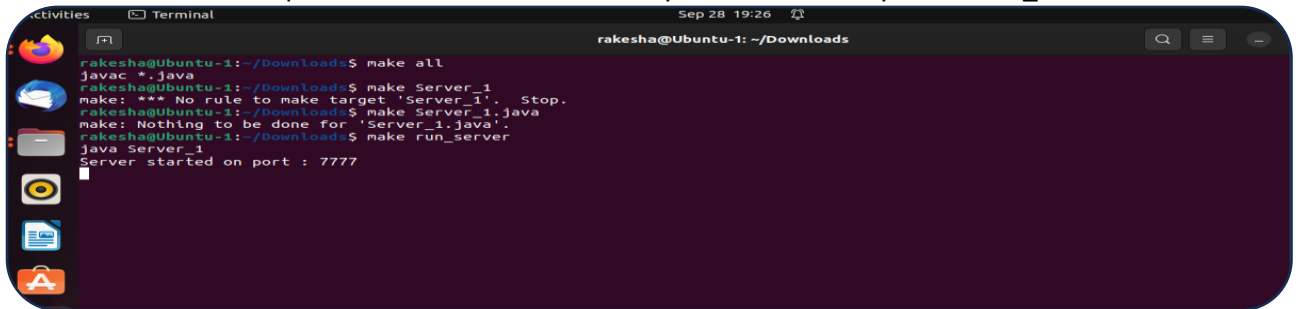


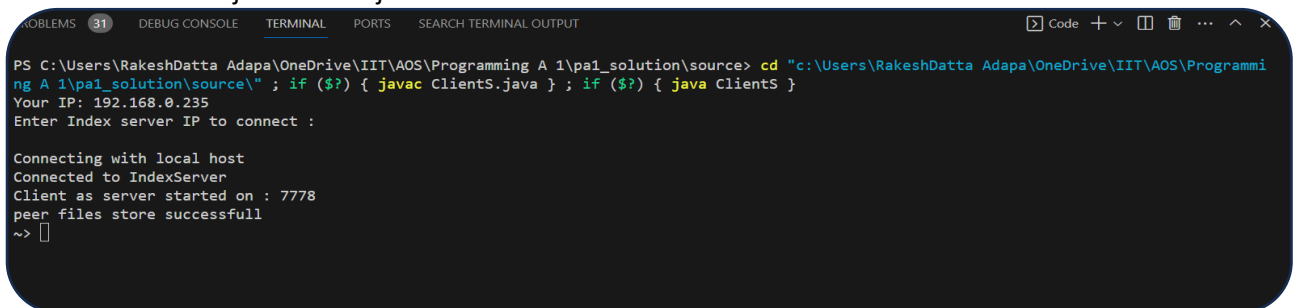
Output - Programming Assignment 1

1. Run the programs in the separate terminals with the command : `javac Server_1.java` to start the server the server is hosted on the port 7777 as it is hard coded or you can also run by “make run_server”



```
rakesha@Ubuntu-1:~/Downloads$ make all
javac *.java
rakesha@Ubuntu-1:~/Downloads$ make Server_1
make: *** No rule to make target 'Server_1'. Stop.
rakesha@Ubuntu-1:~/Downloads$ make Server_1.java
make: Nothing to be done for 'Server_1.java'.
rakesha@Ubuntu-1:~/Downloads$ make run_server
java Server_1
Server started on port : 7777
```

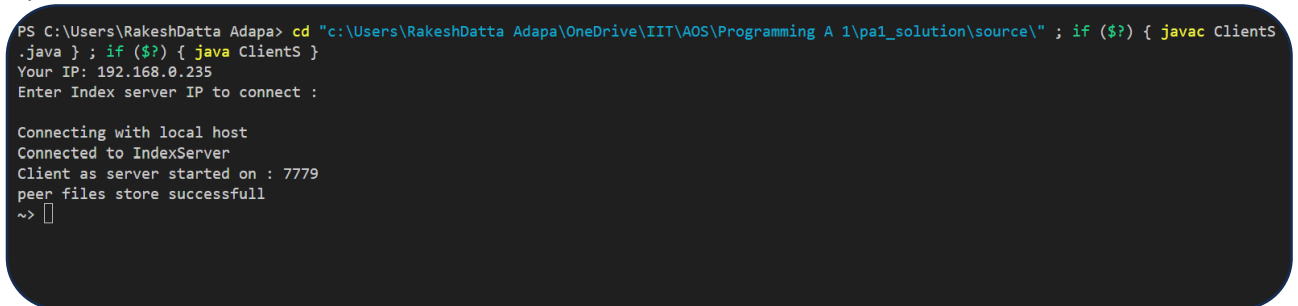
2. Run the command `java ClientS.java` to start the client



```
PS C:\Users\RakeshDatta Adapa\OneDrive\IIT\AOS\Programming A 1\pa1_solution\source> cd "c:\Users\RakeshDatta Adapa\OneDrive\IIT\AOS\Programming A 1\pa1_solution\source\" ; if ($?) { javac ClientS.java } ; if ($?) { java ClientS }
Your IP: 192.168.0.235
Enter Index server IP to connect :

Connecting with local host
Connected to IndexServer
Client as server started on : 7778
peer files store successfull
~>
```

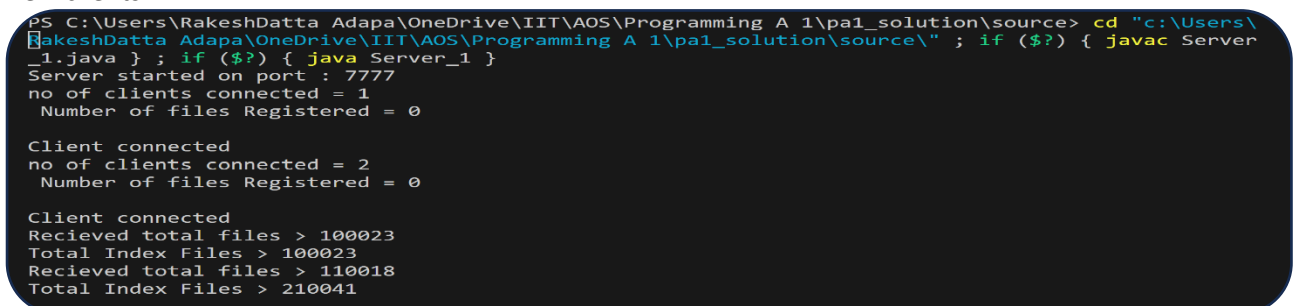
3. Open new Terminal/VM and run the same command to start the second client



```
PS C:\Users\RakeshDatta Adapa\OneDrive\IIT\AOS\Programming A 1\pa1_solution\source> cd "c:\Users\RakeshDatta Adapa\OneDrive\IIT\AOS\Programming A 1\pa1_solution\source\" ; if ($?) { javac ClientS.java } ; if ($?) { java ClientS }
Your IP: 192.168.0.235
Enter Index server IP to connect :

Connecting with local host
Connected to IndexServer
Client as server started on : 7779
peer files store successfull
~>
```

4. Now in the Index server we can see the status of client connections and also server continuously runs for new clients



```
PS C:\Users\RakeshDatta Adapa\OneDrive\IIT\AOS\Programming A 1\pa1_solution\source> cd "c:\Users\RakeshDatta Adapa\OneDrive\IIT\AOS\Programming A 1\pa1_solution\source\" ; if ($?) { javac Server_1.java } ; if ($?) { java Server_1 }
Server started on port : 7777
no of clients connected = 1
Number of files Registered = 0

Client connected
no of clients connected = 2
Number of files Registered = 0

Client connected
Recieved total files > 100023
Total Index Files > 100023
Recieved total files > 110018
Total Index Files > 210041
```

5. Enter the command register in both the terminals to send the meta data of the files to index server. Once the data is sent we will received the message from server as registered.

```
C:\Users\RakeshDatta Adapa> cd "c:\Users\RakeshDatta Adapa\OneDrive\IIT\AOS\Programming A 1\pa1_solution\source\" ; if ($?) { java ClientS } ; if ($?) { javac ClientS.java } ; if ($?) { java ClientS }
Your IP: 192.168.0.235
Enter Index server IP to connect :

Connecting with local host
Connected to IndexServer
Client as server started on : 7779
peer files store successfull
~> register
Registered successfully
~>

PS C:\Users\RakeshDatta Adapa\OneDrive\IIT\AOS\Programming A 1\pa1_solution\source> cd "c:\Users\RakeshDatta Adapa\OneDrive\IIT\AOS\Programming A 1\pa1_solution\source\" ; if ($?) { javac ClientS.java } ; if ($?) { java ClientS }
Your IP: 192.168.0.235
Enter Index server IP to connect :

Connecting with local host
Connected to IndexServer
Client as server started on : 7778
peer files store successfull
~> register
Registered successfully
~>
```

6. Now we can enter the search followed by the file name to send the search query to index server ex: search _txt and also we get all the required information like port and ipadress of the server that holds the information.

```
n
~> search 99988.txt
0) onekbfiles_99988.txt 1 192.168.0.235 7778
1) ptwo_oneKB_99988.txt 2 192.168.0.235 7778
You want to download any file > Enter Y and then index
Y
```

7. If there were no files we receive the message like No results found.

```
~> search text
No results found
~> search _9998
0) onekbfiles_9998.txt 1 192.168.0.235 7778
1) onekbfiles_99980.txt 1 192.168.0.235 7778
2) onekbfiles_99981.txt 1 192.168.0.235 7778
3) onekbfiles_99982.txt 1 192.168.0.235 7778
4) onekbfiles_99983.txt 1 192.168.0.235 7778
```

8. Now you also download the required file by entering y followed by the index number

```
rakesh - Personal > IIT > AOS > Programming A 1 > pa1_solution > source > PeerFiles > downloads
```

Name	Status	Date modified	Type	Size
peer2testfile_9889.txt		9/28/2023 7:59 PM	Text Document	0 KB
1GB.bin		9/28/2023 2:19 PM	BIN File	1,024,000 KB
onekbfiles_99997.txt		9/28/2023 2:14 PM	Text Document	1 KB
p3_4.txt		9/28/2023 9:48 AM	Text Document	0 KB
p1_2.txt		9/28/2023 9:46 AM	Text Document	1 KB
p1_6.txt		9/28/2023 6:09 AM	Text Document	1 KB
p1_1.txt		9/28/2023 6:08 AM	Text Document	1 KB

```

earch 9889.txt
onekbfiles_19889.txt 1 192.168.0.235 7778
1) onekbfiles_29889.txt 1 192.168.0.235 7778
2) onekbfiles_39889.txt 1 192.168.0.235 7778
3) onekbfiles_49889.txt 1 192.168.0.235 7778
4) onekbfiles_59889.txt 1 192.168.0.235 7778
5) onekbfiles_69889.txt 1 192.168.0.235 7778
6) onekbfiles_79889.txt 1 192.168.0.235 7778
7) onekbfiles_89889.txt 1 192.168.0.235 7778
8) onekbfiles_9889.txt 1 192.168.0.235 7778
9) onekbfiles_99889.txt 1 192.168.0.235 7778
10) peer2testfile_9889.txt 2 192.168.0.235 7778
11) ptwo_oneKB_19889.txt 2 192.168.0.235 7778
12) ptwo_oneKB_29889.txt 2 192.168.0.235 7778
13) ptwo_oneKB_39889.txt 2 192.168.0.235 7778
14) ptwo_oneKB_49889.txt 2 192.168.0.235 7778
15) ptwo_oneKB_59889.txt 2 192.168.0.235 7778
16) ptwo_oneKB_69889.txt 2 192.168.0.235 7778
17) ptwo_oneKB_79889.txt 2 192.168.0.235 7778
18) ptwo_oneKB_89889.txt 2 192.168.0.235 7778
19) ptwo_oneKB_9889.txt 2 192.168.0.235 7778
20) ptwo_oneKB_99889.txt 2 192.168.0.235 7778
You want to download any file > Enter Y and then index
y
10 [
Requested file: peer2testfile_9889.txt, has been downloaded to directory: C:\Users\RakeshDatta Adapa\OneDrive\IIT\AOS\Programming A 1\pa1_sol
tion\source\PeerFiles\downloads
Display file peer2testfile_9889.txt

```

9. If you enter any other text it will display as unknown command

```

~> download
Unknown Command
> Register
> Search
~> [

```

10. Enter end to exit the server connection.

```

~> end
Connection Terminated with server
[

```