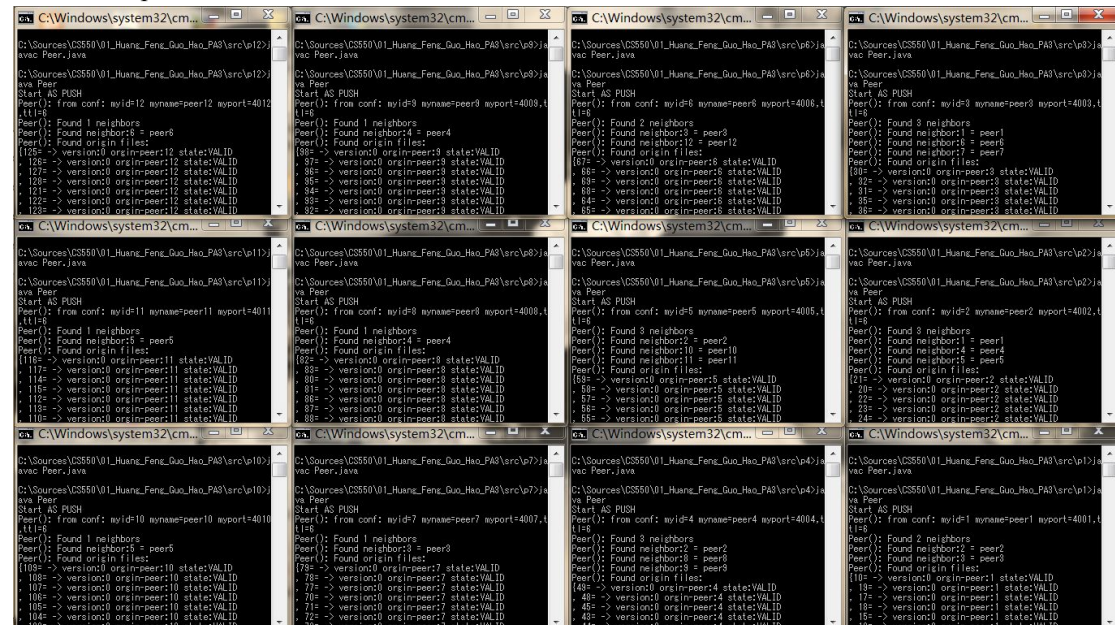


Verification Test

In this file, we will demonstrate why our program works correctly, in other words, how does a file get invalidated. Details about topology and file transferring are talked in previous projects, in this one, we focus on file consistency. How the testing files are located is stated in performance.section1.

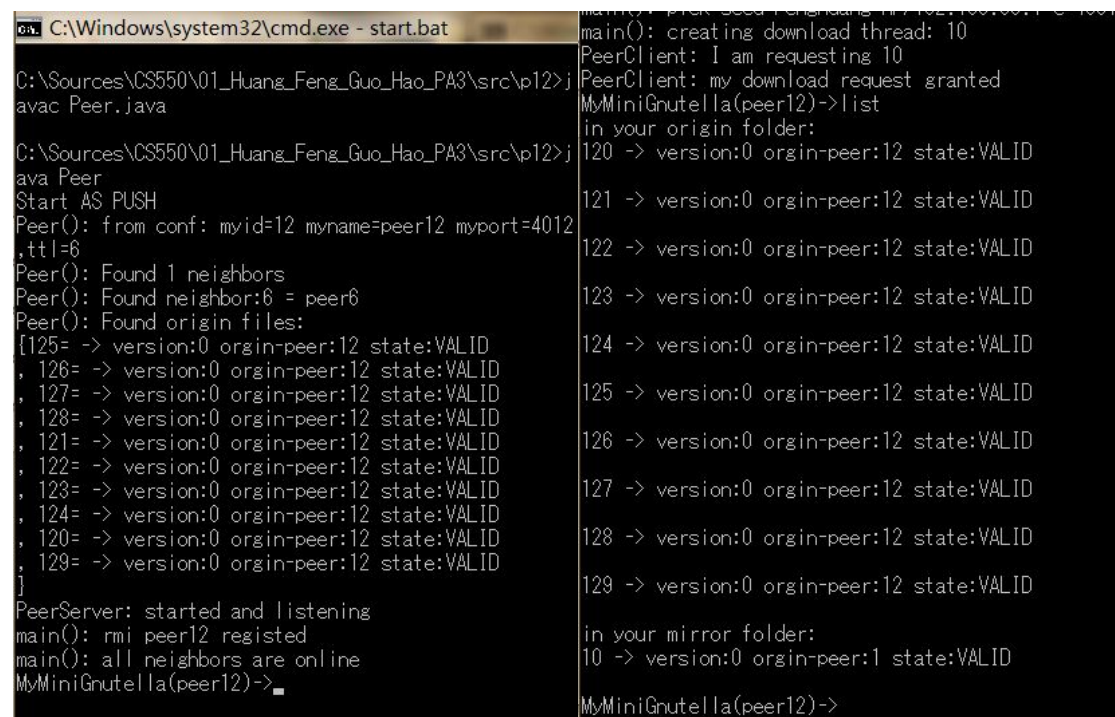
PULL:

After startup:

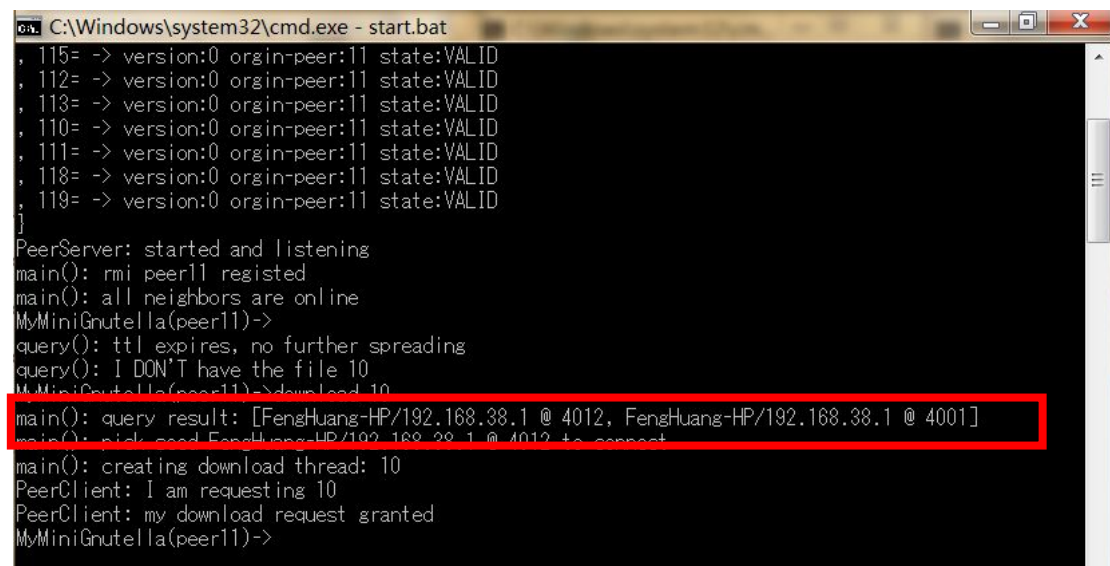


Zoom in to peer 12:

after issued : download 10



At this time we can see peer12 has file 10, in mirror folder, with version and owner. Now we try to download this 10 from peer 11 :



```
C:\Windows\system32\cmd.exe - start.bat
, 115= -> version:0 orgin-peer:11 state:VALID
, 112= -> version:0 orgin-peer:11 state:VALID
, 113= -> version:0 orgin-peer:11 state:VALID
, 110= -> version:0 orgin-peer:11 state:VALID
, 111= -> version:0 orgin-peer:11 state:VALID
, 118= -> version:0 orgin-peer:11 state:VALID
, 119= -> version:0 orgin-peer:11 state:VALID
]
PeerServer: started and listening
main(): rmi peer11 registered
main(): all neighbors are online
MyMiniGnutella(peer11)->
query(): ttl expires, no further spreading
query(): I DON'T have the file 10
MyMiniGnutella(peer11)->download 10
main(): query result: [FengHuang-HP/192.168.38.1 @ 4012, FengHuang-HP/192.168.38.1 @ 4001]
main(): pick peer FengHuang-HP/192.168.38.1 @ 4012 to connect
main(): creating download thread: 10
PeerClient: I am requesting 10
PeerClient: my download request granted
MyMiniGnutella(peer11)->
```

From the red box we see peer11 got to know both peer 1 and peer 12 has the copy. Now we type “modify 10” in peer 1, we will see a lot of invalidation message transferring, and finally in peer 1:



```
C:\Windows\system32\cmd.exe - start.bat
MyMiniGnutella(peer1)->list
in your origin folder:
10 -> version:1 orgin-peer:1 state:VALID
11 -> version:0 orgin-peer:1 state:VALID
12 -> version:0 orgin-peer:1 state:VALID
13 -> version:0 orgin-peer:1 state:VALID
14 -> version:0 orgin-peer:1 state:VALID
15 -> version:0 orgin-peer:1 state:VALID
16 -> version:0 orgin-peer:1 state:VALID
17 -> version:0 orgin-peer:1 state:VALID
18 -> version:0 orgin-peer:1 state:VALID
19 -> version:0 orgin-peer:1 state:VALID
in your mirror folder:
MyMiniGnutella(peer1)->
```

We see that 10's version became 1, not 0. And since invalidation message spread to peer 12 and 11, they are expected to invalidate the copy:

| | |
|--|--|
| <pre> MyMiniGnutella(peer11)->list in your origin folder: 110 -> version:0 orgin-peer:11 state:VALID 111 -> version:0 orgin-peer:11 state:VALID 112 -> version:0 orgin-peer:11 state:VALID 113 -> version:0 orgin-peer:11 state:VALID 114 -> version:0 orgin-peer:11 state:VALID 115 -> version:0 orgin-peer:11 state:VALID 116 -> version:0 orgin-peer:11 state:VALID 117 -> version:0 orgin-peer:11 state:VALID 118 -> version:0 orgin-peer:11 state:VALID 119 -> version:0 orgin-peer:11 state:VALID in your mirror folder: 10 -> version:0 orgin-peer:1 state:INVALID MyMiniGnutella(peer11)-> </pre> | <pre> MyMiniGnutella(peer12)->list in your origin folder: 120 -> version:0 orgin-peer:12 state:VALID 121 -> version:0 orgin-peer:12 state:VALID 122 -> version:0 orgin-peer:12 state:VALID 123 -> version:0 orgin-peer:12 state:VALID 124 -> version:0 orgin-peer:12 state:VALID 125 -> version:0 orgin-peer:12 state:VALID 126 -> version:0 orgin-peer:12 state:VALID 127 -> version:0 orgin-peer:12 state:VALID 128 -> version:0 orgin-peer:12 state:VALID 129 -> version:0 orgin-peer:12 state:VALID in your mirror folder: 11 -> version:0 orgin-peer:1 state:INVALID MyMiniGnutella(peer12)-> </pre> |
|--|--|

At last, if we issue a “download 10” in peer 10, it should know peer 11 and peer 12’s invalidated copies:

```

MyMiniGnutella(peer10)->download 10
main(): query result: [FengHuang-HP/192.168.38.1 @ 4001]
main(): pick seed FengHuang-HP/192.168.38.1 @ 4001 to connect
main(): creating download thread: 10
PeerClient: I am requesting 10
PeerClient: my download request granted
MyMiniGnutella(peer10)->

```

PUSH:

In the push mode, each file has a thread to poll the owner about version of specific file, see the following:

```

C:\Windows\system32\cmd.exe - start.bat
, 127= -> version:0 orgin-peer:12 state:VALID
, 128= -> version:0 orgin-peer:12 state:VALID
, 121= -> version:0 orgin-peer:12 state:VALID
, 122= -> version:0 orgin-peer:12 state:VALID
, 123= -> version:0 orgin-peer:12 state:VALID
, 124= -> version:0 orgin-peer:12 state:VALID
, 120= -> version:0 orgin-peer:12 state:VALID
, 129= -> version:0 orgin-peer:12 state:VALID
}
PeerServer: started and listening
main(): rmi peer12 registered
main(): all neighbors are online
MyMiniGnutella(peer12)->download 10
main(): query result: [FengHuang-HP/192.168.38.1 @ 4001]
main(): pick seed FengHuang-HP/192.168.38.1 @ 4001 to connect
main(): creating download thread: 10
MyMiniGnutella(peer12)->PeerClient: I am requesting 10
PeerClient: my download request granted
MyMiniGnutella(peer12)->
Poll(): check file 10

```

After peer 12 download 10, peer 9 also does it:

```
C:\Windows\system32\cmd.exe - start.bat
main(): neighbor peer4 can not be reached now, wait 5s and retry
main(): neighbor peer4 can not be reached now, wait 5s and retry
main(): neighbor peer4 can not be reached now, wait 5s and retry
main(): neighbor peer4 can not be reached now, wait 5s and retry
main(): all neighbors are online
MyMiniGnutella(peer9)->
query(): ttl expires, no further spreading
query(): I DON'T have the file 10
MyMiniGnutella(peer9)->download 10
main(): query result: [FengHuang-HP/192.168.38.1 @ 4012, FengHuang-HP/192.168.38.1 @ 4001]
main(): pick seed FengHuang-HP/192.168.38.1 @ 4012 to connect
main(): creating download thread: 10
MyMiniGnutella(peer9)->PeerClient: I am requesting 10
PeerClient: my download request granted
MyMiniGnutella(peer9)->
Poll(): check file 10

Poll(): check file 10

Poll(): check file 10

Poll(): check file 10
```

After peer 1 modified its file 10, its version changed:

```
C:\Windows\system32\cmd.exe - start.bat
ServerThread: thread terminate
MyMiniGnutella(peer1)->list
in your origin folder:
10 -> version:1 origin-peer:1 state:VALID
11 -> version:0 origin-peer:1 state:VALID
12 -> version:0 origin-peer:1 state:VALID
13 -> version:0 origin-peer:1 state:VALID
14 -> version:0 origin-peer:1 state:VALID
15 -> version:0 origin-peer:1 state:VALID
16 -> version:0 origin-peer:1 state:VALID
17 -> version:0 origin-peer:1 state:VALID
18 -> version:0 origin-peer:1 state:VALID
19 -> version:0 origin-peer:1 state:VALID

in your mirror folder:
MyMiniGnutella(peer1)->
PeerServerThread: new thread started
ServerThread: report version number of 10 to other: 1
ServerThread: thread terminate
MyMiniGnutella(peer1)->
```

And after the polling interval, peer 9 polled and found its 10 is out-dated:

```
Poll(): check file 10
Poll(): invalidate my file: 10
list
in your origin folder:
90 -> version:0 orgin-peer:9 state:VALID
91 -> version:0 orgin-peer:9 state:VALID
92 -> version:0 orgin-peer:9 state:VALID
93 -> version:0 orgin-peer:9 state:VALID
94 -> version:0 orgin-peer:9 state:VALID
95 -> version:0 orgin-peer:9 state:VALID
96 -> version:0 orgin-peer:9 state:VALID
97 -> version:0 orgin-peer:9 state:VALID
98 -> version:0 orgin-peer:9 state:VALID
99 -> version:0 orgin-peer:9 state:VALID
in your mirror folder:
10 -> version:0 orgin-peer:1 state:INVALID
MyMiniGnutella(peer9)->
```