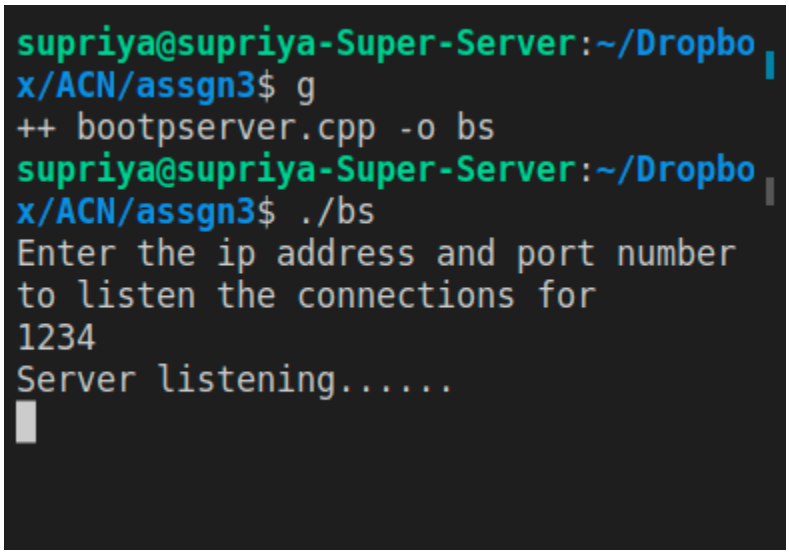
Socket Programming

There are three files, namely, bootpserver.cpp, tcpclient.cpp, fileserver.cpp.

Part A

Bootpserver.cpp

- Bootstrap server that accepts the serving information from different servers in the sequence - MSGTYPE:REGISTRN IP address, port number, servicename(png/jpeg/mp4/txt/ or any other file extension), and a service access token on UDP.
- To compile : g++ bootpserver.cpp -o bs
- To run : ./bs



fileserver.cpp

- Firstly it registers to bootstrap server on UDP with the details given in step1.
- To compile : g++ fileserver.cpp -o fs
- To run : ./fs

```
supriya@supriya-Super-Server:~/Dropbox/ACN/assgn3$ ./fs
Enter the ip address and port number of bootpserver
1234
Enter details: MSGTYPE(REGISTRN,DISCOVERY), IP address, port number, servicena
me, service access token or simply type 'bye' to end
REGISTRN,192.168.126.137,1236,pdf,server1
sent to bootp REGISTRN,192.168.126.137,1236,pdf,server1
server1
message sent successfully
AT is -server1
All done closing socket now
```

Part B

fileserver.cpp

- Once it registers to the BootStrap server successfully in Part A, it listens on TCP socket.

  Server listening on TCP port -

```
supriya@supriya-Super-Server:~/Dropbox/ACN/assgn3$ ./fs
Enter the ip address and port number of bootpserver
1234
Enter details: MSGTYPE(REGISTRN,DISCOVERY), IP address, port number, servicename, servi
ce access token or simply type 'bye' to end
REGISTRN,192.168.126.137,1236,pdf,server1
sent to bootp REGISTRN,192.168.126.137,1236,pdf,server1
server1
message sent successfully
AT is -server1
All done closing socket now
Server server1
Enter the port no
1236
Server socket created successfully.
Listening.....
```

tcpclient.cpp

- client interacts with Bootstrap server on UDP and file servers on TCP. servicename, IP address, port number, and a service access token.
- To compile : g++ tcpclient.cpp -o client
- To run : ./client

- Client connects to bootpserver and get the information of all running servers.

```
supriya@supriya-Super-Server:~/Dropbox/ACN/assgn3$ ./client
Enter the ip address and port number of bootpserver
1234
Enter message to send to server in  or simply type 'bye' to end
DISCOVERY
message sent successfully
Message from server: 192.168.126.137,1236,pdf,server1
ip - 192.168.126.137
0   1236
1   pdf
2   server1
received server infos
Message from server: end
All done closing socket now
All done closing udp bootpsocket now
connecting to server 192.168.126.137 at port 1236
ACCESS TOKEN -server1 server type pdf
Server socket created successfully.
Connected to Server.
want to change the access token
```

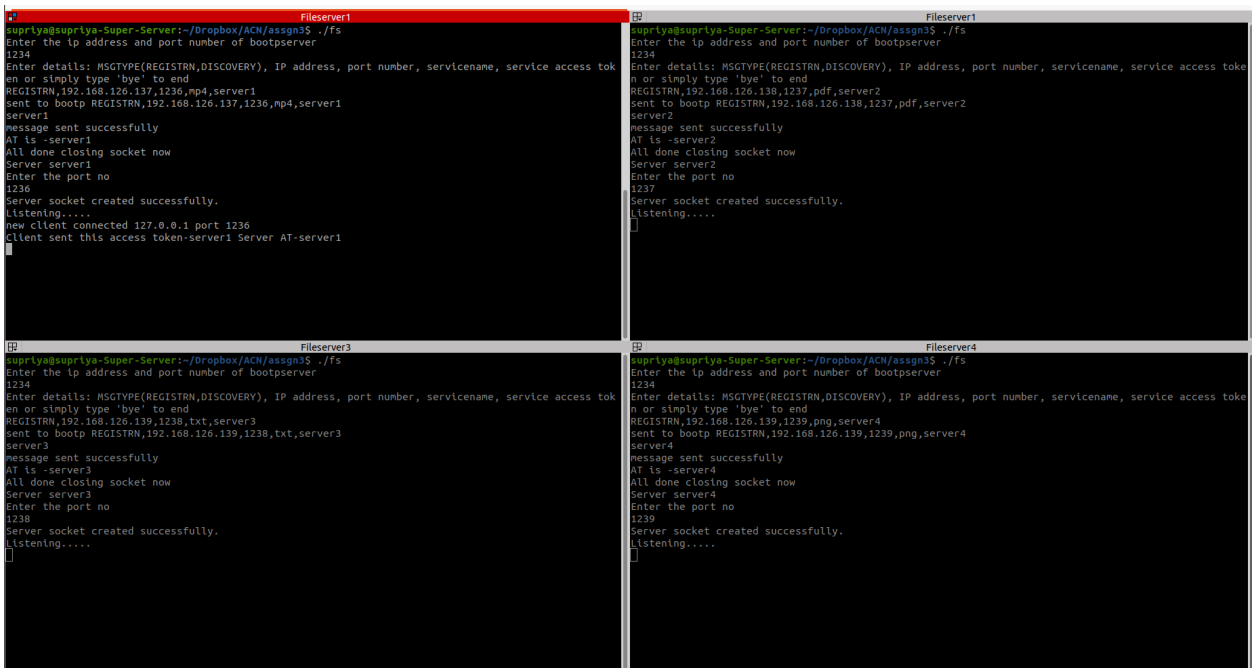
- After that client sends the access token to server and server verifies and confirms that client is valid or not. Then the client sends the filename it wants from the server. Server sends the file and the client exists.

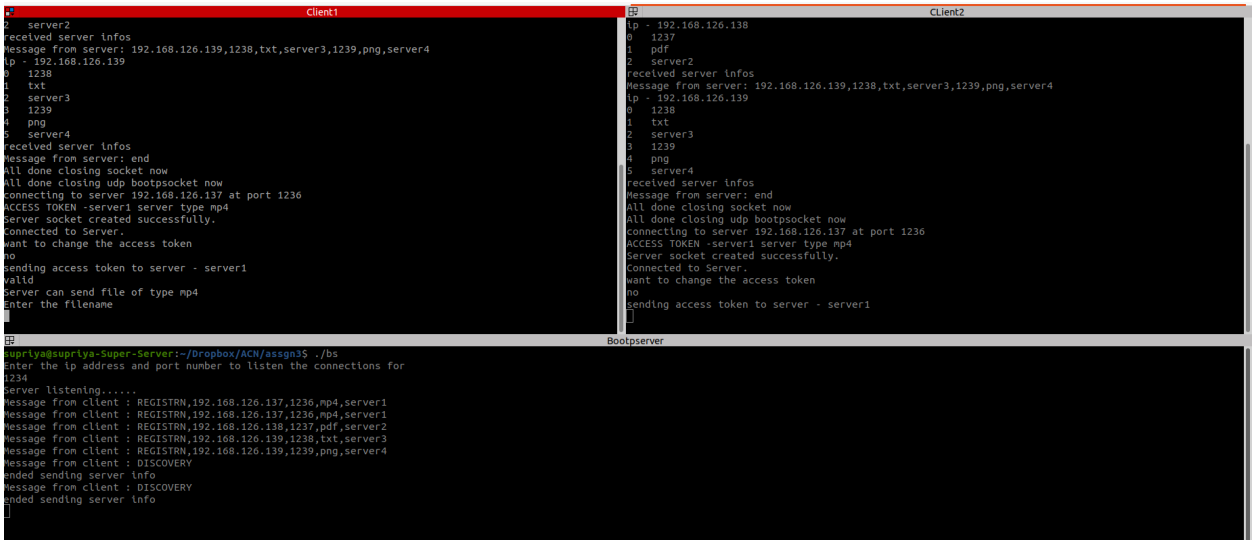
```
CONSOLE  TERMINAL  JUPYTER

Enter details: MSGTYPE(REGISTRN,DISCOVERY), IP address, port
 number, servicename, service access token or simply type 'b
ye' to end
REGISTRN,192.168.126.137,1236,pdf,server1
sent to bootp REGISTRN,192.168.126.137,1236,pdf,server1
server1
message sent successfully
AT is -server1
All done closing socket now
Server server1
Enter the port no
1236
Server socket created successfully.
Listening.....
new client connected 127.0.0.1 port 1236
Client sent this access token-server1 Server AT-server1
client is requesting file-server.pdf
Requested File is found
Total file size: 110592
Bytes read: 4
add to existing sockets list 0
Host disconnected , ip 127.0.0.1 , port 53614

All done closing socket now
All done closing udp bootpsocket now
connecting to server 192.168.126.137 at port 1236
ACCESS TOKEN -server1 server type pdf
Server socket created successfully.
Connected to Server.
want to change the access token
no
sending access token to server - server1
valid
Server can send file of type pdf
Enter the filename
server.pdf
filen server.pdf
filename server.pdf
requesting file server.pdf   server.pdf
found
creating client.pdf
Receiving
file size is - 110592
File successfully Received!
All done closing socket now
supriya@supriya-Super-Server:~/Dropbox/ACN/assgn3$
```

Server connects to multiple clients and sends them files parallely.

Final execution

- A single BootStrap server, 4 different file servers for video/audio files, .pdf, .text files, image files respectively and two clients





- Invalid client check.