28-01-2023

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("\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 recieve")
    connectionSocket, addr = serverSocket.accept()
    sentence = connectionSocket.recv(1024).decode()
    file = open(sentence, "r")
    l = file.read(1024)

    connectionSocket.send(l.encode())
    print("\n Send contents of " + sentence)
    file.close()
    connectionSocket.close()
```

O/P

Server TCP

The server is ready to recieve
Sent contents of ServerTCP.py
The server is ready to recieve

Client TCP

Enter file name: ServerTCP.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 ("Enter The server is ready to recieve")
    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()
>>>
```