LAB-10

1. 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:

Client.py

```
from socket import *
serverName = “127.0.0.1”
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName,serverPort))
sentence = input(&amp;quot;Enter file name&amp;quot;)
clientSocket.send(sentence.encode())
filecontents = clientSocket.recv(1024).decode()
print (&amp;#39;From Server:&amp;#39;, filecontents)
clientSocket.close()
```

Server.py

```
from socket import *
serverName=”127.0.0.1”
serverPort = 12000
serverSocket = socket(AF_INET,SOCK_STREAM)
serverSocket.bind((serverName,serverPort))
serverSocket.listen(1)
print (&amp;quot;The server is ready to receive&amp;quot;)
while 1:
connectionSocket, addr = serverSocket.accept()
sentence = connectionSocket.recv(1024).decode()
file=open(sentence,&amp;quot;r&amp;quot;)
l=file.read(1024)
```

connectionSocket.send(l.encode())

file.close()

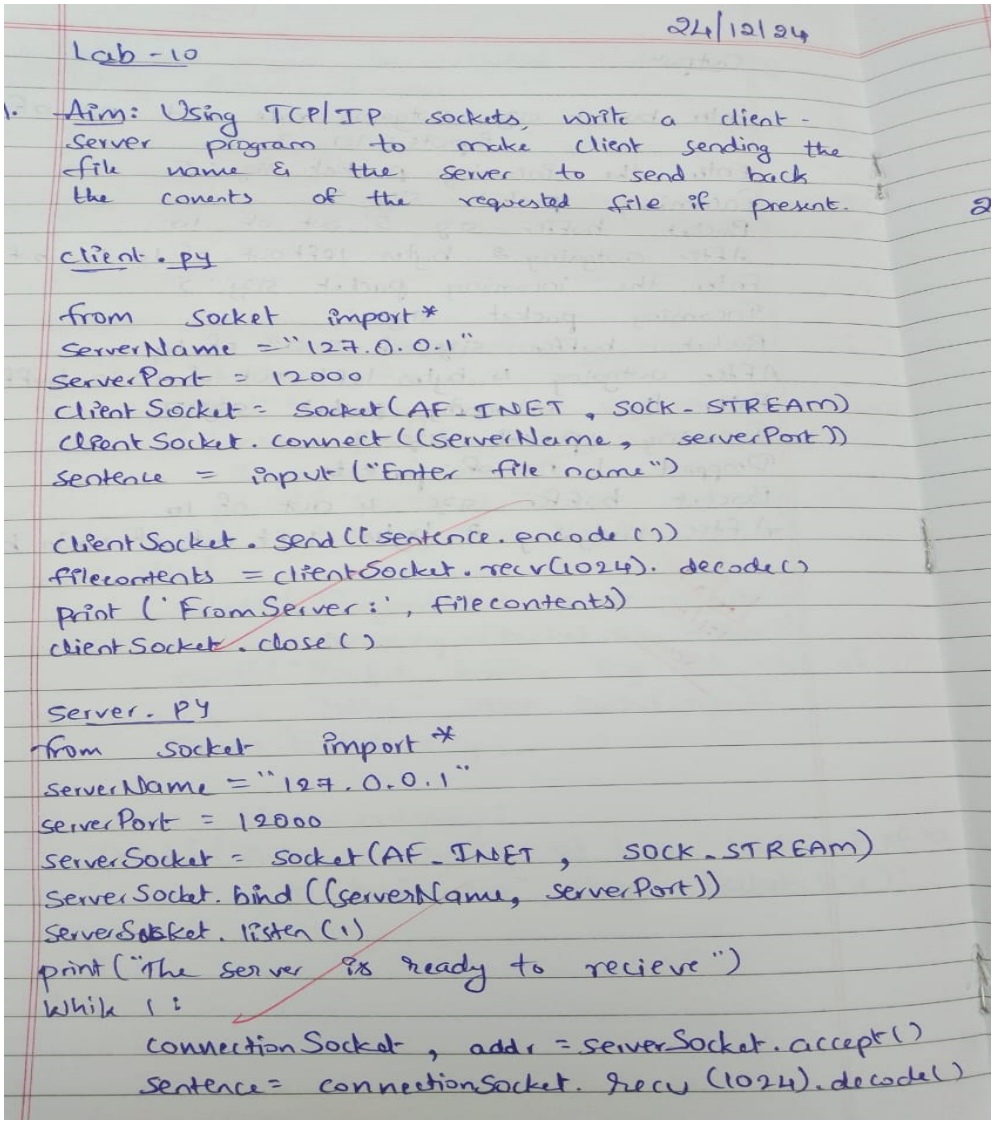
connectionSocket.close()

Output:

```
The server is ready to receive
sent back to client hello world!!

Enter file nameexample.txt
From Server: b'hello world!!'
```

Observation:



24/12/24

Lab - 10

1. Aim: Using TCP/IP sockets, write a client - server program to make client sending the file name & the server to send back the contents of the requested file if present.

client.py

```
from socket import *
serverName = "127.0.0.1"
serverPort = 12000
clientSocket = socket(AF_INET , SOCK_STREAM)
clientSocket.connect((serverName , serverPort))
sentence = input("Enter file name")

clientSocket.send(l.sentence.encode())
filecontents = clientSocket.recv(1024).decode()
print('FromServer:', filecontents)
clientSocket.close()
```

Server.py

```
from socket import *
serverName = "127.0.0.1"
serverPort = 12000
serverSocket = socket(AF_INET , SOCK_STREAM)
serverSocket.bind((serverName, serverPort))
serverSocket.listen(1)
print("The server is ready to recieve")
while 1:
    connectionSocket , addr = serverSocket.accept()
    sentence = connectionSocket.recv(1024).decode()
```

```
file = open(sentence, "r")
l = file.read(1024)
connectionSocket.send(l.encode())
file.close()
connectionSocket.close().
```

Output:

```
Enter file name = example.txt
from server: <<content of example.txt>>
from server: Hello world
Server is ready to recieve
Sent back to client Hello World
```

2. 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.

Code:

ClientUDP.py

```
from socket import *
serverName = &amp;quot;127.0.0.1&amp;quot;
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_DGRAM)
sentence = input(&amp;quot;Enter file name&amp;quot;)
clientSocket.sendto(bytes(sentence,&amp;quot;utf-8&amp;quot;),(serverName, serverPort))
filecontents,serverAddress = clientSocket.recvfrom(2048)
print (&amp;#39;From Server:&amp;#39;, filecontents)
clientSocket.close()
```

ServerUDP.py

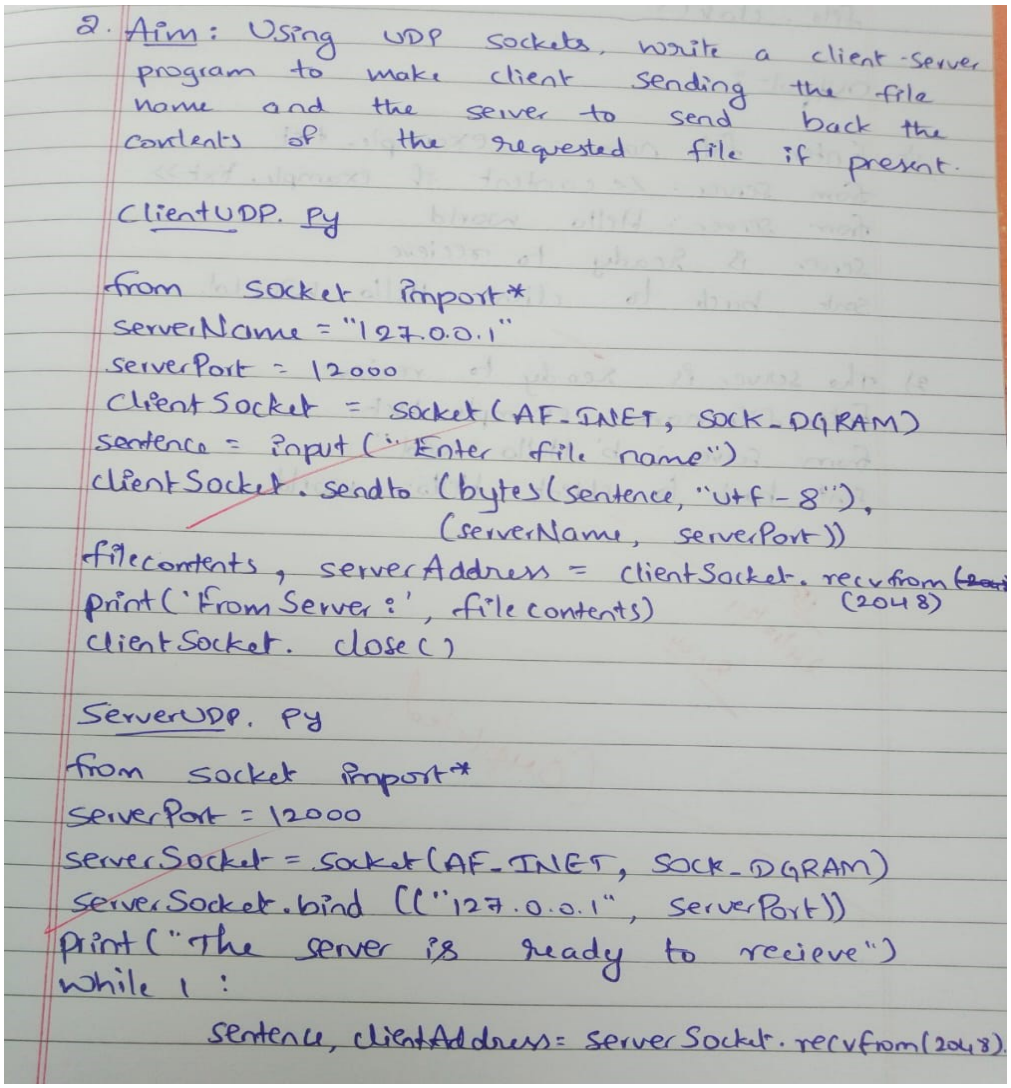
```
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind((&amp;quot;127.0.0.1&amp;quot;, serverPort))
print (&amp;quot;The server is ready to receive&amp;quot;)
while 1:
        sentence,clientAddress = serverSocket.recvfrom(2048)
        file=open(sentence,&amp;quot;r&amp;quot;)
        l=file.read(2048)
        serverSocket.sendto(bytes(l,&amp;quot;utf-8&amp;quot;),clientAddress)
        print(&amp;quot;sent back to client&amp;quot;,l)
file.close()
```

Output:

```
The server is ready to receive
sent back to client hello world!!

Enter file nameexample.txt
From Server: b'hello world!!'
```

Observation:



2. Aim: 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.

ClientUDP.py

```
from socket import*
serverName = "127.0.0.1"
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_DGRAM)
sentence = input("Enter file name")
clientSocket.sendto(bytes(sentence, "utf-8"),
                    (serverName, serverPort))
filecontents, serverAddress = clientSocket.recvfrom(2048)
print('From Server:', filecontents)
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 recieve")
while 1:
        sentence, clientAddress = serverSocket.recvfrom(2048)
```

2) The server is ready to recieve.
Enter file name: example.txt
from server: b' Hello world''
sent back to client Hello. world

15
24/12/24

Completed