WEEK 15

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:

ClientTCP.py

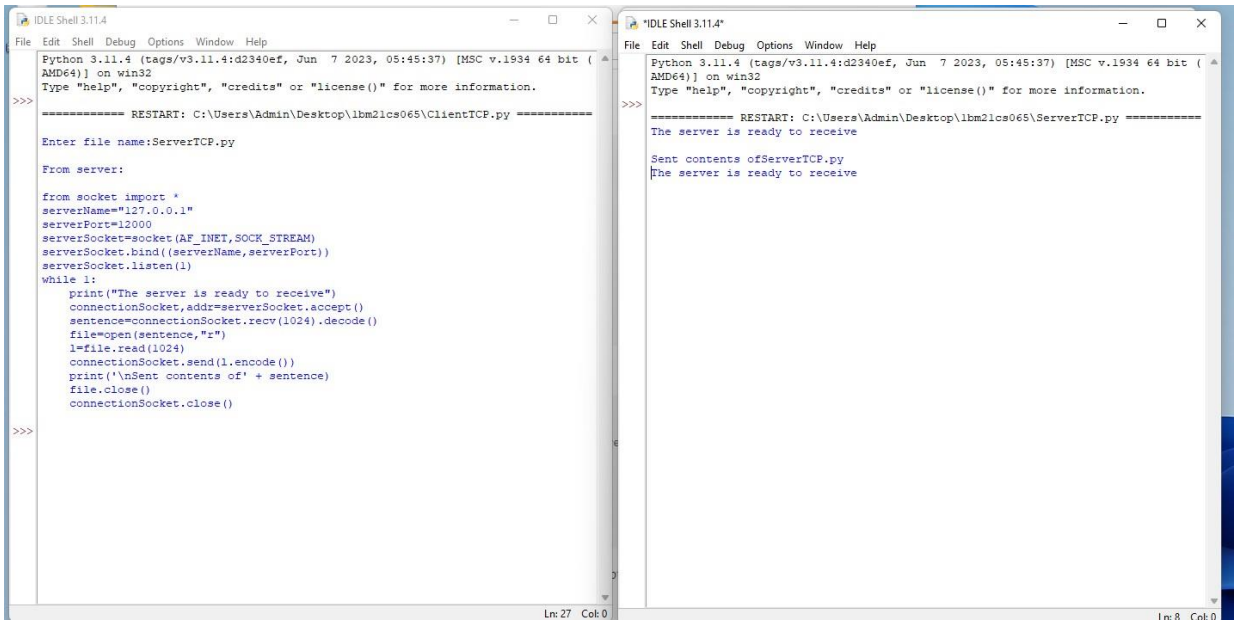
```
from socket import *
serverName = “127.0.0.1”
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName,serverPort))
sentence = input(“\nEnter file name: “)
clientSocket.send(sentence.encode())
filecontents = clientSocket.recv(1024).decode()
print (“\nFrom Server:\n”)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

OUTPUT:



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

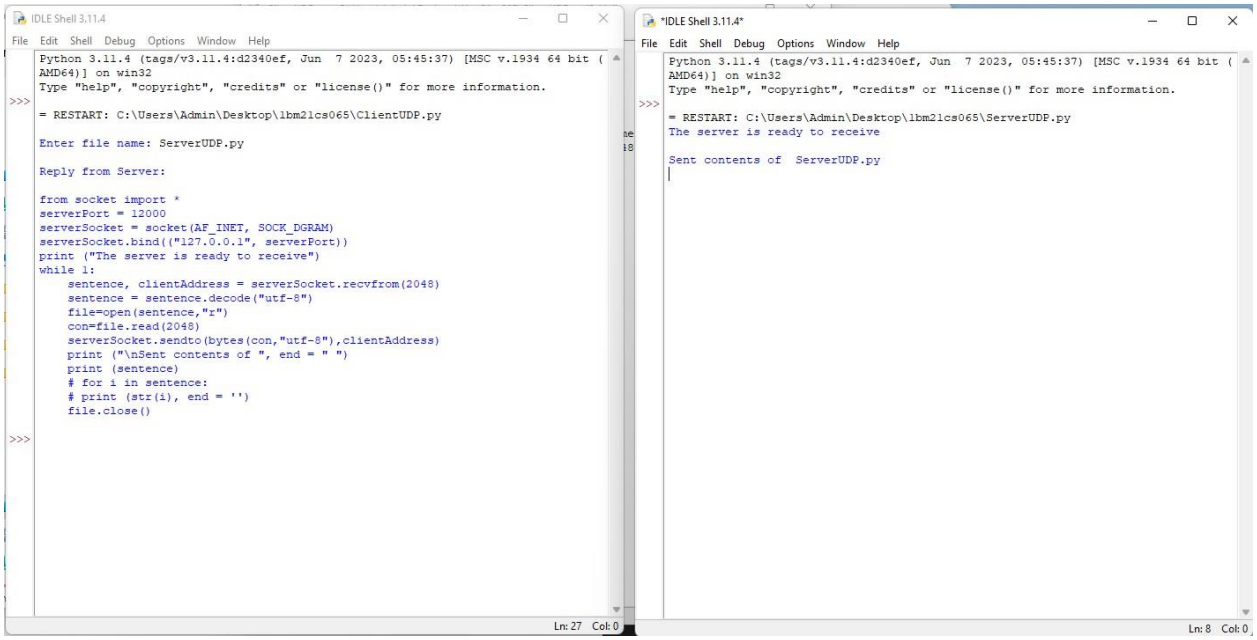
CODE:

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”)
```



observation

DATE PAGE

Cycle - 3

Aim: Using TCP / IP server, 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.

Solution:

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("\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 recive")
connectionSocket, addr = serverSocket.accept
sentence = connectionSocket.recv(1024).decode()
file = open(sentence, 'r')
l = file.read(1024)
connectionSocket.send(l.encode())
print ("\n Sent contents of " + sentence)
file.close()
connectionSocket.close
```

output

Server sede

The server is Ready to recive

Client sede

Enter file name serverTCP.py

From server:

From socket import *

:

(Code under serverTCP.py is Printed as written above)

server sede:

The server is ready to recive

sent contents of serverTCP.py

The server is ready to Recive

Aim: Using UDP Socket, write a client-server program to make client sending the file name and the server to send back the content of the requested file if present.

solution

→ client UPP.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)
file contents, serverAddress = clientSocket.recvf
(2048)
print("\nReply from server:")
print("file contents:", decode("utf-8"))
For i in file contents
    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("\n Sent contents of", end=" ")
    print(sentence)
    for i in sentence:
        print(str(i), end="")
    file.close()
```

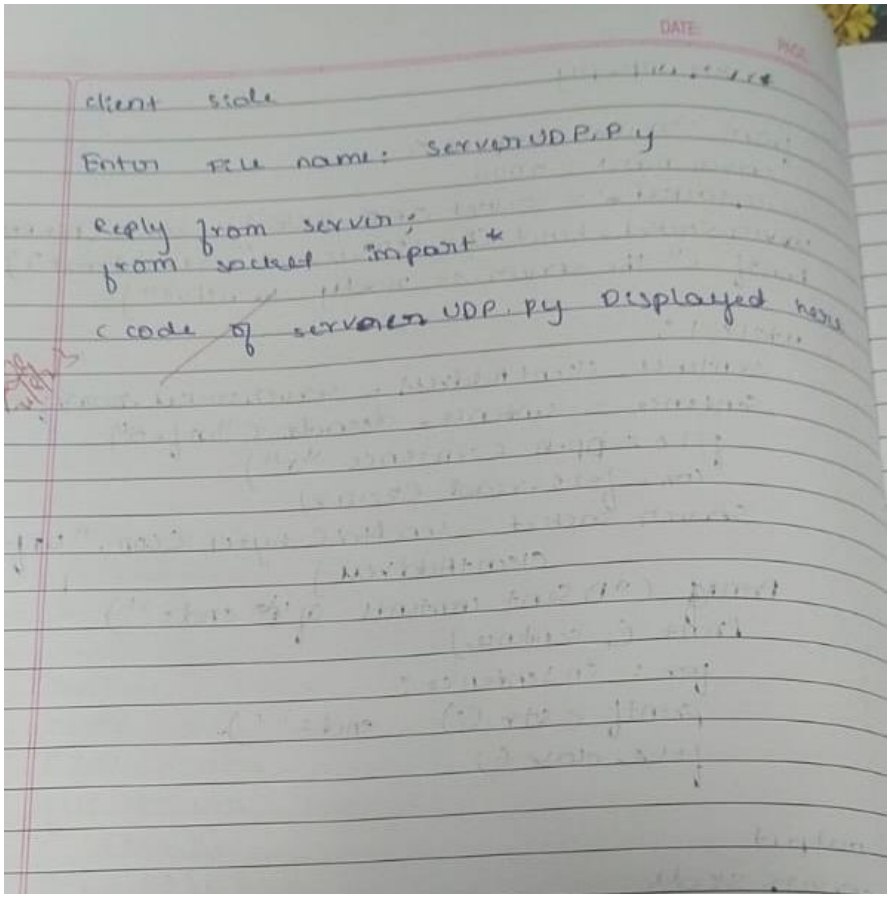
output

~~server side~~

The server is ready to Recive

Sent contents of ServerUDP.py

The Server is ready to receive


client side

Enter file name: serverUDP.py

Reply from server:
from socket import *

code of serverUDP.py Displayed here