

4. Manual

1. Running Environment

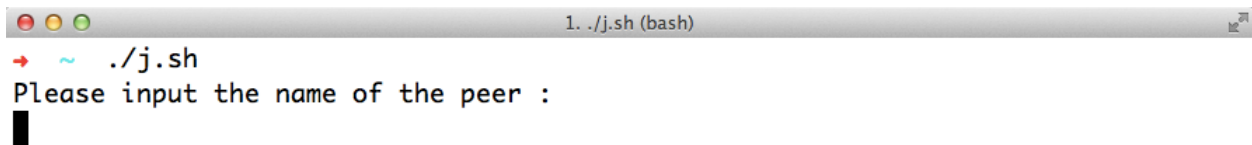
Platform: Java SE-1.6

OS: Window 7, Mac OS X 10.8.5 or Ubuntu 12.04 LTS

2. How to run

Executable jar files: P2PSystem.jar.

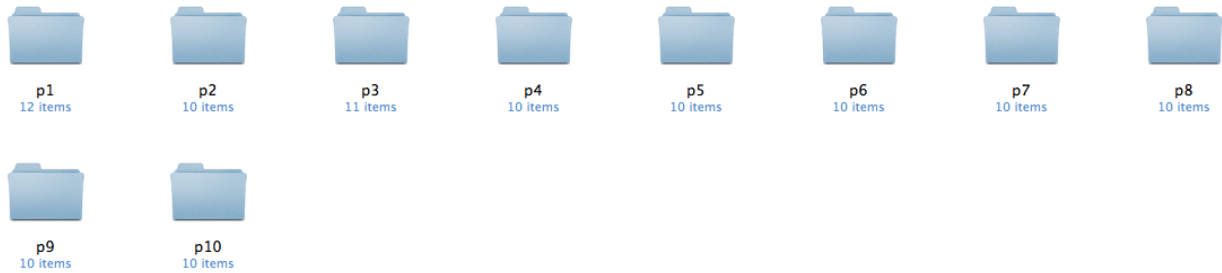
We use a shell script j.sh, thus, when we launch a terminal, we just input ./j.sh.

A screenshot of a terminal window. The title bar at the top reads "1. ./j.sh (bash)". The terminal content shows a red prompt character followed by a tilde symbol and the command "./j.sh". Below this, the text "Please input the name of the peer :" is displayed, followed by a black cursor line.

```
1. ./j.sh (bash)
➔ ~ ./j.sh
Please input the name of the peer :
█
```

3. Input client name:

All the client names are as follows:



When a jar runs, we should first input the client name.

```
1. ./j.sh (bash)
➔ ~ ./j.sh
Please input the name of the peer :
p1
Please input operation index:
1: Create      2: Query      3: Obtain      4: Quit
█
```

As for convenience, we demo our operation with a simple configuration file.

```
configure
p1    127.0.0.1    8101    p2
p2    127.0.0.1    8102    p1    p3
p3    127.0.0.1    8103    p2
```

It is like a chain with 3 nodes. p1 - p2 - p3. Thus, we need open three terminals.

```
1. ./j.sh (java)
./j.sh (bash)  ./j.sh (java)  ./j.sh (java)
➔ ~ ./j.sh
Please input the name of the peer :
p3
Please input operation index:
1: Create      2: Query      3: Obtain      4: Quit
█
```

4. Connections initialization.

After all clients input their names, we should input command 1 in all terminals to create the network.

```
➔ ~ ./j.sh
Please input the name of the peer :
p1
Please input operation index:
1: Create      2: Query      3: Obtain      4: Quit
1
```

5. Query

After the network established, a client can choose query a file or obtain a file. Logically, a client queries a file first.

User is required to input the file name and ttl (we design this because this is a static network).

Please input the file name you are looking for:

tianyang

Please input the TTL:

2

Please input operation index:

1: Create 2: Query 3: Obtain 4: Quit

p2 has the file.

p3 has the file.

Clients who have this file are listed.

Other clients prints whether they have this file.

➔ ~ ./j.sh

Please input the name of the peer :

p2

Please input operation index:

1: Create 2: Query 3: Obtain 4: Quit

1

Please input operation index:

1: Create 2: Query 3: Obtain 4: Quit

I have tianyang

█

6. Obtain.

The user input the filename and client name, then download.

Please input the file name you want to download :

tianyang

Please select the peer name you want to download from :

p2

socket index is 0

P2 displays which file is going to be downloaded.

```
➔ ~ ./j.sh
```

```
Please input the name of the peer :
```

```
p2
```

```
Please input operation index:
```

```
1: Create      2: Query      3: Obtain      4: Quit
```

```
1
```

```
Please input operation index:
```

```
1: Create      2: Query      3: Obtain      4: Quit
```

```
I have tianyang
```

```
Download file is tianyang
```

```
█
```

```
P1 displays
```

```
Please input the file name you want to download :
```

```
tianyang
```

```
Please select the peer name you want to download from :
```

```
p2
```

```
Download file tianyang successfully
```