

### Aim-15

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:

25/08/23 Aim-3

3. 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

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)

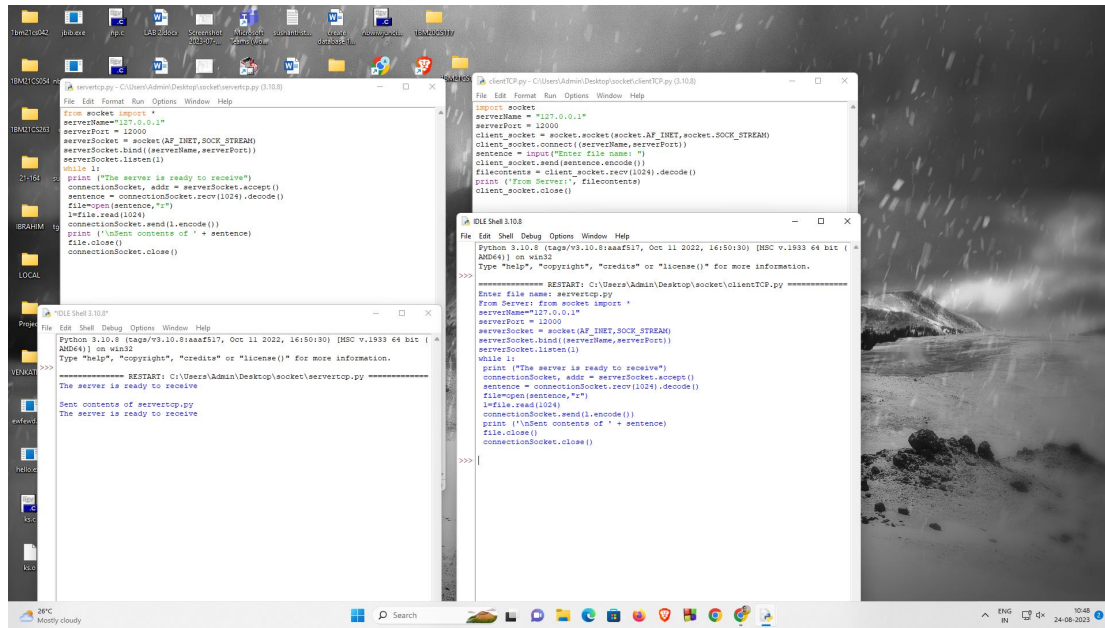
while 1:
    Print ("The Server is ready to receive")
    ConnectionSocket, addr = ServerSocket.accept()
    Sentence = ConnectionSocket.recv(1024).decode()
```

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

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()
```

## Output:



The screenshot displays a Windows desktop environment with a dark, snowy background. Three terminal windows are open, showing the execution of a socket-based program. The top-left window is the 'server.py' script, which sets up a server listening on port 12000. The top-right window is the 'clientTCP.py' script, which connects to the server and sends a file named 'server.py'. The bottom window is a Windows Command Prompt, showing the output of the program, including the message 'The server is ready to receive' and the file contents being received.

```
server.py - C:\Users\Admin\Desktop\socket\server.py (3.10.8)
File Edit Format Run Options Window Help
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")
    i = file.read(1024)
    connectionSocket.send(i.encode())
    print ("Content contents of " + sentence)
    file.close()
    connectionSocket.close()

clientTCP.py - C:\Users\Admin\Desktop\socket\clientTCP.py (3.10.8)
File Edit Format Run Options Window Help
import socket
serverName = "127.0.0.1"
serverPort = 12000
client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
client_socket.connect((serverName, serverPort))
sentence = input("Enter file name: ")
client_socket.send(sentence.encode())
filecontents = client_socket.recv(1024).decode()
print ("From Server:", filecontents)
client_socket.close()

C:\Shell1108
File Edit Shell Debug Options Window Help
Python 3.10.8 (tags/v3.10.8.1000a, Oct 11 2022, 16:50:13) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\Admin\Desktop\socket\clientTCP.py =====
Enter file name: server.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")
    i = file.read(1024)
    connectionSocket.send(i.encode())
    print ("Content contents of " + sentence)
    file.close()
    connectionSocket.close()
>>>
```