OUTPUT FILE

The screenshot of the output is shown below.

In one of the peer's terminal : The filenames are stored in Hash-table. Depending on the hashing of the message, files are be stored in different server. For example with two peers containing different files, the output will be as follows

```
root@ubuntu:/Assignment3# java peer -i 3 -h ubuntu -p 4002 -m 32 -r ubuntu -s 40
00
Hostname: ubuntu Port: 4002 ID: 3 NextHostname: ubuntu NextPort: 4000 NextID: 1
MaxID: 32 isFirstPeer: false Hashtable: {}
Hostname: ubuntu Port: 4002 ID: 3 NextHostname: ubuntu NextPort: 4000 NextID: 1
MaxID: 32 isFirstPeer: false Hashtable: {}
Hostname: ubuntu Port: 4002 ID: 3 NextHostname: ubuntu NextPort: 4000 NextID: 1
MaxID: 32 isFirstPeer: false Hashtable: {}
Hostname: ubuntu Port: 4002 ID: 3 NextHostname: ubuntu NextPort: 4000 NextID: 1
MaxID: 32 isFirstPeer: false Hashtable: {}
Hostname: ubuntu Port: 4002 ID: 3 NextHostname: ubuntu NextPort: 4000 NextID: 1
MaxID: 32 isFirstPeer: false Hashtable: {}
Server 3 Started!
Hostname: ubuntu Port: 4002 ID: 3 NextHostname: ubuntu NextPort: 4000 NextID: 1
MaxID: 32 isFirstPeer: false Hashtable: {24=ADD a.txt}
Hostname: ubuntu Port: 4002 ID: 3 NextHostname: ubuntu NextPort: 4000 NextID: 1
MaxID: 32 isFirstPeer: false Hashtable: {26=ADD c.txt, 24=ADD a.txt}
```

In Peer 1 terminal :

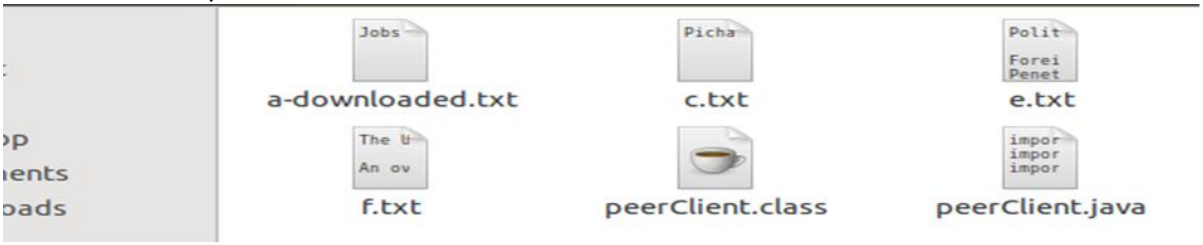
```
 Trying to connect with server. 127.0.1.1
[b.txt, d.txt, peerClient.java~, b.txt~, a.txt, peerClient.java, d.txt~, peerCli
ent.class, peerClient.java~~]
Enter the  filename to be registered from the above list
a.txt
Enter the filename to be searched
c.txt
The requested file found at localhost and can be accessed via port 41430
File received from the peer
Time taken to download 4 milliseconds
```

In Peer 1 Directory

In Peer2 terminal:

```
 Trying to connect with server. 127.0.1.1
[peerClient.java~, c.txt, f.txt, peerClient.java, e.txt, peerClient.class]
Enter the  filename to be registered from the above list
c.txt
Enter the filename to be searched
a.txt
The requested file found at localhost and can be accessed via port 41422
Files Received from the peer
Time taken to download 3 milliseconds
```

In Peer 2 directory.



Once you run the peer code for two or more peers, a Distributed Hash Table log messages will be generated in same directory under (/Assignment3 /dhtmessage.p2plog). where you can see the whole list of commands that are given from multiple peers and their results.

Sample screen shot of the log message is shown below.

a) Peer1 server login.

```
----------------> peer 1 server loging <----------------
[Received - PeerID: 1 Portnum: 4000] PULL 3171_a3/1.0 1 3CRLFubuntu 4002CRLF
[Sent - PeerID: 1 Portnum: 4000]     3171_a3/1.0 PULL 1 200 okCRLFblahCRLF
----------------> peer 1 server loging <----------------
[Received - PeerID: 1 Portnum: 4000] DONE 3171_a3/1.0 0CRLF
[Sent - PeerID: 1 Portnum: 4000]
----------------> peer 1 server loging <----------------
[Received - PeerID: 1 Portnum: 4000] ADD a.txt
----------------> peer 3 server loging <----------------
[Received - PeerID: 3 Portnum: 4002] ADD 3171_a3/1.0 0 1CRLFADD a.txtCRLF
[Sent - PeerID: 3 Portnum: 4002]     3171_a3/1.0 ADD 0 200 okCRLF
[Sent - PeerID: 1 Portnum: 4000]     3171_a3/1.0 ADD 0 400
NotResponsibleCRLFADD a.txtCRLF
----------------> peer 1 server loging <----------------
[Received - PeerID: 1 Portnum: 4000] ID 3171_a3/1.0 0 4CRLF
[Sent - PeerID: 1 Portnum: 4000]     3171_a3/1.0 ID 1 301 redirectCRLFubuntu
4002CRLF
```

Now Peer 3 joins the peer to peer network and the operations are shared concurrently between different peers.

Peer2 server and Peer3 server login

```
----------------> peer 2 server loging <----------------
[Received - PeerID: 2 Portnum: 4001] ADD 3171_a3/1.0 0 1CRLFADD a.txtCRLF
----------------> peer 3 server loging <----------------
[Received - PeerID: 3 Portnum: 4002] ADD 3171_a3/1.0 0 2CRLFADD a.txtCRLF
[Sent - PeerID: 3 Portnum: 4002]     3171_a3/1.0 ADD 0 200 okCRLF
[Sent - PeerID: 2 Portnum: 4001]     3171_a3/1.0 ADD 0 400
NotResponsibleCRLFADD a.txtCRLF
[Sent - PeerID: 1 Portnum: 4000]     3171_a3/1.0 ADD 0 400
NotResponsibleCRLFADD a.txtCRLF
----------------> peer 2 server loging <----------------
[Received - PeerID: 2 Portnum: 4001] ADD c.txt
----------------> peer 3 server loging <----------------
[Received - PeerID: 3 Portnum: 4002] ADD 3171_a3/1.0 0 2CRLFADD c.txtCRLF
[Sent - PeerID: 3 Portnum: 4002]     3171_a3/1.0 ADD 0 200 okCRLF
[Sent - PeerID: 2 Portnum: 4001]     3171_a3/1.0 ADD 0 400
NotResponsibleCRLFADD c.txtCRLF
[Received - PeerID: 1 Portnum: 4000] QUERY c.txt
[Sent - PeerID: 1 Portnum: 4000]     3171_a3/1.0 ADD 0 400
NotResponsibleCRLFQUERY c.txtCRLF
[Received - PeerID: 2 Portnum: 4001] QUERY a.txt
[Sent - PeerID: 2 Portnum: 4001]     3171_a3/1.0 ADD 0 400
NotResponsibleCRLFQUERY a.txtCRLF
```

You can see the list of operations (read(get)/write(put)), given by the peers and its corresponding results with their status message. Similarly, if you login with 8 server (peers) concurrently, overall logs for 8 peers and their results will be generated in this file.