2nd approach execution manual

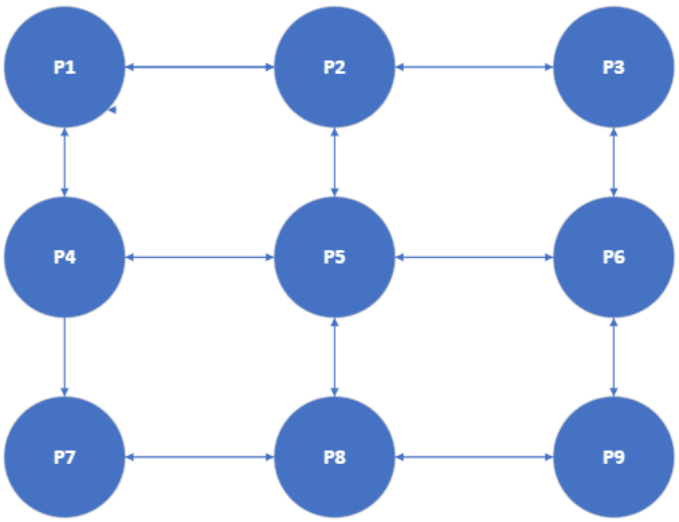
I. Introduction :

In this document we'll present the execution of the Gnutella Peer-2-Peer application (which is our second approach for this project work [using rmi registry]).

We will run the program with different topologies:

- Mesh topology with 9 peers : 3×3
- Star topology
- Two different machines connected to the same network
  - ⇨ To change the topology we simply edit the config file

II. Execution process for the first topology : Mesh 3×3



In the config file:

src > config.properties

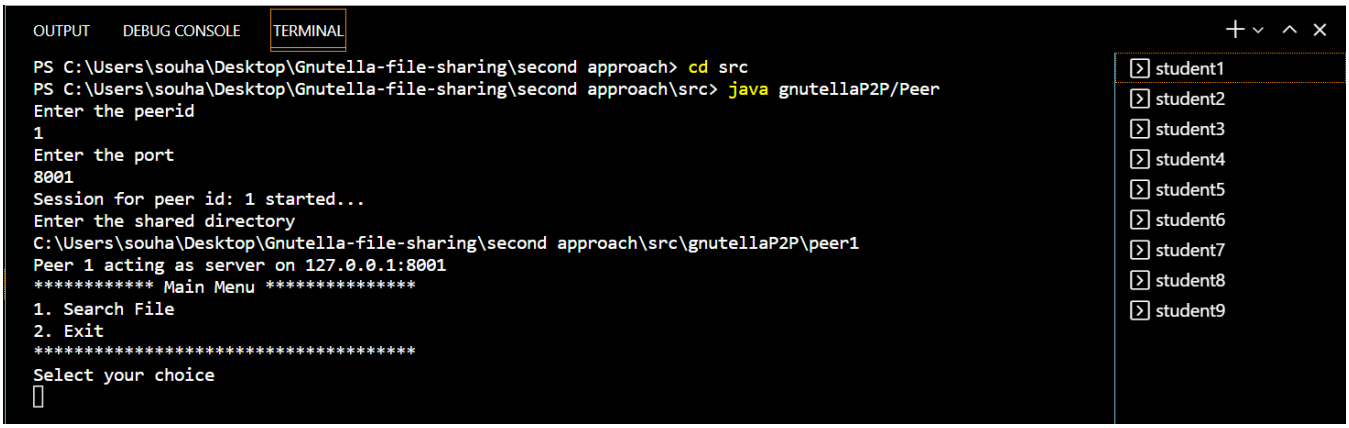
```
# Peer details
peerid.1.ip=127.0.0.1
peerid.1.port=8001
peerid.2.ip=127.0.0.1
peerid.2.port=8002
peerid.3.ip=127.0.0.1
peerid.3.port=8003
peerid.4.ip=127.0.0.1
peerid.4.port=8004
peerid.5.ip=127.0.0.1
peerid.5.port=8005
peerid.6.ip=127.0.0.1
peerid.6.port=8006
peerid.7.ip=127.0.0.1
peerid.7.port=8007
peerid.8.ip=127.0.0.1
peerid.8.port=8008
peerid.9.ip=127.0.0.1
peerid.9.port=8009
```

```
# Neighbor details
#3X3 Mesh
peerid.1.neighbors=peerid.2,peerid.4
peerid.2.neighbors=peerid.1,peerid.3,peerid.5
peerid.3.neighbors=peerid.2,peerid.6
peerid.4.neighbors=peerid.1,peerid.5,peerid.7
peerid.5.neighbors=peerid.2,peerid.4,peerid.6,peerid.8
peerid.6.neighbors=peerid.3,peerid.5,peerid.9
peerid.7.neighbors=peerid.4,peerid.8
peerid.8.neighbors=peerid.5,peerid.7,peerid.9
peerid.9.neighbors=peerid.6,peerid.8
```

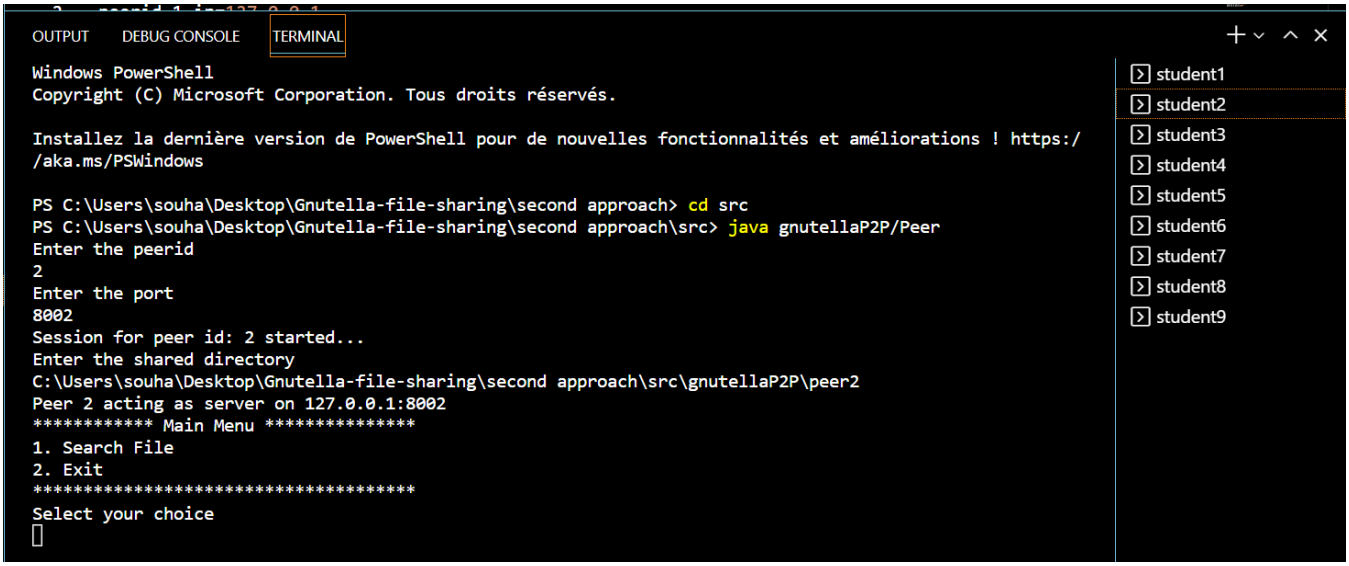
First, since we have 9 peers, we'll open 9 command line interfaces "CMD" from the path of the source code. (we worked with VS code so we opened 9 terminals)

Then we run the Peer.java class using the command java Peer. For all 9 peers, we enter the peer ID, Port, and its equivalent shared directory. (That we already created so that every peer (student) has its own files)

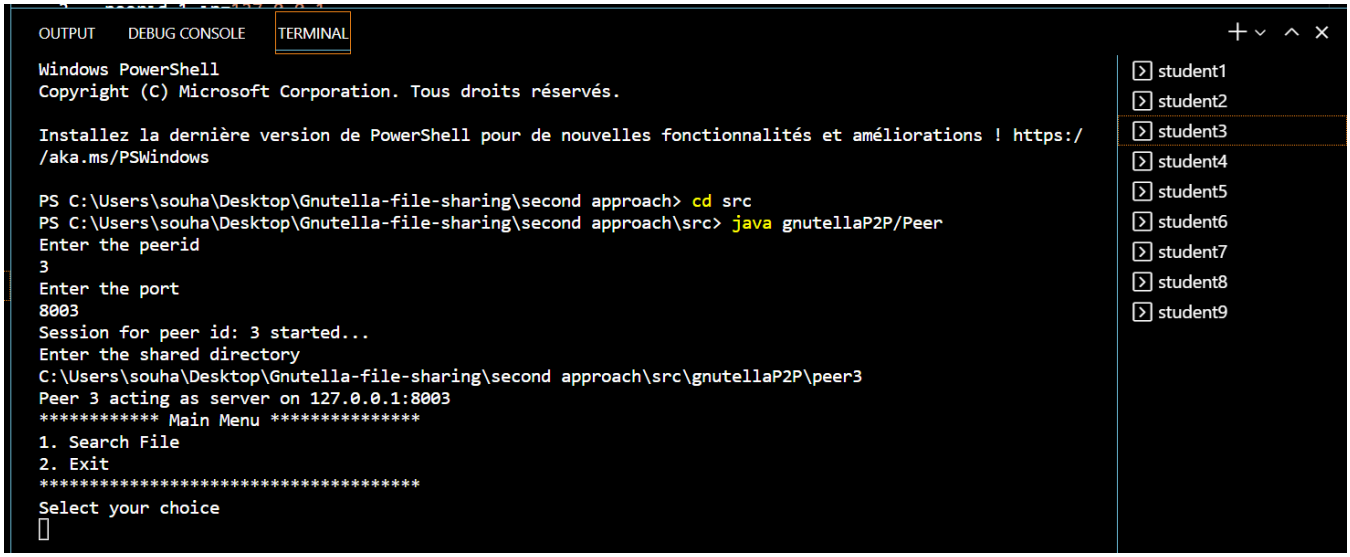
Student 1:



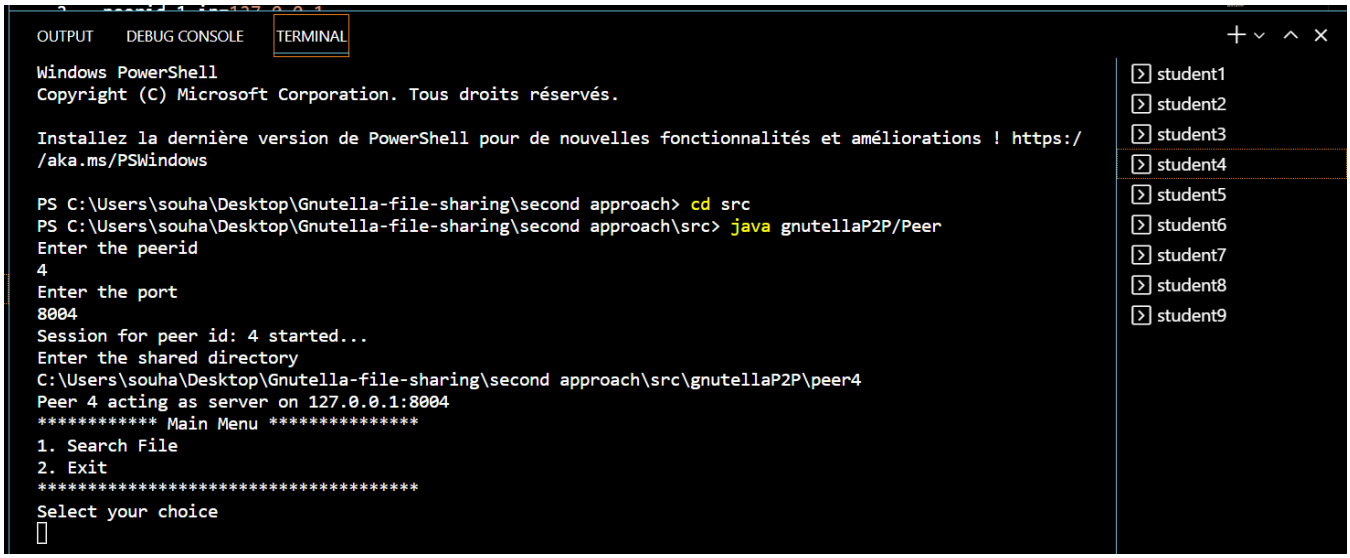
Student 2:



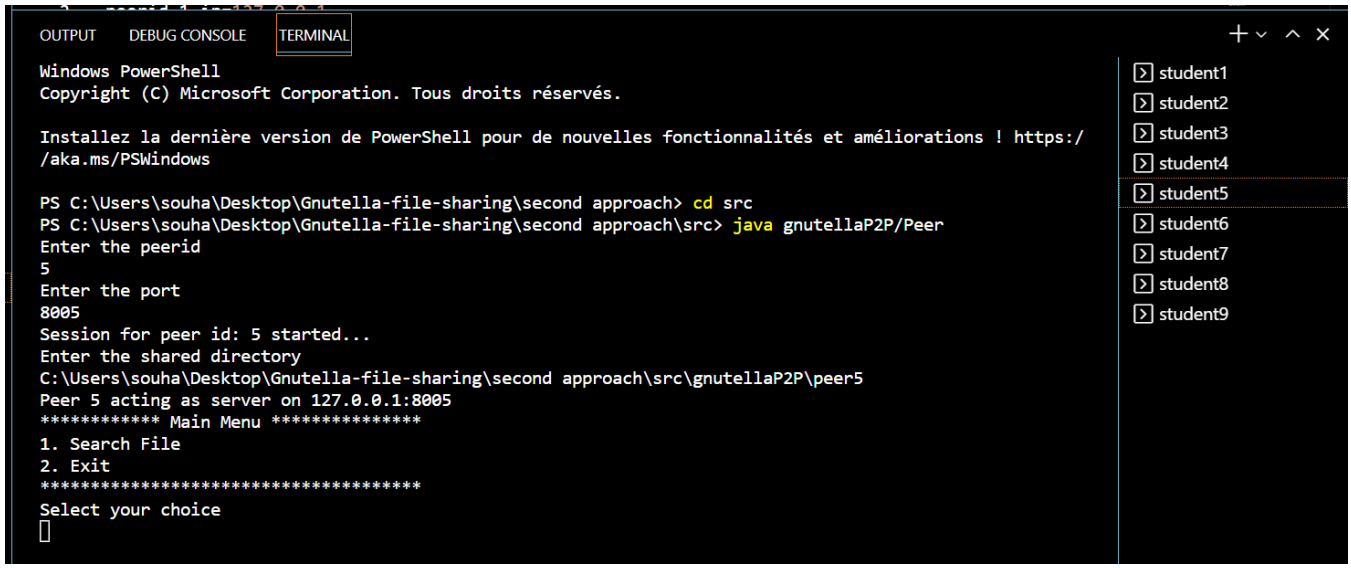
Student 3:



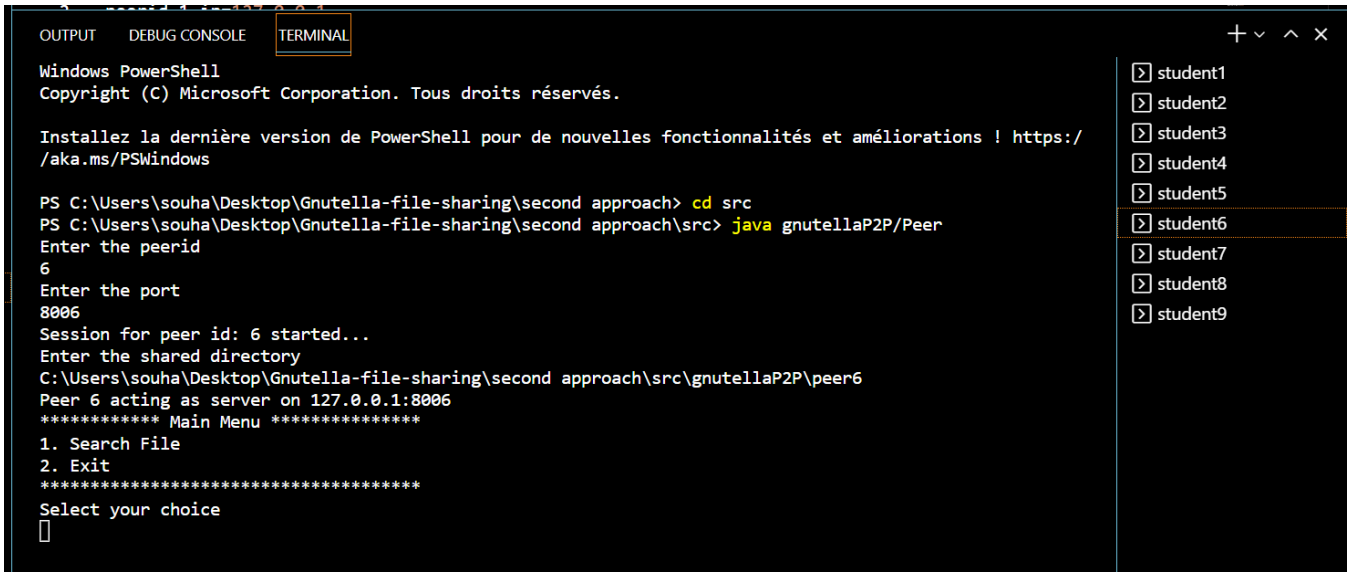
Student 4:



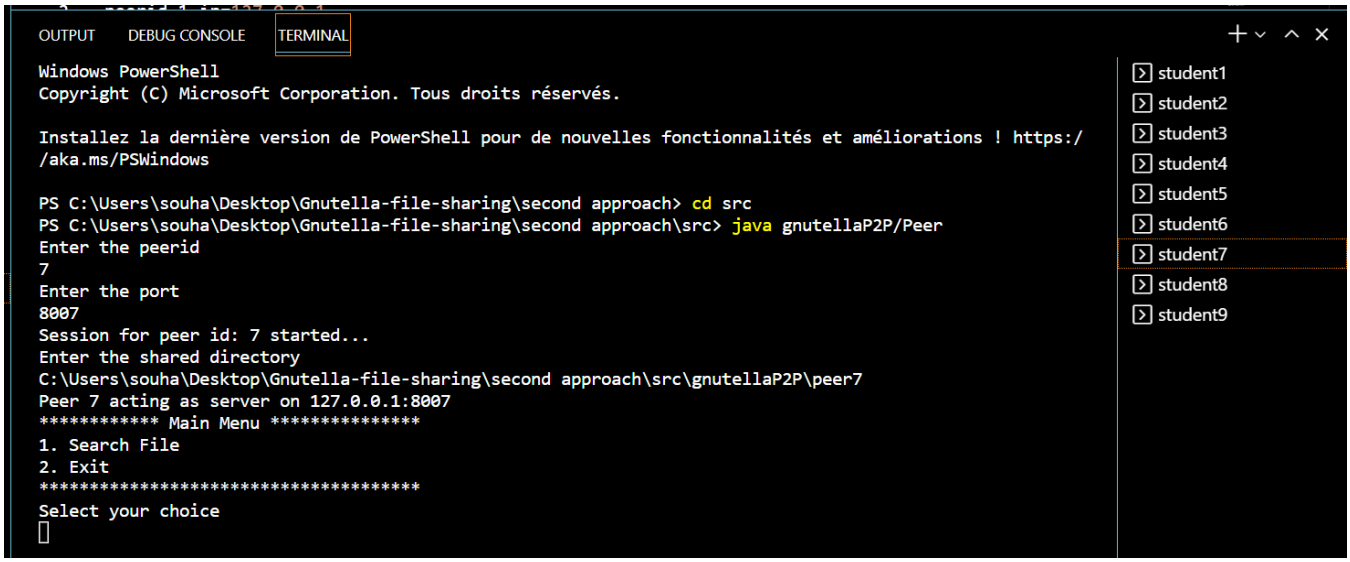
Student 5:



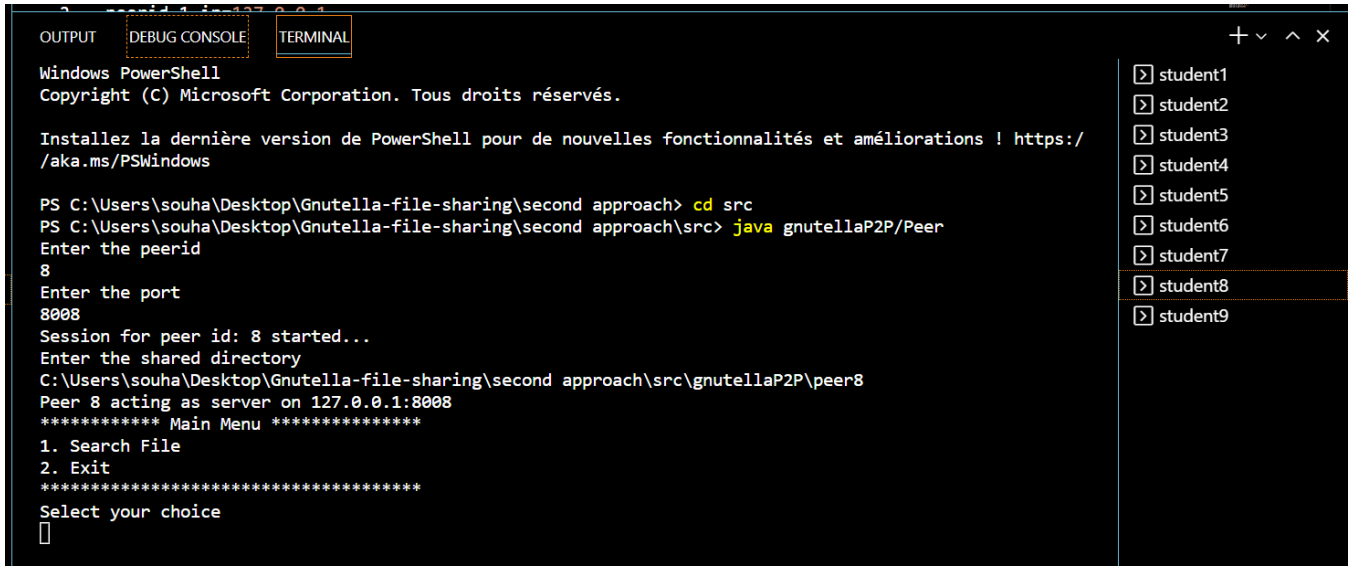
Student 6:



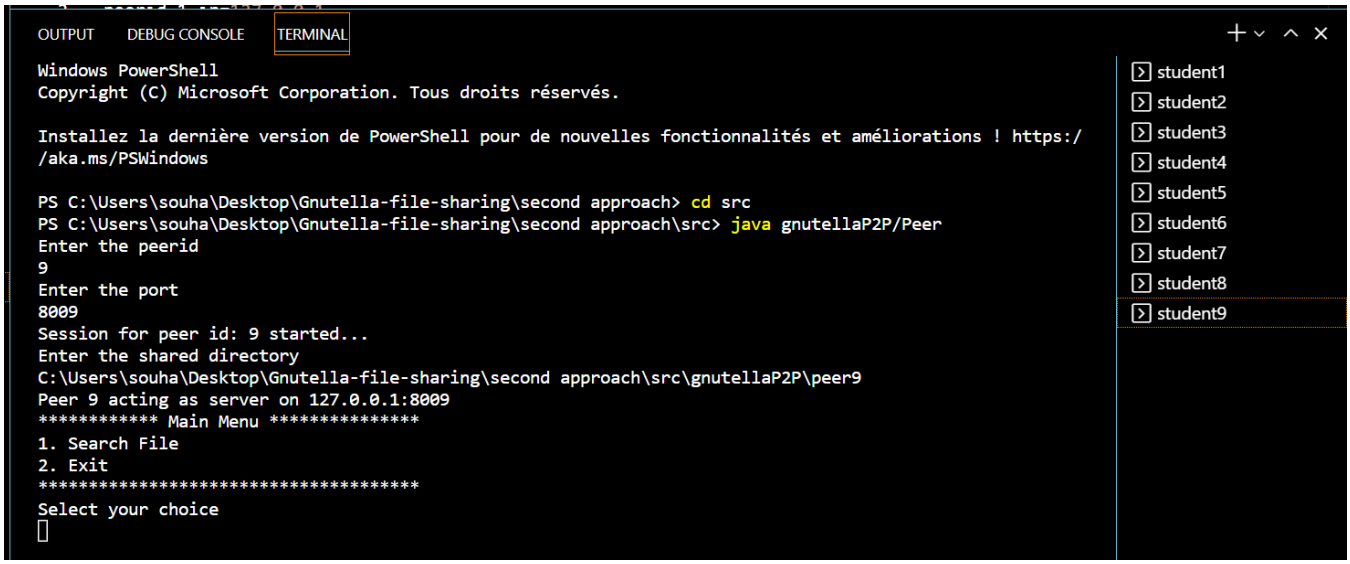
Student 7:



Student 8:

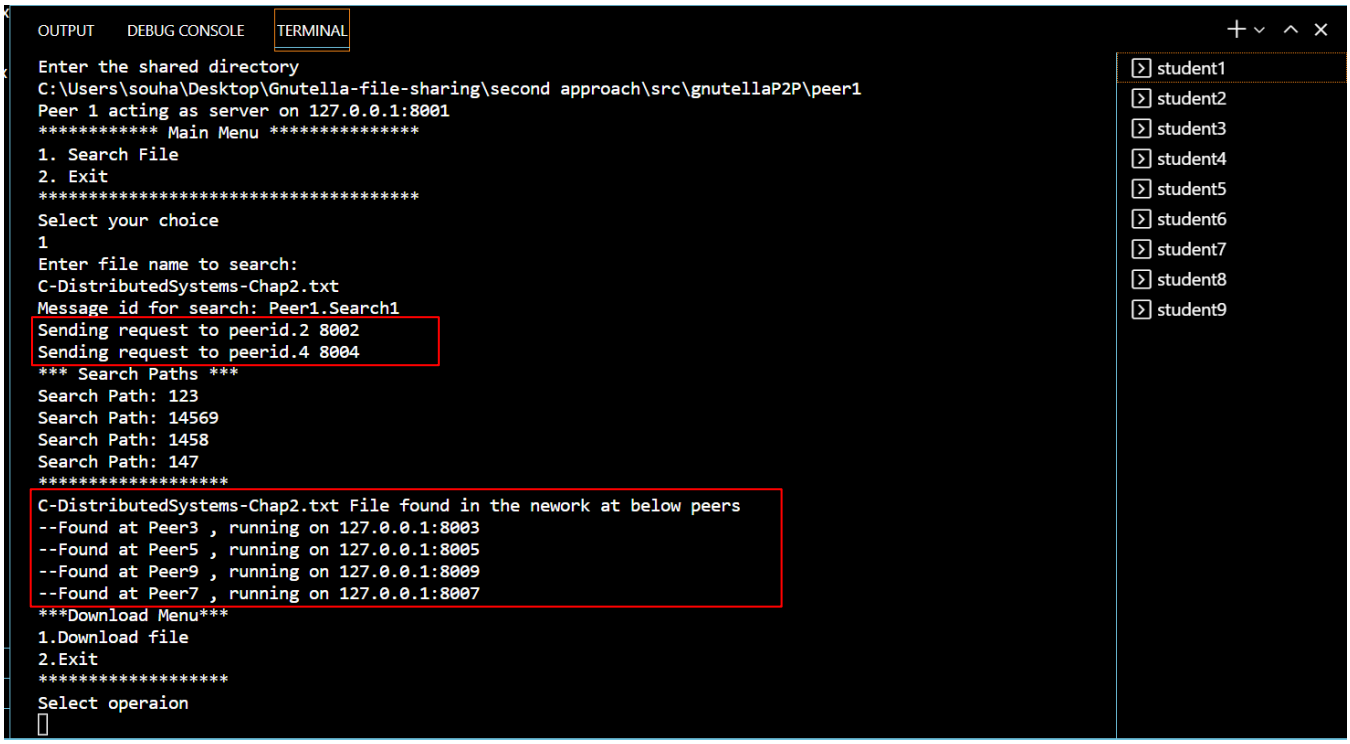


Student 9:



Now student1 wants to search for the chap2 of the course “distributed systems” which we added to student3, 5, 7 ,9 directories.

*- The search result in Peer1’s CMD : student1



After asking his neighbors 2 and 4, it’s clear that the file was found at peer 3, 5, 7 and 9.

*- The search result in Peer2's CMD : student2

```
Peer 2 acting as server on 127.0.0.1:8002
************ Main Menu ***************
1. Search File
2. Exit
************************************
Select your choice
Incoming Request to peer 2: From - 1 Search locally and send request to neighbours for msg id- Peer1.Search1
Local Search: File not found in the current peer
Outgoing Request from peer 2: Sending request to peerid.3 8003
Outgoing Request from peer 2: Sending request to peerid.5 8005
Incoming Request to peer 2: From - 5 Duplicate Request - Already searched in this peer- with message id - Peer1.Search1
HitQuery: Send following result back to 1
--Found at Peer3 on localhost:8003
```

*- The search result in Peer3's CMD : student3

```
Peer 3 acting as server on 127.0.0.1:8003
************ Main Menu ***************
1. Search File
2. Exit
************************************
Select your choice
Incoming Request to peer 3: From - 2 Search locally and send request to neighbours for msg id- Peer1.Search1
Local Search: File Found in the current peer
Outgoing Request from peer 3: Sending request to peerid.6 8006
Incoming Request to peer 3: From - 6 Duplicate Request - Already searched in this peer- with message id - Peer1.Search1
HitQuery: Send following result back to 2
--Found at Peer3 on localhost:8003
```

*- The search result in Peer4's CMD : student4

```
Peer 4 acting as server on 127.0.0.1:8004
************ Main Menu ***************
1. Search File
2. Exit
************************************
Select your choice
Incoming Request to peer 4: From - 1 Search locally and send request to neighbours for msg id- Peer1.Search1
Local Search: File not found in the current peer
Outgoing Request from peer 4: Sending request to peerid.5 8005
Outgoing Request from peer 4: Sending request to peerid.7 8007
HitQuery: Send following result back to 1
--Found at Peer5 on localhost:8005
--Found at Peer9 on localhost:8009
--Found at Peer7 on localhost:8007
```

*- The search result in Peer5's CMD : student5

```
Peer 5 acting as server on 127.0.0.1:8005
************ Main Menu ***************
1. Search File
2. Exit
************************************
Select your choice
Incoming Request to peer 5: From - 4 Search locally and send request to neighbours for msg id- Peer1.Search1
Incoming Request to peer 5: From - 2 Duplicate Request - Already searched in this peer- with message id - Peer1.Search1
Local Search: File Found in the current peer
Outgoing Request from peer 5: Sending request to peerid.2 8002
Outgoing Request from peer 5: Sending request to peerid.6 8006
Outgoing Request from peer 5: Sending request to peerid.8 8008
HitQuery: Send following result back to 4
--Found at Peer5 on localhost:8005
--Found at Peer9 on localhost:8009
```

*- The search result in Peer6's CMD : student6

```
Peer 6 acting as server on 127.0.0.1:8006
********** Main Menu **************
1. Search File
2. Exit
************************************
Select your choice
Incoming Request to peer 6: From - 5 Search locally and send request to neighbours for msg id- Peer1.Search1
Local Search: File not found in the current peer
Incoming Request to peer 6: From - 3 Duplicate Request - Already searched in this peer- with message id - Peer1.Search1
Outgoing Request from peer 6: Sending request to peerid.3 8003
Outgoing Request from peer 6: Sending request to peerid.9 8009
HitQuery: Send following result back to 5
--Found at Peer9 on localhost:8009
```

*- The search result in Peer7's CMD : student7

```
Peer 7 acting as server on 127.0.0.1:8007
********** Main Menu **************
1. Search File
2. Exit
************************************
Select your choice
Incoming Request to peer 7: From - 4 Search locally and send request to neighbours for msg id- Peer1.Search1
Local Search: File Found in the current peer
Outgoing Request from peer 7: Sending request to peerid.8 8008
HitQuery: Send following result back to 4
--Found at Peer7 on localhost:8007
Incoming Request to peer 7: From - 8 Duplicate Request - Already searched in this peer- with message id - Peer1.Search1
```

*- The search result in Peer8's CMD : student8

```
Peer 8 acting as server on 127.0.0.1:8008
********** Main Menu **************
1. Search File
2. Exit
************************************
Select your choice
Incoming Request to peer 8: From - 5 Search locally and send request to neighbours for msg id- Peer1.Search1
Local Search: File not found in the current peer
Incoming Request to peer 8: From - 7 Duplicate Request - Already searched in this peer- with message id - Peer1.Search1
Outgoing Request from peer 8: Sending request to peerid.7 8007
Outgoing Request from peer 8: Sending request to peerid.9 8009
HitQuery: Send following result back to 5
Incoming Request to peer 8: From - 9 Duplicate Request - Already searched in this peer- with message id - Peer1.Search1
```

*- The search result in Peer9's CMD : student9

```
Peer 9 acting as server on 127.0.0.1:8009
********** Main Menu **************
1. Search File
2. Exit
************************************
Select your choice
Incoming Request to peer 9: From - 6 Search locally and send request to neighbours for msg id- Peer1.Search1
Incoming Request to peer 9: From - 8 Duplicate Request - Already searched in this peer- with message id - Peer1.Search1
Local Search: File Found in the current peer
Outgoing Request from peer 9: Sending request to peerid.8 8008
HitQuery: Send following result back to 6
--Found at Peer9 on localhost:8009
```

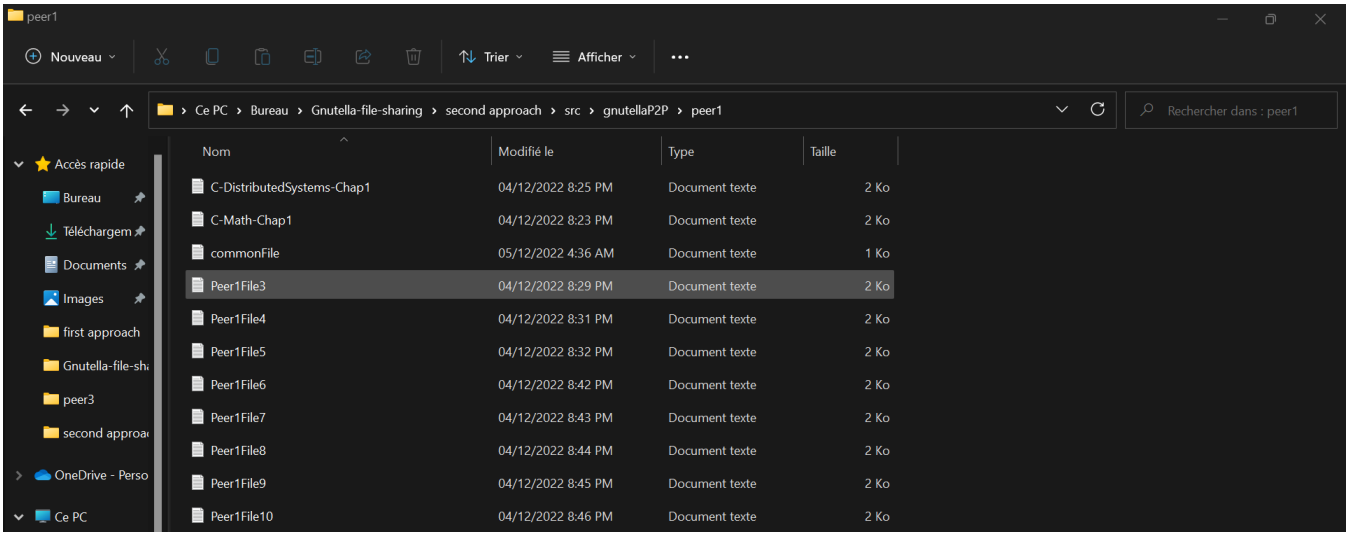
*- The download result in Peer1's CMD : student1

We select the peer id that we want to download the file from:

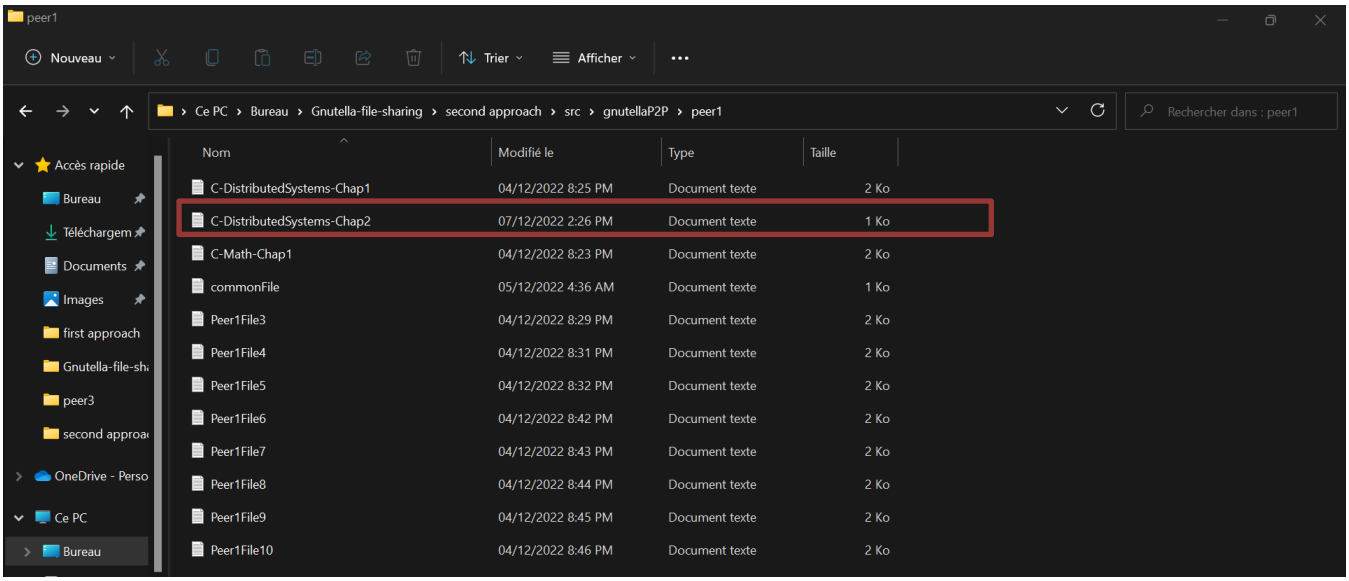
```
***Download Menu***
1.Download file
2.Exit
******************
Select operaion
1
Enter peer id to connect and download the file
3
Downloading from localhost:8003
"C-DistributedSystems-Chap2.txt" downloaded to path: C:\Users\souha\Desktop\Gnutella-file-sharing\second approach\src\gnu
tellaP2P\peer1
************ Main Menu ***************
1. Search File
2. Exit
************************************
Select your choice
```

Peer1 directory:

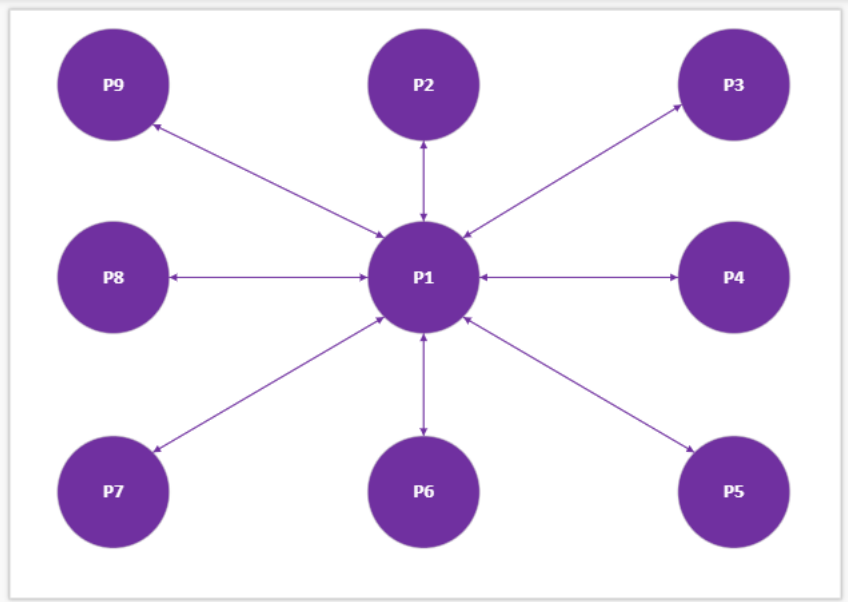
Before:



After:



III. Execution process for the second topology :



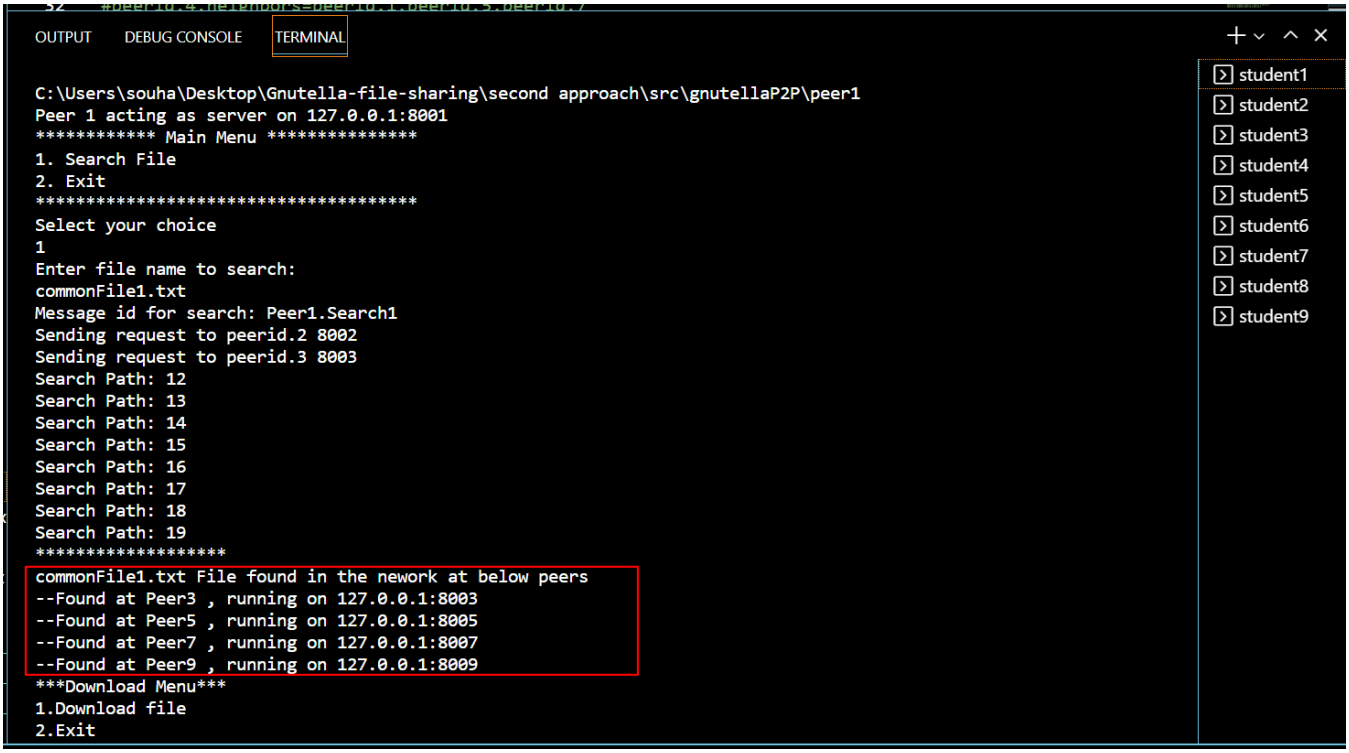
Config file:

```
39  #6 star network
40  peerid.1.neighbors=peerid.2,peerid.3,peerid.4,peerid.5,peerid.6,peerid.7,peerid.8,peerid.9
41  peerid.2.neighbors=peerid.1
42  peerid.3.neighbors=peerid.1
43  peerid.4.neighbors=peerid.1
44  peerid.5.neighbors=peerid.1
45  peerid.6.neighbors=peerid.1
46  peerid.7.neighbors=peerid.1
47  peerid.8.neighbors=peerid.1
48  peerid.9.neighbors=peerid.1
49
```

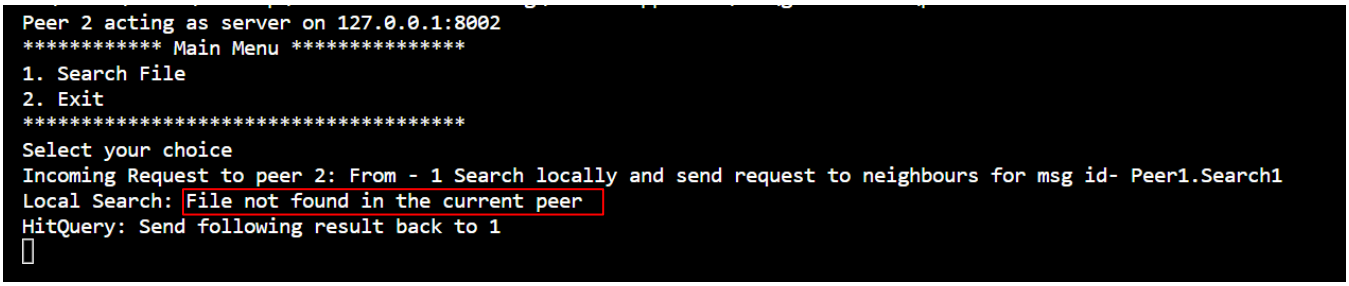
First, we do the same process defined in the first execution where we open 9 terminals corresponding to 9 peers and we initialize then by defining their ID, port and shared directory.

Then, we choose search file : student1 is searching for commonFile1.txt

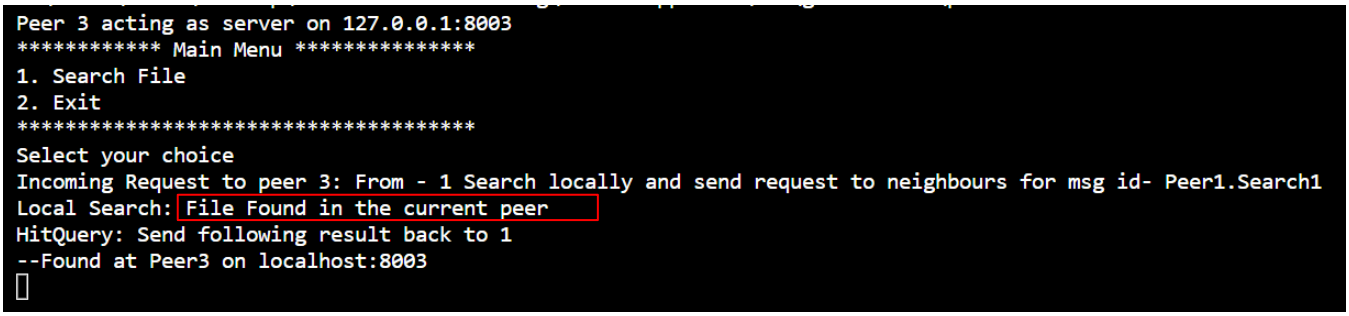
*- The search result in Peer1's CMD : student1



*- The search result in Peer2's CMD : student2



*- The search result in Peer3's CMD : student3



*- The search result in Peer4's CMD : student4

```
Peer 4 acting as server on 127.0.0.1:8004
************ Main Menu ***************
1. Search File
2. Exit
**************************************
Select your choice
Incoming Request to peer 4: From - 1 Search locally and send request to neighbours for msg id- Peer1.Search1
Local Search: File not found in the current peer
HitQuery: Send following result back to 1
```

*- The search result in Peer5's CMD : student5

```
Peer 5 acting as server on 127.0.0.1:8005
************ Main Menu ***************
1. Search File
2. Exit
**************************************
Select your choice
Incoming Request to peer 5: From - 1 Search locally and send request to neighbours for msg id- Peer1.Search1
Local Search: File Found in the current peer
HitQuery: Send following result back to 1
--Found at Peer5 on localhost:8005
```

*- The search result in Peer6's CMD : student6

```
Peer 6 acting as server on 127.0.0.1:8006
************ Main Menu ***************
1. Search File
2. Exit
**************************************
Select your choice
Incoming Request to peer 6: From - 1 Search locally and send request to neighbours for msg id- Peer1.Search1
Local Search: File not found in the current peer
HitQuery: Send following result back to 1
```

*- The search result in Peer7's CMD : student7

```
Peer 7 acting as server on 127.0.0.1:8007
************ Main Menu ***************
1. Search File
2. Exit
**************************************
Select your choice
Incoming Request to peer 7: From - 1 Search locally and send request to neighbours for msg id- Peer1.Search1
Local Search: File Found in the current peer
HitQuery: Send following result back to 1
--Found at Peer7 on localhost:8007
```

*- The search result in Peer8's CMD : student8

```
Peer 8 acting as server on 127.0.0.1:8008
************ Main Menu ***************
1. Search File
2. Exit
**************************************
Select your choice
Incoming Request to peer 8: From - 1 Search locally and send request to neighbours for msg id- Peer1.Search1
Local Search: File not found in the current peer
HitQuery: Send following result back to 1
```

*- The search result in Peer9's CMD : student9

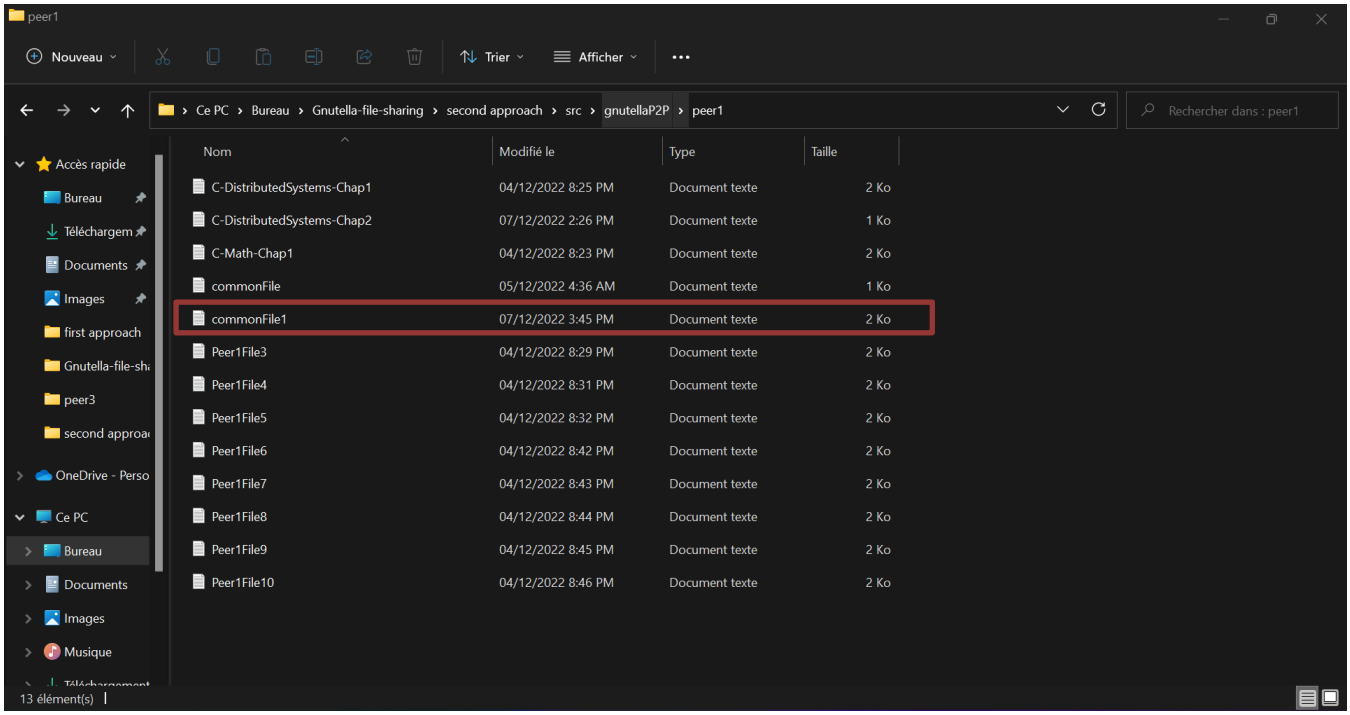
```
Peer 9 acting as server on 127.0.0.1:8009
************ Main Menu ***************
1. Search File
2. Exit
**************************************
Select your choice
Incoming Request to peer 9: From - 1 Search locally and send request to neighbours for msg id- Peer1.Search1
Local Search: File Found in the current peer
HitQuery: Send following result back to 1
--Found at Peer9 on localhost:8009
```

⇨ For the star topology, only one request from peer1 to all the other peers will be executed which is clear in the captures above.

*- The download result in Peer1's CMD : student1

```
***Download Menu***
1.Download file
2.Exit
*******************
Select operaion
1
Enter peer id to connect and download the file
7
Downloading from localhost:8007
"commonFile1.txt" downloaded to path: C:\Users\souha\Desktop\Gnutella-file-sharing\second approach\src\gnutellaP
2P\peer1
************ Main Menu ***************
1. Search File
2. Exit
**************************************
Select your choice
```

Peer1 Directory:



IV. Execution process for two different machines :

Machines must be connected to the same network.

We change the config file:

```
# Peer details
peerid.1.ip=192.168.43.140
peerid.1.port=8001
peerid.2.ip=192.168.43.215
peerid.2.port=8002


peerid.1.neighbors=peerid.2
```

Peer 1: @ip: 192.168.43.140

```
C:\Windows\System32\cmd.exe - java gnutellaP2P/Peer

C:\Users\Lenovo\Desktop\P2PGnutella-master\P2PGnutella-master\src>java gnutellaP2P/Peer
Enter the peerid
1
Enter the port
8001
Session for peer id: 1 started...
Enter the shared directory
C:\Users\Lenovo\Desktop\TPs_RT4\shared
Peer 1 acting as server on 127.0.0.1:8001
********** Main Menu **************
1. Search File
2. Exit
************************************
Select your choice
1
Enter file name to search:
Peer2File2.txt
Message id for search: Peer1.Search1
Sending request to peerid.2 8002
*** Search Paths ***
Search Path: 12
******************
Peer2File2.txt File found in the nework at below peers
--Found at Peer2 , running on 127.0.0.1:8002
***Download Menu***
1.Download file
2.Exit
******************
Select operaion
1
Enter peer id to connect and download the file
2
Downloading from localhost:8002
```

Peer2: @ip: 192.168.43.215

```
PS C:\Users\souha\Desktop\P2PGnutella-master\src> java gnutellaP2P/Peer
Enter the peerid
2
Enter the port
8002
Session for peer id: 2 started...
Enter the shared directory
C:\Users\souha\Desktop\P2PGnutella-master\src\gnutellaP2P\peer2
--Found at Peer2 on localhost:8002
```