

Output

1) Start Server

Command: `java IndexServer`

Output:

```
F:\IIT course\CS 550\P2P\P2P\IndexServer>java IndexServer
start Index Server ...
RMI registry at port 1099.
Bind Succussfully.
Waitting calls for clients...
```

2) Run Peer

Command: `java Peer`

Peer Output:

```
F:\IIT course\CS 550\P2P\P2P\Peer1>java Peer
start Peer ...
RMI registry at port 1001.
Bind Succussfully.
Notice: Peer (rmi://127.0.1.1:1001/Peer) automatically update registry.

-----
The menu options for user:
1.Set loop times (now is: 1)
2.Lookup (and download) files
3.Set a timer to call Lookup (So as to run with other peer simultaneously)
4.Exit
-----
Please select a option number:
```

We also start other 2 Peers. The Peer outputs are similar. So there we ignore their outputs.

IndexServer Output:

```
Peer with IP: "rmi://127.0.1.1:1001/Peer" request registering.
Peer with IP: "rmi://127.0.1.1:1001/Peer" register these files:
a.txt  b.txt  c.txt  d.txt  e.txt
f.txt  g.txt  h.txt  i.txt  j.txt
Peer with IP: "rmi://127.0.1.1:1001/Peer" registered successfully.

Peer with IP: "rmi://127.0.1.2:1002/Peer" request registering.
Peer with IP: "rmi://127.0.1.2:1002/Peer" register these files:
e.txt  f.txt  g.txt  h.txt  i.txt
j.txt  k.txt  l.txt  m.txt  n.txt
Peer with IP: "rmi://127.0.1.2:1002/Peer" registered successfully.

Peer with IP: "rmi://127.0.1.3:1003/Peer" request registering.
Peer with IP: "rmi://127.0.1.3:1003/Peer" register these files:
i.txt  j.txt  k.txt  l.txt  m.txt
n.txt  o.txt  p.txt  q.txt  r.txt
Peer with IP: "rmi://127.0.1.3:1003/Peer" registered successfully.
```

We can see all the files updated.

3) Lookup and download from one Peer.

Commands:

Option: 2

File name: `j.txt`

Choose Server: 2

Peer Output:

```
Please select a option number:
2
Please enter the file name:
k.txt
The avg time for 1 sequence loopup is: 3.757 ms

There are all address of the peers keeping file k.txt:
1. rmi://127.0.1.3:1003/Peer
2. rmi://127.0.1.2:1002/Peer

You can choose one to download the file: <Or you can enter 0 to cancel>
2
The file k.txt has been saved in the path: files/ Successfully.
The download speed is: 0.169 MB/s
```

Because Peer 1 add new file k.txt, the registry will automatically update.

The IndexServer Output will be:

```
A peer request looking up file k.txt .
Searching for file k.txt complete.
Peer with IP: "rmi://127.0.1.1:1001/Peer" request registering.
Peer with IP: "rmi://127.0.1.1:1001/Peer" register these files:
a.txt  b.txt  c.txt  d.txt  e.txt
f.txt  g.txt  h.txt  i.txt  j.txt
k.txt
Peer with IP: "rmi://127.0.1.1:1001/Peer" registered successfully.
```

4) Delete a file from Peer directory.

From example, we delete k.txt from Peer 3(port 1003), the IndexServer output:

```
Peer with IP: "rmi://127.0.1.3:1003/Peer" request registering.
Peer with IP: "rmi://127.0.1.3:1003/Peer" register these files:
i.txt  j.txt  l.txt  m.txt  n.txt
o.txt  p.txt  q.txt  r.txt
Peer with IP: "rmi://127.0.1.3:1003/Peer" registered successfully.
```

We lookup k.txt:

```
Please select a option number:
2
Please enter the file name:
k.txt
The avg time for 1 sequence loopup is: 6.627 ms

There are all address of the peers keeping file k.txt:
1. rmi://127.0.1.1:1001/Peer
2. rmi://127.0.1.2:1002/Peer

You can choose one to download the file: <Or you can enter 0 to cancel>
```

Only Peer1(port 1001) and Peer2(port 1002) keep that file.

5) Set loop time

You can choose **option 1** in menu and set the loop time:

```
Please select a option number:
1
Please enter the loop time number:
```

Enter the times of loop you want, for example, 1000:

```
Please enter the loop time number:
1000
The loop time has been set as: 1000.

-----
The menu options for user:
1.Set loop times <now is: 1000>
2.Lookup <and download> files
3.Set a timer to call Lookup <So as to run with other peer simultaneously>
4.Exit
-----
```

You can see in the option 1 of menu, the default loop time is 1000.

If you run lookup or download again, you can see it give average time of the request.

```
Please select a option number:
2
Please enter the file name:
m.txt
The avg time for 1000 sequence lookup is: 1.266 ms
```

6) Set a timer

Select option: 3

Input the timer in format, for example: 8 30 00.

Input a file name, for example: b.txt

```
Please select a option number:
3
Please input the timer in format HH MM SS <ex. 11 30 00 means 11:30:00>
8 30 00
Please enter the searching file name:
b.txt
You create a lookup task for b.txt
It will start at Mon Jan 30 08:30:00 CST 2017
```

It will show that the task will start at today's 08:30:00 (the millisecond is 000).

When reaching the time, the task will automatically start:

```
Timer for execting lookup b.txt start!
The avg time for 1000 sequence lookup is: 1.958 ms
```

7) Exit Peer

For example, we exit Peer 2 which keep k.txt.

```
Please select a option number:
4
Register Monitor: sleep interrupted
Peer <rmi://127.0.1.1:1001/Peer> is exiting...
```

The IndexServer output:

```
Peer with IP: "rmi://127.0.1.2:1002/Peer" request registering.
Peer with IP: "rmi://127.0.1.2:1002/Peer" exit.
```

We lookup k.txt:

```
Please select a option number:
2
Please enter the file name:
k.txt
The avg time for 1 sequence loopup is: 5.350 ms

There are all address of the peers keeping file k.txt:
1. rmi://127.0.1.1:1001/Peer

You can choose one to download the file: <Or you can enter 0 to cancel>
```

Only Peer1(port 1001) keep this file.