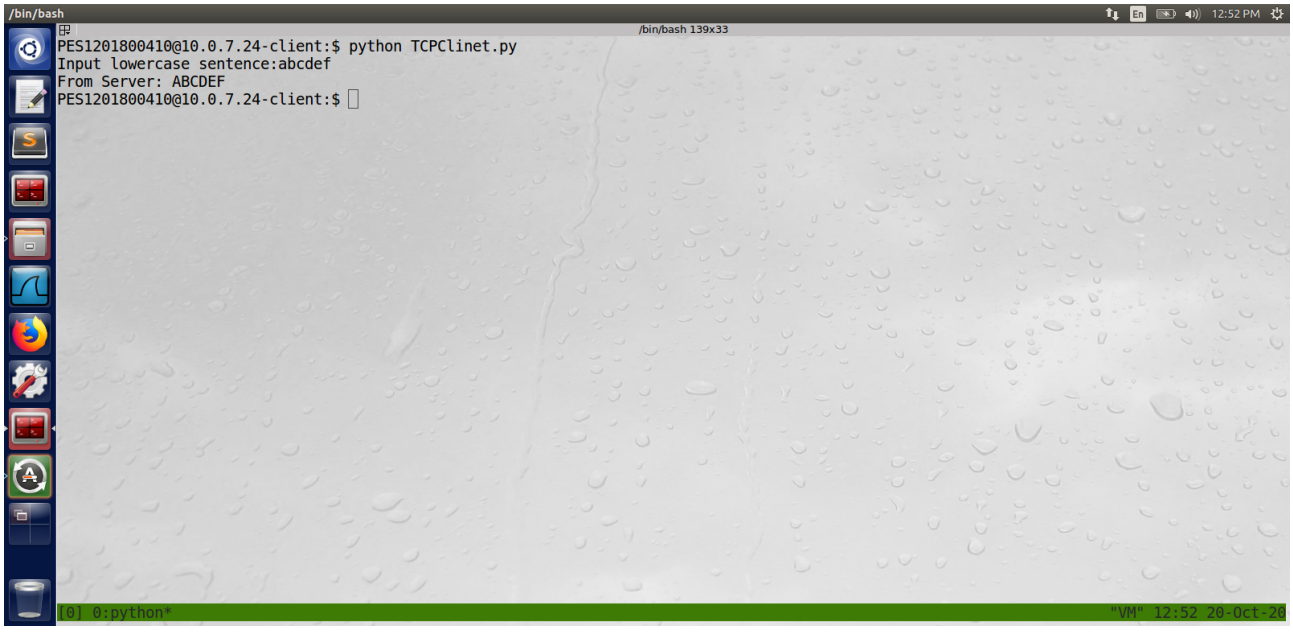
wireshark udp client output



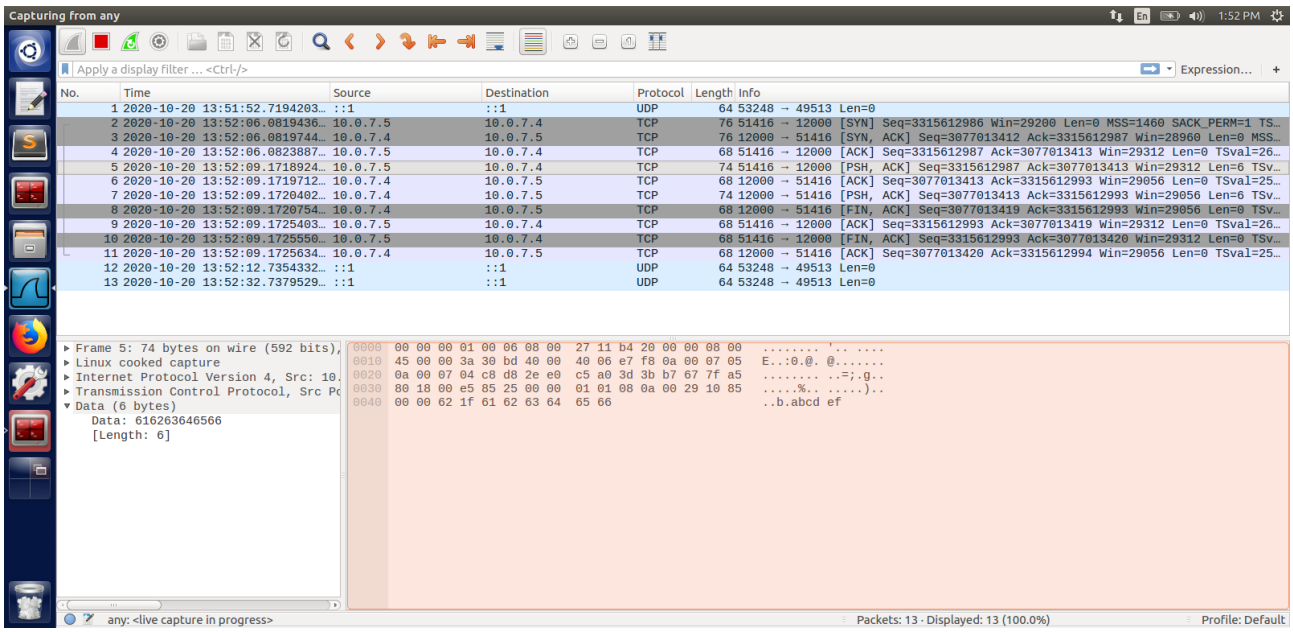
TCP Server



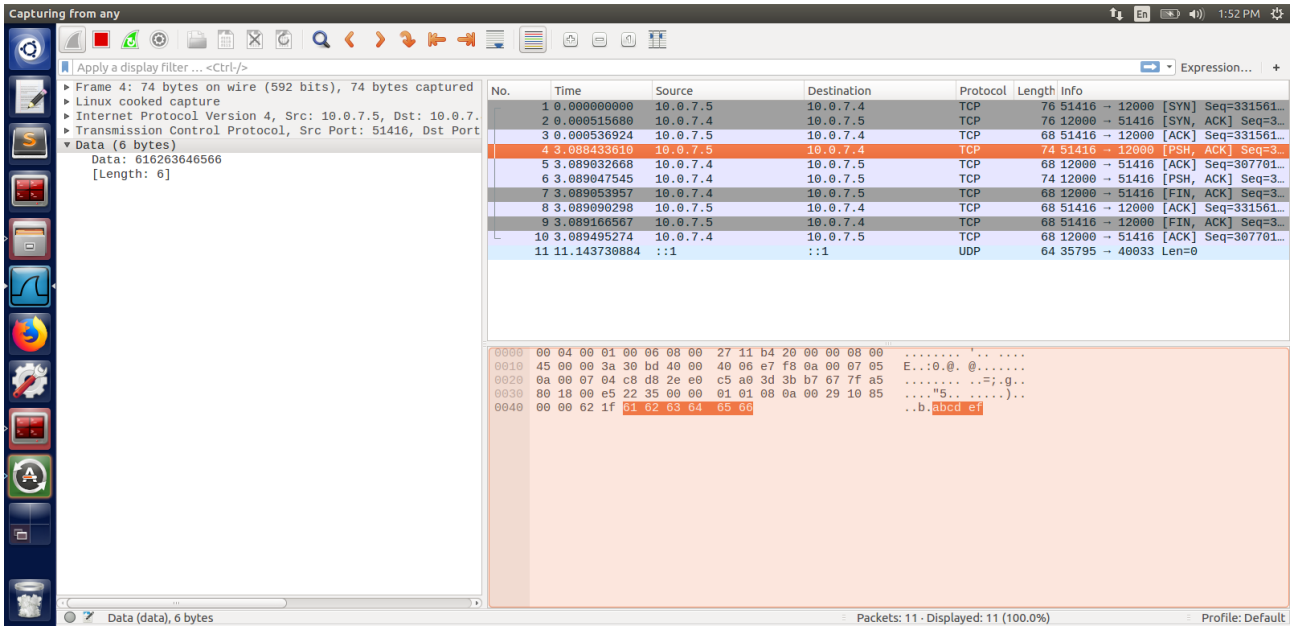
TCP Client



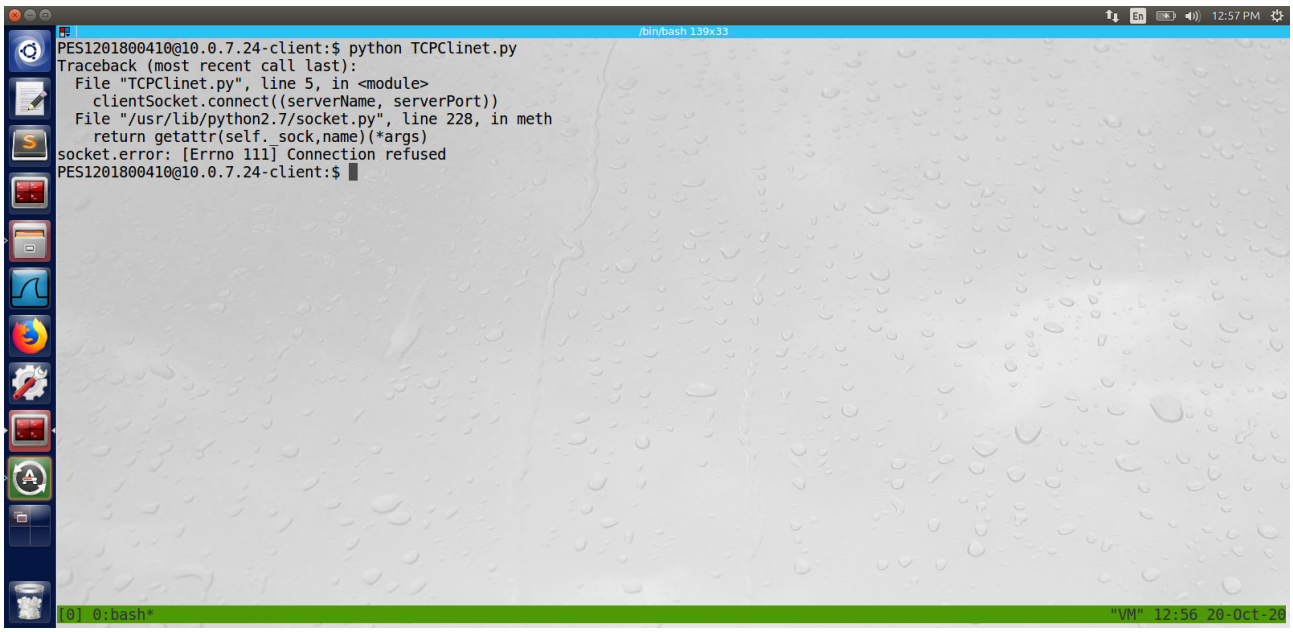
Wireshark TCP server output



Wireshark TCP client output

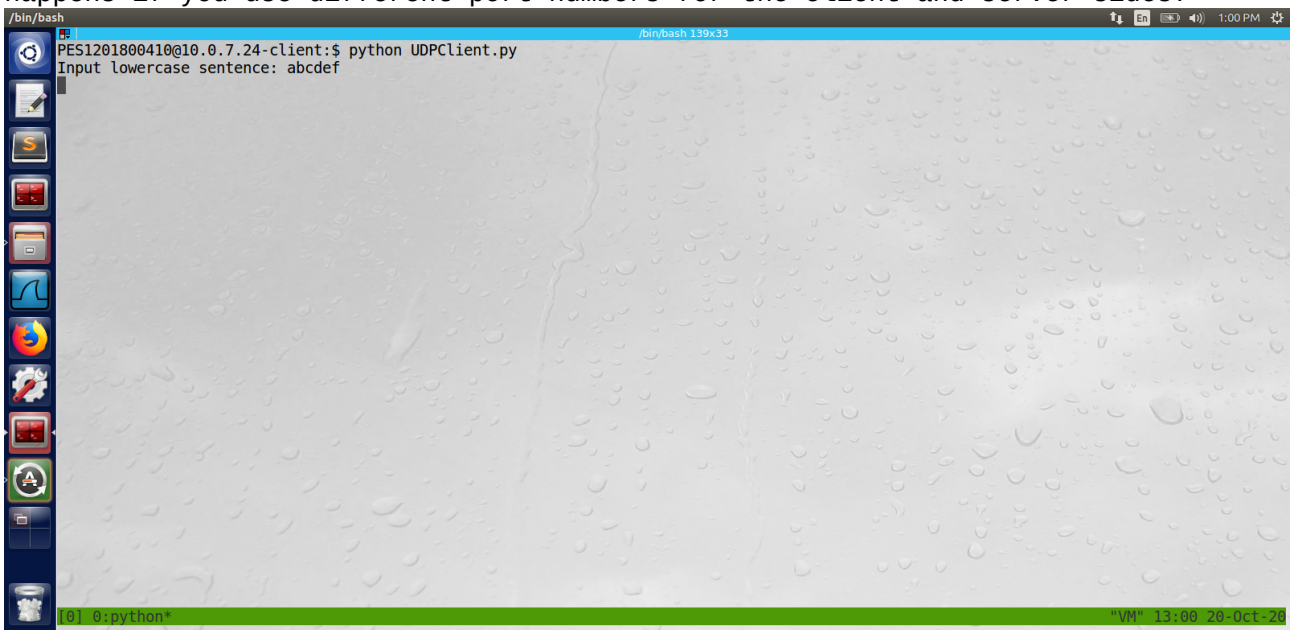


1.Suppose you run TCPClient before you run TCPServer. What happens? Why?
Ans :



we get a connection refused error because there is no server running to send/acknoledge the request sent by the tcp client

2.Suppose you run UDPClient before you run UDPServer. What happens? Why?3.What happens if you use different port numbers for the client and server sides?



We wont recieve a reply back because there is no server to send the response as it is a udp connection it doesn’t really check if the server is running or in close state.

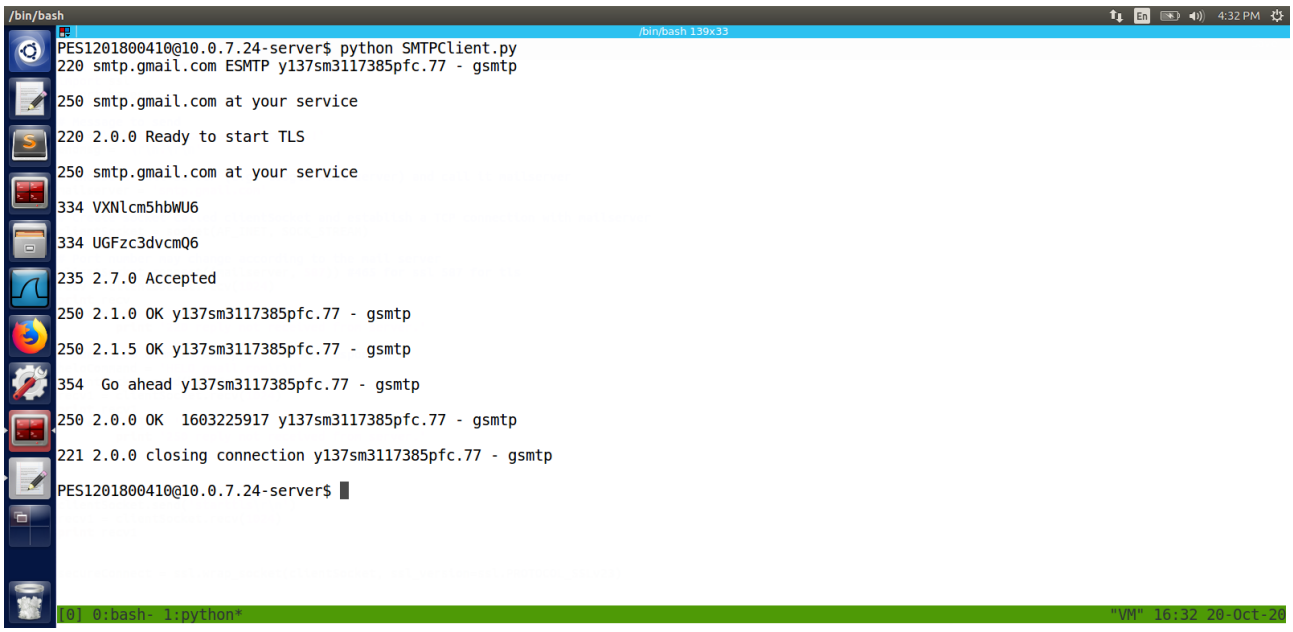
3.What happens if you use different port numbers for the client and server sides?

Same as the above answers tcp client will raise a connection refused error and udp client waits forever for the response to come form the server(which is running in different port so there won’t be any reply)

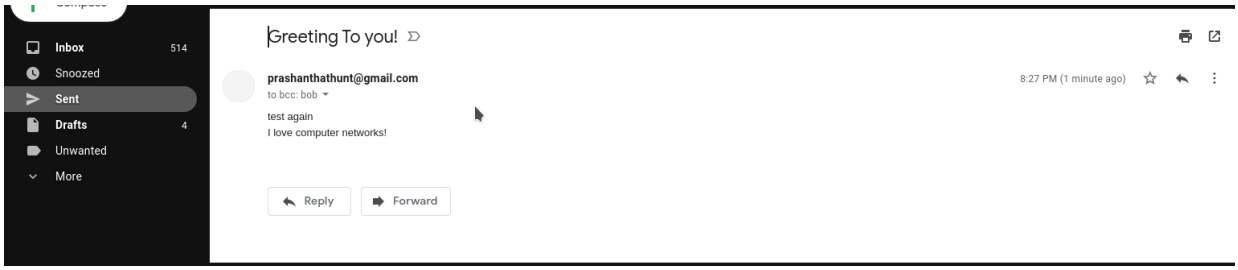
Task 4 Mail Client

```
from socket import *
import ssl
import base64

# Message to send
msg = '\r\nI love computer networks!'
endmsg = '\r\n.\r\n'

# Choose a mail server (e.g. Google mail server) and call it mailserver
mailserver = 'smtp.gmail.com'

# Create socket called clientSocket and establish a TCP connection with
mailserver
clientSocket = socket(AF_INET, SOCK_STREAM)

# Port number may change according to the mail server
clientSocket.connect((mailserver, 587)) #465 for ssl 587 for tls
recv = clientSocket.recv(1024)
print recv
if recv[:3] != '220':
      print '220 reply not received from server.'

# Send HELO command and print server response.
heloCommand = 'HELO gmail.com\r\n'
clientSocket.send(heloCommand)
recv1 = clientSocket.recv(1024)
print recv1
if recv1[:3] != '250':
      print '250 reply not received from server.'



clientSocket.send('starttls\r\n')
recv1 = clientSocket.recv(1024)
print recv1


secureConnect = ssl.wrap_socket(clientSocket, ssl_version=ssl.PROTOCOL_SSLv23)


secureConnect.send(heloCommand)
recv1 = secureConnect.recv(1024)
print recv1

secureConnect.send('AUTH LOGIN\r\n')
recv1 = secureConnect.recv(1024)

print recv1

Username = 'prashanthathunt@gmail.com'
Password = "xxxxxxxxxxxxxxxxxxxxxxxxx”

#print base64.b64encode(Username) + "  username"
#print base64.b64encode(Password) + "  Password"
secureConnect.send(base64.b64encode(Username)+'\r\n')
secureConnect.send(base64.b64encode(Password)+'\r\n')

recv_auth = secureConnect.recv(1024)
print recv_auth

print secureConnect.recv(1024)
```

```
# Send MAIL FROM command and print server response.
mailfrom = 'MAIL FROM: <prashanthathunt@gmail.com>\r\n'
secureConnect.send(mailfrom)
recv2 = secureConnect.recv(1024)
print recv2
if recv2[:3] != '250':
      print '250 reply not received from server.'


# Send RCPT TO command and print server response.
rcptto = 'RCPT TO: <bob@yahoo.com>\r\n'
secureConnect.send(rcptto)
recv3 = secureConnect.recv(1024)
print recv3
if recv3[:3] != '250':
      print '250 reply not received from server.'

# Send DATA command and print server response.
data = 'DATA\r\n'
secureConnect.send(data)
recv4 = secureConnect.recv(1024)
print recv4
if recv4[:3] != '354':
      print '354 reply not received from server.'

# Send message data.
secureConnect.send('SUBJECT: Greeting To you!\r\n')
secureConnect.send('test again')
secureConnect.send(msg)

# Message ends with a single period.
secureConnect.send(endmsg)
recv5 = secureConnect.recv(1024)
print recv5
if recv5[:3] != '250':
      print '250 reply not received from server.'

# Send QUIT command and get server response.
quitcommand = 'QUIT\r\n'
secureConnect.send(quitcommand)
recv6 = secureConnect.recv(1024)
print recv6
if recv6[:3] != '221':
      print '221 reply not received from server.'


secureConnect.close()
clientSocket.close()
```



cross check in gmail using web browser



wireshark output