

COMPUTER NETWORKS LABORATORY

WEEK #5

NAME : SIRI S

SEMESTER : 4

SECTION :H

SRN : PESIUG19CS485

Objective:

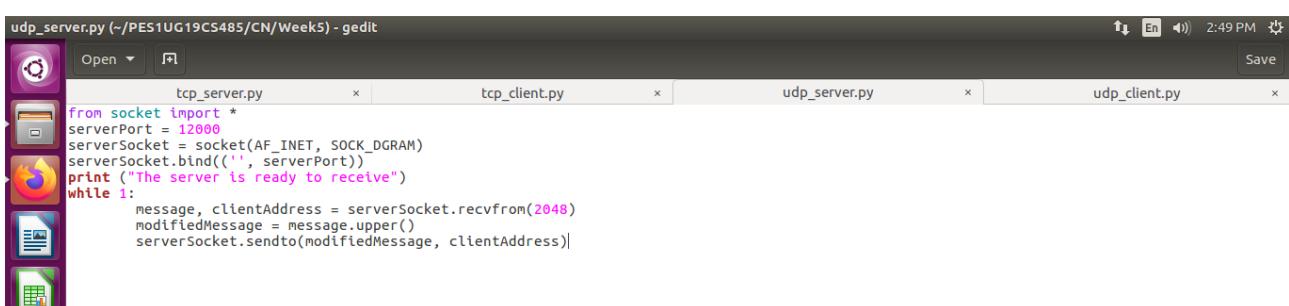
To develop a simple Client-Server application using TCP and UDP.

Task 1:

1. Create an application that will
 - a. Convert lowercase letters to uppercase
 - e.g. [a...z] to [A...Z]
 - code will not change any special characters, e.g. &*!
 - b. If the character is in uppercase, the program must not alter
2. Create Socket API both for client and server.
3. Must take the server address and port from the Command Line Interface (CLI).

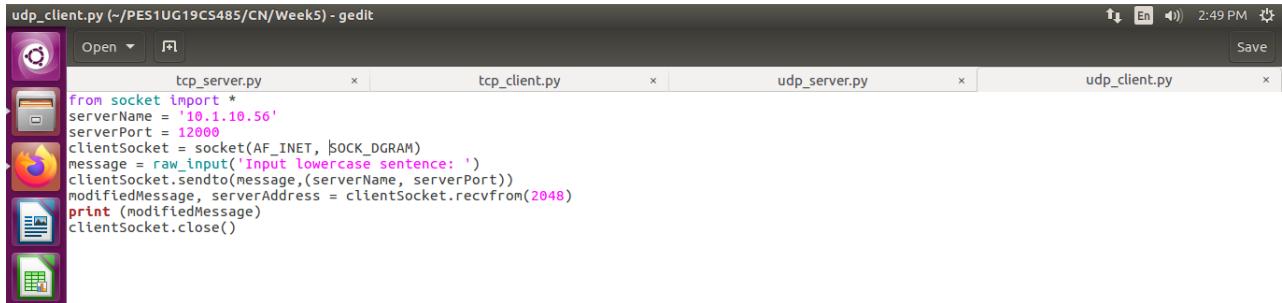
1. UDP CONNECTION:

UDP Server:



```
tcp_server.py (~-/PESIUG19CS485/CN/Week5) - gedit
Open  Save
tcp_server.py  x  tcp_client.py  x  udp_server.py  x  udp_client.py  x
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(("", serverPort))
print ("The server is ready to receive")
while 1:
    message, clientAddress = serverSocket.recvfrom(2048)
    modifiedMessage = message.upper()
    serverSocket.sendto(modifiedMessage, clientAddress)
```

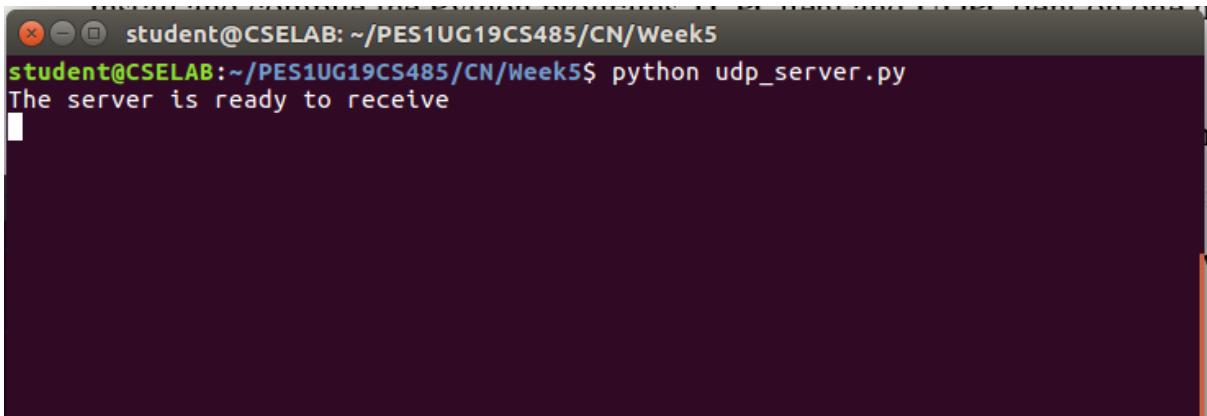
UDP Client:



```
udp_client.py (~-/PES1UG19CS485/CN/Week5) - gedit
tcp_server.py      x |      tcp_client.py      x |      udp_server.py      x |      udp_client.py      x |
Save
From socket import *
serverName = '10.1.10.56'
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_DGRAM)
message = raw_input('Input lowercase sentence: ')
clientSocket.sendto(message,(serverName, serverPort))
modifiedMessage, serverAddress = clientSocket.recvfrom(2048)
print (modifiedMessage)
clientSocket.close()
```

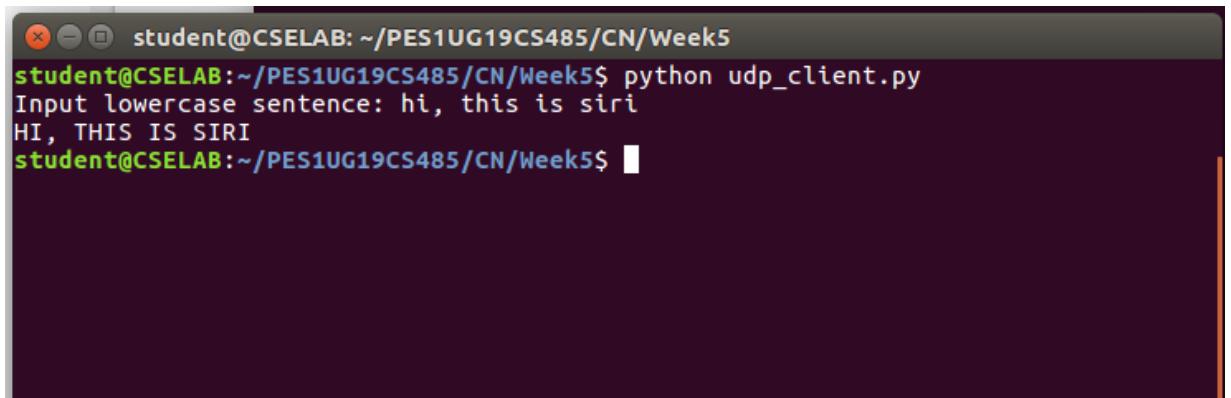
UDP CONNECTION BETWEEN SERVER AND CLIENT:

UDP Server:



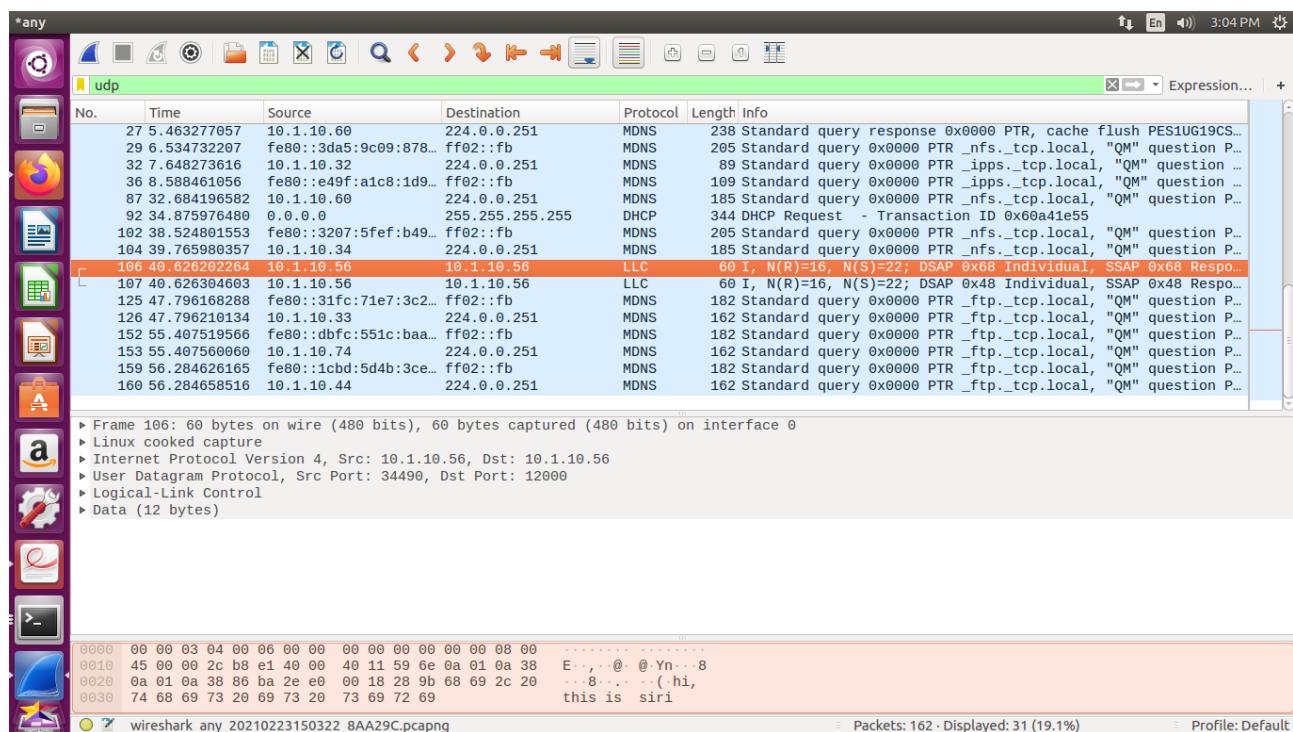
```
student@CSELAB: ~/PES1UG19CS485/CN/Week5
student@CSELAB:~/PES1UG19CS485/CN/Week5$ python udp_server.py
The server is ready to receive
```

UDP Client:



```
student@CSELAB: ~/PES1UG19CS485/CN/Week5
student@CSELAB:~/PES1UG19CS485/CN/Week5$ python udp_client.py
Input lowercase sentence: hi, this is siri
HI, THIS IS SIRI
student@CSELAB:~/PES1UG19CS485/CN/Week5$
```

WIRESHARK CAPTURE FOR UDP CONNECTION:



2. TCP CONNECTION:

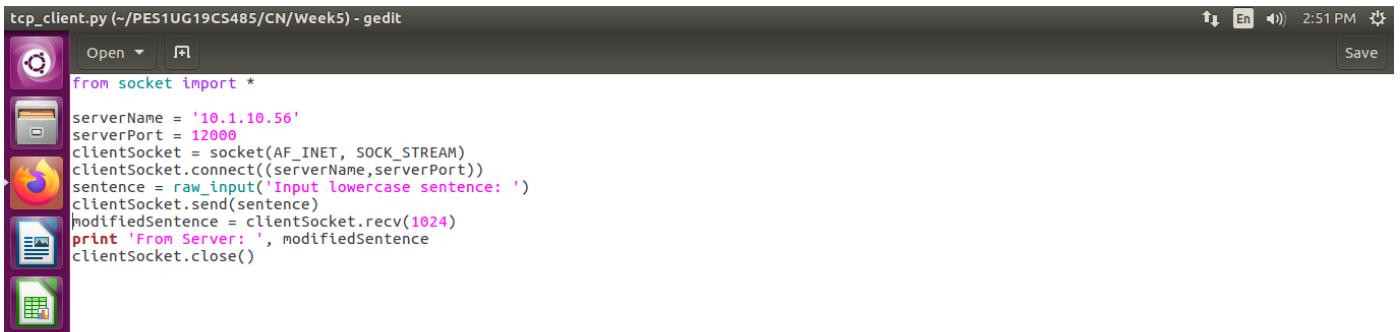
TCP Server:

```
tcp_server.py (~/PES1UG19CS485/CN/Week5) - gedit
from socket import *

serverPort = 12000
serverSocket = socket(AF_INET, SOCK_STREAM)
serverSocket.bind(('',serverPort))
serverSocket.listen(1)
print 'The server is ready to receive'

while 1:
    connectionSocket, addr = serverSocket.accept()
    sentence = connectionSocket.recv(1024)
    capitalizedSentence = sentence.upper()
    connectionSocket.send(capitalizedSentence)
    connectionSocket.close()
```

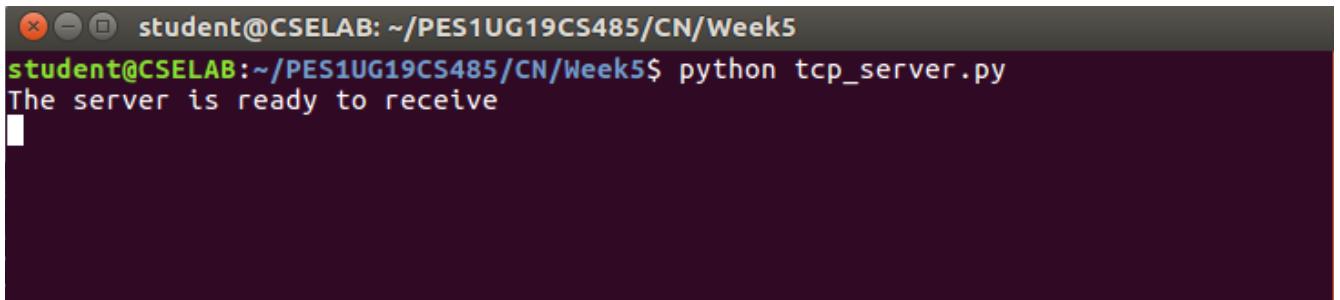
TCP Client:



```
tcp_client.py (~/PES1UG19CS485/CN/Week5) - gedit
from socket import *
serverName = '10.1.10.56'
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName,serverPort))
sentence = raw_input('Input lowercase sentence: ')
clientSocket.send(sentence)
modifiedSentence = clientSocket.recv(1024)
print 'From Server: ', modifiedSentence
clientSocket.close()
```

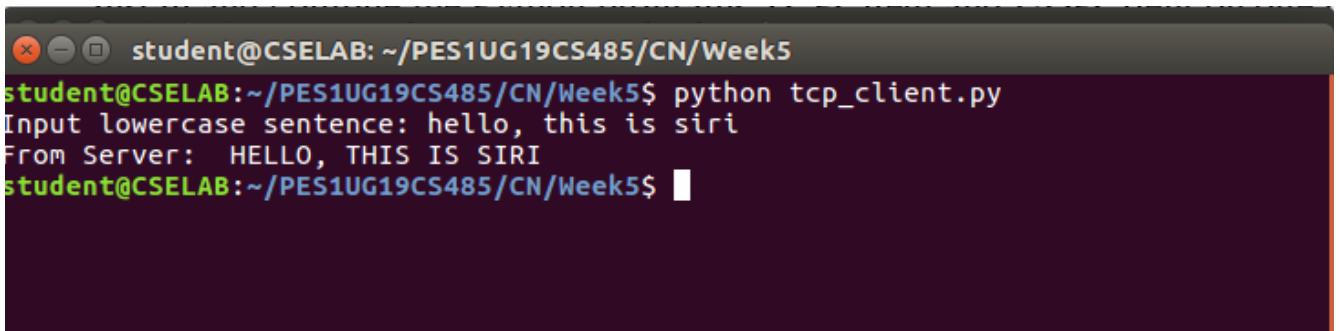
TCP CONNECTION BETWEEN SERVER AND CLIENT:

TCP Server:



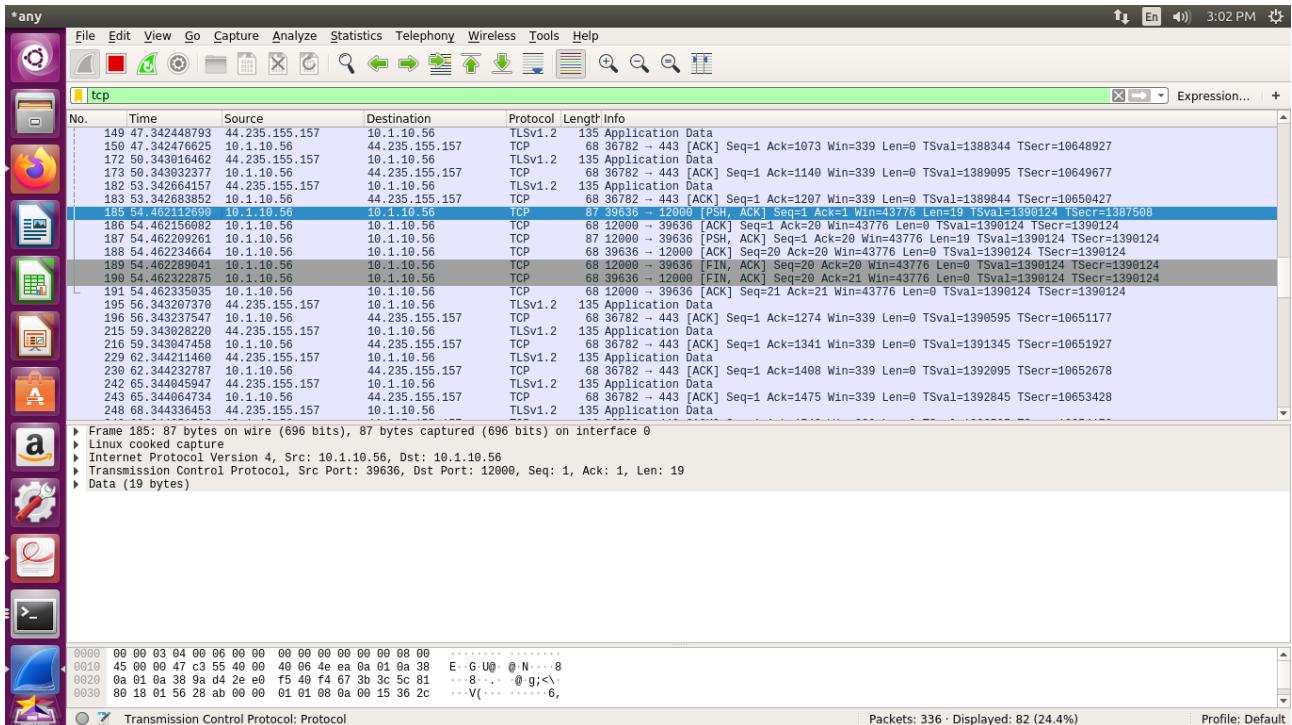
```
student@CSELAB: ~/PES1UG19CS485/CN/Week5
student@CSELAB:~/PES1UG19CS485/CN/Week5$ python tcp_server.py
The server is ready to receive
```

TCP Client:



```
student@CSELAB: ~/PES1UG19CS485/CN/Week5
student@CSELAB:~/PES1UG19CS485/CN/Week5$ python tcp_client.py
Input lowercase sentence: hello, this is siri
From Server: HELLO, THIS IS SIRI
student@CSELAB:~/PES1UG19CS485/CN/Week5$
```

WIRESHARK CAPTURE FOR TCP CONNECTION:



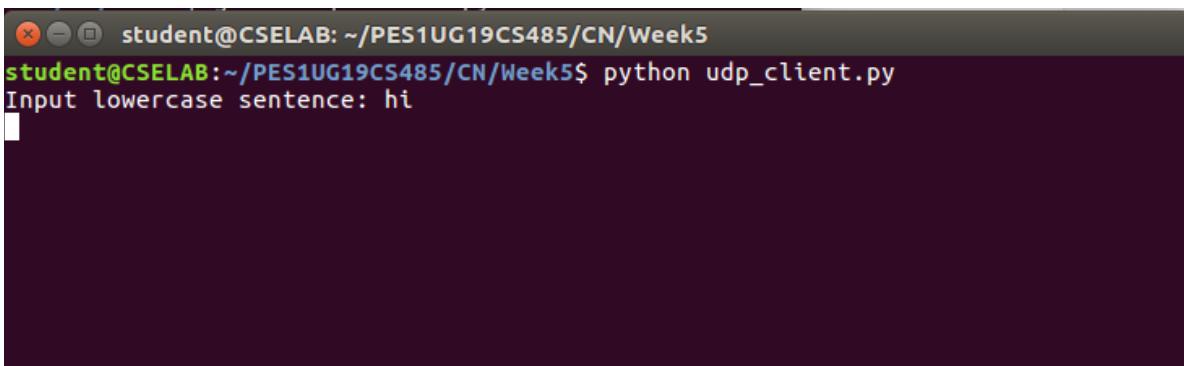
3. PROBLEMS:

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

```
student@CSELAB: ~/PES1UG19CS485/CN/Week5
student@CSELAB:~/PES1UG19CS485/CN/Week5$ python tcp_client.py
Traceback (most recent call last):
  File "tcp_client.py", line 6, in <module>
    clientSocket.connect((serverName,serverPort))
  File "/usr/lib/python2.7/socket.py", line 228, in meth
    return getattr(self._sock,name)(*args)
socket.error: [Errno 111] Connection refused
student@CSELAB:~/PES1UG19CS485/CN/Week5$
```

This is because a TCP Connection between two socket interfaces only when a host machine listens to requests on a given IP address and port number and accepts connections made by another machine at the same address and port.

2. Suppose you run UDPClient before you run UDPServer. What happens? Why?



A screenshot of a terminal window titled "student@CSELAB: ~/PES1UG19CS485/CN/Week5". The window shows the command "python udp_client.py" being run, followed by the input "Input lowercase sentence: hi".

No error occurs because UDP does not require a prior connection to be established between the host machines for data transfer to begin.

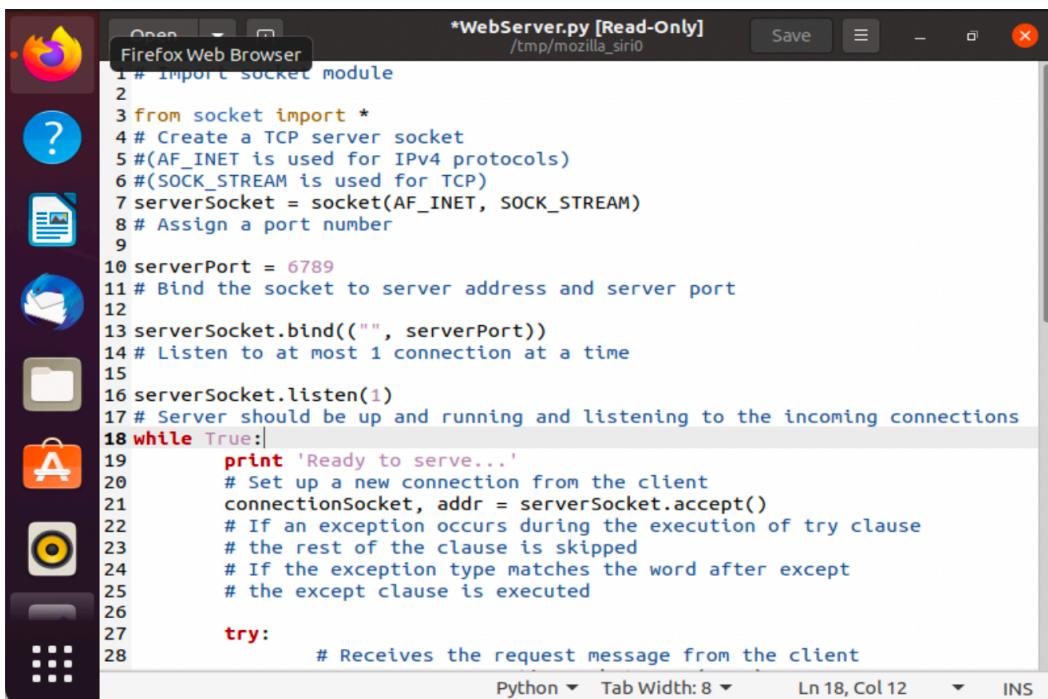
3. What happens if you use different port numbers for the client and server sides?

In case of TCP Connection, it leads to a ConnectionRefusedError because the server socket application is not listening to requests at the same port number as the client socket is trying to connect with.

In case of UDP Connection, no error is obtained as no prior connection is required to be established .

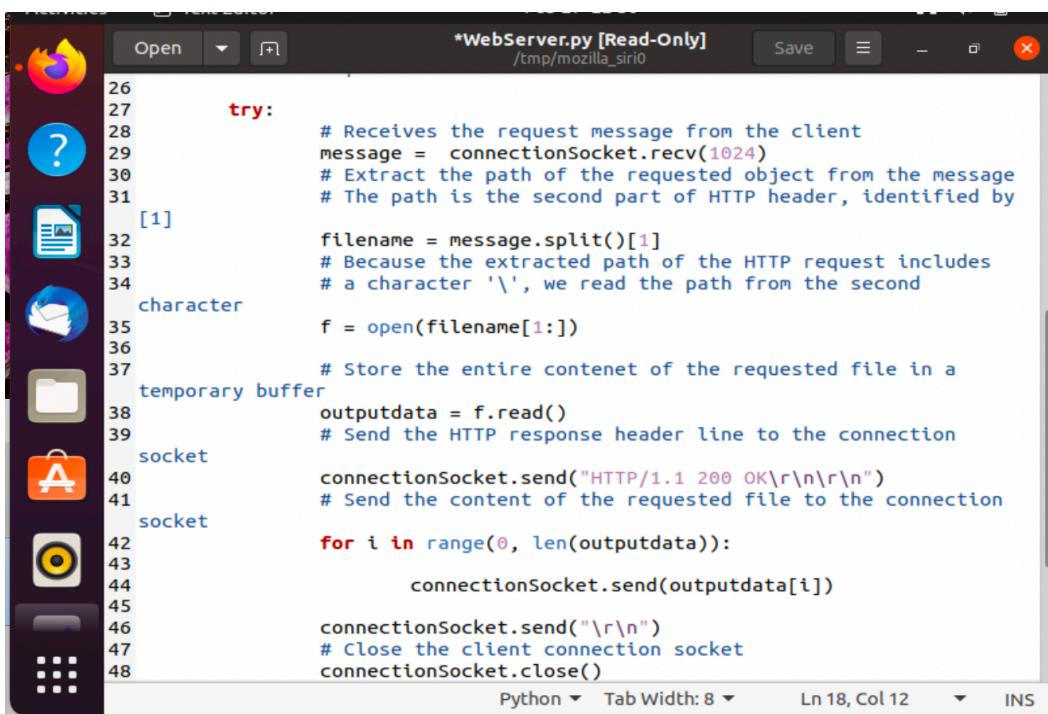
Task 2: WEB SERVER

WebServer.py



```
1 # Import socket module
2
3 from socket import *
4 # Create a TCP server socket
5 #(AF_INET is used for IPv4 protocols)
6 #(SOCK_STREAM is used for TCP)
7 serverSocket = socket(AF_INET, SOCK_STREAM)
8 # Assign a port number
9
10 serverPort = 6789
11 # Bind the socket to server address and server port
12
13 serverSocket.bind(("", serverPort))
14 # Listen to at most 1 connection at a time
15
16 serverSocket.listen(1)
17 # Server should be up and running and listening to the incoming connections
18 while True:
19     print 'Ready to serve...'
20     # Set up a new connection from the client
21     connectionSocket, addr = serverSocket.accept()
22     # If an exception occurs during the execution of try clause
23     # the rest of the clause is skipped
24     # If the exception type matches the word after except
25     # the except clause is executed
26
27     try:
28         # Receives the request message from the client
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
```

Python Tab Width: 8 Ln 18, Col 12 INS

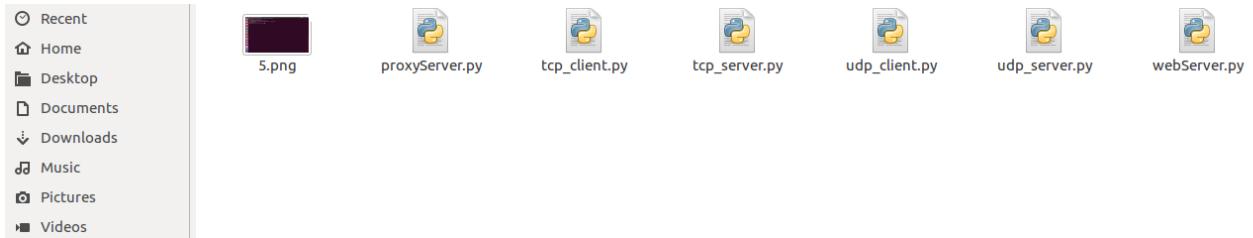


```
26
27     try:
28         # Receives the request message from the client
29         message = connectionSocket.recv(1024)
30         # Extract the path of the requested object from the message
31         # The path is the second part of HTTP header, identified by
32         [1]
33         filename = message.split()[1]
34         # Because the extracted path of the HTTP request includes
35         # a character '\', we read the path from the second
36         character
37         f = open(filename[1:])
38
39         # Store the entire contenet of the requested file in a
40         # temporary buffer
41         outputdata = f.read()
42         # Send the HTTP response header line to the connection
43         connectionSocket.send("HTTP/1.1 200 OK\r\n\r\n")
44         # Send the content of the requested file to the connection
45         socket
46         for i in range(0, len(outputdata)):
47             connectionSocket.send(outputdata[i])
48
49
50         except:
51             # Send HTTP response message for file not found
52             connectionSocket.send("HTTP/1.1 404 Not Found\r\n\r\n")
53             connectionSocket.send("<html><head></head><body><h1>404 Not
54             Found</h1></body></html>\r\n")
55             # Close the client connection socket
56             connectionSocket.close()
57
58
59
```

Python Tab Width: 8 Ln 18, Col 12 INS

```
59
```

Saving an image in the same directory as WebServer.py



On accessing localhost with the port number specified in the code from Firefox:

