

CYCLE 2

AIM:

Using TCP/IP sockets, write a client-server program to make client sending the file name and the server to send back the contents of the requested file if present.

Using UDP sockets, write a client-server program to make client sending the file name and the server to send back the contents of the requested file if present.

PROGRAM:

Aim:

Using TCP/IP sockets, write a client-server program to make client sending the file name and the server to send back the contents of the requested file if present.

CODE:

1. ClientTCP.py.

```
from socket import *
ServerName = '127.0.0.1'
ServerPort = 12000
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((ServerName, ServerPort))
sentence = input("\n Enter file name: ")
clientSocket.send(sentence.encode())
filecontents = clientSocket.recv(1024).decode()
print('\n From Server: \n')
print(filecontents)
clientSocket.close()
```

2. ServerTCP.py.

```
from socket import *
ServerName = "127.0.0.1"
ServerPort = 12000
ServerSocket = socket(AF_INET, SOCK_STREAM)
ServerSocket.bind((ServerName, ServerPort))
ServerSocket.listen(1)
while 1:
    print("The server is ready to receive.")
```



```

connectionSocket, addr = serverSocket.accept()
sentence = connectionSocket.recv(1024).decode()

file = open(sentence, "a")
L = file.read(1024)

connectionSocket.send(L.encode())
print('In Sent contents of ' + sentence)
file.close()
connectionSocket.close()

```

OUTPUT:

* When you run ServerTCP.py.

The server is ready to receive

* When you run ClientTCP.py

Enter file name: ServerTCP.py

From Server:

(The files from ServerTCP.py will be copied and displayed here).

* In ServerTCP.py

The server is ready to receive

Sent contents of ServerTCP.py

The server is ready to receive.

10/10

25/8/23

Aim:

Using UDP sockets, write a client-server program to make client sending the file name and the server to send back the contents of the requested file if present.

Code:

1. ClientUDP.py

```
from socket import *
serverName = "127.0.0.1"
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_DGRAM)

sentence = input("Enter file name: ")
clientSocket.sendto(bytes(sentence, "utf-8"), (serverName, serverPort))

filecontents, serverAddress = clientSocket.recvfrom(2048)
print('In Reply from server: In')
print(filecontents.decode("utf-8"))
# for i in filecontents:
#     print(i, end=" ")
clientSocket.close()
clientSocket.close()
```

2. ServerUDP.py

```
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(("127.0.0.1", serverPort))
```



```
print("The server is ready to receive")
```

```
while 1:
```

```
    sentence, clientAddress = serverSocket.recvfrom(2048)
```

```
    sentence = sentence.decode("utf-8")
```

```
    file = open(sentence, "x")
```

```
    con = file.read(2048)
```

```
    serverSocket.sendto(bytes(con, "utf-8"), clientAddress)
```

```
print("\n Sent contents of", end="")
```

```
print(sentence)
```

```
# for i in sentence:
```

```
    # print(stuli), end="")
```

```
file.close()
```

OUTPUT:

When you run ServerUDP.py

The server is ready to receive

When you run ClientUDP.py

Enter file name: ServerUDP.py

Reply from server:

(The files from ServerUDP.py will be copied and displayed here)

In ServerUDP.py

The server is ready to receive

Sent contents of ServerUDP.py

The server is ready to receive

10/10

28/8/23

PROGRAM:

1.

ClientTCP.py

```
from socket import *
serverName = '127.0.0.1'
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName,serverPort))
sentence = input("\nEnter file name: ")
clientSocket.send(sentence.encode())
filecontents = clientSocket.recv(1024).decode()
print ("\nFrom Server:\n")
print(filecontents)
clientSocket.close()
```

ServerTCP.py

```
from socket import *
serverName="127.0.0.1"
serverPort = 12000
serverSocket = socket(AF_INET,SOCK_STREAM)
serverSocket.bind((serverName,serverPort))
serverSocket.listen(1)
while 1:
    print ("The server is ready to receive")
    connectionSocket, addr = serverSocket.accept()
    sentence = connectionSocket.recv(1024).decode()
    file=open(sentence,"r")

    l=file.read(1024)
    connectionSocket.send(l.encode())
    print ('\ nSent contents of '+ sentence)
    file.close()
    connectionSocket.close()
```

2.

ClientUDP.py

```
from socket import *
```

```
serverName = "127.0.0.1"
```

```
serverPort = 12000
```

```
clientSocket = socket(AF_INET, SOCK_DGRAM)
```

```
sentence = input("\nEnter file name: ")
```

```
clientSocket.sendto(bytes(sentence,"utf-8"),(serverName, serverPort))
```

```
filecontents,serverAddress = clientSocket.recvfrom(2048)
```

```
print ("\nReply from Server:\n")
```

```
print (filecontents.decode("utf-8"))
```

```
# for i in filecontents:
```

```
# print(str(i), end = " ")
```

```
clientSocket.close()
```

```
clientSocket.close()
```

ServerUDP.py

```
from socket import *
```

```
serverPort = 12000
```

```
serverSocket = socket(AF_INET, SOCK_DGRAM)
```

```
serverSocket.bind(("127.0.0.1", serverPort))
```

```
print ("The server is ready to receive")
```

```
while 1:
```

```
sentence, clientAddress = serverSocket.recvfrom(2048)
```

```
sentence = sentence.decode("utf-8")
```

```
file=open(sentence,"r")
```

```
con=file.read(2048)
```

```
serverSocket.sendto(bytes(con,"utf-8"),clientAddress)
```

```
print ("\nSent contents of ", end = " ")
```

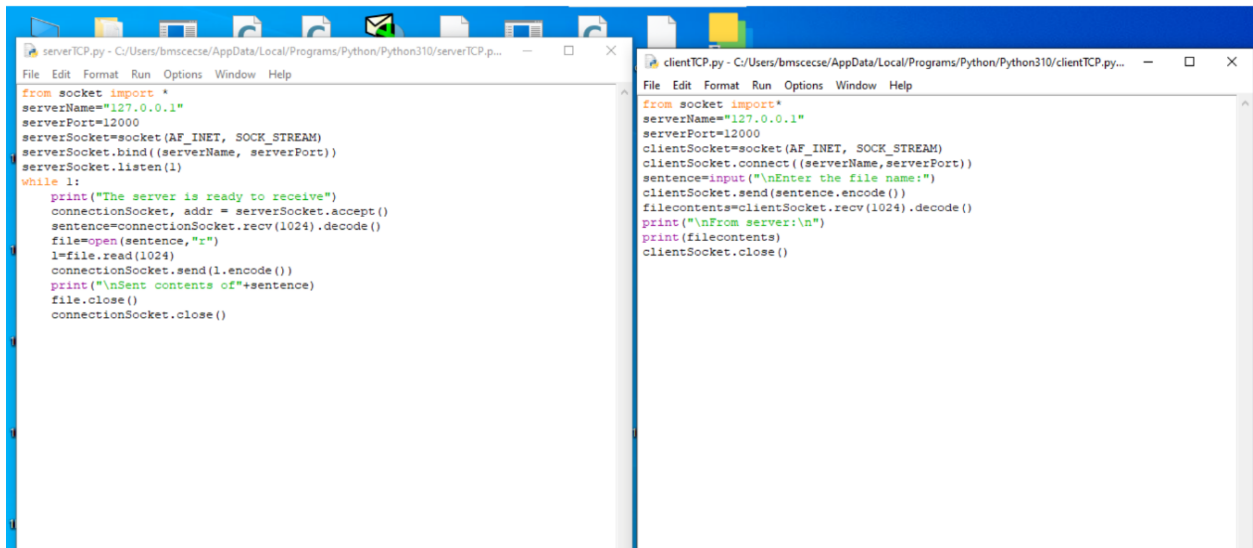
```
print (sentence)
```

```
# for i in sentence:
```

```
# print (str(i), end = " ")
```

```
file.close()
```

Output:

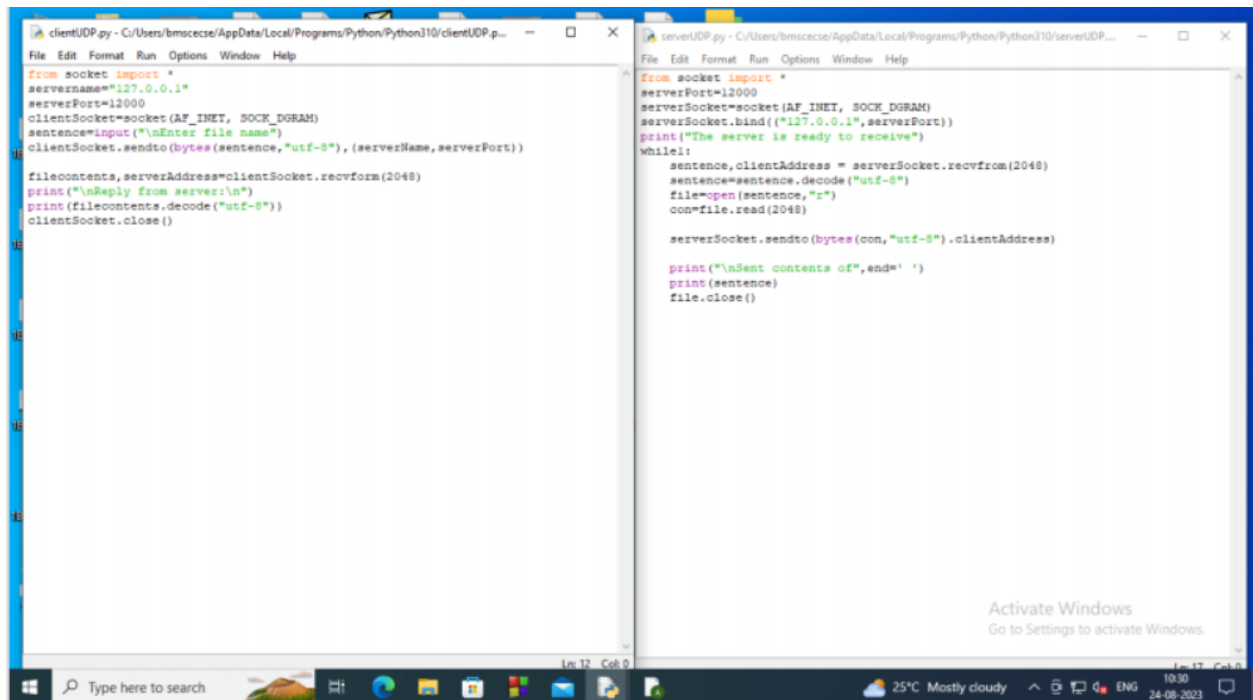


The image shows two side-by-side windows of a Python IDE. The left window, titled 'serverTCP.py', contains the following code:

```
from socket import *
serverName="127.0.0.1"
serverPort=12000
serverSocket=socket(AF_INET, SOCK_STREAM)
serverSocket.bind((serverName, serverPort))
serverSocket.listen(1)
while 1:
    print("The server is ready to receive")
    connectionSocket, addr = serverSocket.accept()
    sentence=connectionSocket.recv(1024).decode()
    file=open(sentence,"r")
    l=file.read(1024)
    connectionSocket.send(l.encode())
    print("\nSent contents of"+sentence)
    file.close()
    connectionSocket.close()
```

The right window, titled 'clientTCP.py', contains the following code:

```
from socket import *
serverName="127.0.0.1"
serverPort=12000
clientSocket=socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName,serverPort))
sentence=input("\nEnter the file name:")
clientSocket.send(sentence.encode())
filecontents=clientSocket.recv(1024).decode()
print("\nFrom server:\n")
print(filecontents)
clientSocket.close()
```



The image shows two side-by-side windows of a Python IDE. The left window, titled 'clientUDP.py', contains the following code:

```
from socket import *
serverName="127.0.0.1"
serverPort=12000
clientSocket=socket(AF_INET, SOCK_DGRAM)
sentence=input("\nEnter file name")
clientSocket.sendto(bytes(sentence,"utf-8"), (serverName,serverPort))

filecontents, serverAddress=clientSocket.recvfrom(2048)
print("\nReply from server:\n")
print(filecontents.decode("utf-8"))
clientSocket.close()
```

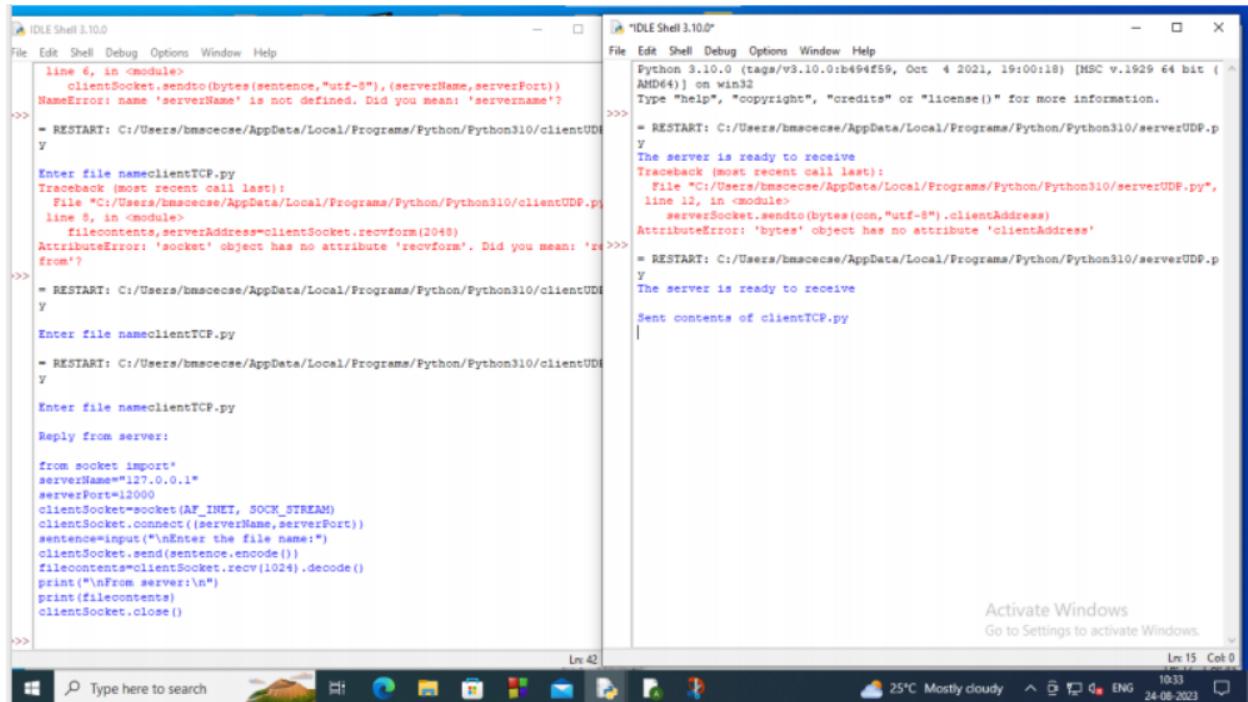
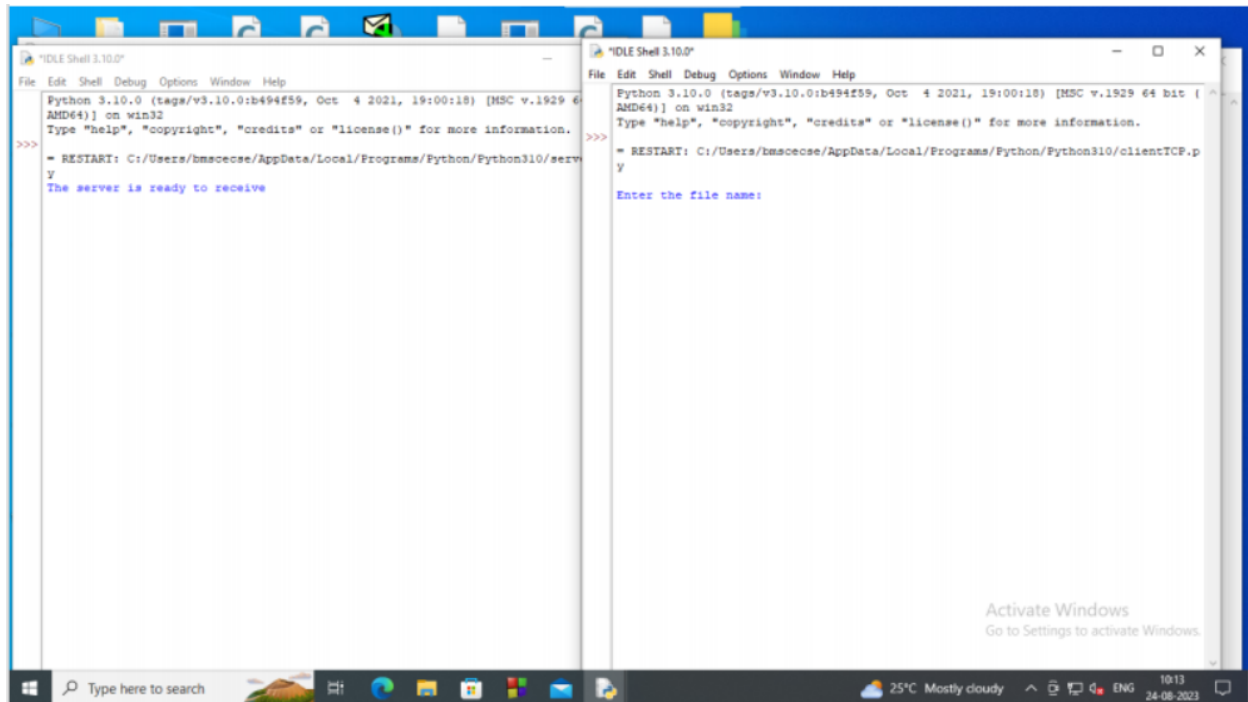
The right window, titled 'serverUDP.py', contains the following code:

```
from socket import *
serverPort=12000
serverSocket=socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(("127.0.0.1",serverPort))
print("The server is ready to receive")
while 1:
    sentence,clientAddress = serverSocket.recvfrom(2048)
    sentence=sentence.decode("utf-8")
    file=open(sentence,"r")
    con=file.read(2048)

    serverSocket.sendto(bytes(con,"utf-8"),clientAddress)

    print("\nSent contents of",end=' ')
    print(sentence)
    file.close()
```

An "Activate Windows" watermark is visible in the bottom right corner of the IDE windows.



"IDLE Shell 3.10.0"

File Edit Shell Debug Options Window Help

Python 3.10.0 (tags/v3.10.0:b494f59, Oct 4 2021, 19:00:18) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/bmscece/AppData/Local/Programs/Python/Python310/serverTCP.py
The server is ready to receive
Traceback (most recent call last):
 File "C:/Users/bmscece/AppData/Local/Programs/Python/Python310/serverTCP.py", line 10, in <module>
 sentence=connectionSocket.recv(1024).decoder()
AttributeError: 'bytes' object has no attribute 'decoder'. Did you mean: 'd'
>>>
===== RESTART: C:/Users/bmscece/AppData/Local/Programs/Python/Python310/serverTCP.py =====
The server is ready to receive
Sent contents of serverTCP.py
The server is ready to receive

IDLE Shell 3.10.0

File Edit Shell Debug Options Window Help

Python 3.10.0 (tags/v3.10.0:b494f59, Oct 4 2021, 19:00:18) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/bmscece/AppData/Local/Programs/Python/Python310/clientTCP.py
Enter the file name:serverTCP.py
From server:
from socket import *
serverName="127.0.0.1"
serverPort=12000
serverSocket=socket(AF_INET, SOCK_STREAM)
serverSocket.bind((serverName, serverPort))
serverSocket.listen(1)
while 1:
 print("The server is ready to receive")
 connectionSocket, addr = serverSocket.accept()
 sentence=connectionSocket.recv(1024).decode()
 file=open(sentence,"r")
 l=file.read(1024)
 connectionSocket.send(l.encode())
 print("\nSent contents of "+sentence)
 file.close()
 connectionSocket.close()
>>> |

Activate Windows
Go to Settings to activate Windows.