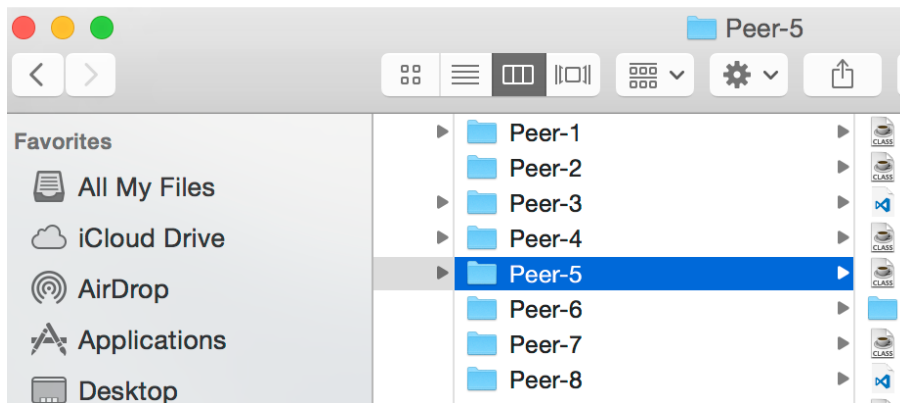
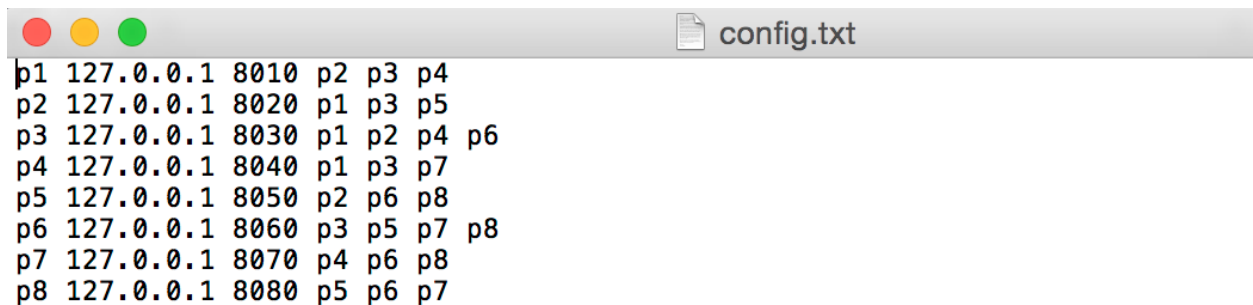


OUTPUT FILE

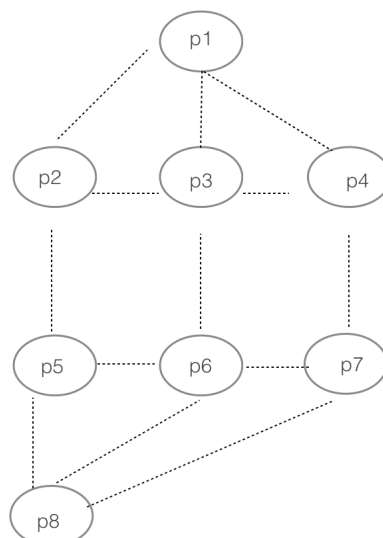
1) In this assignment, we are using 8 Peers and 8 Servers. Below is the basic setup for this. Server code is inside of Peer code.



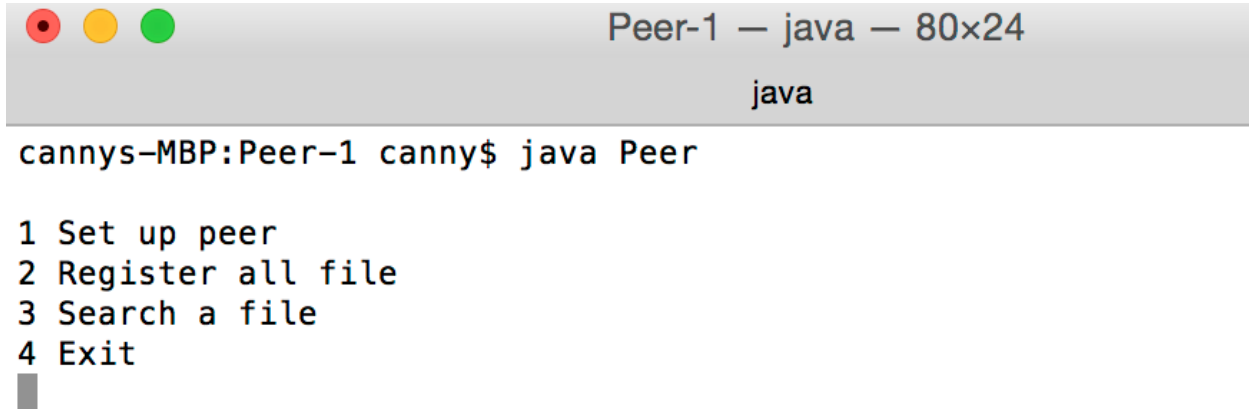
→ config.txt file is set up for all peer as a common. Every Peer will read their neighbor Peer information from here.



-Peer Structure



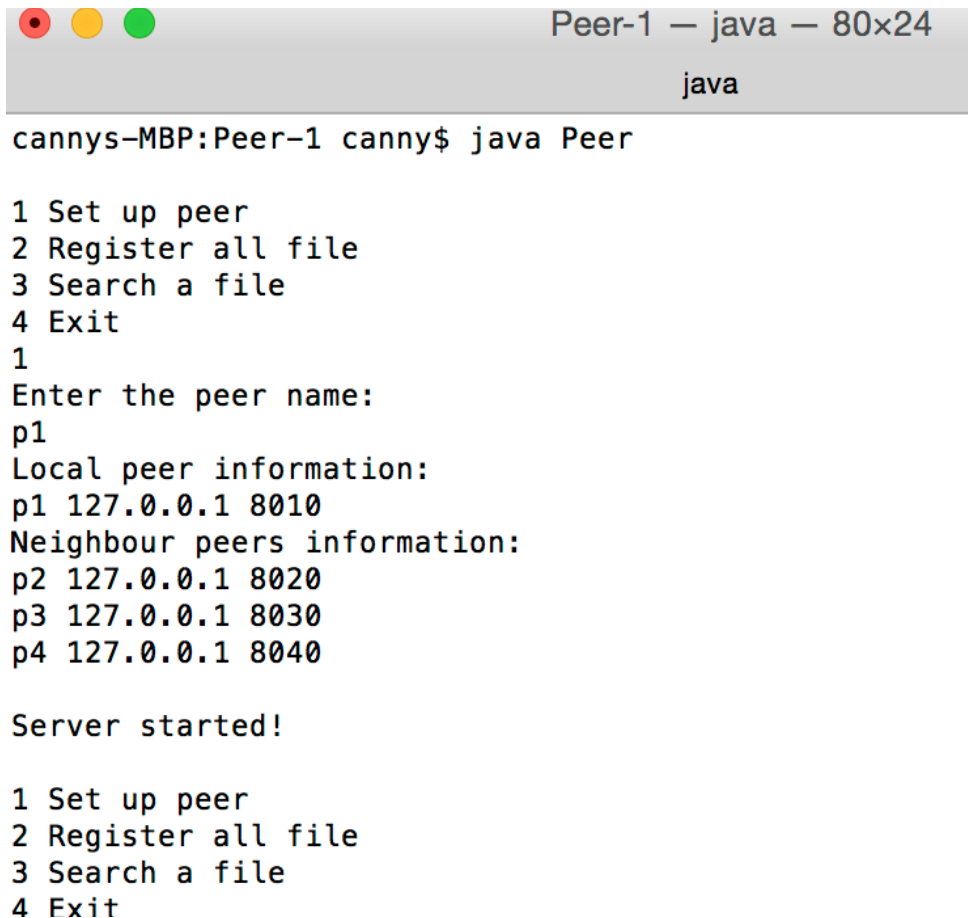
2)Set up pear name



```
Peer-1 — java — 80x24
java
cannys-MBP:Peer-1 canny$ java Peer

1 Set up peer
2 Register all file
3 Search a file
4 Exit
█
```

—>select option 1 to give peer a name(P1,p2,p3...p8) ,after entering name,server will automatically start and show the neighbor peers information.



```
Peer-1 — java — 80x24
java
cannys-MBP:Peer-1 canny$ java Peer

1 Set up peer
2 Register all file
3 Search a file
4 Exit
1
Enter the peer name:
p1
Local peer information:
p1 127.0.0.1 8010
Neighbour peers information:
p2 127.0.0.1 8020
p3 127.0.0.1 8030
p4 127.0.0.1 8040

Server started!

1 Set up peer
2 Register all file
3 Search a file
4 Exit
```

→All the Peers must be set up names.

Peer-1	Peer-2	Peer-3	Peer-4
<pre>cannys-MBP:Peer-1 canny\$ 1 Set up peer 2 Register all file 3 Search a file 4 Exit 1 Enter the peer name: p1 Local peer information: p1 127.0.0.1 8010 Neighbour peers information: p2 127.0.0.1 8020 p3 127.0.0.1 8030 p4 127.0.0.1 8040 Server started! 1 Set up peer 2 Register all file 3 Search a file 4 Exit 2 File .DS_Store is do_re</pre>	<pre>cannys-MBP:Peer-2 canny\$ cannys-MBP:Peer-2 canny\$ 1 Set up peer 2 Register all file 3 Search a file 4 Exit 1 Enter the peer name: p2 Local peer information: p2 127.0.0.1 8020 Neighbour peers information: p1 127.0.0.1 8010 p3 127.0.0.1 8030 p5 127.0.0.1 8050 Server started! 1 Set up peer 2 Register all file 3 Search a file 4 Exit</pre>	<pre>cannys-MBP:Peer-3 canny\$ j 1 Set up peer 2 Register all file 3 Search a file 4 Exit 1 Enter the peer name: p3 Local peer information: p3 127.0.0.1 8030 Neighbour peers information: p1 127.0.0.1 8010 p2 127.0.0.1 8020 p4 127.0.0.1 8040 p6 127.0.0.1 8060 Server started! 1 Set up peer 2 Register all file 3 Search a file 4 Exit</pre>	<pre>Peer-4 — java — java cannys-MBP:Peer-4 canny\$ java Peer 1 Set up peer 2 Register all file 3 Search a file 4 Exit 1 Enter the peer name: p4 Local peer information: p4 127.0.0.1 8040 Neighbour peers information: p1 127.0.0.1 8010 p3 127.0.0.1 8030 p7 127.0.0.1 8070 Server started! 1 Set up peer 2 Register all file 3 Search a file 4 Exit 2</pre>
<pre>cannys-MBP:Peer-5 canny\$ 1 Set up peer 2 Register all file 3 Search a file 4 Exit 1 Enter the peer name: p5 Local peer information: p5 127.0.0.1 8050 Neighbour peers information: p2 127.0.0.1 8020 p6 127.0.0.1 8060 p8 127.0.0.1 8080 Server started! 1 Set up peer 2 Register all file 3 Search a file 4 Exit 2</pre>	<pre>cannys-MBP:Peer-6 canny\$ 1 Set up peer 2 Register all file 3 Search a file 4 Exit 1 Enter the peer name: p6 Local peer information: p6 127.0.0.1 8060 Neighbour peers information: p3 127.0.0.1 8030 p5 127.0.0.1 8050 p7 127.0.0.1 8070 p8 127.0.0.1 8080 Server started! 1 Set up peer 2 Register all file 3 Search a file 4 Exit</pre>	<pre>cannys-MBP:Peer-7 canny\$ j 1 Set up peer 2 Register all file 3 Search a file 4 Exit 1 Enter the peer name: p7 Local peer information: p7 127.0.0.1 8070 Neighbour peers information: p4 127.0.0.1 8040 p6 127.0.0.1 8060 p8 127.0.0.1 8080 Server started! 1 Set up peer 2 Register all file 3 Search a file 4 Exit 2</pre>	<pre>Peer-8 — java — java cannys-MBP:Peer-8 canny\$ java Peer 1 Set up peer 2 Register all file 3 Search a file 4 Exit 1 Enter the peer name: p8 Local peer information: p8 127.0.0.1 8080 Neighbour peers information: p5 127.0.0.1 8050 p6 127.0.0.1 8060 p7 127.0.0.1 8070 Server started! 1 Set up peer 2 Register all file 3 Search a file 4 Exit</pre>

3) Register Function

Now we will see register file function.

press 2 to register all file from folder.

```
1 Set up peer
2 Register all file
3 Search a file
4 Exit
2
File .DS_Store is do_registered !
File 1.txt is do_registered !
File 13.txt is do_registered !
File 19.txt is do_registered !
File 2.txt is do_registered !
File 3.txt is do_registered !
File 44.txt is do_registered !
```

4) Search

→we will search file which is on peer-3 from peer-1.

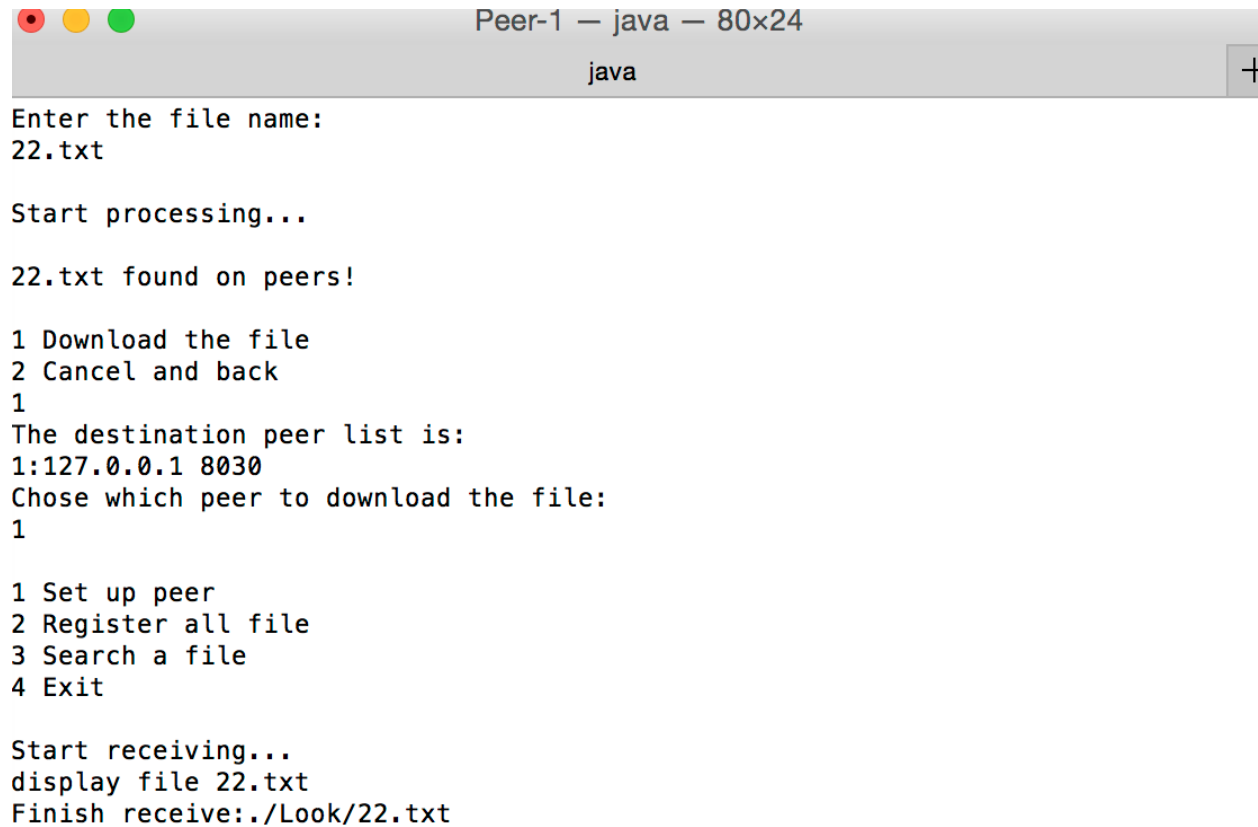
```
1 Set up peer
2 Register all file
3 Search a file
4 Exit
3
Enter the file name:
22.txt

Start processing...

22.txt found on peers!

1 Download the file
2 Cancel and back
```

—>Press 1 for download that file from source peer.



```
Peer-1 - java - 80x24
java
Enter the file name:
22.txt

Start processing...

22.txt found on peers!

1 Download the file
2 Cancel and back
1
The destination peer list is:
1:127.0.0.1 8030
Chose which peer to download the file:
1

1 Set up peer
2 Register all file
3 Search a file
4 Exit

Start receiving...
display file 22.txt
Finish receive:./Look/22.txt
```

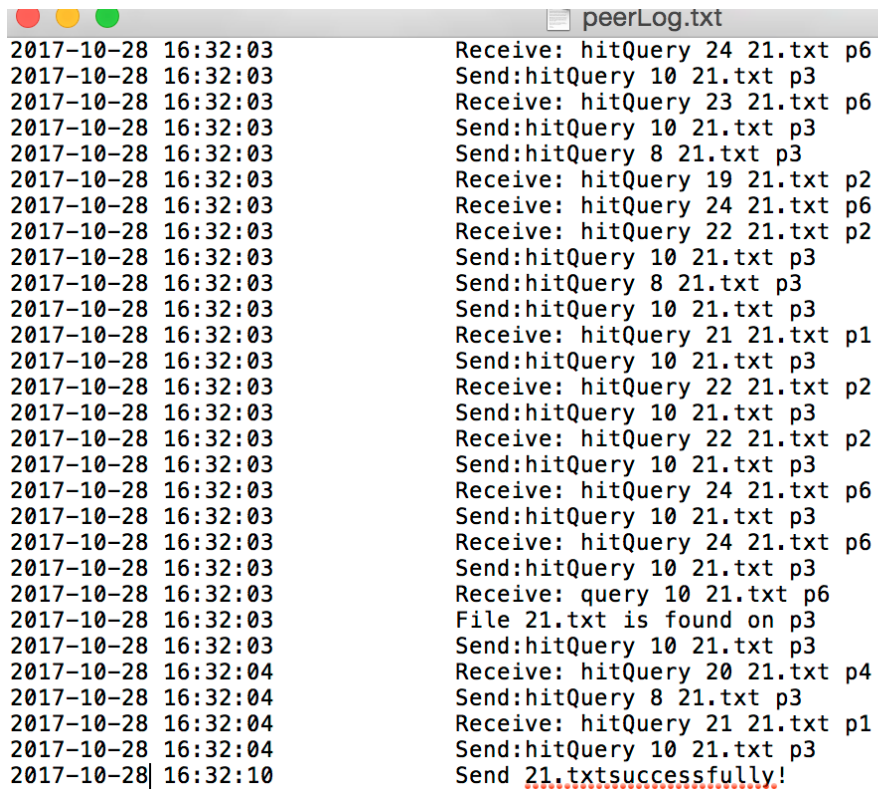
5)Concurrency

—>when we try to download file to two different peer from common peer.

Common peer terminal.

```
1 Set up peer
2 Register all file
3 Search a file
4 Exit
22.txt is on p3
22.txt is on p3
22.txt is on p3
Sent:100.0%
Success
21.txt is on p3
21.txt is on p3
21.txt is on p3
21.txt is on p3
Sent:100.0%
Success
```

→Common peer log file is shown below.



```
peerLog.txt
2017-10-28 16:32:03      Receive: hitQuery 24 21.txt p6
2017-10-28 16:32:03      Send:hitQuery 10 21.txt p3
2017-10-28 16:32:03      Receive: hitQuery 23 21.txt p6
2017-10-28 16:32:03      Send:hitQuery 10 21.txt p3
2017-10-28 16:32:03      Send:hitQuery 8 21.txt p3
2017-10-28 16:32:03      Receive: hitQuery 19 21.txt p2
2017-10-28 16:32:03      Receive: hitQuery 24 21.txt p6
2017-10-28 16:32:03      Receive: hitQuery 22 21.txt p2
2017-10-28 16:32:03      Send:hitQuery 10 21.txt p3
2017-10-28 16:32:03      Send:hitQuery 8 21.txt p3
2017-10-28 16:32:03      Send:hitQuery 10 21.txt p3
2017-10-28 16:32:03      Receive: hitQuery 21 21.txt p1
2017-10-28 16:32:03      Send:hitQuery 10 21.txt p3
2017-10-28 16:32:03      Receive: hitQuery 22 21.txt p2
2017-10-28 16:32:03      Send:hitQuery 10 21.txt p3
2017-10-28 16:32:03      Receive: hitQuery 22 21.txt p2
2017-10-28 16:32:03      Send:hitQuery 10 21.txt p3
2017-10-28 16:32:03      Receive: hitQuery 24 21.txt p6
2017-10-28 16:32:03      Send:hitQuery 10 21.txt p3
2017-10-28 16:32:03      Receive: hitQuery 24 21.txt p6
2017-10-28 16:32:03      Send:hitQuery 10 21.txt p3
2017-10-28 16:32:03      Receive: query 10 21.txt p6
2017-10-28 16:32:03      File 21.txt is found on p3
2017-10-28 16:32:03      Send:hitQuery 10 21.txt p3
2017-10-28 16:32:04      Receive: hitQuery 20 21.txt p4
2017-10-28 16:32:04      Send:hitQuery 8 21.txt p3
2017-10-28 16:32:04      Receive: hitQuery 21 21.txt p1
2017-10-28 16:32:04      Send:hitQuery 10 21.txt p3
2017-10-28| 16:32:10      Send 21.txtsuccessfully!
```