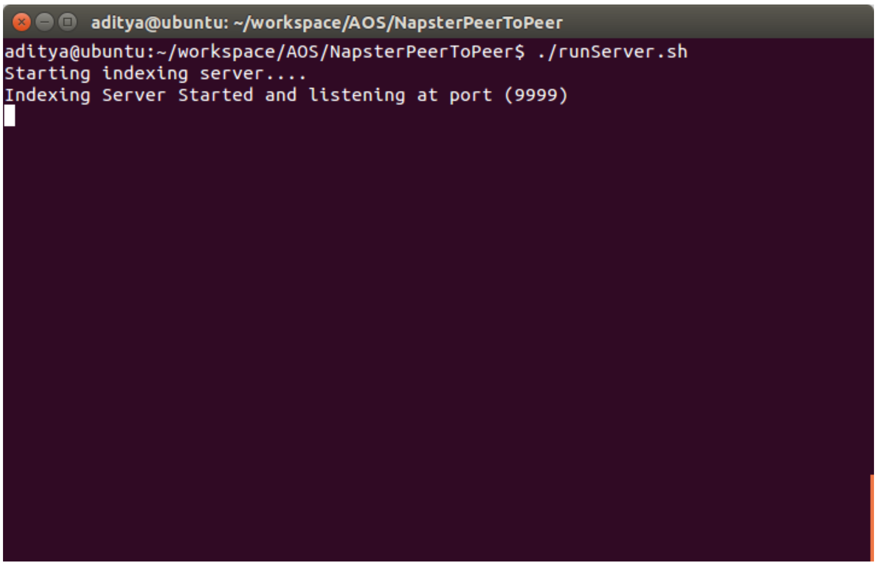
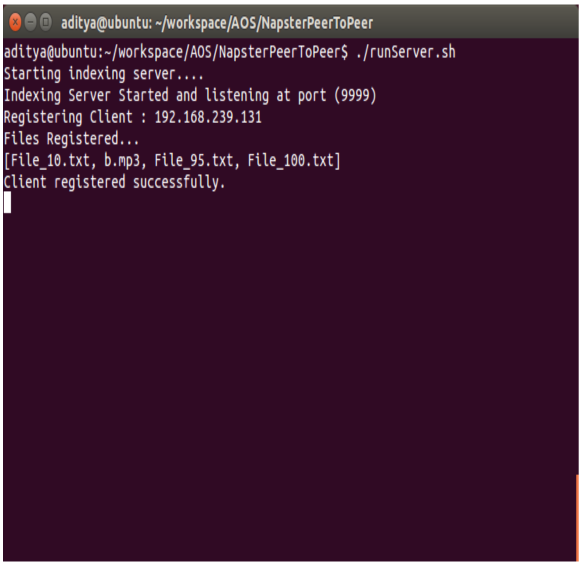
Output after running the Client And Server application.

- **Running a Server.**
  - Execute **command** /NapsterPeerToPeer$ **./runServer.sh**
    - Server gets start and ready to listen other peer request.

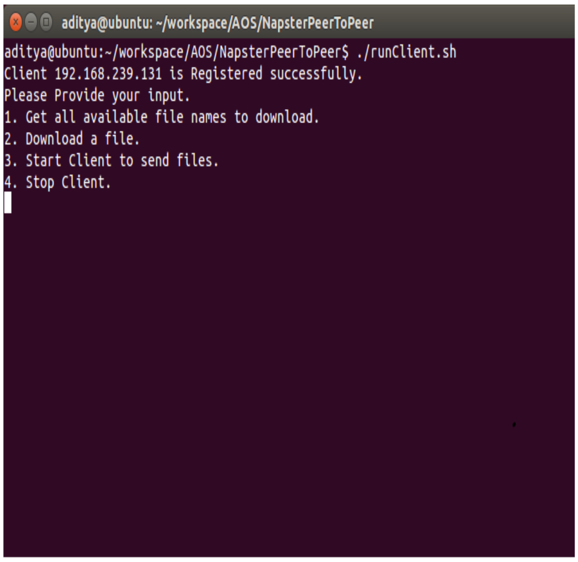


```
aditya@ubuntu:~/workspace/AOS/NapsterPeerToPeer$ ./runServer.sh
Starting indexing server....
Indexing Server Started and listening at port (9999)
```

- **Running a Client**
  - Execute **command** /NapsterPeerToPeer$ **./runClient.sh**
    - Clients gets Start and all the files from folder "ClientData" will be registered in an Indexing Server.



```
aditya@ubuntu:~/workspace/AOS/NapsterPeerToPeer$ ./runServer.sh
Starting indexing server....
Indexing Server Started and listening at port (9999)
Registering Client : 192.168.239.131
Files Registered...
[File_10.txt, b.mp3, File_95.txt, File_100.txt]
Client registered successfully.
```



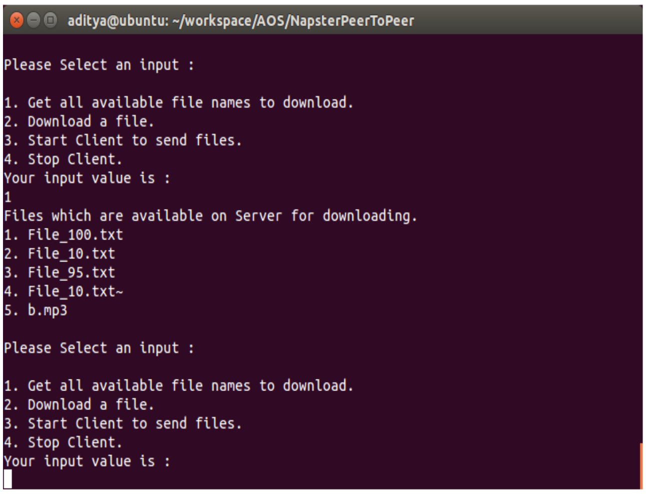
```
aditya@ubuntu:~/workspace/AOS/NapsterPeerToPeer$ ./runClient.sh
Client 192.168.239.131 is Registered successfully.
Please Provide your input.
1. Get all available file names to download.
2. Download a file.
3. Start Client to send files.
4. Stop Client.
```

  - Now you can give input from client.
    1. **Get all available file names to download** → Displays all file names which are registered by all different clients.
    2. **Download a file** → Will ask for another input as file name, which you want to download. And if the Peer is up, will download the file in **Downloaded** folder.

3. **Start Client to send files** → Will make current Client as server. All the files which are registered to Indexing Server will become available to download for other peers.
4. **Stop Client** → Will terminate the program.
5. Also can Use **ctrl + C** to terminate the program.

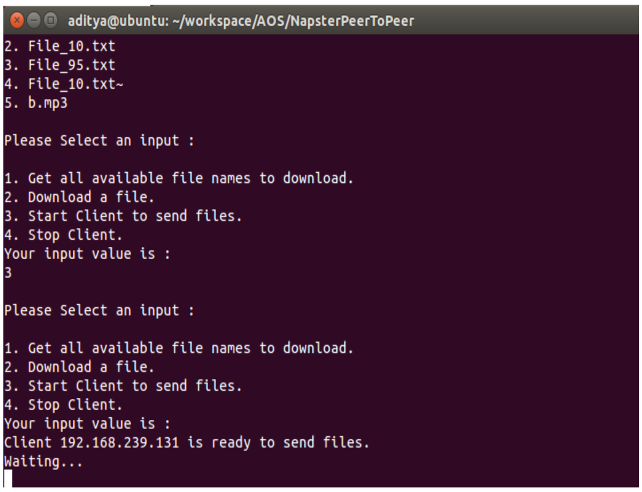
- Below screenshot shows the output after all the options are selected.

Selected option 1 for getting file names.



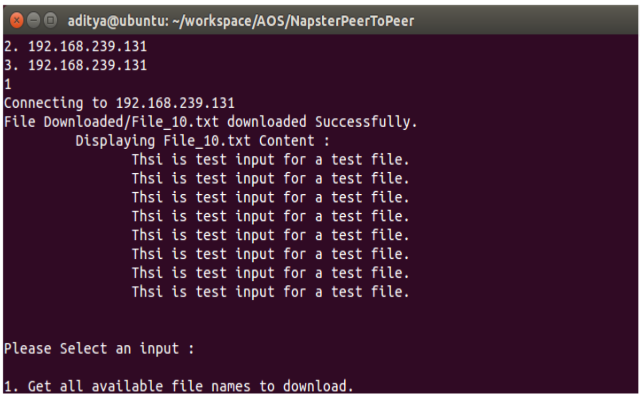
```
Please Select an input :

1. Get all available file names to download.
2. Download a file.
3. Start Client to send files.
4. Stop Client.
Your input value is :
1
Files which are available on Server for downloading.
1. File_100.txt
2. File_10.txt
3. File_95.txt
4. File_10.txt~
5. b.mp3

Please Select an input :

1. Get all available file names to download.
2. Download a file.
3. Start Client to send files.
4. Stop Client.
Your input value is :
```

Selected Ooption 3 : Waiting for another peer to send files.



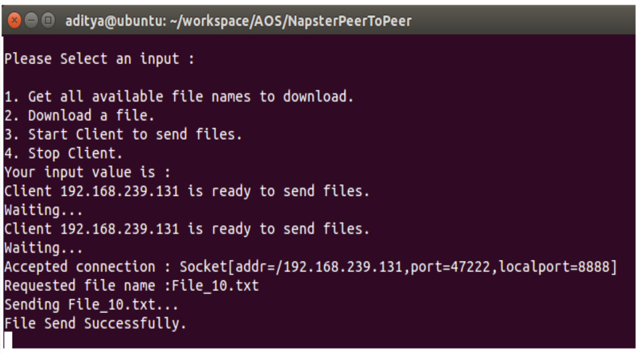
```
2. File_10.txt
3. File_95.txt
4. File_10.txt~
5. b.mp3

Please Select an input :

1. Get all available file names to download.
2. Download a file.
3. Start Client to send files.
4. Stop Client.
Your input value is :
3

Please Select an input :

1. Get all available file names to download.
2. Download a file.
3. Start Client to send files.
4. Stop Client.
Your input value is :
Client 192.168.239.131 is ready to send files.
Waiting...
```

Selected Option 2 : Download a file and Displaying file Content



```
2. 192.168.239.131
3. 192.168.239.131
1
Connecting to 192.168.239.131
File Downloaded/File_10.txt downloaded Successfully.
         Displaying File_10.txt Content :
                Thsi is test input for a test file.
                Thsi is test input for a test file.
                Thsi is test input for a test file.
                Thsi is test input for a test file.
                Thsi is test input for a test file.
                Thsi is test input for a test file.
                Thsi is test input for a test file.
                Thsi is test input for a test file.


Please Select an input :

1. Get all available file names to download.
```

Client acting as Server. Sending files to another peer.



```
Please Select an input :

1. Get all available file names to download.
2. Download a file.
3. Start Client to send files.
4. Stop Client.
Your input value is :
Client 192.168.239.131 is ready to send files.
Waiting...
Client 192.168.239.131 is ready to send files.
Waiting...
Accepted connection : Socket[addr=/192.168.239.131,port=47222,localport=8888]
Requested file name :File_10.txt
Sending File_10.txt...
File Send Successfully.
```

Cases where this program will not work:

1. If serverIp is not configure properly in config.properties. Indexing Server will not start.
2. If the port is not available, or already in use by some other applications. Then you need to kill the other applications which is using that port and has to make sure that same port number is available for all clients. If all peers are not running on same port, then communication between peers is not possible. File Downloading is not possible in this scenario.
3. If file is available on multiple clients and system gives multiple option to download, please give numeric input as display on Terminal. If given input other than displayed value then program will break.