Manual

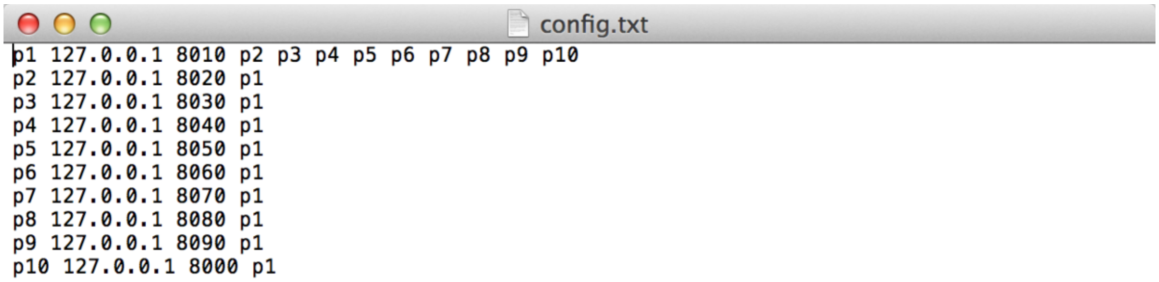
Please make sure that your machine has java runtime environment.

1. Open Terminals and Run the Code

First, you need to open 10 terminals to represent 10 peers.

Under each peer’s folder, there must be a config.txt file, which stores the topology information of peers.

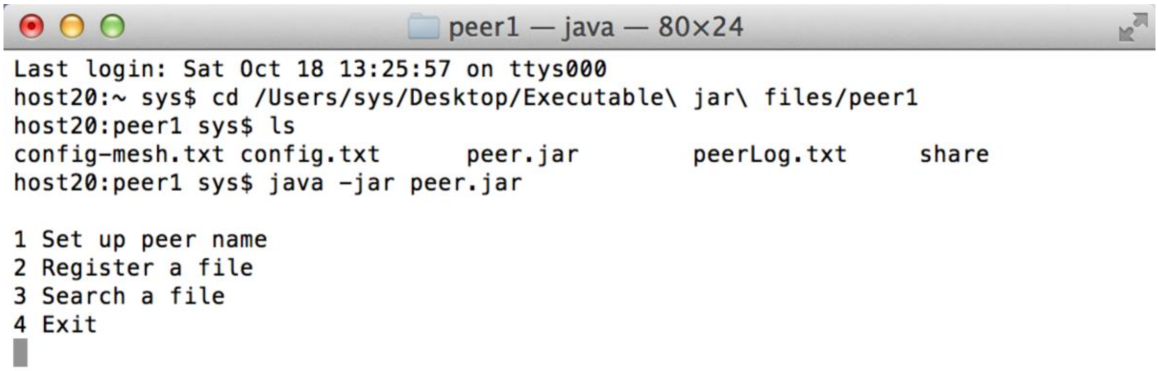
The config.txt file is as following:



```
p1 127.0.0.1 8010 p2 p3 p4 p5 p6 p7 p8 p9 p10
p2 127.0.0.1 8020 p1
p3 127.0.0.1 8030 p1
p4 127.0.0.1 8040 p1
p5 127.0.0.1 8050 p1
p6 127.0.0.1 8060 p1
p7 127.0.0.1 8070 p1
p8 127.0.0.1 8080 p1
p9 127.0.0.1 8090 p1
p10 127.0.0.1 8000 p1
```

For each peer, use the following command to open the peer:

Java –jar peer.jar

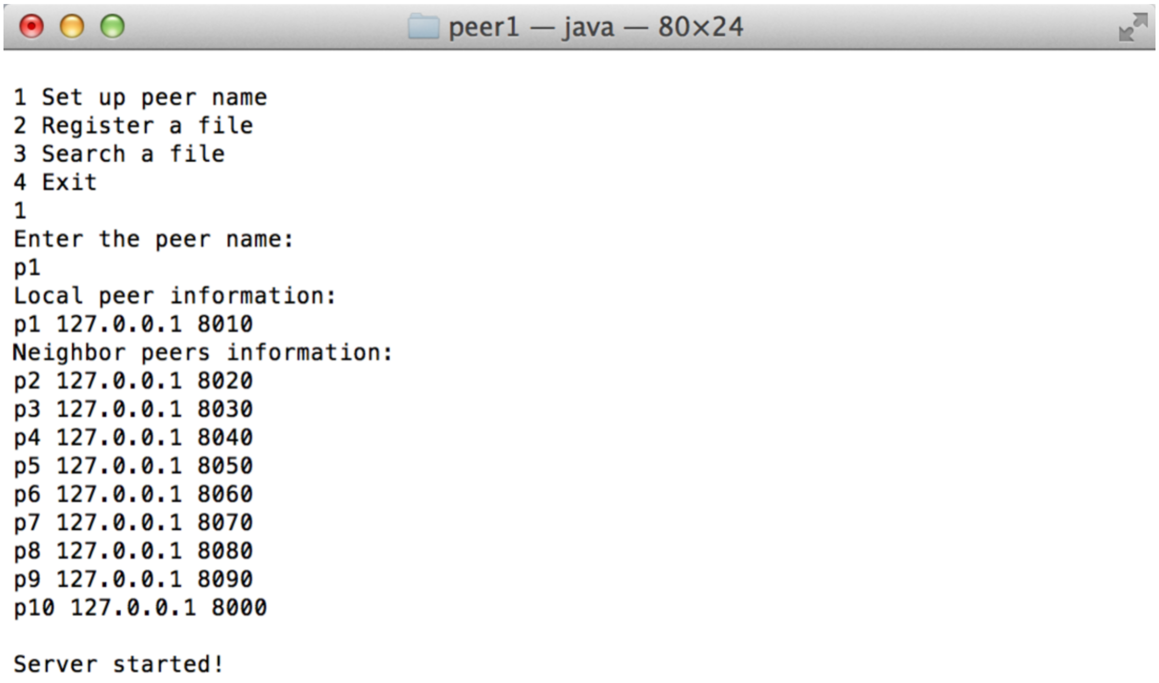


```
Last login: Sat Oct 18 13:25:57 on ttys000
host20:~ sys$ cd /Users/sys/Desktop/Executable\ jar\ files/peer1
host20:peer1 sys$ ls
config-mesh.txt config.txt      peer.jar        peerLog.txt     share
host20:peer1 sys$ java -jar peer.jar

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

2. Set up Peer Name

After opening peers, type ‘1’ and press enter button to set up the peer name, the program will automatically read the config.txt and get the neighbor peers information.

For example, set up the name of peer 1:



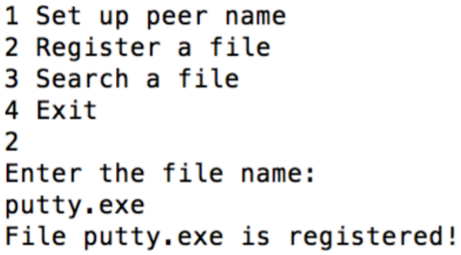
```
1 Set up peer name
2 Register a file
3 Search a file
4 Exit
1
Enter the peer name:
p1
Local peer information:
p1 127.0.0.1 8010
Neighbor peers information:
p2 127.0.0.1 8020
p3 127.0.0.1 8030
p4 127.0.0.1 8040
p5 127.0.0.1 8050
p6 127.0.0.1 8060
p7 127.0.0.1 8070
p8 127.0.0.1 8080
p9 127.0.0.1 8090
p10 127.0.0.1 8000

Server started!
```

The neighbor peers' information is shown on the screen.

Every peer should be set up name first before it can work properly.

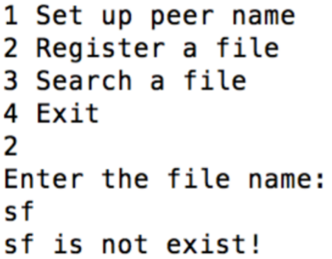
3. Register Files

In the main menu, type '2', press enter and input the file name to register file:



```
1 Set up peer name
2 Register a file
3 Search a file
4 Exit
2
Enter the file name:
putty.exe
File putty.exe is registered!
```

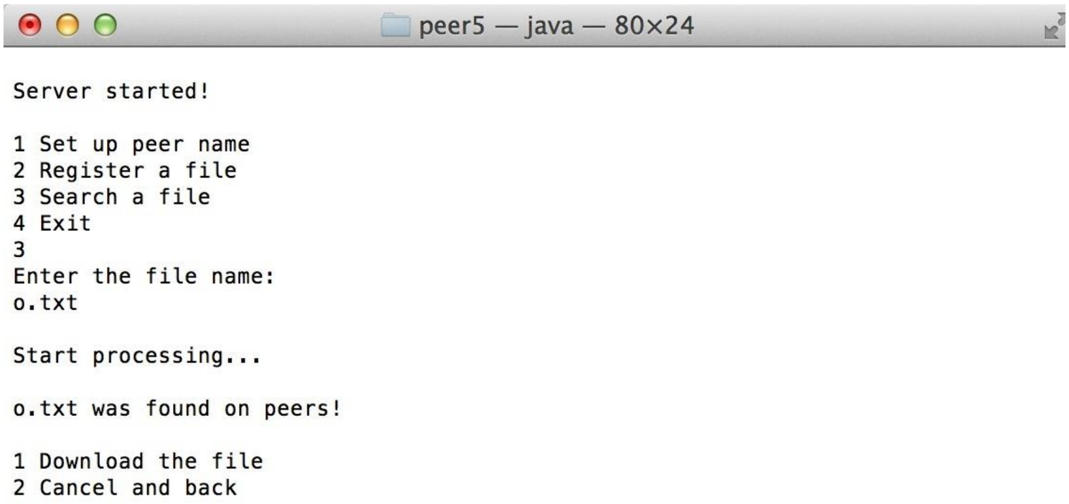
If the file exists, then the file will be successfully registered. However, if the file does not exist, the error report message will be shown on the screen:



```
1 Set up peer name
2 Register a file
3 Search a file
4 Exit
2
Enter the file name:
sf
sf is not exist!
```

4. Search a File

In the main menu, choose command ‘3’ and input the file name to search for a file. For example, use peer 10 to search for the file “putty.exe”, which has been registered in peer 1:



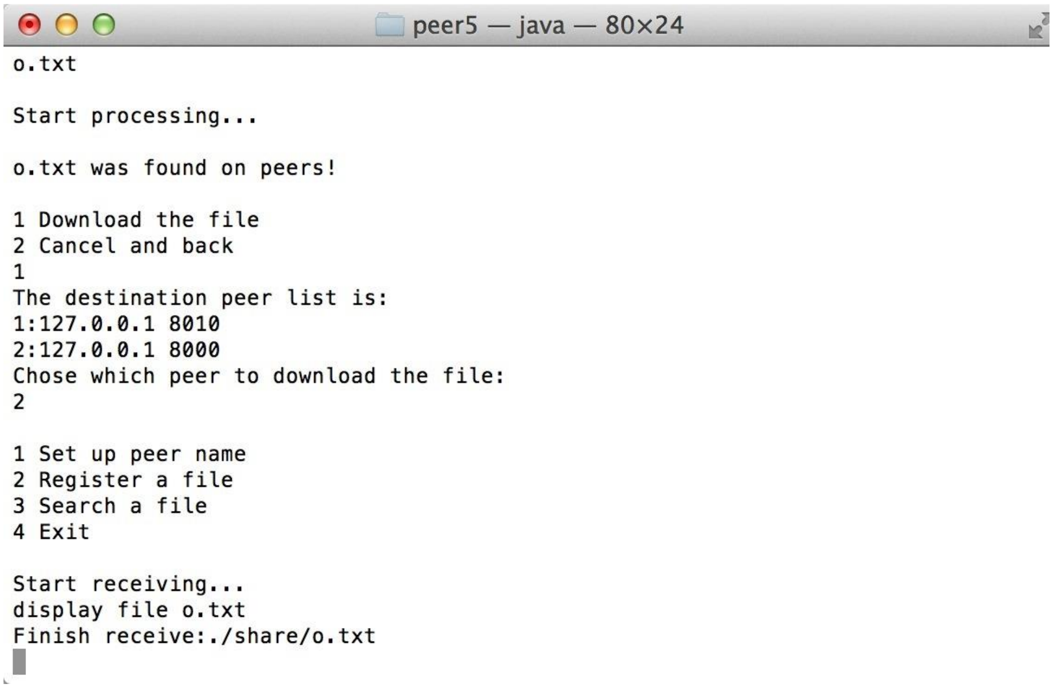
```
Server started!

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

Start processing...

o.txt was found on peers!

1 Download the file
2 Cancel and back
```

5. Download

If the file was found on peers, the screen will display a menu that provides the download command. Then you can choose ‘1’ to download the file, or choose ‘2’ to cancel and back.

When downloading the file, then the program will let you choose which peer to download. Choose one peer and press enter button to process downloading.

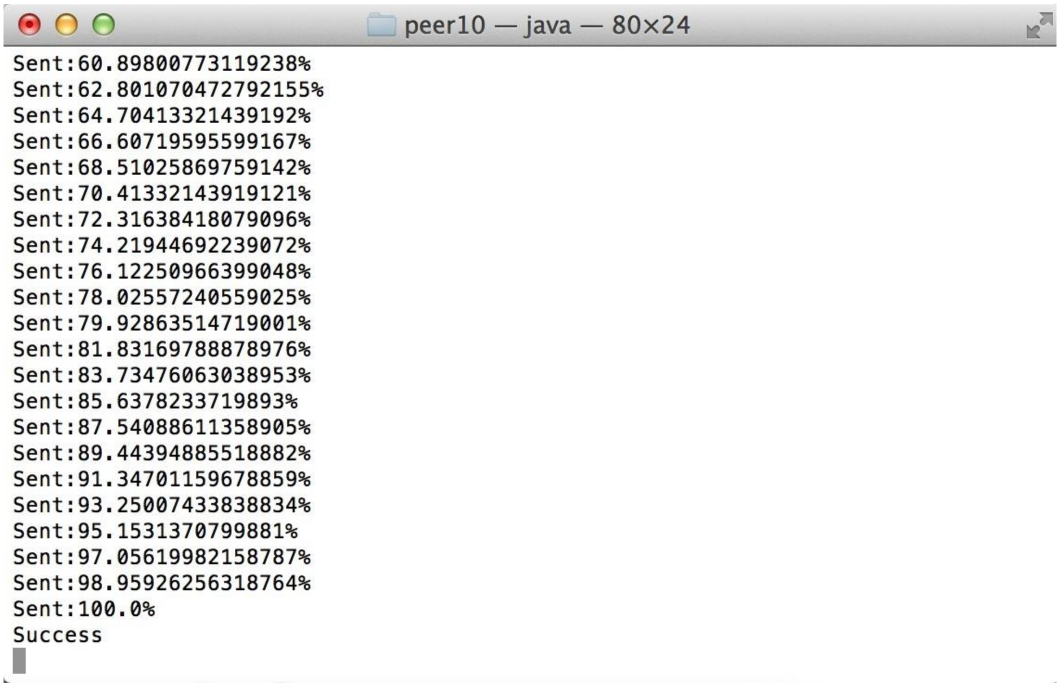
Receive End:



```
o.txt

Start processing...

o.txt was found on peers!

1 Download the file
2 Cancel and back
1
The destination peer list is:
1:127.0.0.1 8010
2:127.0.0.1 8000
Chose which peer to download the file:
2

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

Start receiving...
display file o.txt
Finish receive:./share/o.txt
```

If the screen displays like the above picture, the file has been received.

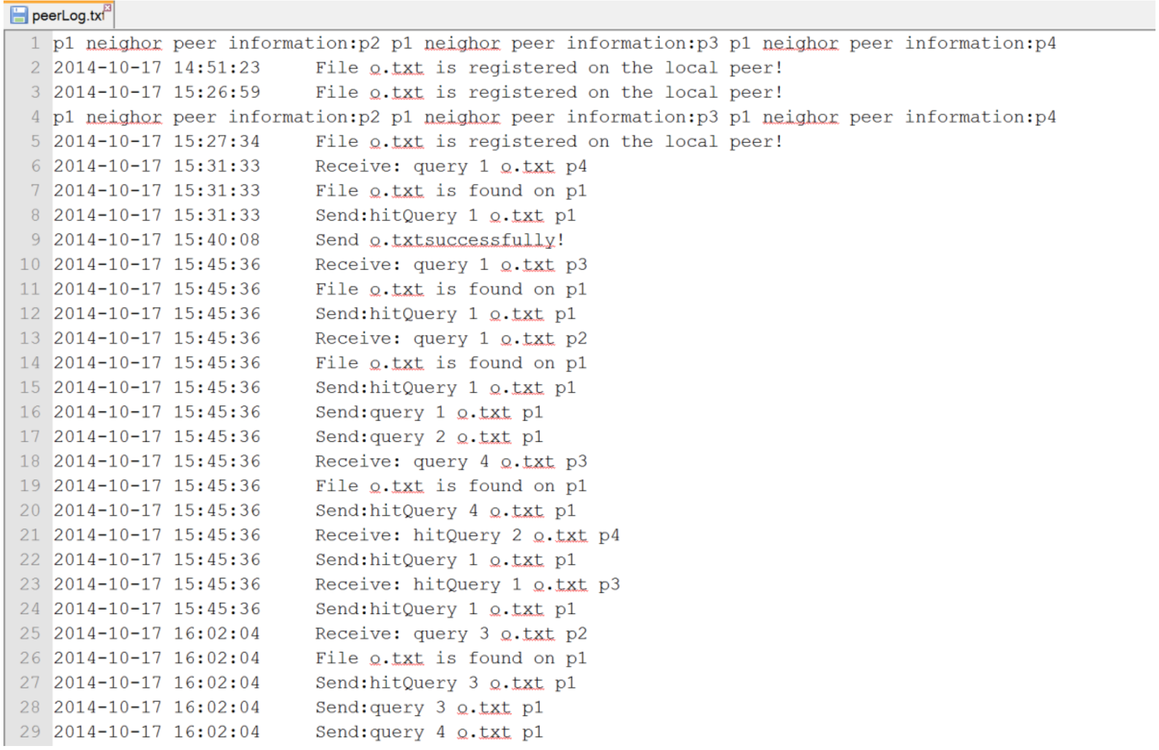
Send End:



```
Sent:60.89800773119238%
Sent:62.801070472792155%
Sent:64.70413321439192%
Sent:66.60719595599167%
Sent:68.51025869759142%
Sent:70.41332143919121%
Sent:72.31638418079096%
Sent:74.21944692239072%
Sent:76.12250966399048%
Sent:78.02557240559025%
Sent:79.92863514719001%
Sent:81.83169788878976%
Sent:83.73476063038953%
Sent:85.6378233719893%
Sent:87.54088611358905%
Sent:89.44394885518882%
Sent:91.34701159678859%
Sent:93.25007433838834%
Sent:95.1531370799881%
Sent:97.05619982158787%
Sent:98.95926256318764%
Sent:100.0%
Success
```

6. Logs

You can see operations from logs which have been recorded in “peerLog.txt” for each peer.



```
p1 neighor peer information:p2 p1 neighor peer information:p3 p1 neighor peer information:p4
2014-10-17 14:51:23     File o.txt is registered on the local peer!
2014-10-17 15:26:59     File o.txt is registered on the local peer!
p1 neighor peer information:p2 p1 neighor peer information:p3 p1 neighor peer information:p4
2014-10-17 15:27:34     File o.txt is registered on the local peer!
2014-10-17 15:31:33     Receive: query 1 o.txt p4
2014-10-17 15:31:33     File o.txt is found on p1
2014-10-17 15:31:33     Send:hitQuery 1 o.txt p1
2014-10-17 15:40:08     Send o.txtsuccessfully!
2014-10-17 15:45:36     Receive: query 1 o.txt p3
2014-10-17 15:45:36     File o.txt is found on p1
2014-10-17 15:45:36     Send:hitQuery 1 o.txt p1
2014-10-17 15:45:36     Receive: query 1 o.txt p2
2014-10-17 15:45:36     File o.txt is found on p1
2014-10-17 15:45:36     Send:hitQuery 1 o.txt p1
2014-10-17 15:45:36     Send:query 1 o.txt p1
2014-10-17 15:45:36     Send:query 2 o.txt p1
2014-10-17 15:45:36     Receive: query 4 o.txt p3
2014-10-17 15:45:36     File o.txt is found on p1
2014-10-17 15:45:36     Send:hitQuery 4 o.txt p1
2014-10-17 15:45:36     Receive: hitQuery 2 o.txt p4
2014-10-17 15:45:36     Send:hitQuery 1 o.txt p1
2014-10-17 15:45:36     Receive: hitQuery 1 o.txt p3
2014-10-17 15:45:36     Send:hitQuery 1 o.txt p1
2014-10-17 16:02:04     Receive: query 3 o.txt p2
2014-10-17 16:02:04     File o.txt is found on p1
2014-10-17 16:02:04     Send:hitQuery 3 o.txt p1
2014-10-17 16:02:04     Send:query 3 o.txt p1
2014-10-17 16:02:04     Send:query 4 o.txt p1
```