

## 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:

The image shows two terminal windows side-by-side, both running Python 3.11.4 on Windows 10. The left window is titled "ClientTCP" and the right window is titled "ServerTCP". Both windows display the same Python code for a TCP client and server respectively.

```
File Edit Shell Debug Options Window Help
Python 3.11.4 (tags/v3.11.4+d34def, Jan 7 2023, 08:48:37) [GCC v.1004 64 bit]
AMD64 on win32
Type "help", "copyright", "credits" or "license()" for more information.

----- RESTART: C:\Users\Admin\Desktop\file\file048\ClientTCP.py -----
from socket import *
serverName="127.0.0.1"
serverPort=12345
connectionSocket=socket(AF_INET,SOCK_STREAM)
connectionSocket.bind((serverName,serverPort))
connectionSocket.listen(1)
while 1:
    print("The server is ready to receive")
    connectionSocket,addr=connectionSocket.accept()
    file=open("file048","w")
    file.write("Hello")
    file.close()
    connectionSocket.send(b"Hello")
    connectionSocket.close()

----- RESTART: C:\Users\Admin\Desktop\file\file048\ServerTCP.py -----
from socket import*
print("The server is ready to receive")
connectionSocket,addr=socket().accept()
file=open("file048","w")
file.write("Hello")
file.close()
connectionSocket.send(b"Hello")
connectionSocket.close()
```

The output from the "ClientTCP" window shows the server's response "Hello" being printed to the console. The output from the "ServerTCP" window shows the server's response "Hello" being printed to the console.

## OBSERVATION:

15

Using TCP/IP Sockets, write a client-server program.  
Client sending the file name & the Server to send  
the contents of the generated file if present.

### Solution

Client TCP By

from socket import \*

Server Name = "122.0.0.1"

Server Port = 12000

Client\_Socket = socket (AF\_INET, SOCK\_STREAM)

Client\_Socket. connect ((server Name, Server Port))

Sentence = input ("\\n Enter the name")

Client\_Socket. send (Sentence. encode())

file contents = Client\_Socket. recv (1024). decode()

Print (file contents)

Client\_Socket. close()

Server TCP By

from socket import \*

Server Name = "127.0.0.1"

Server Port = 12000

Server\_Socket = socket (server Name, Server Port))

Server\_Socket. listen()

while 1:

Reut ("The Server is ready to receive")

ConnectionSocket, address = Server\_Socket. accept()

Sentence = ConnectionSocket. recv (1024). decode()

file = open (Sentence, "r")

```
l = file.read(1024)
connectionSocket.send(l.encode())
print("Sent contents of "+sentence)
file.close()
connectionSocket.close()
```

### Output

```
Restart: c:\Users\Admin\AppData\Local\Programs\Python\Python310\server-tcp.py
```

The server is ready to receive

Sent contents of the server -tcp.py

The server is ready to receive

```
Restart: c:\Users\Admin\AppData\Local\Programs\Python\Python310\client-tcp.py
```

Enter file name: server-tcp.py

from socket import

ServerName = "127.0.0.1"

ServerPort = 12000

ServerSocket = socket(AF\_INET, SOCK\_STREAM)

→ Content sent by the server displayed here