FILE SERVER IN C

Problem Statement :

Build a C based file server that serves multiple clients using socket programming. For serving multiple clients implementation must be based on multithreading. Basic working is as follows :

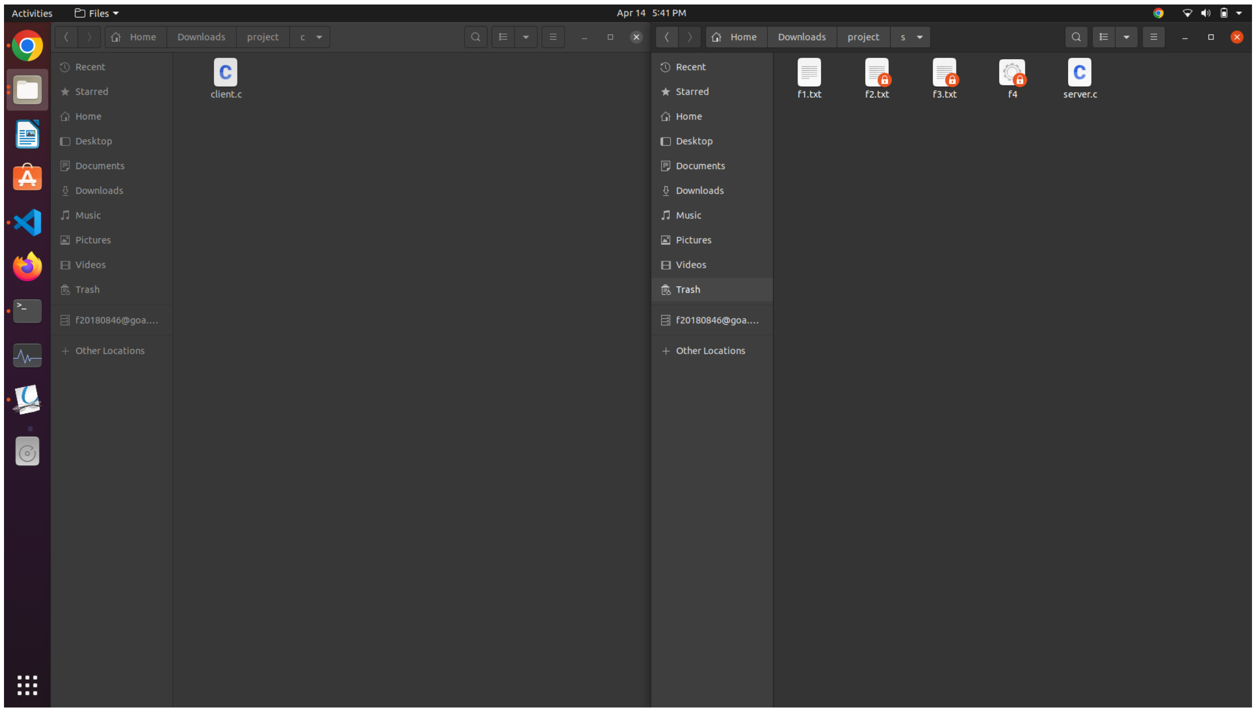
- Accept a port number from the user to bind a server using TCP/IP to that port. The server prints its IP address and port number after binding.
- Accept the server's IP address and the port number at the client's end. The client connects to the server.
- For every client connected, the server spawns a new thread or uses an existing empty thread. The client reads a filename from the terminal and sends it to the server.The thread serving the client prints its thread ID and the received filename. The server sends the file to the client. The client stores the received file locally. This continues until the user types the string "quit".
- If a client disconnects, the server terminates the corresponding thread/reuses it for another client. You may assume that there can be at most five clients in parallel.

Main Challenges :

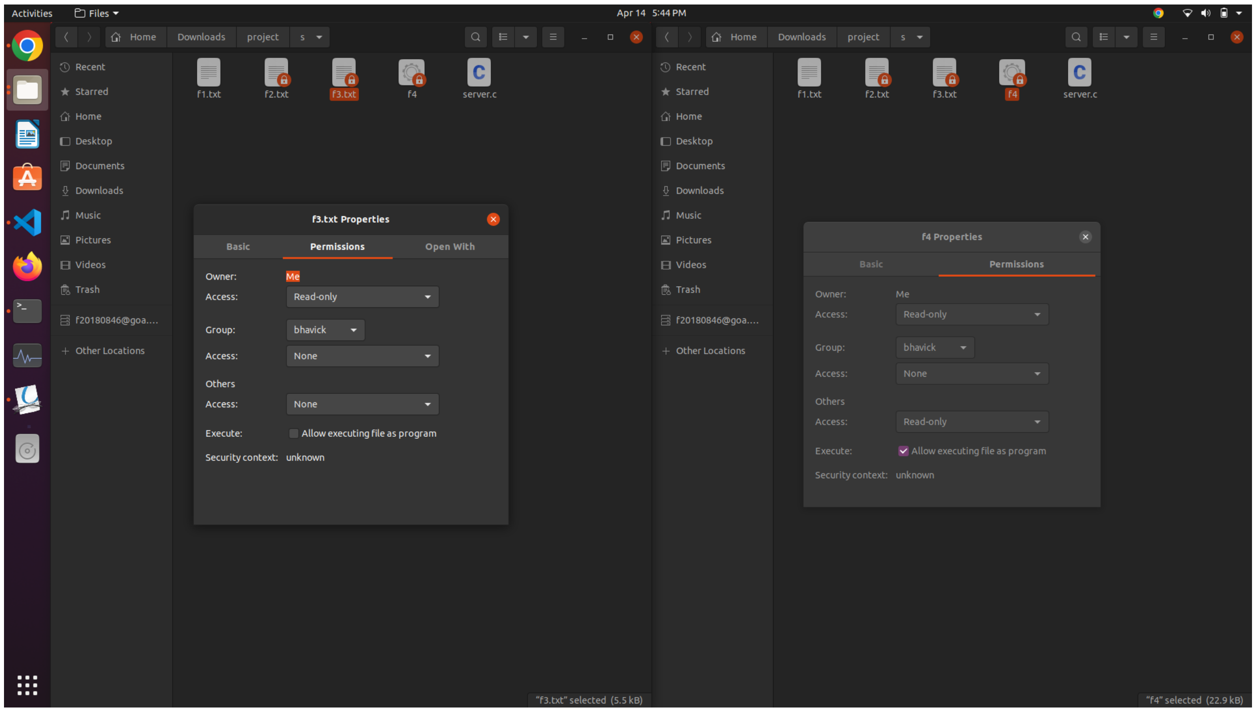
- The file can be of large sizes (>1GB) which must be sent through TCP/IP port using blocks.
- All the file permissions(Read/Write/Execute) must be consistent for the file present in the server and the file sent to the client.

Implementation :

Initially I have created two directories(one each for client and server). Client dir contains client.c file and server dir contains server.c file, 3 text files (f1.txt, f2.txt. f3.txt) and 1 exec file f4.



Also all files have different permissions set by me. Shown in screenshots below :



f3.txt Properties

Owner: Me
Access: Read-only
Group: bhavick
Access: None
Others
Access: None
Execute: Allow executing file as program
Security context: unknown

"f3.txt" selected (5.5 kB)

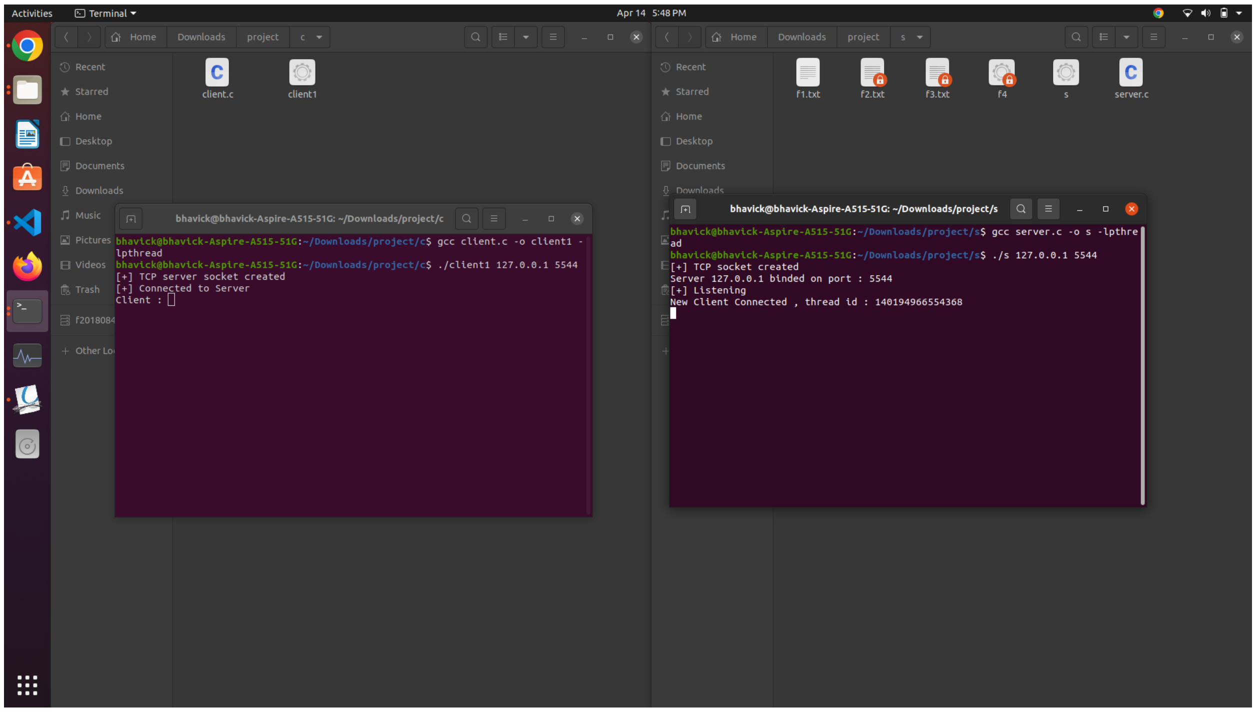
f4 Properties

Owner: Me
Access: Read-only
Group: bhavick
Access: None
Others
Access: Read-only
Execute: [✓ Allow executing file as program]
Security context: unknown

"f4" selected (22.9 kB)

I will implement the project by showing two clients connecting to the server and requesting different files. The respective files are sent to the client dir and permissions are consistent with the file. After the execution each client quits and eventually the server also exits by Ctrl+C.

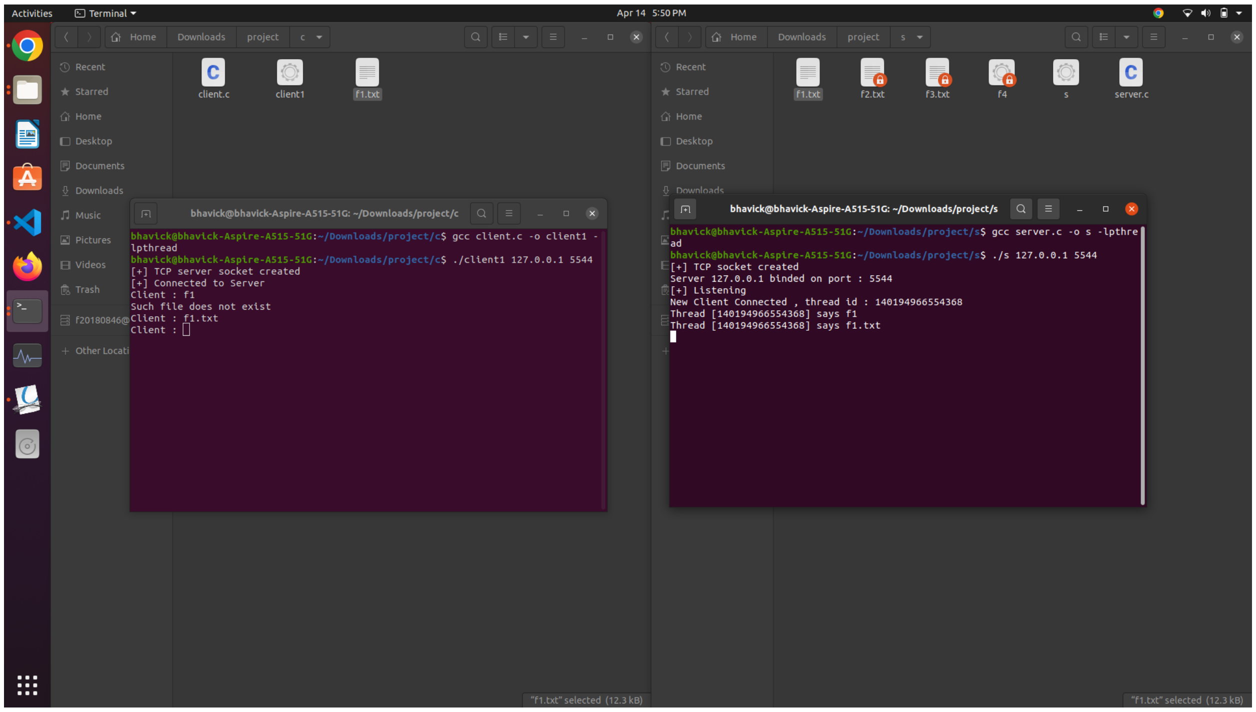
1. Binding server to the local host and port 5544. Similarly binding client1 to the local host on port 5544.



```
bhavick@bhavick-Aspire-A515-51G:~/Downloads/project/c$ gcc client.c -o client1 -
lpthread
bhavick@bhavick-Aspire-A515-51G:~/Downloads/project/c$ ./client1 127.0.0.1 5544
[+] TCP server socket created
[+] Connected to Server
Client :
```

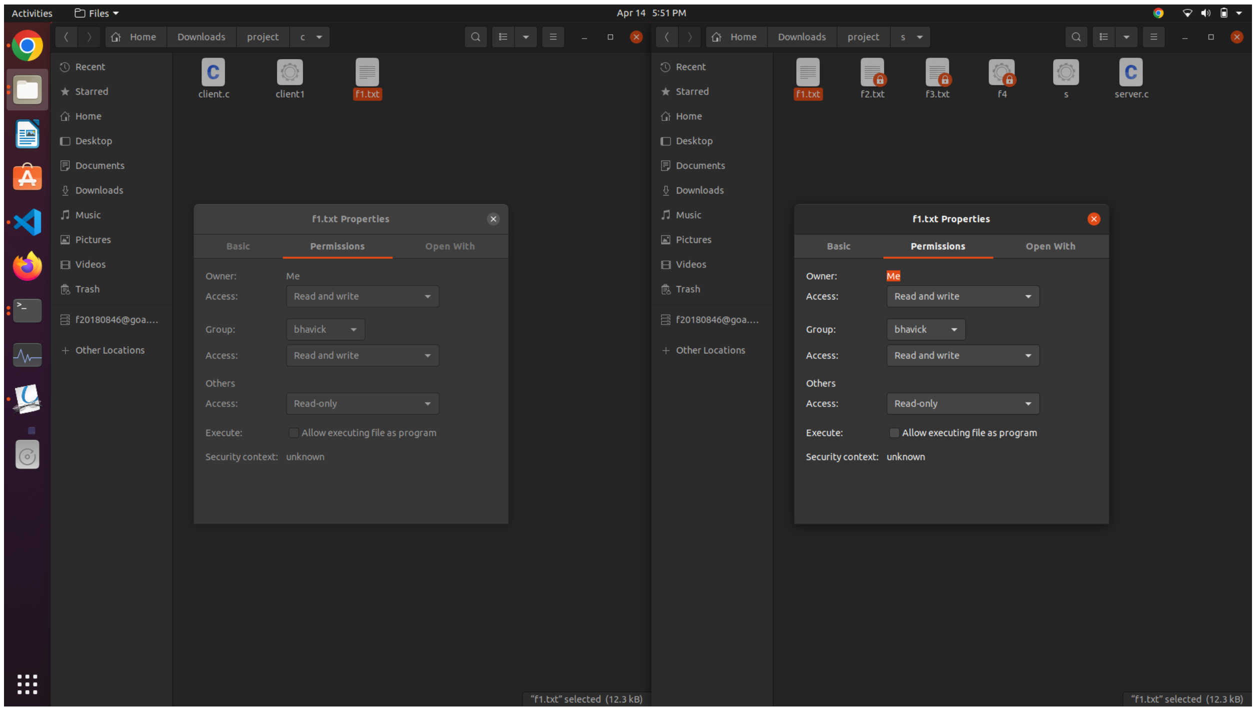
```
bhavick@bhavick-Aspire-A515-51G:~/Downloads/project/s$ gcc server.c -o s -lpthre
ad
bhavick@bhavick-Aspire-A515-51G:~/Downloads/project/s$ ./s 127.0.0.1 5544
[+] TCP socket created
Server 127.0.0.1 binded on port : 5544
[+] Listening
New Client Connected , thread id : 140194966554368
```

2. Client1 requests for f1(file does not exist) and then requests for f1.txt(file exists) and f1.txt is created in client dir with the same permissions.



```
bhavick@bhavick-Aspire-A515-51G:~/Downloads/project/c$ gcc client.c -o client1 -
lpthread
bhavick@bhavick-Aspire-A515-51G:~/Downloads/project/c$ ./client1 127.0.0.1 5544
[+] TCP server socket created
[+] Connected to Server
Client : f1
Such file does not exist
Client : f1.txt
Client :
```

```
bhavick@bhavick-Aspire-A515-51G:~/Downloads/project/s$ gcc server.c -o s -lpthre
ad
bhavick@bhavick-Aspire-A515-51G:~/Downloads/project/s$ ./s 127.0.0.1 5544
[+] TCP socket created
Server 127.0.0.1 binded on port : 5544
[+] Listening
New Client Connected , thread id : 140194966554368
Thread [140194966554368] says f1
Thread [140194966554368] says f1.txt
```



f1.txt Properties

Basic | Permissions | Open With

Owner: Me
Access: Read and write

Group: bhavick
Access: Read and write

Others
Access: Read-only

Execute: Allow executing file as program

Security context: unknown

f1.txt Properties

Basic | Permissions | Open With

Owner: Me
Access: Read and write

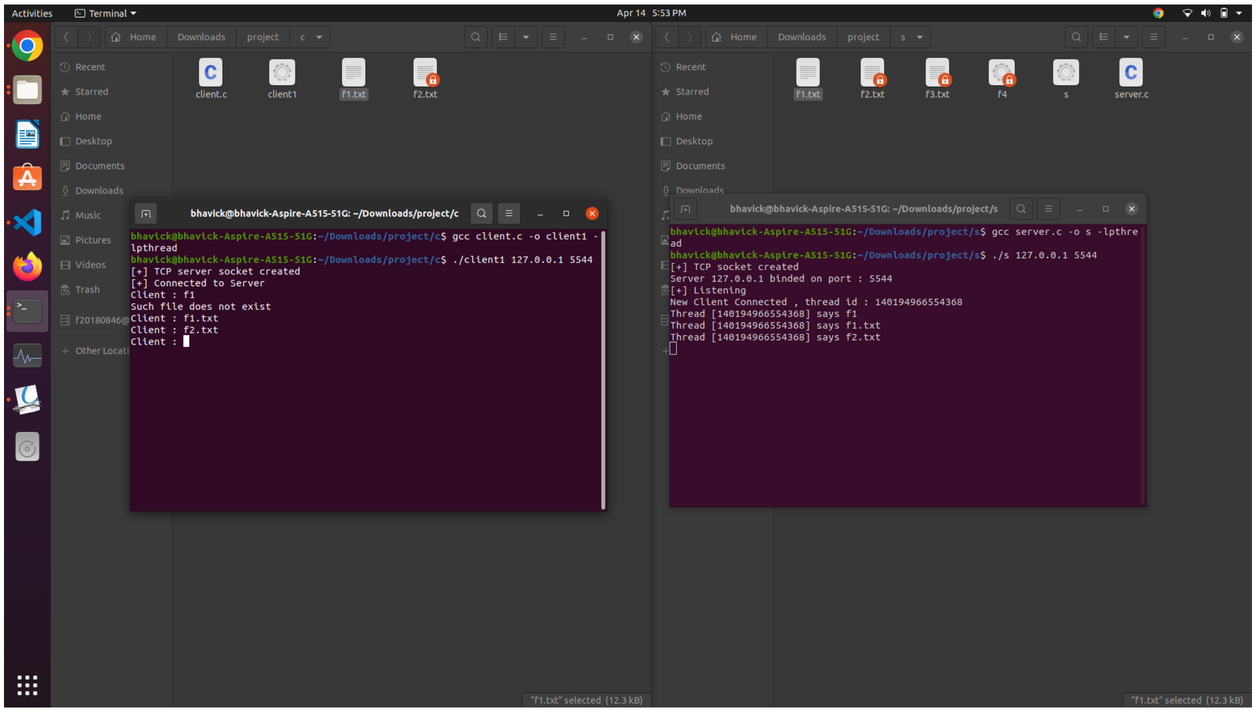
Group: bhavick
Access: Read and write

Others
Access: Read-only

Execute: Allow executing file as program

Security context: unknown

3. Client1 requests for f2.txt(file exists) and f2.txt is created in client dir with the same permissions.

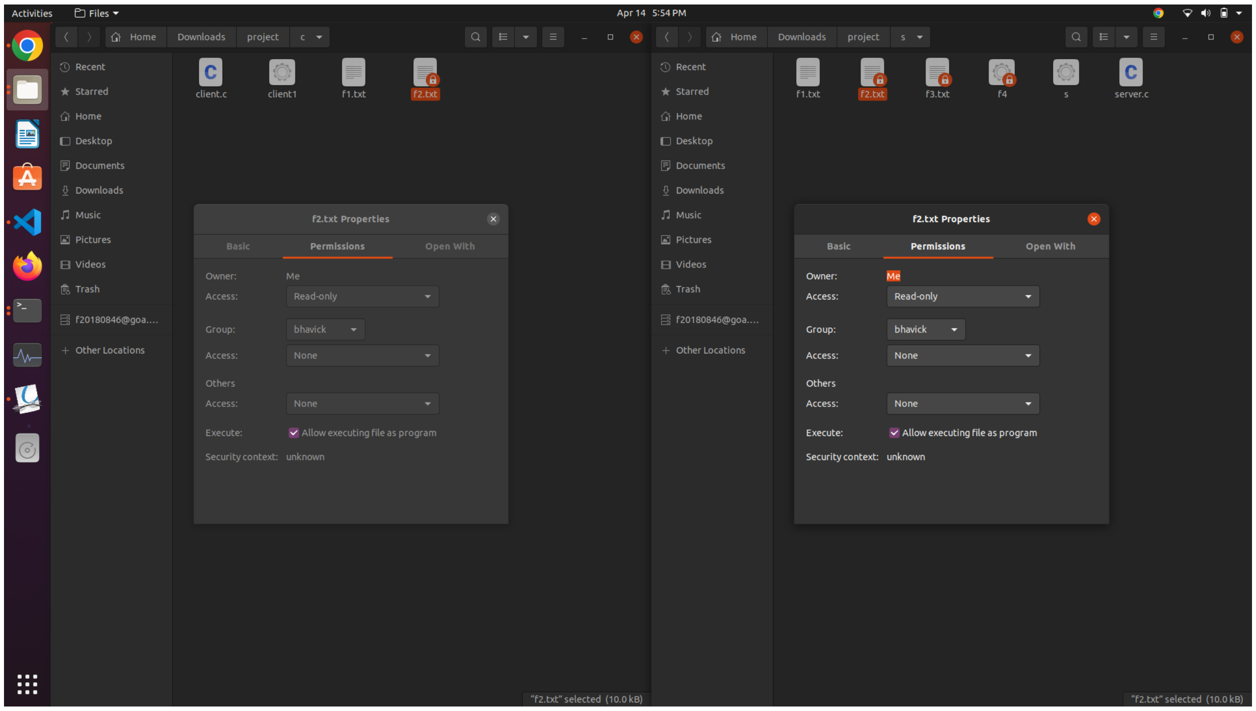


bhavick@bhavick-Aspire-A515-51G: ~/Downloads/project/c

```
bhavick@bhavick-Aspire-A515-51G:~/Downloads/project/c$ gcc client.c -o client1 -
lpthread
bhavick@bhavick-Aspire-A515-51G:~/Downloads/project/c$ ./client1 127.0.0.1 5544
[+] TCP server socket created
[+] Connected to Server
Client : f1
Such file does not exist
Client : f1.txt
Client : f2.txt
Client : 
```

bhavick@bhavick-Aspire-A515-51G: ~/Downloads/project/s

```
bhavick@bhavick-Aspire-A515-51G:~/Downloads/project/s$ gcc server.c -o s -lpthre
ad
bhavick@bhavick-Aspire-A515-51G:~/Downloads/project/s$ ./s 127.0.0.1 5544
[+] TCP socket created
Server 127.0.0.1 binded on port : 5544
[+] Listening
New Client Connected , thread id : 140194966554368
Thread [140194966554368] says f1
Thread [140194966554368] says f1.txt
Thread [140194966554368] says f2.txt
```



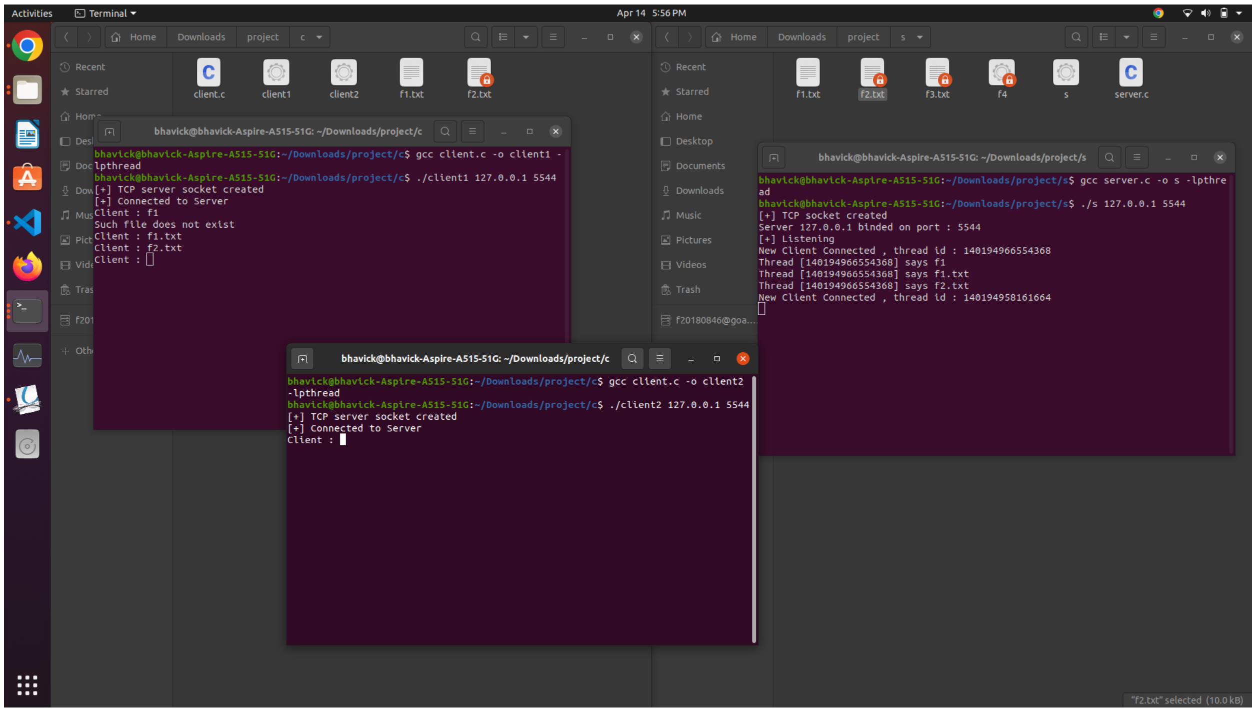
```
f2.txt Properties
Basic   Permissions   Open With

Owner:            Me
Access:           Read-only
Group:            bhavick
Access:           None
Others
Access:           None
Execute:          ☑ Allow executing file as program
Security context: unknown
```

```
f2.txt Properties
Basic   Permissions   Open With

Owner:            Me
Access:           Read-only
Group:            bhavick
Access:           None
Others
Access:           None
Execute:          ☑ Allow executing file as program
Security context: unknown
```

4. Now another client - Client2 binds to the local host and port 5544 and establishes connection with the server.

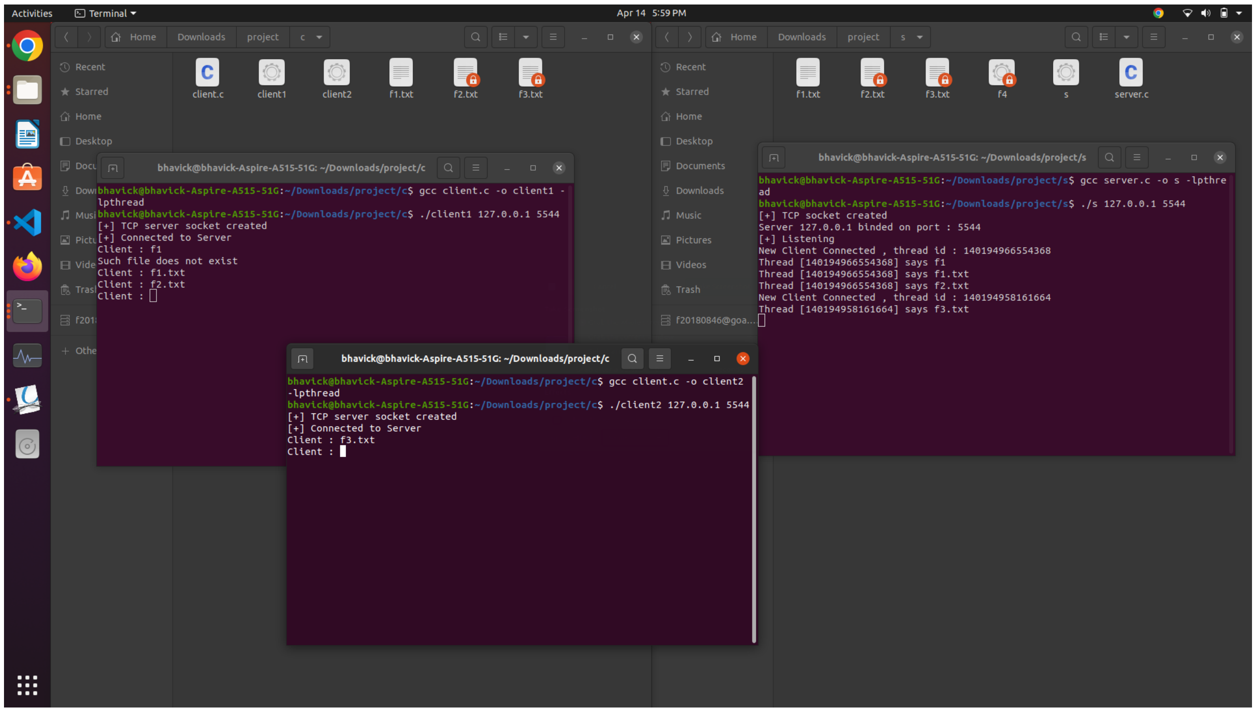


```
bhavick@bhavick-Aspire-A515-51G:~/Downloads/project/c$ gcc client.c -o client1 -
lpthread
bhavick@bhavick-Aspire-A515-51G:~/Downloads/project/c$ ./client1 127.0.0.1 5544
[+] TCP server socket created
[+] Connected to Server
Client : f1
Such file does not exist
Client : f1.txt
Client : f2.txt
Client :
```

```
bhavick@bhavick-Aspire-A515-51G:~/Downloads/project/c$ gcc client.c -o client2
-lpthread
bhavick@bhavick-Aspire-A515-51G:~/Downloads/project/c$ ./client2 127.0.0.1 5544
[+] TCP server socket created
[+] Connected to Server
Client :
```

```
bhavick@bhavick-Aspire-A515-51G:~/Downloads/project/s$ gcc server.c -o s -lpthre
ad
bhavick@bhavick-Aspire-A515-51G:~/Downloads/project/s$ ./s 127.0.0.1 5544
[+] TCP socket created
Server 127.0.0.1 binded on port : 5544
[+] Listening
New Client Connected , thread id : 140194966554368
Thread [140194966554368] says f1
Thread [140194966554368] says f1.txt
Thread [140194966554368] says f2.txt
New Client Connected , thread id : 140194958161664
```

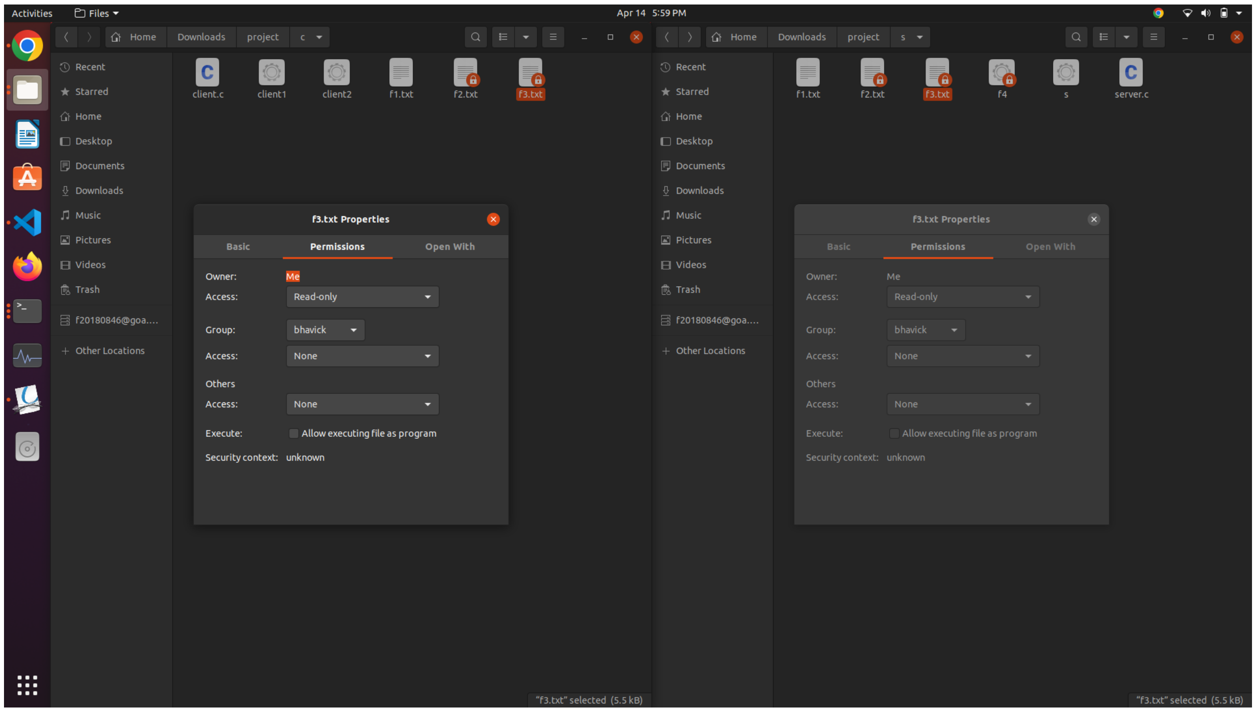
5. Now Client2 requests for f3.txt(file exists) and f3.txt is created in client dir with the same permissions.



```
bhavick@bhavick-Aspire-A515-51G:~/Downloads/project/c$ gcc client.c -o client1 -
lpthread
bhavick@bhavick-Aspire-A515-51G:~/Downloads/project/c$ ./client1 127.0.0.1 5544
[+] TCP server socket created
[+] Connected to Server
Client : f1
Such file does not exist
Client : f1.txt
Client : f2.txt
Client :
```

```
bhavick@bhavick-Aspire-A515-51G:~/Downloads/project/c$ gcc client.c -o client2
-lpthread
bhavick@bhavick-Aspire-A515-51G:~/Downloads/project/c$ ./client2 127.0.0.1 5544
[+] TCP server socket created
[+] Connected to Server
Client : f3.txt
Client :
```

```
bhavick@bhavick-Aspire-A515-51G:~/Downloads/project/s$ gcc server.c -o s -lpthre
ad
bhavick@bhavick-Aspire-A515-51G:~/Downloads/project/s$ ./s 127.0.0.1 5544
[+] TCP socket created
Server 127.0.0.1 binded on port : 5544
[+] Listening
New Client Connected , thread id : 140194966554368
Thread [140194966554368] says f1
Thread [140194966554368] says f1.txt
Thread [140194966554368] says f2.txt
New Client Connected , thread id : 140194958161664
Thread [140194958161664] says f3.txt
```



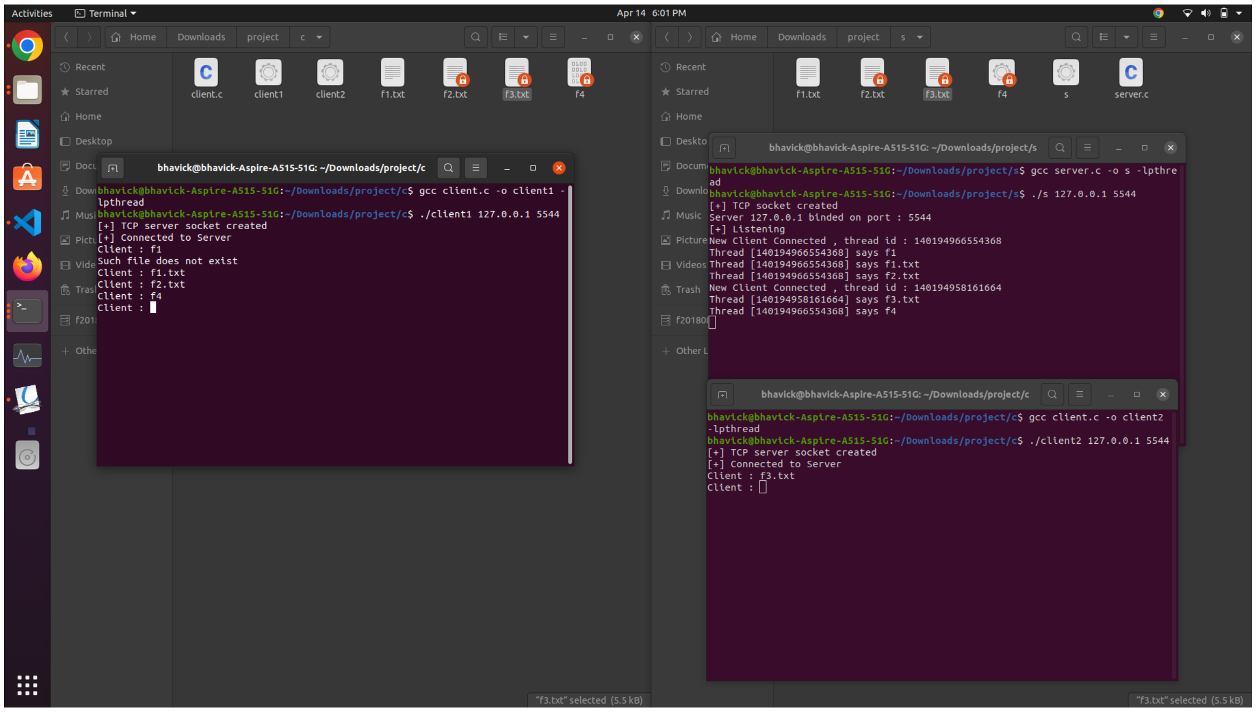
f3.txt Properties — Permissions

Owner: Me
Access: Read-only
Group: bhavick
Access: None
Others
Access: None
Execute: Allow executing file as program
Security context: unknown

f3.txt Properties — Permissions

Owner: Me
Access: Read-only
Group: bhavick
Access: None
Others
Access: None
Execute: Allow executing file as program
Security context: unknown

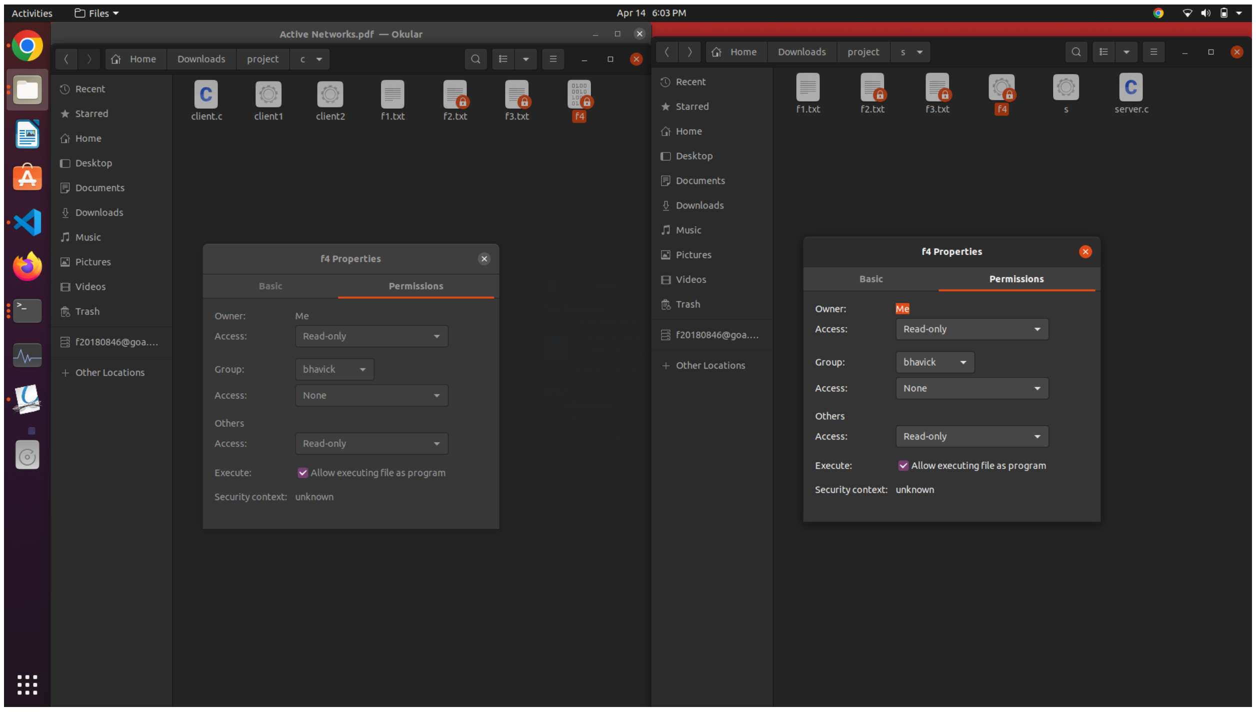
6. Now Client1 requests for f4 (exec file)(file exists) and f4 is created in client dir with the same permissions.



```
bhavick@bhavick-Aspire-A515-51G:~/Downloads/project/c$ gcc client.c -o client1 -
lpthread
bhavick@bhavick-Aspire-A515-51G:~/Downloads/project/c$ ./client1 127.0.0.1 5544
[+] TCP server socket created
[+] Connected to Server
Client : f1
Such file does not exist
Client : f1.txt
Client : f2.txt
Client : f4
Client : 
```

```
bhavick@bhavick-Aspire-A515-51G:~/Downloads/project/s$ gcc server.c -o s -lpthre
ad
bhavick@bhavick-Aspire-A515-51G:~/Downloads/project/s$ ./s 127.0.0.1 5544
[+] TCP socket created
Server 127.0.0.1 binded on port : 5544
[+] Listening
New Client Connected , thread id : 140194966554368
Thread [140194966554368] says f1
Thread [140194966554368] says f1.txt
Thread [140194966554368] says f2.txt
New Client Connected , thread id : 140194958161664
Thread [140194958161664] says f3.txt
Thread [140194966554368] says f4
```

```
bhavick@bhavick-Aspire-A515-51G:~/Downloads/project/c$ gcc client.c -o client2
-lpthread
bhavick@bhavick-Aspire-A515-51G:~/Downloads/project/c$ ./client2 127.0.0.1 5544
[+] TCP server socket created
[+] Connected to Server
Client : f3.txt
Client : 
```



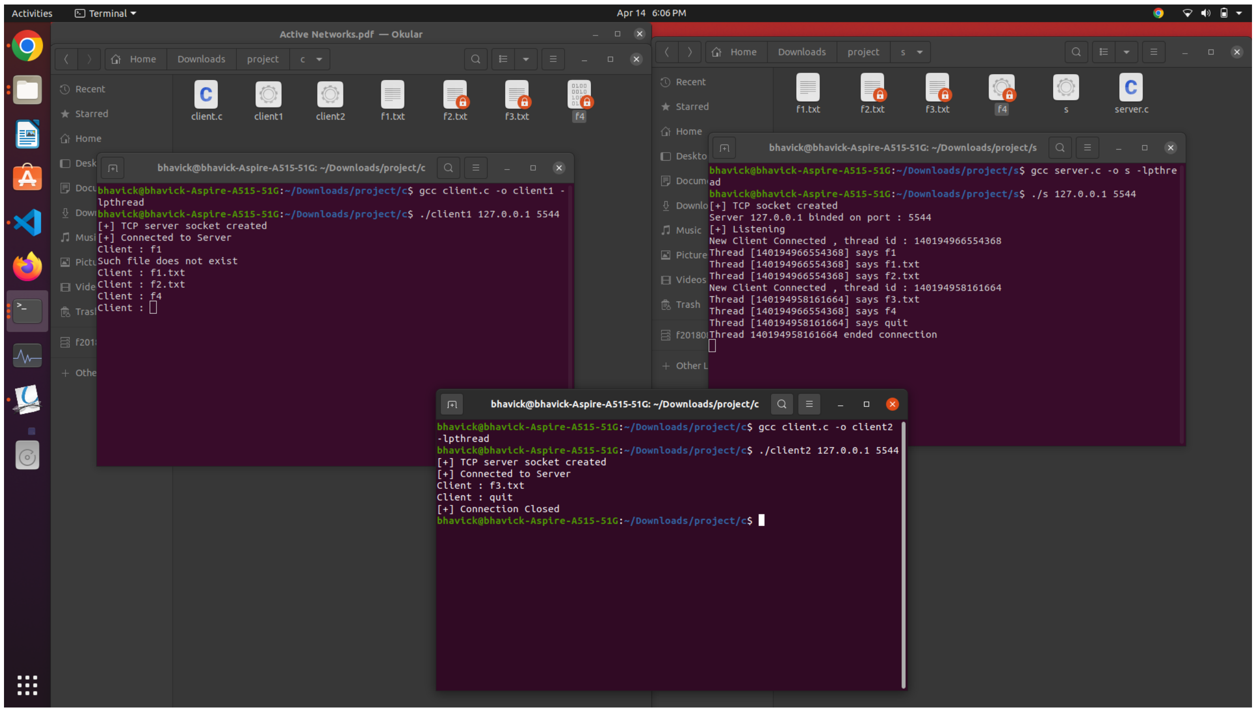
f4 Properties (client directory)

Owner: Me
Access: Read-only
Group: bhavick
Access: None
Others
Access: Read-only
Execute: ✓ Allow executing file as program
Security context: unknown

f4 Properties (server directory)

Owner: Me
Access: Read-only
Group: bhavick
Access: None
Others
Access: Read-only
Execute: ✓ Allow executing file as program
Security context: unknown

7. Now Client2 exits by “quit” command in stdin and the server responds and the thread spawned by server for the Client2 is exited.

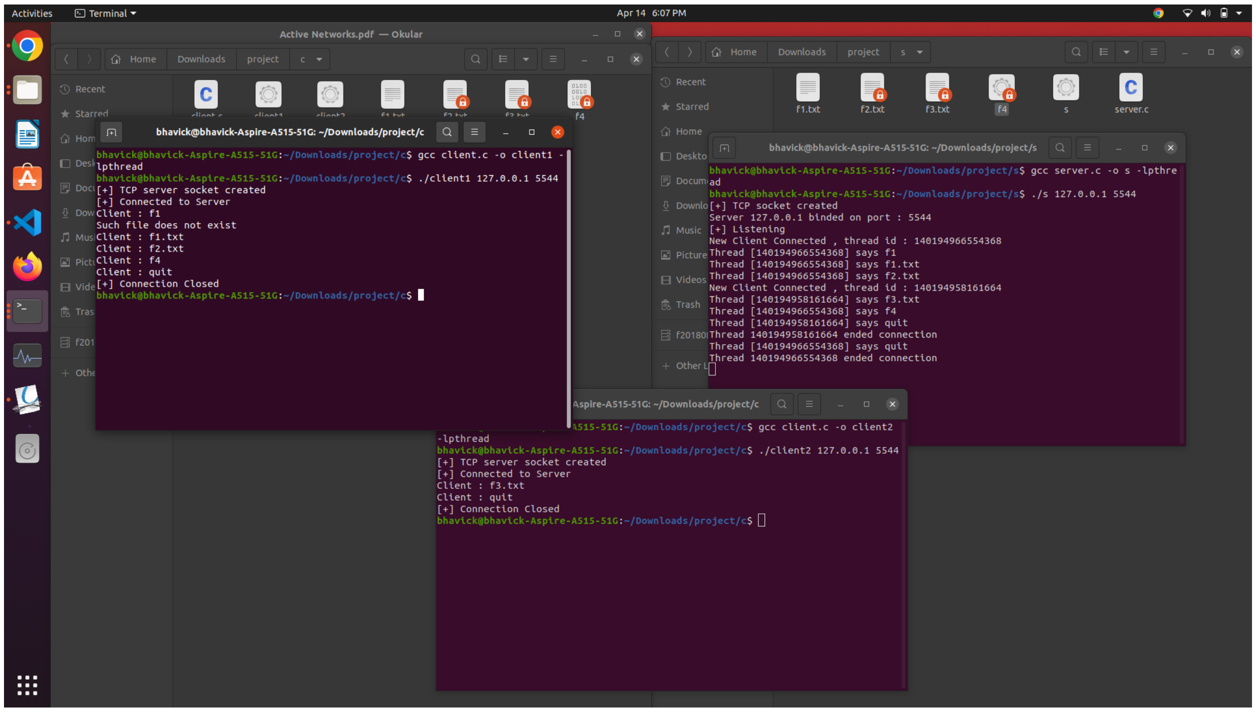


```
bhavick@bhavick-Aspire-A515-51G:~/Downloads/project/c$ gcc client.c -o client1 -
lpthread
bhavick@bhavick-Aspire-A515-51G:~/Downloads/project/c$ ./client1 127.0.0.1 5544
[+] TCP server socket created
[+] Connected to Server
Client : f1
Such file does not exist
Client : f1.txt
Client : f2.txt
Client : f4
Client :
```

```
bhavick@bhavick-Aspire-A515-51G:~/Downloads/project/s$ gcc server.c -o s -lpthre
ad
bhavick@bhavick-Aspire-A515-51G:~/Downloads/project/s$ ./s 127.0.0.1 5544
[+] TCP socket created
Server 127.0.0.1 binded on port : 5544
[+] Listening
New Client Connected , thread id : 140194966554368
Thread [140194966554368] says f1
Thread [140194966554368] says f1.txt
Thread [140194966554368] says f2.txt
New Client Connected , thread id : 140194958161664
Thread [140194958161664] says f3.txt
Thread [140194966554368] says f4
Thread [140194958161664] says quit
Thread 140194958161664 ended connection
```

```
bhavick@bhavick-Aspire-A515-51G:~/Downloads/project/c$ gcc client.c -o client2
-lpthread
bhavick@bhavick-Aspire-A515-51G:~/Downloads/project/c$ ./client2 127.0.0.1 5544
[+] TCP server socket created
[+] Connected to Server
Client : f3.txt
Client : quit
[+] Connection Closed
bhavick@bhavick-Aspire-A515-51G:~/Downloads/project/c$
```

8. Now Client1 exits by “quit” command in stdin and the server responds and the thread spawned by server for the Client1 is exited.

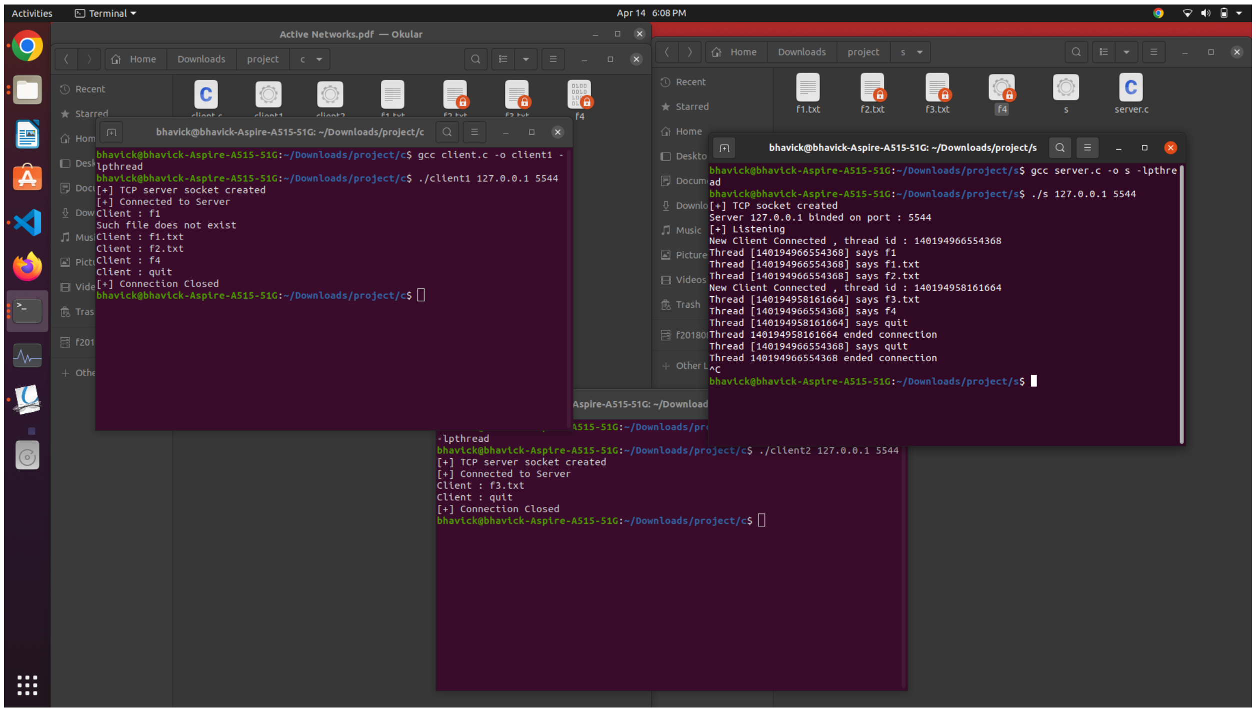


```
bhavick@bhavick-Aspire-A515-51G:~/Downloads/project/c$ gcc client.c -o client1 -
lpthread
bhavick@bhavick-Aspire-A515-51G:~/Downloads/project/c$ ./client1 127.0.0.1 5544
[+] TCP server socket created
[+] Connected to Server
Client : f1
Such file does not exist
Client : f1.txt
Client : f2.txt
Client : f4
Client : quit
[+] Connection Closed
bhavick@bhavick-Aspire-A515-51G:~/Downloads/project/c$
```

```
bhavick@bhavick-Aspire-A515-51G:~/Downloads/project/s$ gcc server.c -o s -lpthre
ad
bhavick@bhavick-Aspire-A515-51G:~/Downloads/project/s$ ./s 127.0.0.1 5544
[+] TCP socket created
Server 127.0.0.1 binded on port : 5544
[+] Listening
New Client Connected , thread id : 140194966554368
Thread [140194966554368] says f1
Thread [140194966554368] says f1.txt
Thread [140194966554368] says f2.txt
New Client Connected , thread id : 140194958161664
Thread [140194958161664] says f3.txt
Thread [140194966554368] says f4
Thread [140194958161664] says quit
Thread 140194958161664 ended connection
Thread [140194966554368] says quit
Thread 140194966554368 ended connection
```

```
515-51G:~/Downloads/project/c$ gcc client.c -o client2
-lpthread
bhavick@bhavick-Aspire-A515-51G:~/Downloads/project/c$ ./client2 127.0.0.1 5544
[+] TCP server socket created
[+] Connected to Server
Client : f3.txt
Client : quit
[+] Connection Closed
bhavick@bhavick-Aspire-A515-51G:~/Downloads/project/c$
```

9. Server terminates execution when Ctrl+C keys are pressed.



```
bhavick@bhavick-Aspire-A515-51G:~/Downloads/project/c$ gcc client.c -o client1 -
lpthread
bhavick@bhavick-Aspire-A515-51G:~/Downloads/project/c$ ./client1 127.0.0.1 5544
[+] TCP server socket created
[+] Connected to Server
Client : f1
Such file does not exist
Client : f1.txt
Client : f2.txt
Client : f4
Client : quit
[+] Connection Closed
bhavick@bhavick-Aspire-A515-51G:~/Downloads/project/c$
```

```
bhavick@bhavick-Aspire-A515-51G:~/Downloads/project/s$ gcc server.c -o s -lpthre
ad
bhavick@bhavick-Aspire-A515-51G:~/Downloads/project/s$ ./s 127.0.0.1 5544
[+] TCP socket created
Server 127.0.0.1 binded on port : 5544
[+] Listening
New Client Connected , thread id : 140194966554368
Thread [140194966554368] says f1
Thread [140194966554368] says f1.txt
Thread [140194966554368] says f2.txt
New Client Connected , thread id : 140194958161664
Thread [140194958161664] says f3.txt
Thread [140194966554368] says f4
Thread [140194958161664] says quit
Thread 140194958161664 ended connection
Thread [140194966554368] says quit
Thread 140194966554368 ended connection
^C
bhavick@bhavick-Aspire-A515-51G:~/Downloads/project/s$
```

```
-lpthread
bhavick@bhavick-Aspire-A515-51G:~/Downloads/project/c$ ./client2 127.0.0.1 5544
[+] TCP server socket created
[+] Connected to Server
Client : f3.txt
Client : quit
[+] Connection Closed
bhavick@bhavick-Aspire-A515-51G:~/Downloads/project/c$
```

The above scenario can be depicted for at max 5 users and the Max Limit Reached for serving works completely fine.

Answering a few questions which I think may clarify my code and the reader may gain a better understanding of the working :

Q.What is new in this project than previous assignments?
Ans : I have created a multithreaded server and single threaded client model. The basic socket creation and TCP connection establishment is done the same as previous assignments.
In this project we have to work with files and thus file handling is necessary. I have used various functions like fopen(), fclose(), fgets(), fputs(), fputc(), fgetc() for file handling.
Also my code uses different inbuilt functions like stat(), realpath(), chmod command, etc.

Q. Are big files (>2KB size) sent from server to client and how?
Ans : Yes, I have considered that possibility and hence I send files from server to client in chunks/blocks of size 2 KB/ 2048 bytes. My code reads char by char and stores it in a buffer array(of size 2 KB). When the buffer array gets full, the server sends it to the client and again initializes it with '\0' and starts reading char by char and stor.ing again for starting index in buffer array. Same working is done on the client side which reads the buffer array and stores it in the new_file/modified_file char by char.

Q. Is permission consistency considered and how?
Ans : Yes, I have maintained the permissions consistency of the files sent to the client. For that I have used the stat_library. I send the stat struct for the file from server to the client and after modifying/creating the file on the client side the code modifies the “mode”(which contains permission bits) for the file using the struct sent before by using the chmod command.