COMPUTER NETWORKS – ASSIGNMENT

The file transfer application using TCP sockets

Name: Jonnala Naga Yaswanth Reddy

Roll no: SE20UARI106

LINK : https://github.com/yasreddi/TCP-File-Transfer-Application

These are the steps to be followed while coding the server, in this assignment we have coded our server in python using the socket library, the code is at the end to the steps and this has to be run in a separate terminal from client, the output screenshots are pasted in the report.

1. Make a TCP socket first.
2. Combine IP address and PORT to the server socket.
3. Listen to the clients.
4. Confirm the client's connection.
5. Create a text file after getting the client's filename.
6. Respond to the client with a reply.
7. Collect the client's text data.
8. Enter the data into the text file (save it).
9. Respond to the client with a message.
10. Delete/save the text document.
11. Disconnect the connection.

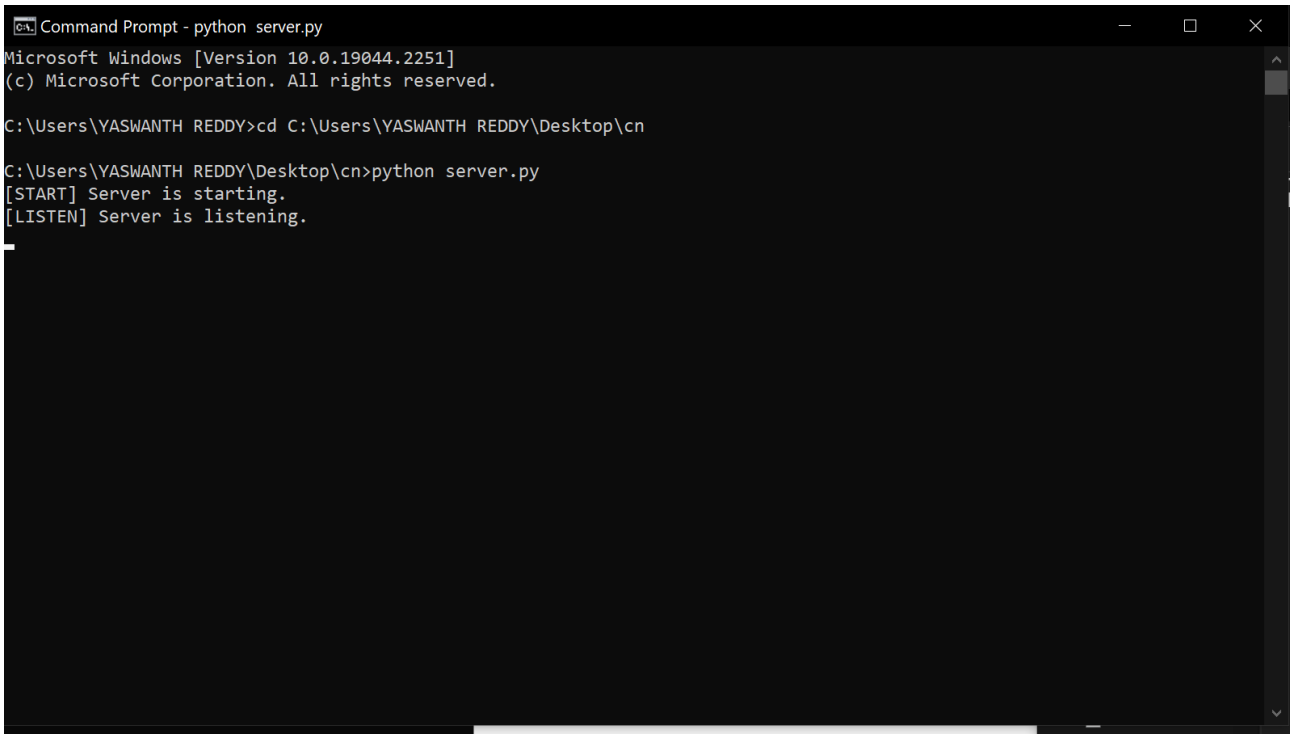
CODE

```
import socket

Ip = socket.gethostbyname(socket.gethostname())
Size = 1024
FORMAT = "utf-8"
```

```
Port = 8888
ADDR = (Ip, Port)



def main():
    print("[START] Server is starting.")
    server = socket.socket(socket.AF_INET, socket.SOCK_STREAM) #Staring a TCP
socket.
    server.bind(ADDR) # connecting the server socket to the Ip address and
Port
    server.listen() # listening to the client (wait for the client to respond)
    print("[LISTEN] Server is listening.")

    while True:
        conn, addr = server.accept() #accepting the connection
        print(f"[NEW CONNECTION] {addr} connected.")

        filename = conn.recv(Size).decode(FORMAT) #getting the filename
        print(f"[RECVIVE] Receiving the filename.")
        file = open(filename, "w")
        conn.send("Filename received.".encode(FORMAT))

        data = conn.recv(Size).decode(FORMAT)# Receiving the data from the
client
        print(f"[RECVIVE] Receiving the file data.")
        file.write(data)
        conn.send("File data received".encode(FORMAT))

        file.close() #close the file

        conn.close() #close the connection
        print(f"[DISCONNECTED] {addr} disconnected.")

if __name__ == "__main__":
    main()
```

Now, save this code in server.py and run this code in terminal.

Now, if we run this code in terminal we will get (server - terminal)



The server is started and listening for the client,

Now, we have to code the client and send file over to the server and get a conformation from the server. So, these are the steps that should be followed while coding the client and the client performs following tasks.

1. Make a TCP socket for the client.
2. Connect to the server.
3. Read the text file.
4. Send the filename to the server.
5. Receive the response from the server.
6. Send the text file data to the server.
7. Receive the response from the server.
8. Close the file.
9. Disconnect the connection.

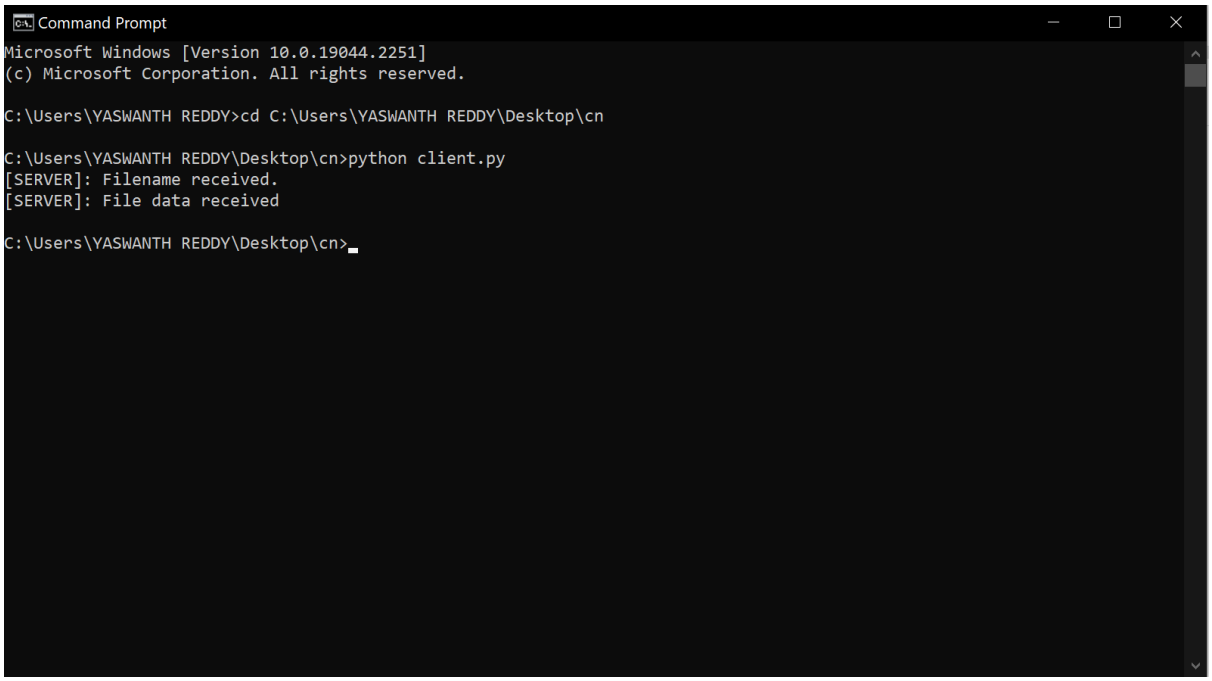
Also create a text file with some data in it; in this case “thefile.txt”

CODE

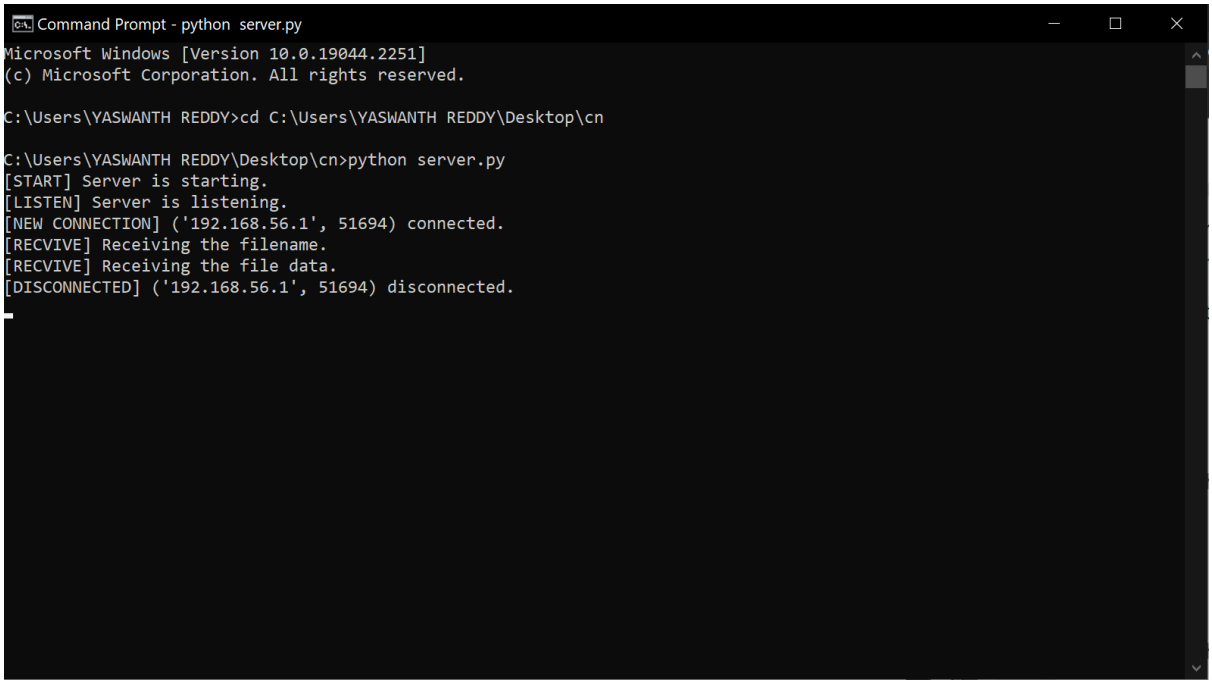
```
import socket

Ip = socket.gethostbyname(socket.gethostname())
FORMAT = "utf-8"
Size = 1024
Port = 8888
Address = (Ip, Port)


def main():
    client = socket.socket(socket.AF_INET, socket.SOCK_STREAM) #starting the
socket

    client.connect(Address) #connecting to server

    file = open("thefile.txt", "r") # opening the file
    data = file.read()

    client.send("thefile.txt".encode(FORMAT)) #send the filename to server
    msg = client.recv(Size).decode(FORMAT)
    print(f"[SERVER]: {msg}")

    client.send(data.encode(FORMAT)) #receiving the data from the server
    msg = client.recv(Size).decode(FORMAT)
    print(f"[SERVER]: {msg}")

    file.close() #closing the file

    client.close() #disconnecting


if __name__ == "__main__":
    main()
```

Now, save this code in client.py and run this code in new terminal (keep running the server terminal)

This message will be shown in the terminal



If we check the server terminal it will be



We, can see that there is conformation with the client that the file is received and the file data is received by the server sent by the client. At the end the connection is disconnected.