KARNATAK LAW SOCIETY’S

GOGTE INSTITUTE OF TECHNOLOGY

UDYAMBAG, BELAGAVI-590008
(An Autonomous Institution under Visvesvaraya Technological University, Belagavi)

(APPROVED BY AICTE, NEW DELHI)

Course Activity Report
Submitted in the partial fulfilment for the academic requirement of
7th Semester B.E. Computer Science &

Engineering in

Network Programming Lab

Submitted by

Shreyas Shivakumar – 2GI18CS139

GUIDE

Prof. Naitik Suryavanshi

Problem Statement : Implement a simple file server using sockets. The file server should be able to take the request from any client and return the requested file to the client or return error message, status to the client. Consider all the possible inputs for the file server. Implement using programming. Compare this result with FTP by using suitable tools.

Theory:

i. Sockets : A socket is one endpoint of a two way communication link between two programs running on the network. The socket mechanism provides a means of inter-process communication (IPC) by establishing named contact points between which the communication take place.

Like 'Pipe' is used to create pipes and sockets is created using 'socket' system call. The socket provides bidirectional FIFO Communication facility over the network. A socket connecting to the network is created at each end of the communication. Each socket has a specific address. This address is composed of an IP address and a port number.

ii. TCP file server : TCP refers to the Transmission Control Protocol, which is a highly efficient and reliable protocol designed for end-to-end data transmission over an unreliable network.

A TCP connection uses a three-way handshake to connect the client and the server. It is a process that requires both the client and the server to exchange synchronization (SYN) and acknowledge (ACK) packets before the data transfer takes place.

Some important features of TCP:

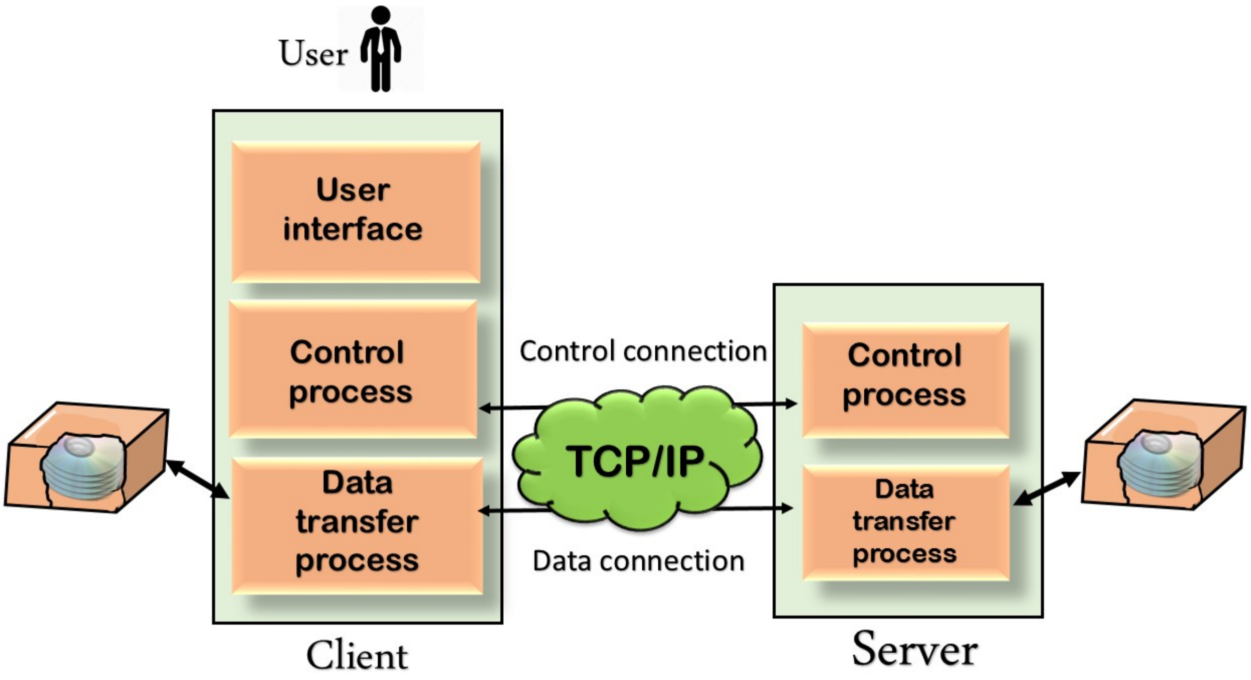
It's a connection-oriented protocol.
It provides error-checking and recovery mechanisms.
It helps in end-to-end communication.

Iii. FTP: stands for File transfer protocol.
FTP is a standard internet protocol provided by TCP/IP used for transmitting the files from one host to another.
It is mainly used for transferring the web page files from their creator to the computer that acts as a server for other computers on the internet.
It is also used for downloading the files to computer from other servers.

Objectives of FTP
It provides the sharing of files.
It is used to encourage the use of remote computers.
It transfers the data more reliably and efficiently.

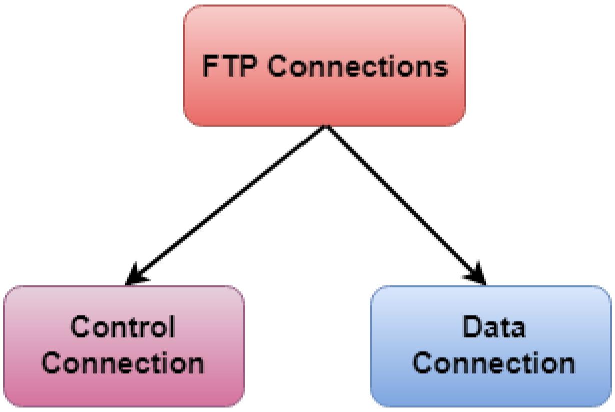
Although transferring files from one system to another is very simple and straightforward, but sometimes it can cause problems. For example, two systems may have different file conventions. Two systems may have different ways to represent text and data. Two systems may have different directory structures. FTP protocol overcomes these problems by establishing two connections between hosts. One connection is used for data transfer, and another connection is used for the control connection.

Mechanism of FTP



The above figure shows the basic model of the FTP. The FTP client has three components: the user interface, control process, and data transfer process. The server has two components: the server control process and the server data transfer process.

There are two types of connections in FTP:



Control Connection: The control connection uses very simple rules for communication. Through control connection, we can transfer a line of command or line of response at a time. The control connection is made between the control processes. The control connection remains connected during the entire interactive FTP session.

Data Connection: The Data Connection uses very complex rules as data types may vary. The data connection is made between data transfer processes. The data connection opens when a command comes for transferring the files and closes when the file is transferred.

FTP Clients

FTP client is a program that implements a file transfer protocol which allows you to transfer files between two hosts on the internet.

It allows a user to connect to a remote host and upload or download the files.

It has a set of commands that we can use to connect to a host, transfer the files between you and your host and close the connection.

The FTP program is also available as a built-in component in a Web browser. This GUI based FTP client makes the file transfer very easy and also does not require to remember the FTP commands.

Advantages of FTP:

Speed: One of the biggest advantages of FTP is speed. The FTP is one of the fastest way to transfer the files from one computer to another computer.

Efficient: It is more efficient as we do not need to complete all the operations to get the entire file.

Security: To access the FTP server, we need to login with the username and password. Therefore, we can say that FTP is more secure.

Back & forth movement: FTP allows us to transfer the files back and forth. Suppose you are a manager of the company, you send some information to all the employees, and they all send information back on the same server.

Disadvantages of FTP:

The standard requirement of the industry is that all the FTP transmissions should be encrypted. However, not all the FTP providers are equal and not all the providers offer encryption. So, we will have to look out for the FTP providers that provides encryption.

FTP serves two operations, i.e., to send and receive large files on a network. However, the size limit of the file is 2GB that can be sent. It also doesn't allow you to run simultaneous transfers to multiple receivers.

Passwords and file contents are sent in clear text that allows unwanted eavesdropping. So, it is quite possible that attackers can carry out the brute force attack by trying to guess the FTP password.

It is not compatible with every system.

Approach towards developing a simple File Server using TCP

On CLIENT side:
The client performs the following functions.

-Start the program
-Declare the variables and structures required.
-A socket is created and the connect function is executed.
-The file is opened.
-The data from the file is read and sent to the server.
-The socket is closed.
-The program is stopped

On SERVER side:
The server performs the following functions.

-Start the program.
-Declare the variables and structures required.
-The socket is created using the socket function.
-The socket is binded to the specific port.
-Start listening for the connections.
-Accept the connection from the client.
-Create a new file.
-Receives the data from the client.
-Write the data into the file.
-The program is stopped.

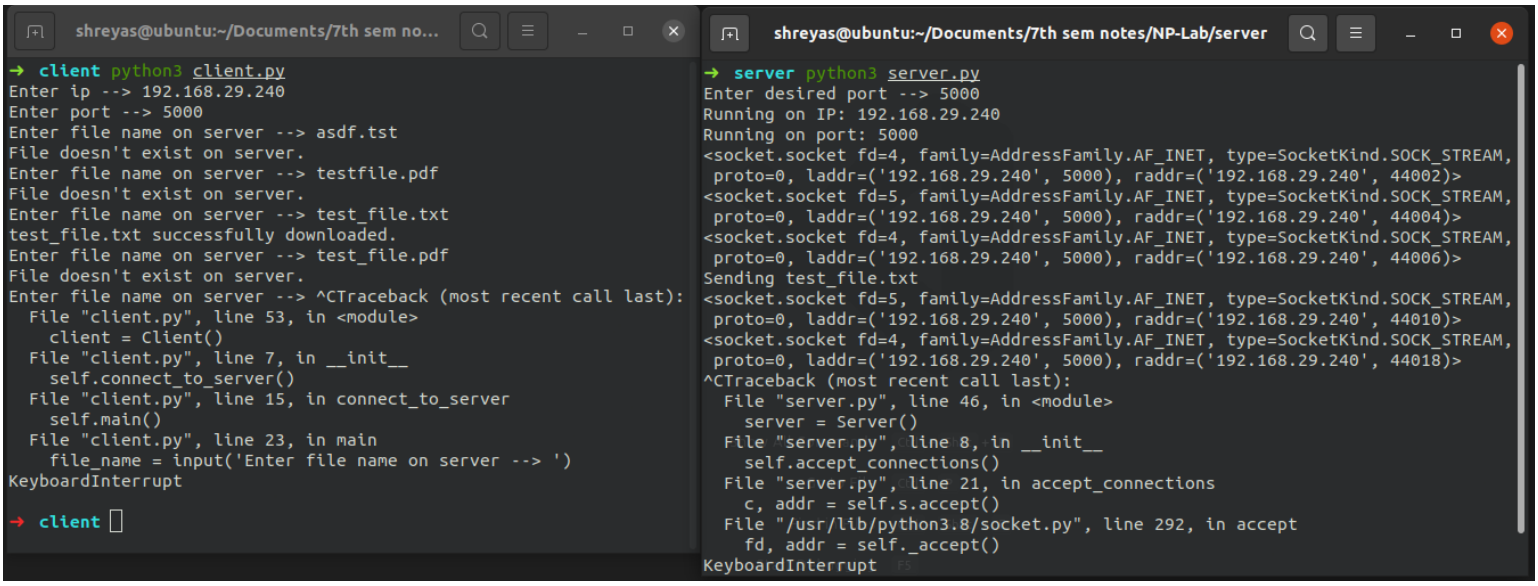
Implementation:

www.github.com/shreyasssk/FileTCP

Output :

For file sharing using TCP

Running

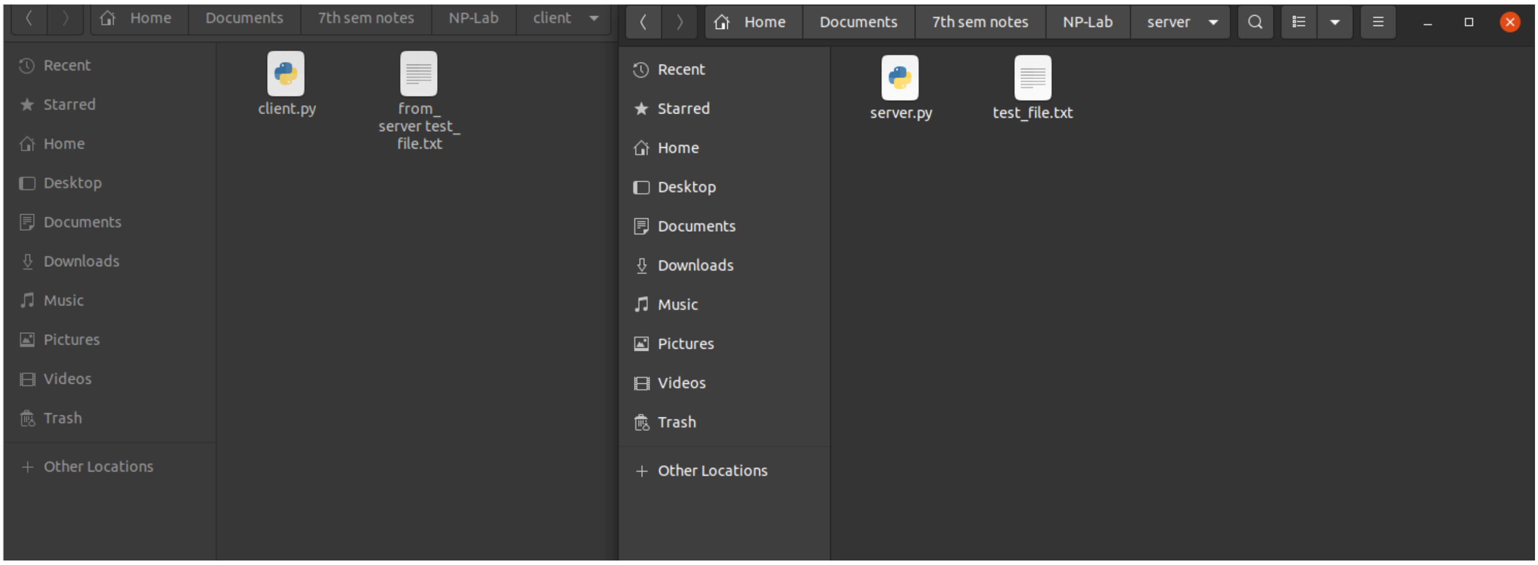


shreyas@ubuntu:~/Documents/7th sem no...
shreyas@ubuntu:~/Documents/7th sem notes/NP-Lab/server

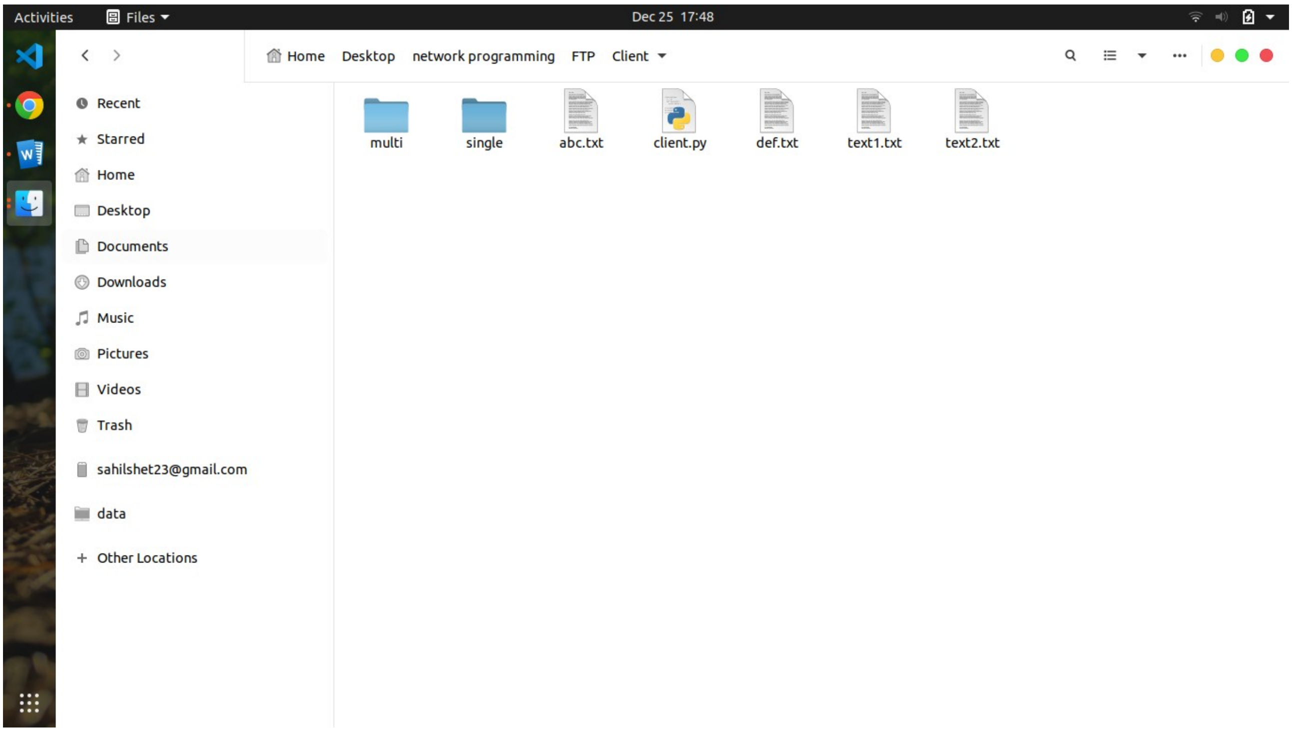
```
➜ client python3 client.py
Enter ip --> 192.168.29.240
Enter port --> 5000
Enter file name on server --> asdf.tst
File doesn't exist on server.
Enter file name on server --> testfile.pdf
File doesn't exist on server.
Enter file name on server --> test_file.txt
test_file.txt successfully downloaded.
Enter file name on server --> test_file.pdf
File doesn't exist on server.
Enter file name on server --> ^CTraceback (most recent call last):
  File "client.py", line 53, in <module>
    client = Client()
  File "client.py", line 7, in __init__
    self.connect_to_server()
  File "client.py", line 15, in connect_to_server
    self.main()
  File "client.py", line 23, in main
    file_name = input('Enter file name on server --> ')
KeyboardInterrupt

➜ client
```

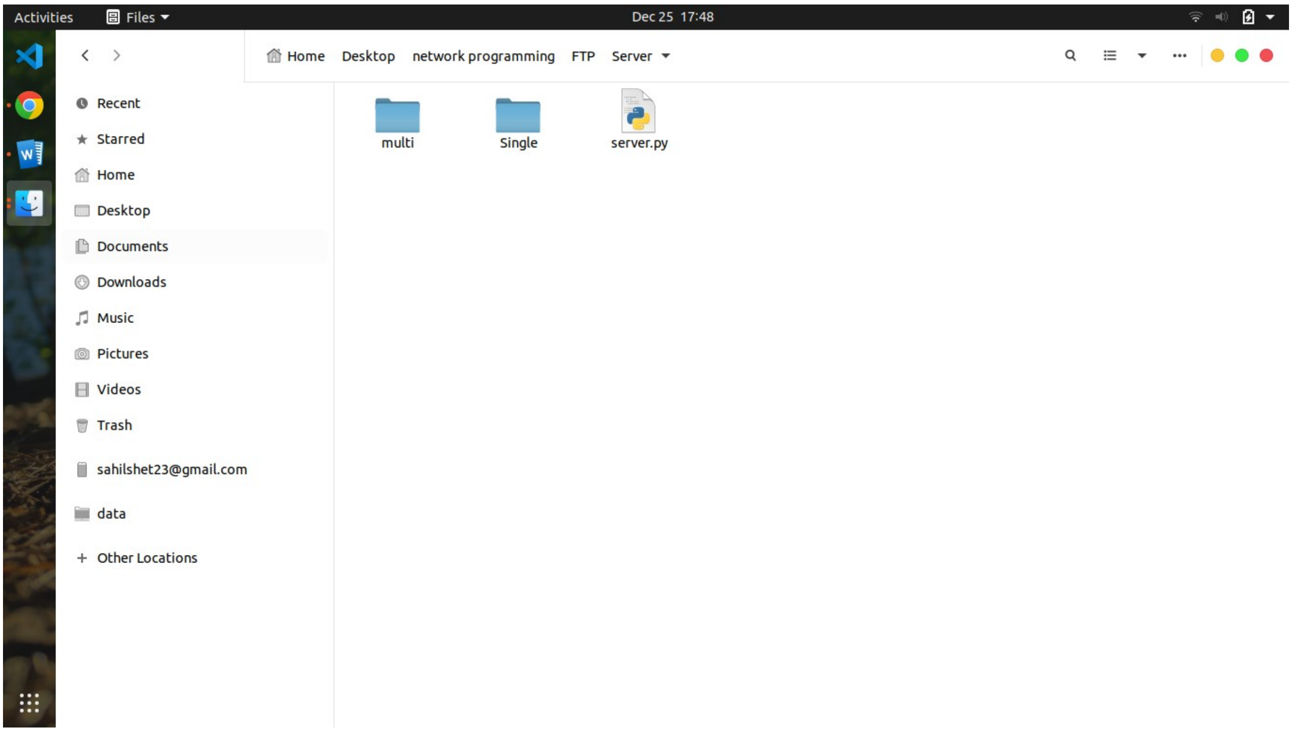
```
➜ server python3 server.py
Enter desired port --> 5000
Running on IP: 192.168.29.240
Running on port: 5000
<socket.socket fd=4, family=AddressFamily.AF_INET, type=SocketKind.SOCK_STREAM,
 proto=0, laddr=('192.168.29.240', 5000), raddr=('192.168.29.240', 44002)>
<socket.socket fd=5, family=AddressFamily.AF_INET, type=SocketKind.SOCK_STREAM,
 proto=0, laddr=('192.168.29.240', 5000), raddr=('192.168.29.240', 44004)>
<socket.socket fd=4, family=AddressFamily.AF_INET, type=SocketKind.SOCK_STREAM,
 proto=0, laddr=('192.168.29.240', 5000), raddr=('192.168.29.240', 44006)>
Sending test_file.txt
<socket.socket fd=5, family=AddressFamily.AF_INET, type=SocketKind.SOCK_STREAM,
 proto=0, laddr=('192.168.29.240', 5000), raddr=('192.168.29.240', 44010)>
<socket.socket fd=4, family=AddressFamily.AF_INET, type=SocketKind.SOCK_STREAM,
 proto=0, laddr=('192.168.29.240', 5000), raddr=('192.168.29.240', 44018)>
^CTraceback (most recent call last):
  File "server.py", line 46, in <module>
    server = Server()
  File "server.py", line 8, in __init__
    self.accept_connections()
  File "server.py", line 21, in accept_connections
    c, addr = self.s.accept()
  File "/usr/lib/python3.8/socket.py", line 292, in accept
    fd, addr = self._accept()
KeyboardInterrupt
```

Post Execution

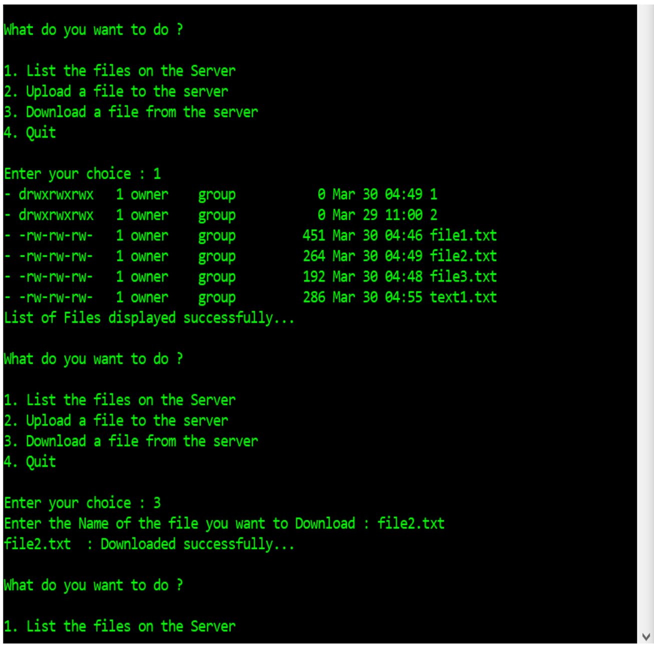


For FTP using pyftpdlib and ftplib

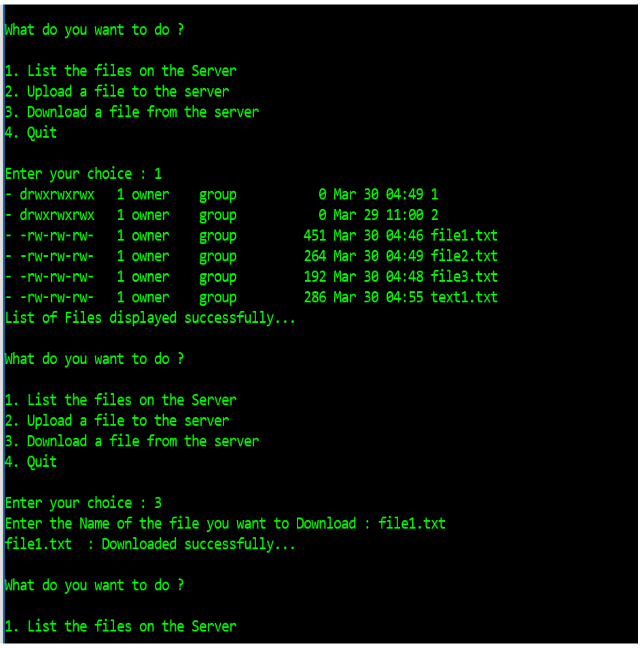




After



```
What do you want to do ?

1. List the files on the Server
2. Upload a file to the server
3. Download a file from the server
4. Quit

Enter your choice : 1
- drwxrwxrwx   1 owner    group           0 Mar 30 04:49 1
- drwxrwxrwx   1 owner    group           0 Mar 29 11:00 2
- -rw-rw-rw-   1 owner    group         451 Mar 30 04:46 file1.txt
- -rw-rw-rw-   1 owner    group         264 Mar 30 04:49 file2.txt
- -rw-rw-rw-   1 owner    group         192 Mar 30 04:48 file3.txt
- -rw-rw-rw-   1 owner    group         286 Mar 30 04:55 text1.txt
List of Files displayed successfully...

What do you want to do ?

1. List the files on the Server
2. Upload a file to the server
3. Download a file from the server
4. Quit

Enter your choice : 3
Enter the Name of the file you want to Download : file2.txt
file2.txt  : Downloaded successfully...

What do you want to do ?

1. List the files on the Server
```



```
What do you want to do ?

1. List the files on the Server
2. Upload a file to the server
3. Download a file from the server
4. Quit

Enter your choice : 1
- drwxrwxrwx   1 owner    group           0 Mar 30 04:49 1
- drwxrwxrwx   1 owner    group           0 Mar 29 11:00 2
- -rw-rw-rw-   1 owner    group         451 Mar 30 04:46 file1.txt
- -rw-rw-rw-   1 owner    group         264 Mar 30 04:49 file2.txt
- -rw-rw-rw-   1 owner    group         192 Mar 30 04:48 file3.txt
- -rw-rw-rw-   1 owner    group         286 Mar 30 04:55 text1.txt
List of Files displayed successfully...

What do you want to do ?

1. List the files on the Server
2. Upload a file to the server
3. Download a file from the server
4. Quit

Enter your choice : 3
Enter the Name of the file you want to Download : file1.txt
file1.txt  : Downloaded successfully...

What do you want to do ?

1. List the files on the Server
```

Conclusion : Both FTP and TCP- based file transfer server can be used to transfer files, although FTP is much faster when large sized files are needed to be transported

References :

1. https://www.khanacademy.org/computing/computers-and-internet/xcae6f4a7ff015e7d:the-internet/xcae6f4a7ff015e7d:transporting-packets/a/transmission-control-protocol--tcp#:~:text=The%20Transmission%20Control%20Protocol%20(TCP,duplicate%20packets%2C%20and%20corrupted%20packets.

2. https://www.javatpoint.com/computer-network-ftp

3. https://docs.oracle.com/javase/tutorial/networking/sockets/definition.html