

Program 4

Aim of the program:

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 = "127.0.0.1"
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_DGRAM)
sentence = input("\nEnter file name: ")
clientSocket.sendto(bytes(sentence, "utf-8"), (serverName, serverPort))
filecontents, serverAddress = clientSocket.recvfrom(2048)
print("\nReply from Server:\n')
print(filecontents.decode("utf-8"))
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 receive")
while 1:
    sentence, clientAddress = serverSocket.recvfrom(2048)
    sentence = sentence.decode("utf-8")
    file = open(sentence, "r")
    con = file.read(2048)
    serverSocket.sendto(bytes(con, "utf-8"), clientAddress)
    print("\nSent contents of ', end=' ')
    print(sentence)
    file.close()
```

The screenshot shows the VS Code editor with three files open: ClientUDP.py, ServerUDP.py, and ClientTCP.py. The ServerUDP.py file is active, displaying the following Python code:

```
1 from socket import *
2 serverPort = 12000
3 serverSocket = socket(AF_INET, SOCK_DGRAM)
4 serverSocket.bind(('127.0.0.1', serverPort))
5 print ("The server is ready to receive")
6 while 1:
7     sentence, clientAddress = serverSocket.recvfrom(2048)
8     sentence = sentence.decode("utf-8")
9     file=open(sentence,"r")
10    con=file.read(2048)
11    serverSocket.sendto(bytes(con,"utf-8"),clientAddress)
12    print ('\nSent contents of', end='')
13    print (sentence)
14    # for i in sentence:
15    #     # print (str(i), end = " ")
16    file.close()
```

The terminal window at the bottom shows the command prompt running the script. The output indicates that the server is ready to receive and has sent the contents of the file ServerUDP.py.

```
PS C:\Users\dell\Desktop\CN_LAB> & C:/Users/dell/AppData/Local/Programs/Python/Python312/python.exe c:/Users/dell/Desktop/CN_LAB/ClientUDP.py
Enter file name:ServerUDP.py
Reply from Server:
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(('127.0.0.1', serverPort))
print ("The server is ready to receive")
while 1:
    sentence, clientAddress = serverSocket.recvfrom(2048)
    sentence = sentence.decode("utf-8")
    file=open(sentence,"r")
```

The screenshot shows the VS Code editor with three files open: ClientUDP.py, ServerUDP.py, and ClientTCP.py. The ServerUDP.py file is active, displaying the following Python code:

```
1 from socket import *
2 serverPort = 12000
3 serverSocket = socket(AF_INET, SOCK_DGRAM)
4 serverSocket.bind(('127.0.0.1', serverPort))
5 print ("The server is ready to receive")
6 while 1:
7     sentence, clientAddress = serverSocket.recvfrom(2048)
8     sentence = sentence.decode("utf-8")
9     file=open(sentence,"r")
10    con=file.read(2048)
11    serverSocket.sendto(bytes(con,"utf-8"),clientAddress)
12    print ('\nSent contents of', end='')
13    print (sentence)
14    # for i in sentence:
15    #     # print (str(i), end = " ")
16    file.close()
```

The terminal window at the bottom shows the command prompt running the script. The output indicates that the server is ready to receive and has sent the contents of the file ServerUDP.py.

```
PS C:\Users\dell\Desktop\CN_LAB> & C:/Users/dell/AppData/Local/Programs/Python/Python312/python.exe c:/Users/dell/Desktop/CN_LAB/ServerUDP.py
The server is ready to receive
Sent contents ofServerUDP.py
```

Observation:

classmate
Date _____
Page _____

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.

```
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 receive")
while 1:
    sentence, clientAddress = serverSocket.recvfrom(2048)
    file = open(sentence, "r")
    data = file.read(2048)
    serverSocket.sendto(bytes(data, "utf-8"), clientAddress)
    print("sent back to client", data)
file.close()
```

OUTPUT:

```
The server is ready to receive
connection from ('127.0.0.1', 63844)
Enter the filename: sample.txt
From server: Hello world!
```

Handwritten notes:

- IP 127.0.0.1
- Port 12000
- Client socket = socket(AF_INET, SOCK_DGRAM)
- 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()

Red annotations:

- A red circle around "Output" with an arrow pointing to the output text.
- A red circle around "IP" with an arrow pointing to "127.0.0.1".
- A red circle around "Port" with an arrow pointing to "12000".
- A red circle around "Client socket" with an arrow pointing to "socket(AF_INET, SOCK_DGRAM)".
- A red circle around "serverName" with an arrow pointing to "'127.0.0.1'".
- A red circle around "serverPort" with an arrow pointing to "12000".
- A red circle around "sentence" with an arrow pointing to "input('Enter file name')".
- A red circle around "clientSocket" with an arrow pointing to "socket(AF_INET, SOCK_DGRAM)".
- A red circle around "serverName, serverPort" with an arrow pointing to "(serverName, serverPort)".
- A red circle around "filecontents, serverAddress" with an arrow pointing to "clientSocket.recvfrom(2048)".
- A red circle around "print" with an arrow pointing to "print('From server', filecontents)".
- A red circle around "clientSocket" with an arrow pointing to "clientSocket.close()".