LAB PROGRAM-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.

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("\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)
```

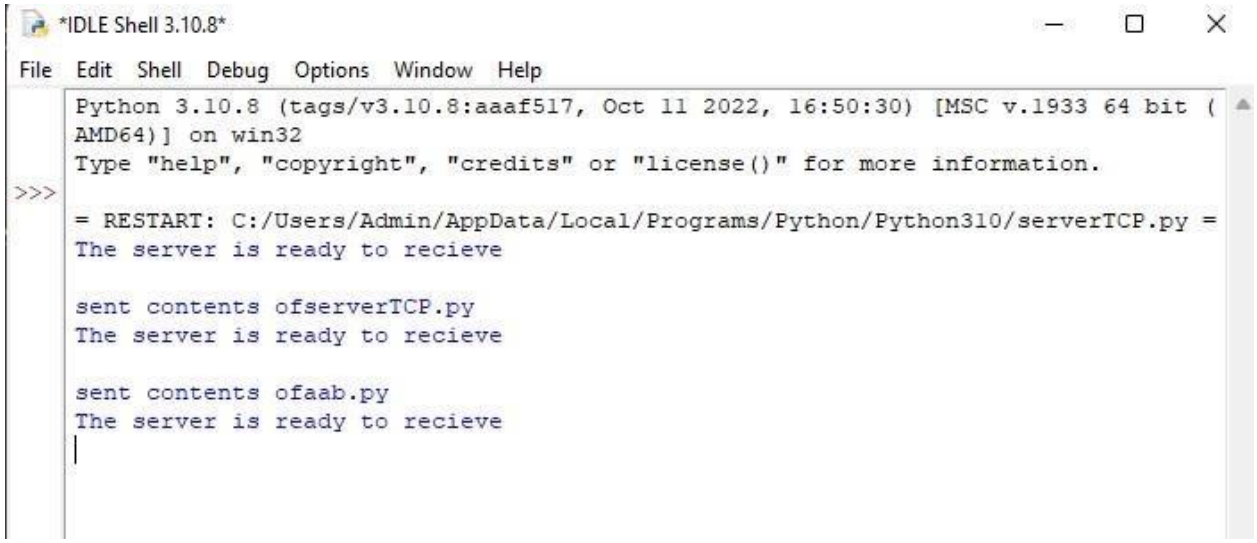
file.close() connectionSocket.close()

OUTPUT:

Client:

```
>>>
= RESTART: C:/Users/Admin/AppData/Local/Programs/Python/Python310/clientTCP.py =

Enter file name:aab.py

From Server:

Python 3.10.8 (tags/v3.10.8:aaaf517, Oct 11 2022, 16:50:30) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
class Node:
    def _init_(self,data):
        self.data=data
        self.left=None
        self.right=None
        self.height=1

class AVL Tree:
    def getHeight(self,root):
        if not root:
            return 0
        return root.height

    def getBalance(self,root):
        if not root:
            return 0
        return self.getHeight(root.left)-self.getHeight(root.right)

    def rightRotate(self,z):
        y=z.left
        T3=y.right

        y.right=z
        z.left=T3

        z.height=1+max(self.getHeight(z.left),self.getHeight(z.right))
        y.height=1+max(self.getHeight(y.left),self.getHeight(y.right))

        return y

    def insert(self,root,data):
        if not root:
            return Node(data)
        if data < root.data
            root.left=self.insert(root.left,data)
        else:
            root.right=se
>>>
```

Server:



LAB PROGRAM-16

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.

SOLUTION:

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:

Client:

```
= RESTART: C:/Users/Admin/AppData/Local/Programs/Python/Python310/clientUDP.py =

Enter file name:  serverUDP.py

Reply from Server:

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()

>>>
```

Server:

```
>>>
= RESTART: C:/Users/Admin/AppData/Local/Programs/Python/Python310/serverUDP.py =
The server is ready to receive

Sent contents of  serverUDP.py
```