Follow the below steps to execute Assignment no 3 step by step:-

1) Change the Servers.txt file under each client directory and put the server details (IP and Port) as per below.

   - If you want to execute this assignment for 8 clients and 8 Servers then keep the details (IP and Port) of 8th Server at top and reset as per sequence.

     Example:-

     192.168.1.101 25008

     192.168.1.101 25001

     192.168.1.101 25002

     192.168.1.101 25003

     192.168.1.101 25004

     192.168.1.101 25005

     192.168.1.101 25006

     192.168.1.101 25007

   - In above example you can see that ip is 192.168.1.101 and 25008 is port number.

2) If you want to execute this program from multiple machines then copy the IP of that machine in server.txt file as per below.

   Example:-

   If client 2 (port 25002) and 3 (port 25003) are running from different machine which has IP address **192.168.1.123** and all other six clients running from one machine which has IP 192.168.1.101 then update the Server.txt file as below

   192.168.1.101 25008

   192.168.1.101 25001

   192.168.1.123 25002

   192.168.1.123 25003

   192.168.1.101 25004

   192.168.1.101 25005

192.168.1.101 25006

192.168.1.101 25007

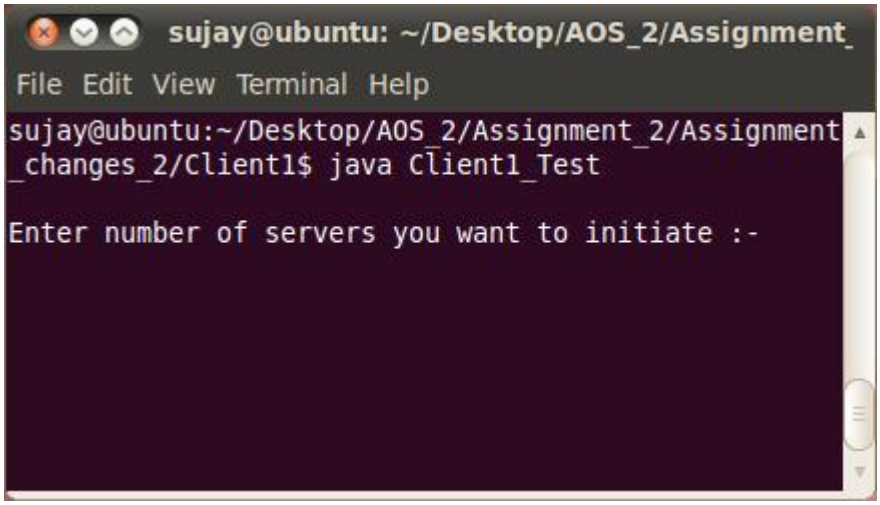
3) Follow below steps to execute program.

- Start the new terminal and go to the path where client directory is copied and follow below steps. Suppose we are executing for client1 then follow below steps.

  - cd Client1
  - javac Client1_Test.java
  - java Client1_Test

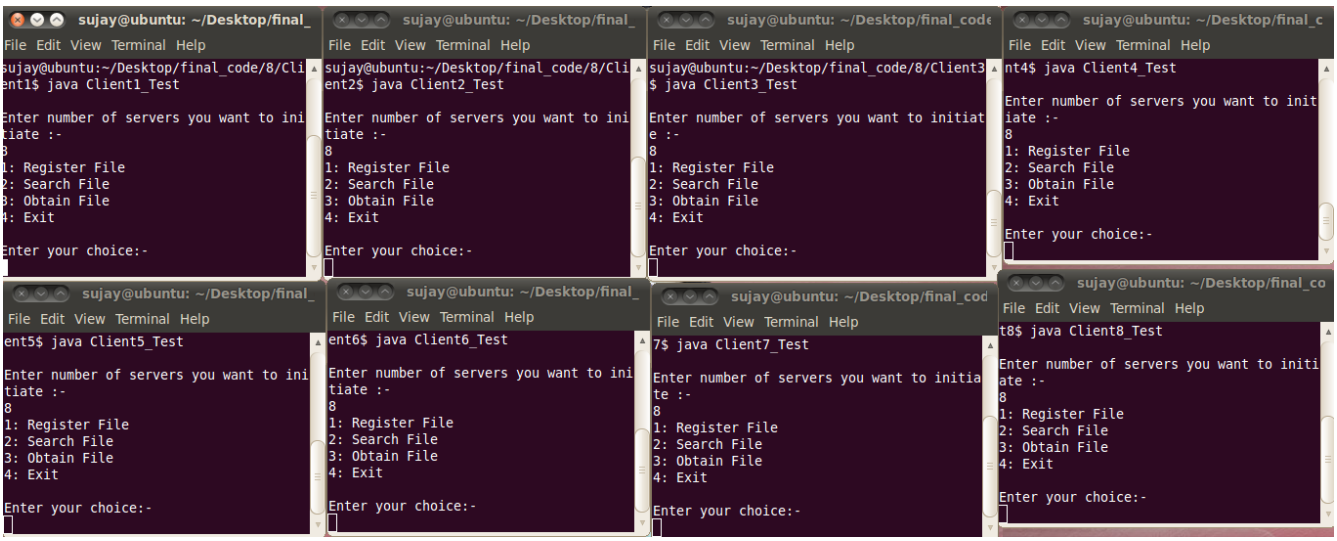
4) Follow step 3 for all 8 clients and do not enter anything on console.

5) You will Get below screen on all 8 console. You will be asked to "Enter number of servers you want to initiate:-" . Do not enter anything till all 8 clients get started otherwise you will get connection refused exception.

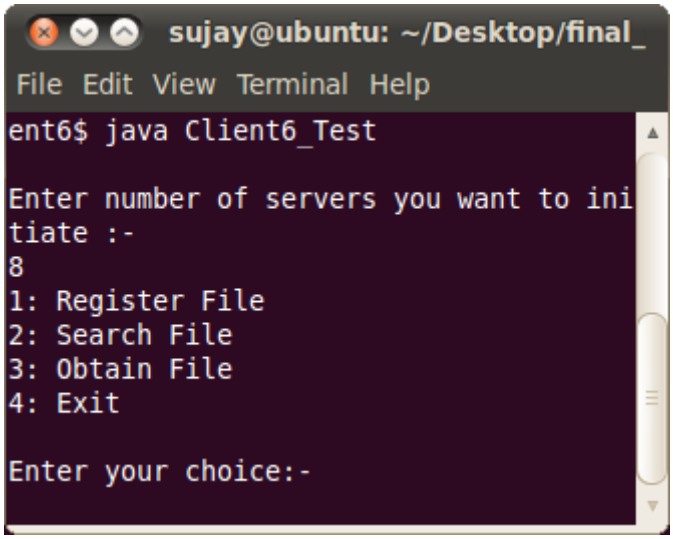


6) Once all Clients get started then enter 8 on all console. All 8 Consoles will look like below.

7) Now all 8 clients and Server are working and they are connected to each other. You will see the client window and server's are running in background.



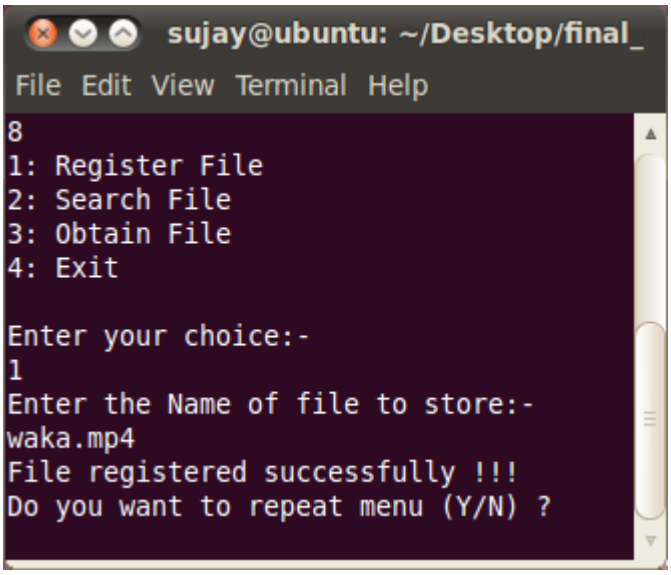
8) User will be asked to enter his choice to perform put, get and delete operations as shown below.



9) All client files should be kept inside Myfiles directory provided for each client.

1) Register Operation:-

- User should enter 1 on console to do put operation



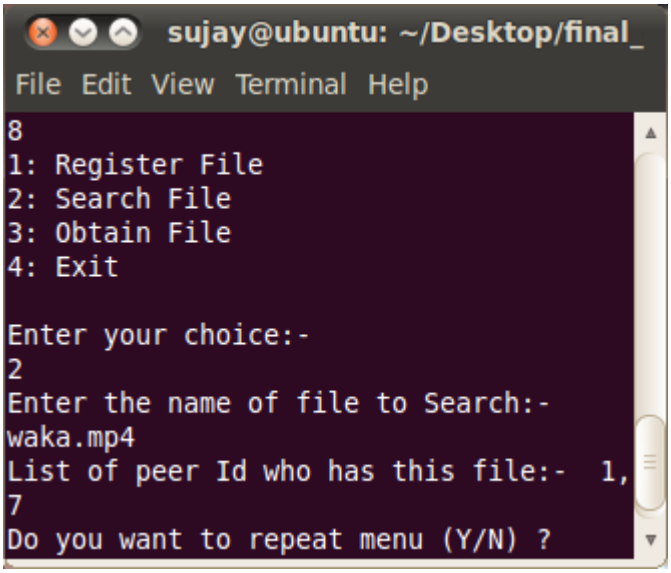
- User will be asked to enter filename to register.
- Once file name get registered on primary and replica server user will get message that "File registered sucessfully" from server.
- User will be asked if he want to continue any more operations. If user wish to do then he can type 'Y' to repeat menu else user can enter 'N' to exit.

Register Resilience:-

- Users file name will be stored at multiple locations primary and replica server.
- Also file will be sent to another server to maintain replica. This replica will be retrieved when primary server fails to deliver the file.

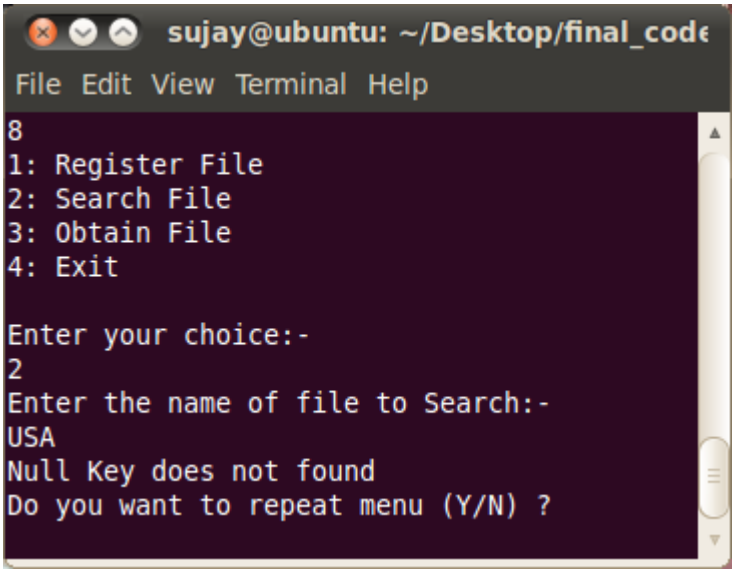
2) Search Operation:-

– User should enter 2 to do search operation as shown below.



– User will be asked to enter file name to search. We can see above, user entered waka.mp4 and peer Id who has this file is returned.

– If user enters a key which is not registered then get will return NULL and message that "Key does not found" as shown below.

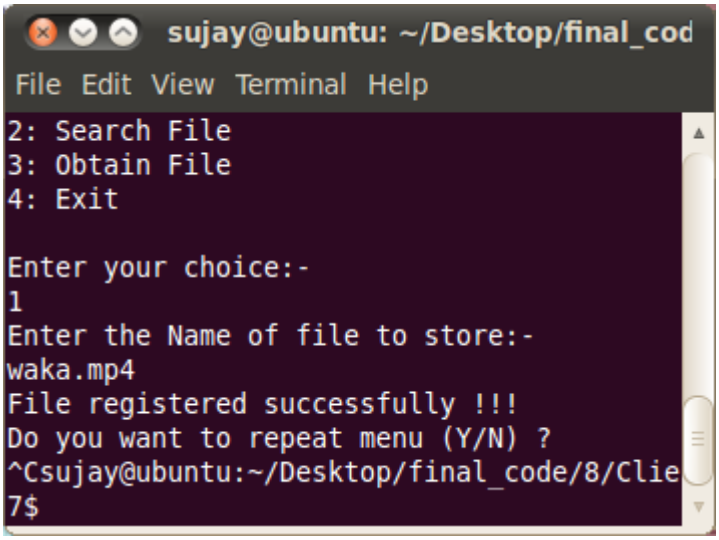


Search Resilience:-

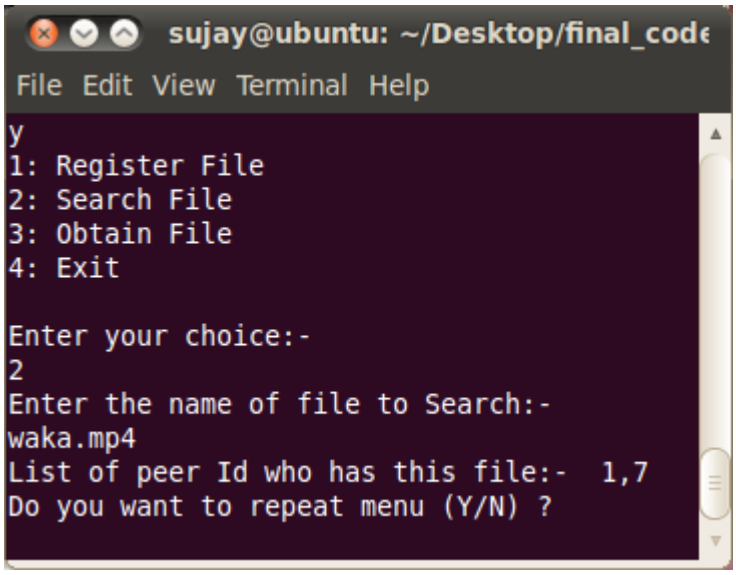
If primary source failed to deliver peerid due to connection issue, then replica server will provide the file name and peer id to user as shown below.

Example:-

- Suppose user stores file name waka.mp4 and peer id 1 and 7.
- Client will calculate which server to connect based on below Hash Function.
- Hash Value = (Sum of ascii value of waka.mp4) % 8;
- Hash value = (230) % 8;
- Hash Value = 7;
- Primary server is 7 in this case. We can kill Client & Server 7 to see if replica provides the value associated with file name waka.mp4.
- In Below screenshot i have killed server 7 which was primary server to store filename waka.mp4.

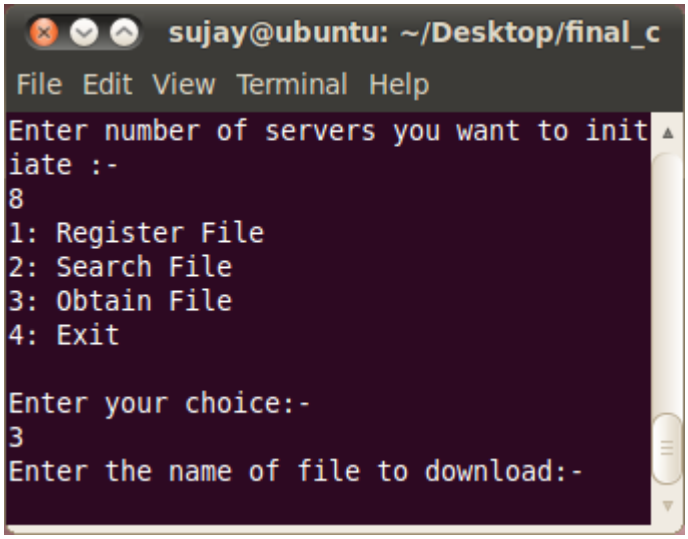


- Now I will try to get the file name from Replica Server. Below is the screen shot that peer id 1,7 returned by replica server.

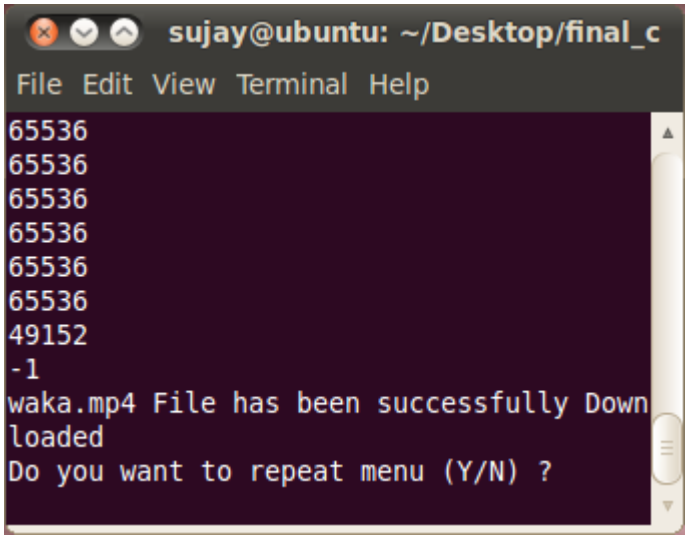


3) Obtain Operation:-

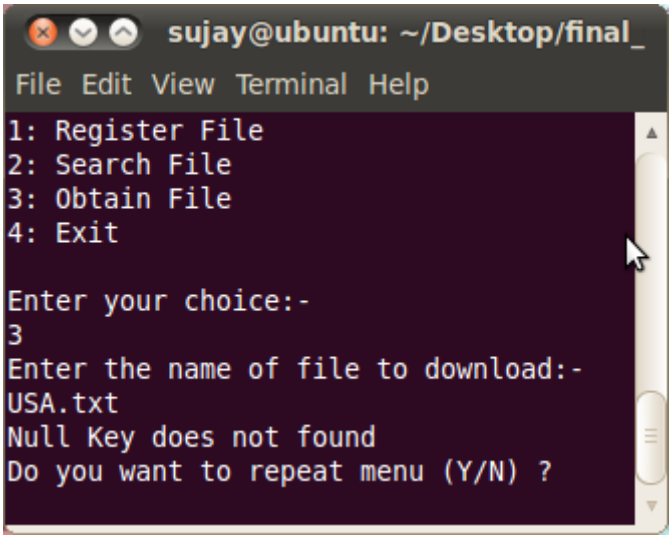
- User should enter 3 to do obtain operation. As shown below.



- User will be asked to enter file name to obtain as shown above. Once user enters filename, peer id who has this file will be returned to user.
- Then user will be asked to entre peerid from which he want to receive file.
- Once user gets the peerid that peer server will be contacted to get value.



- If filename does not found then user will get message that “Failed key does not found” as below.



Obtain Resilience:-

If user performs obtain operation and primary server is shutdown then replica server will be contacted to obtain file. All the details to connect to replica will be kept hidden from user.

When user goes for registration file will be sent to other server to store replica.

Example:

1. If user wants to register “xyz.txt” file.
2. Generated hashcode is 1.
3. Now file is going to register at PEER_SERVER_1.
4. Replica of file is going to save in SERVER_7.
5. All files will be stored at Myfiles directory
6. When user get SocketConnection exception from primary server while trying to download file then user will connect to replica server automatically to get file.