28/1/23

LAB-11

Socket (TCP/IP)

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

Client TCP.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:")
Client Socket.send(sentence.encode())
filecontents = clientSocket.recv(1024).decode()
print("\n From Server:\n")
print(filecontents)
ClientSocket.close()
```

Server TCP.py

```
from socket import *
ServerName = "127.0.0.1"
ServerPort = 12000
ServerSocket = socket(AF_INET, SOCK-STREAM)
ServerSocket.bind((ServerName, Server Port))
ServerSocket.listen(1)
while(1):
    print("The Server is ready to receive")
    connectionSocket, addr = ServerSocket.accept()
    Sentence = connectionSocket.recv(1024).decode()
    file = open(sentence, "y")
    l = file.read(1024)
    connectionSocket.send(l.encode())
    print("\n Send contents of" + Sentence)
    file.close()
    connectionSocket.close()
```

Output:

Server TCP

The server is ready to receive
Sent contents of Server TCP.py
The server is ready to recieve

Client TCP:

Enter file name : Server TCP.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 = serverSock.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()
```

>>>