Programming Project – 4

CS5352 Advanced Operating Systems

Design Gnutella File Sharing System with push based approach Output Screenshots

Rishika Reddy A

Mesh Topology(mesh.properties):

Command: vi mesh.properties

disci.hpcc.ttu.edu - PuTTY

```
# To change this license header, choose License Headers in Project Properties.
# To change this template file, choose Tools | Templates
# and open the template in the editor.
#Mon Apr 04 12:18:41 CDT 2016

#port number for each peer of mesh topology


peerone=1234
peertwo=2345
peerthree=3456
peerfour=4567
peerfive=5678
peersix=6789
peerseven=1111
peereight=5498
peernine=3649
peerten=3939

#neighours for star topology network

1234=peertwo,peerfour
2345=peerone,peerthree,peerfive
3456=peertwo,peersix
4567=peerone,peerfive,peerseven
5678=peertwo,peerfour,peersix,peereight
6789=peerthree,peerfive,peernine
1111=peerfour,peereight,peerten
5498=peerseven,peerfive,peernine
3649=peersix,peereight
3939=peerseven


~
~
~
~
~
```

Star Topology(star.properties):
Command: vi star.properties

```
#port number for each peer of star topology
peerone=1234
peertwo=2345
peerthree=3456
peerfour=4567
peerfive=5678
peersix=6789
peerseven=1111
peereight=5498
peernine=3649
peerten=3939


#neighours for star topology network
1234=peertwo,peerthree,peerfour,peerfive,peersix,peerseven,peereight,peernine,peerten
2345=peerone
3456=peerone
4567=peerone
5678=peerone
6789=peerone
1111=peerone
5498=peerone
3649=peerone
3939=peerten
~
~
~
```

Program execution:
Start peerone server:
Command:

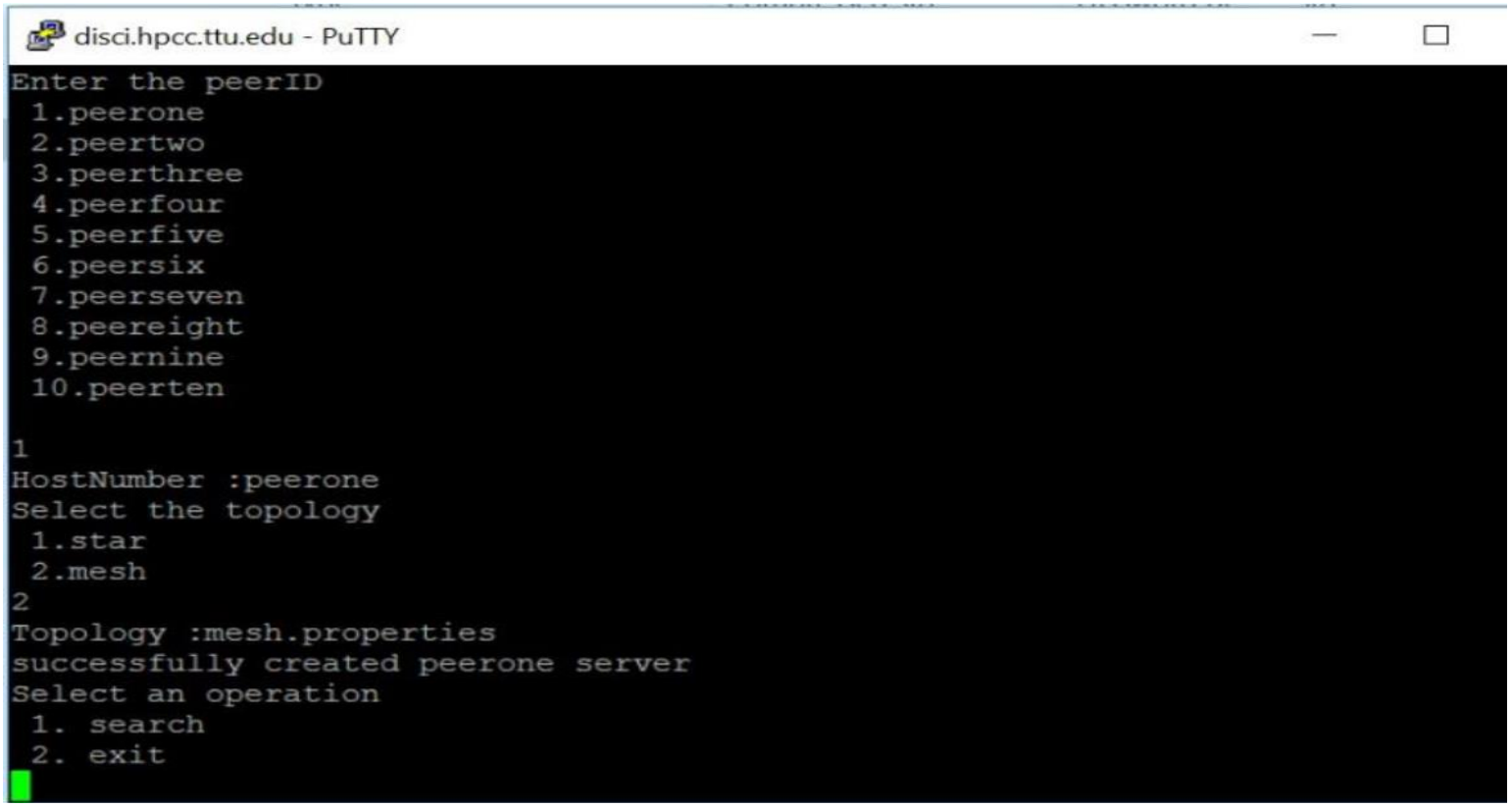
- Compiling – javac MainThread.java
- Execution – java MainThread
- Select – 1(peerone)

```
or service and University disciplinary procedure and/or criminal
prosecution.

If you have questions concerning this notification screen, please
contact IT Help Central at 742-HELP (4357).
-bash-4.1$ cd javaapplication12
-bash-4.1$ vi star.properties
-bash-4.1$ vi mesh.properties
-bash-4.1$ javac MainThread.java
-bash-4.1$ java MainThread
Enter the peerID
 1.peerone
 2.peertwo
 3.peerthree
 4.peerfour
 5.peerfive
 6.peersix
 7.peerseven
 8.peereight
 9.peernine
 10.peerten

1
HostNumber :peerone
```

- Select – 1(star.properties) or 2(mesh.properties)



Note: Run all the peers in different terminals

Start peertwo server:

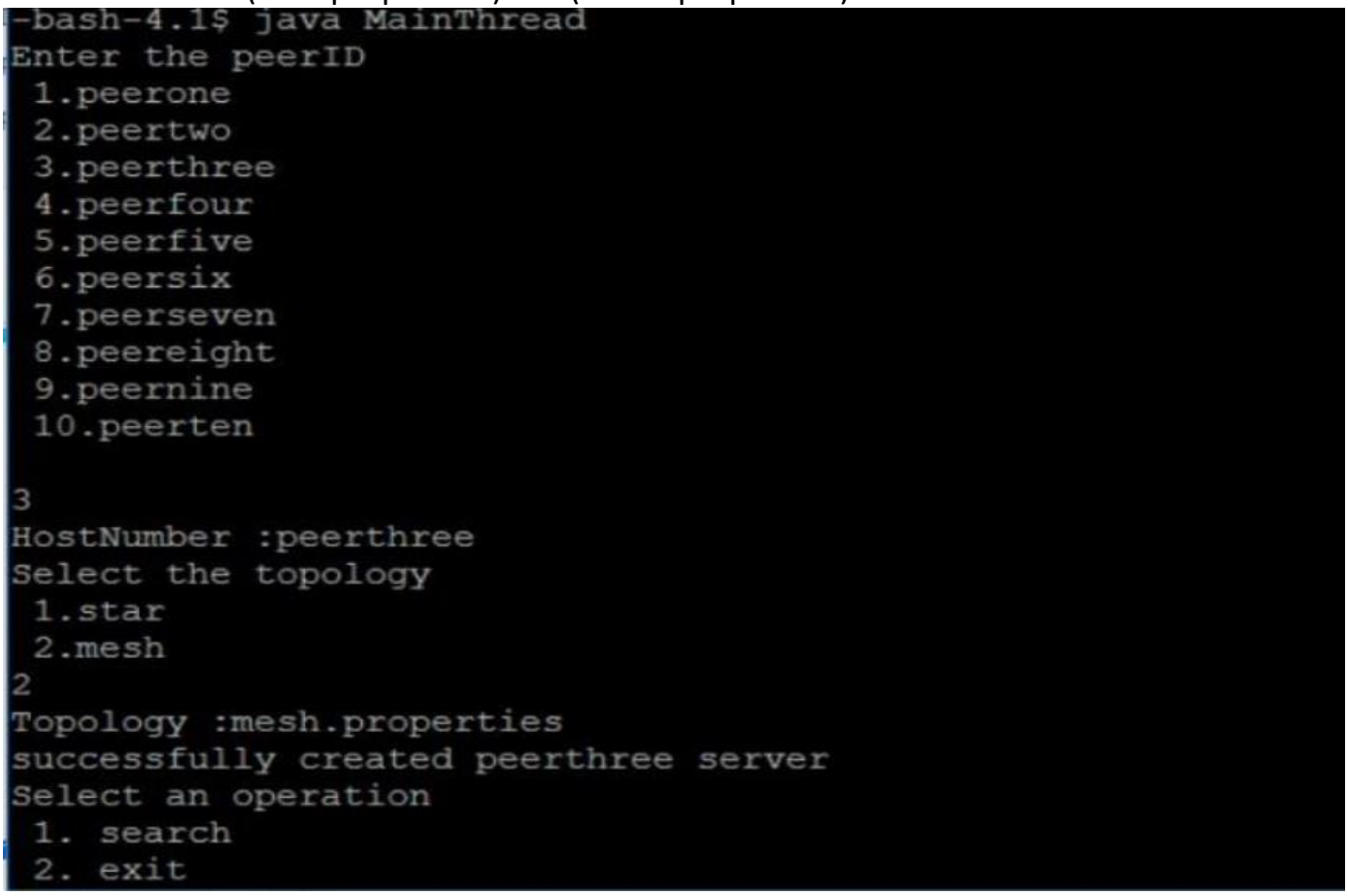
Command:

- Execution – java MainThread
- Select – 2(peertwo)
- Select – 1(star.properties) or 2(mesh.properties)

Start peerthree server:

Command:

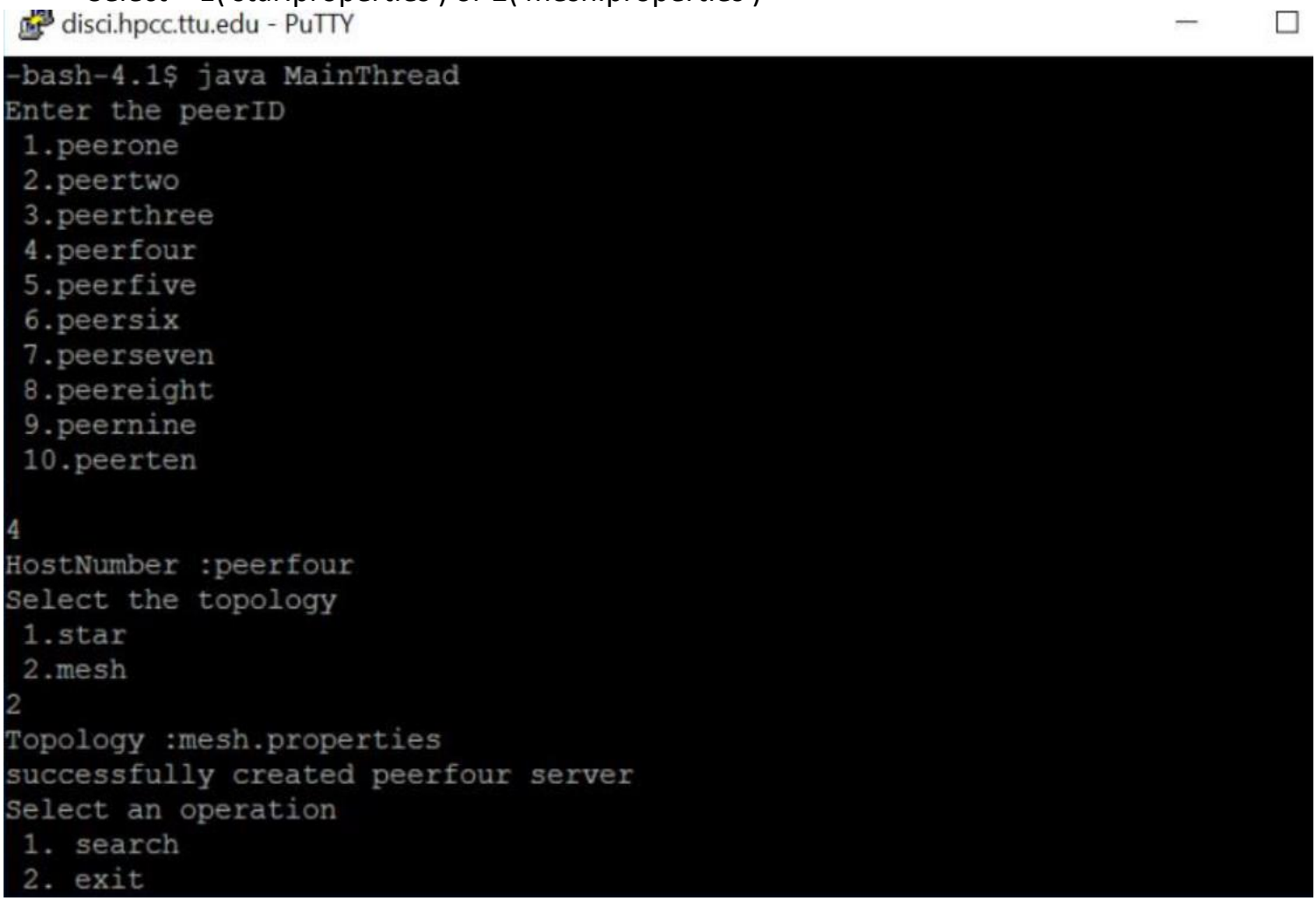
- Execution – java MainThread
- Select – 3(peerthree)
- Select – 1(star.properties) or 2(mesh.properties)



Start peerfour server:

Command:

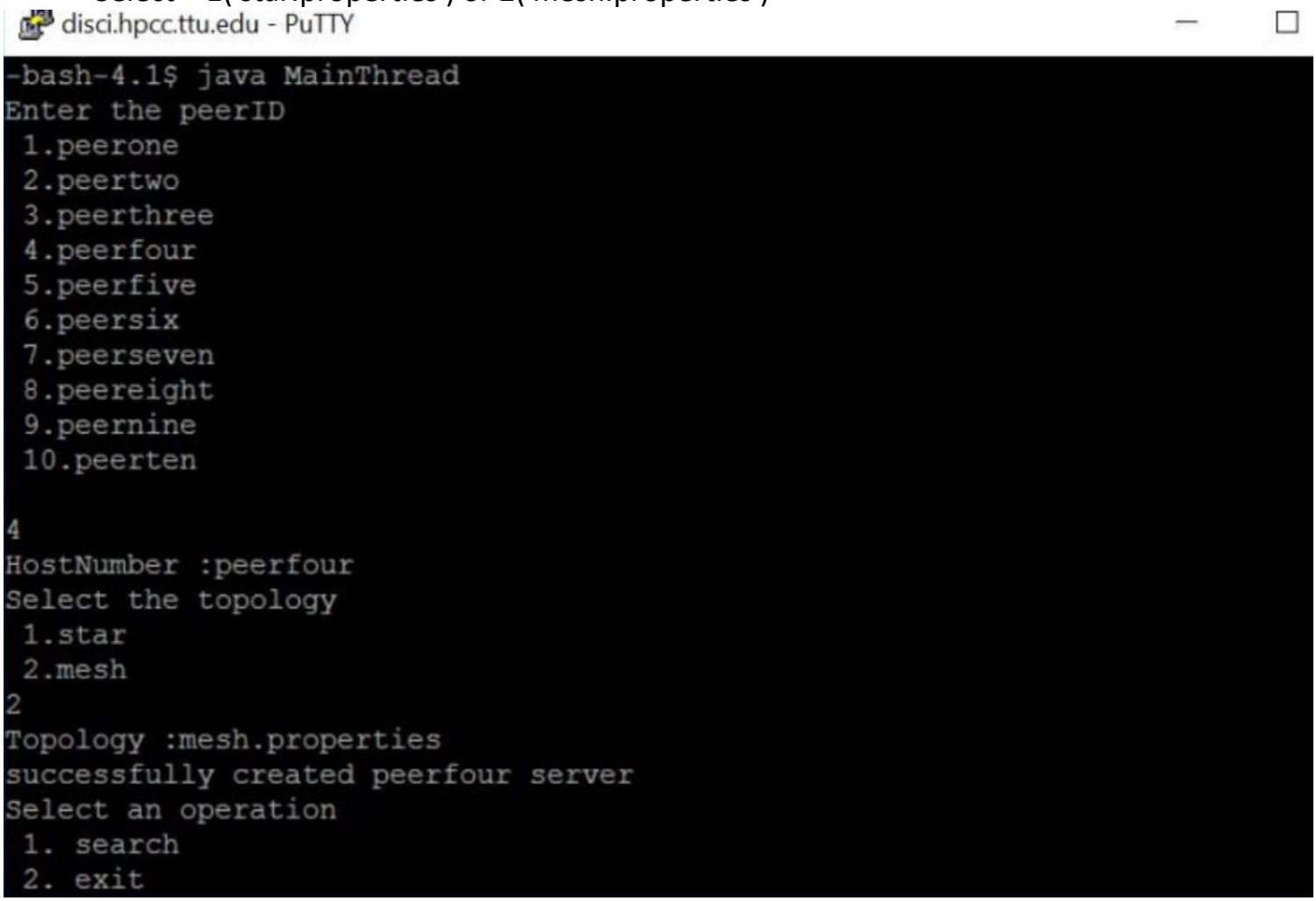
- Execution – java MainThread
- Select – 4(peerfour)
- Select – 1(star.properties) or 2(mesh.properties)



Start peerfive server:

Command:

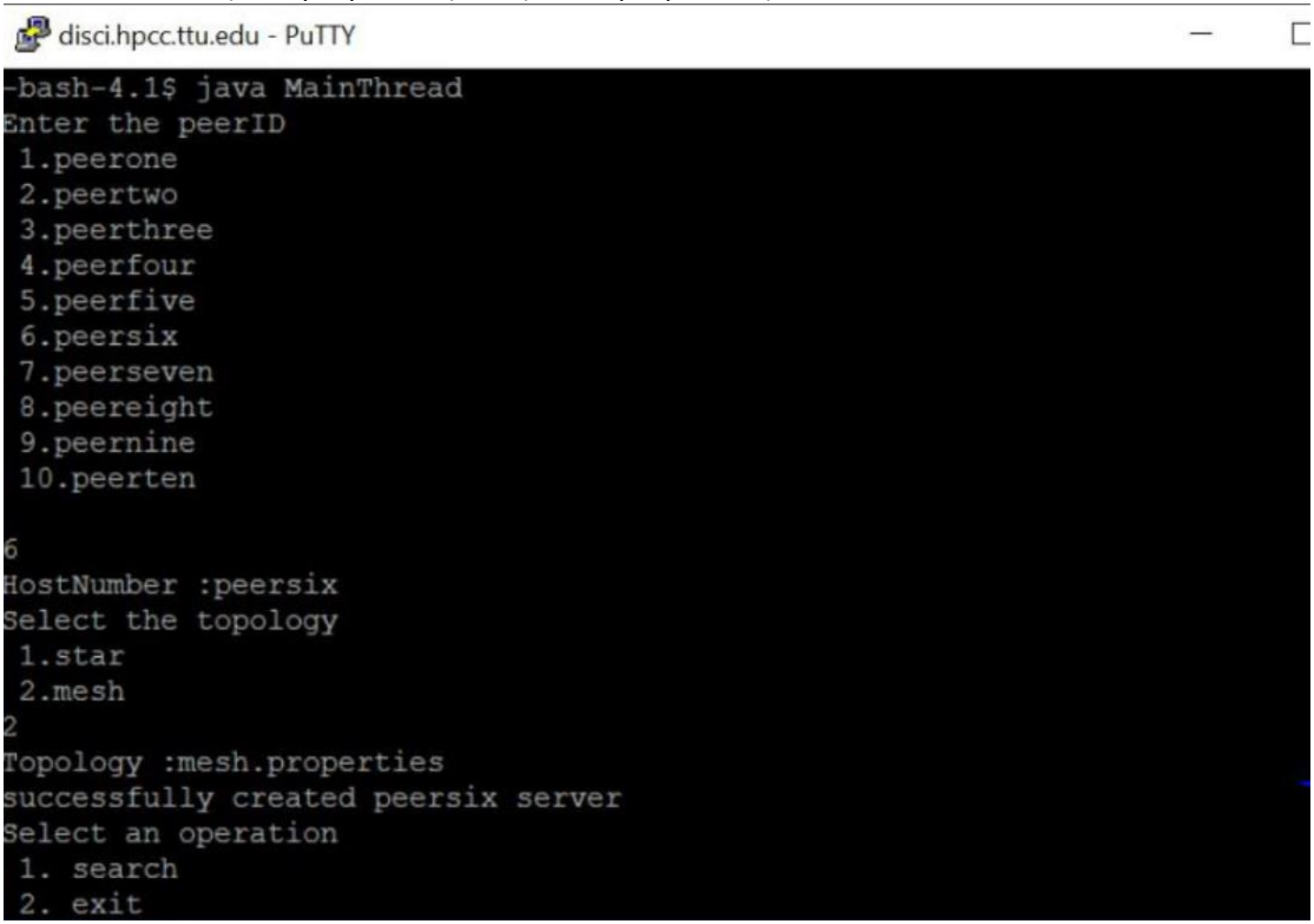
- Execution – java MainThread
- Select – 5(peerfive)
- Select – 1(star.properties) or 2(mesh.properties)



Start peersix server:

Command:

- Execution – java MainThread
- Select – 6(peersix)
- Select – 1(star.properties) or 2(mesh.properties)



Start peerseven server:

Command:

- Execution – java MainThread
- Select – 7(peerseven)
- Select – 1(star.properties) or 2(mesh.properties)

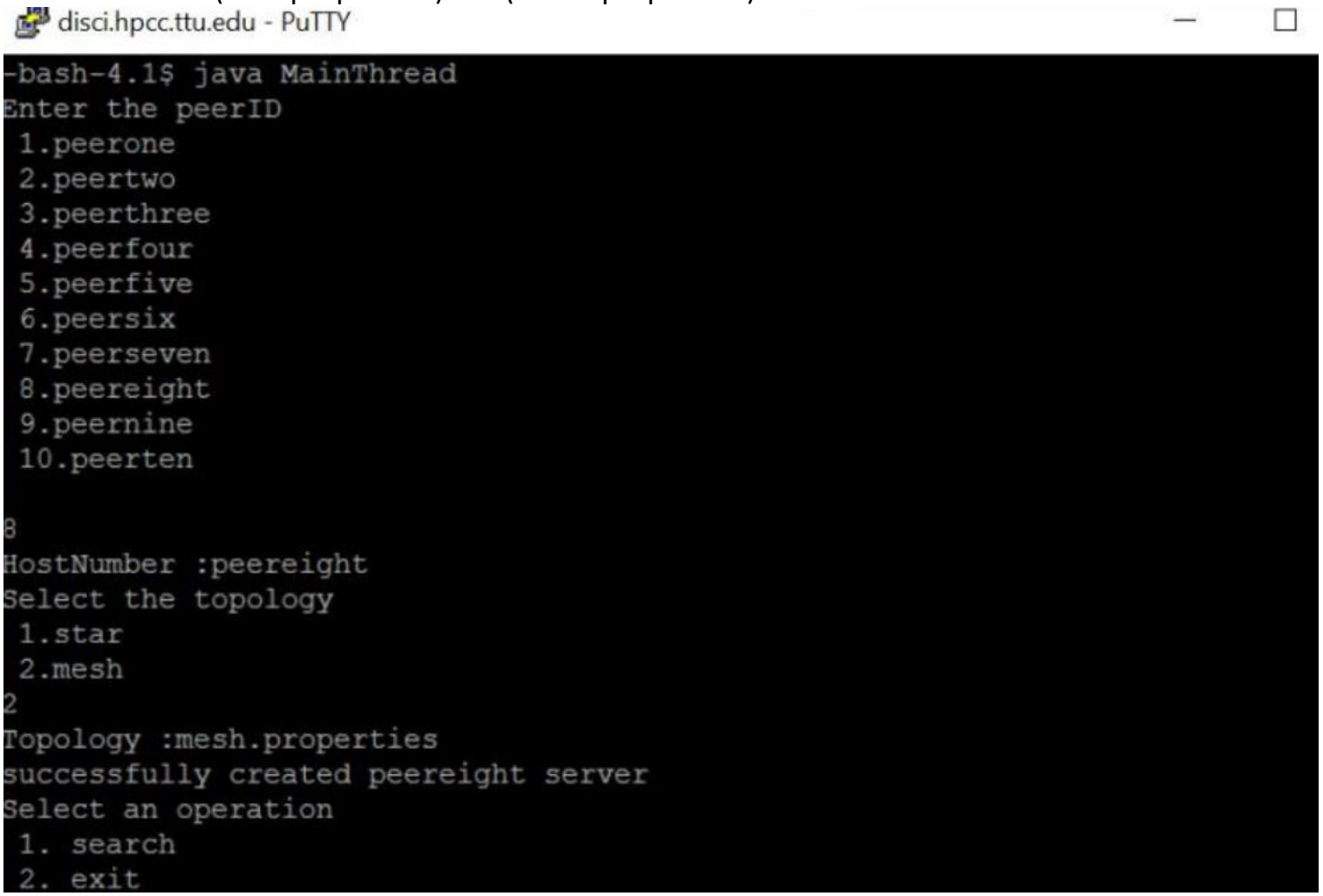
```
-bash-4.1$ java MainThread
Enter the peerID
 1.peerone
 2.peertwo
 3.peerthree
 4.peerfour
 5.peerfive
 6.peersix
 7.peerseven
 8.peereight
 9.peernine
 10.peerten

7
HostNumber :peerseven
Select the topology
 1.star
 2.mesh
2
Topology :mesh.properties
successfully created peerseven server
Select an operation
 1. search
 2. exit
```

Start peereight server:

Command:

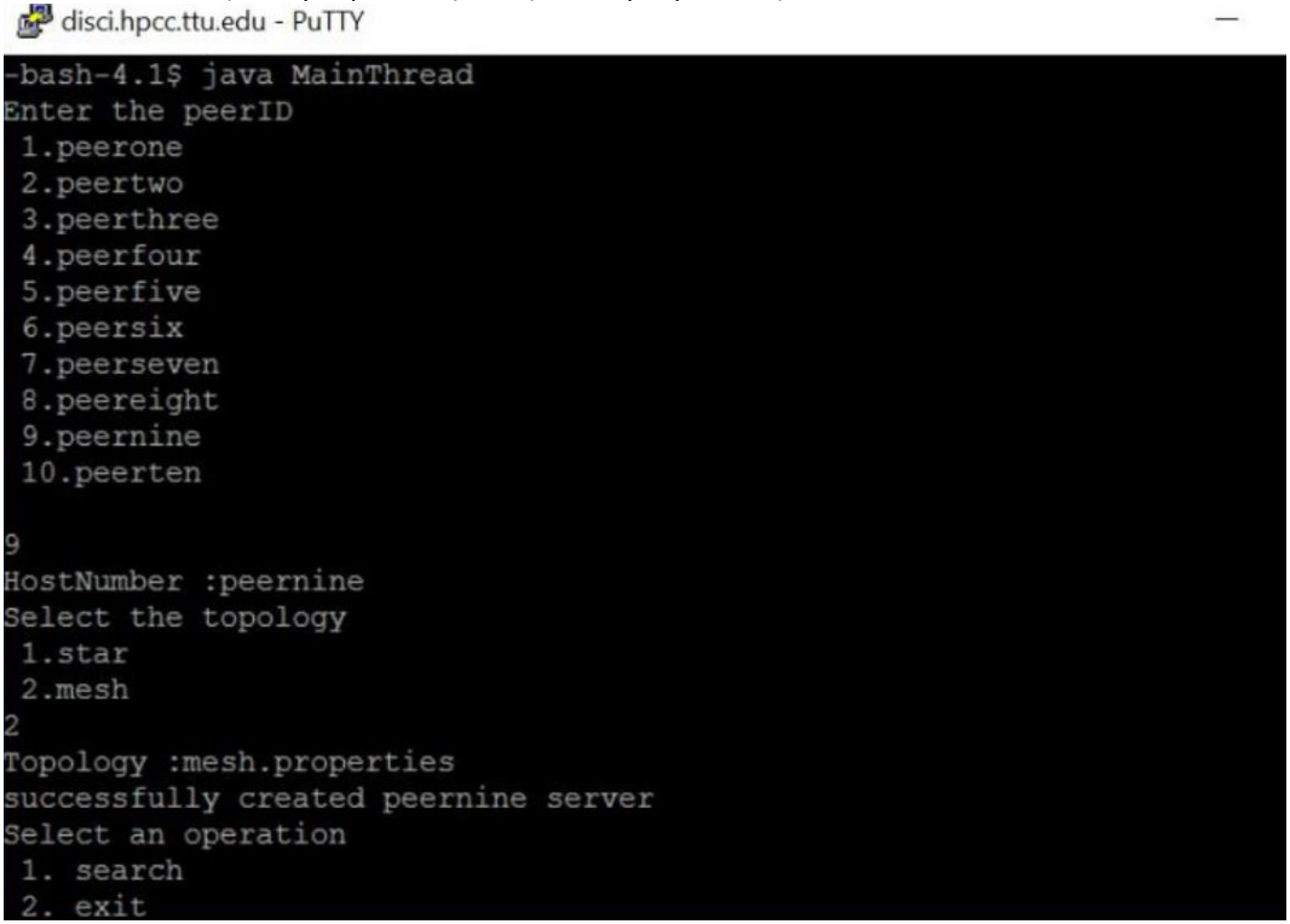
- Execution – java MainThread
- Select – 8(peereight)
- Select – 1(star.properties) or 2(mesh.properties)



Start peernine server:

Command:

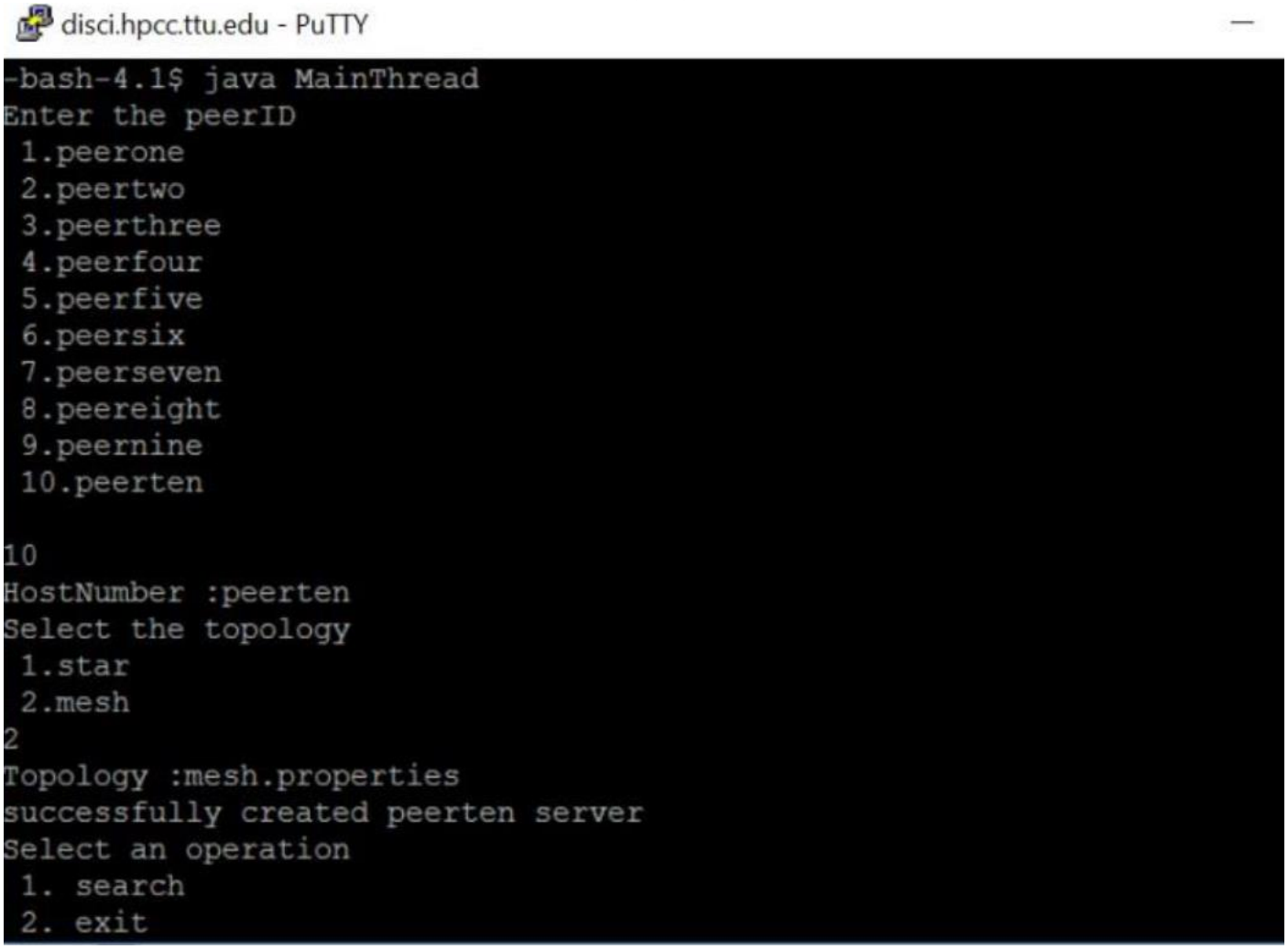
- Execution – java MainThread
- Select – 9(peernine)
- Select – 1(star.properties) or 2(mesh.properties)



Start peerten server:

Command:

- Execution – java MainThread
- Select – 10(peerten)
- Select – 1(star.properties) or 2(mesh.properties)

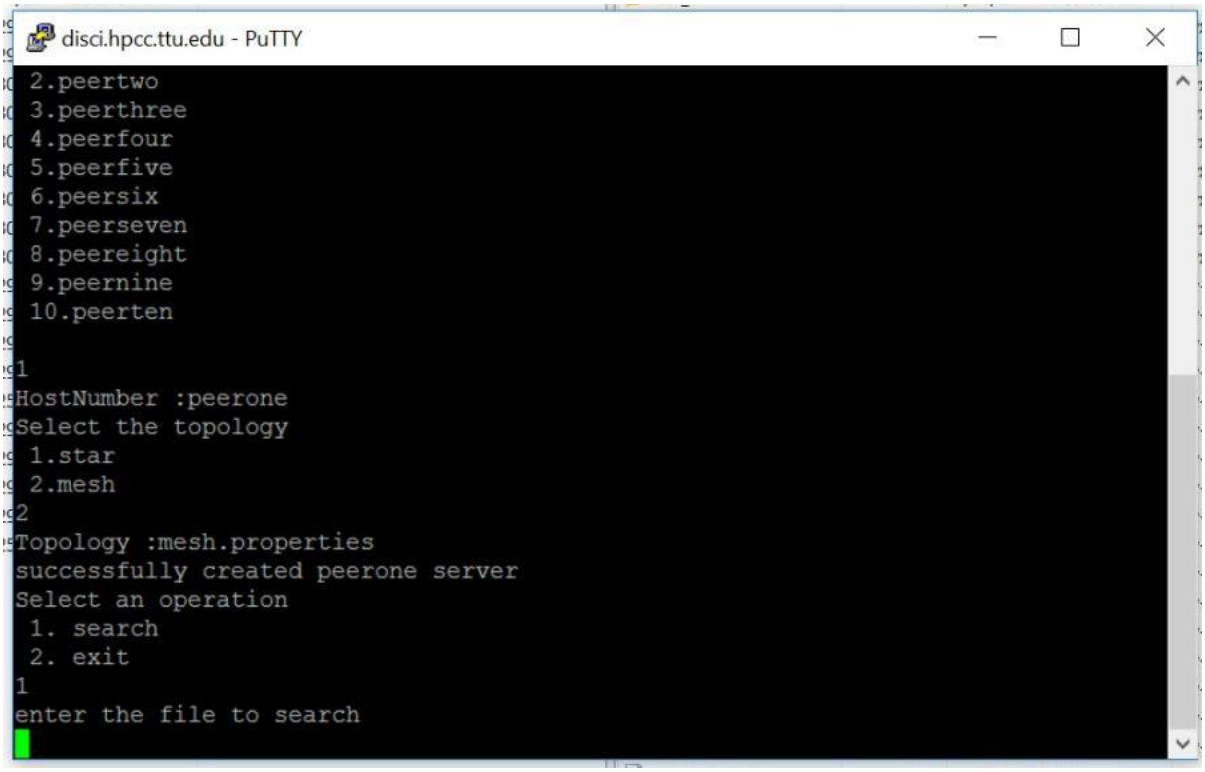


Searching file operation

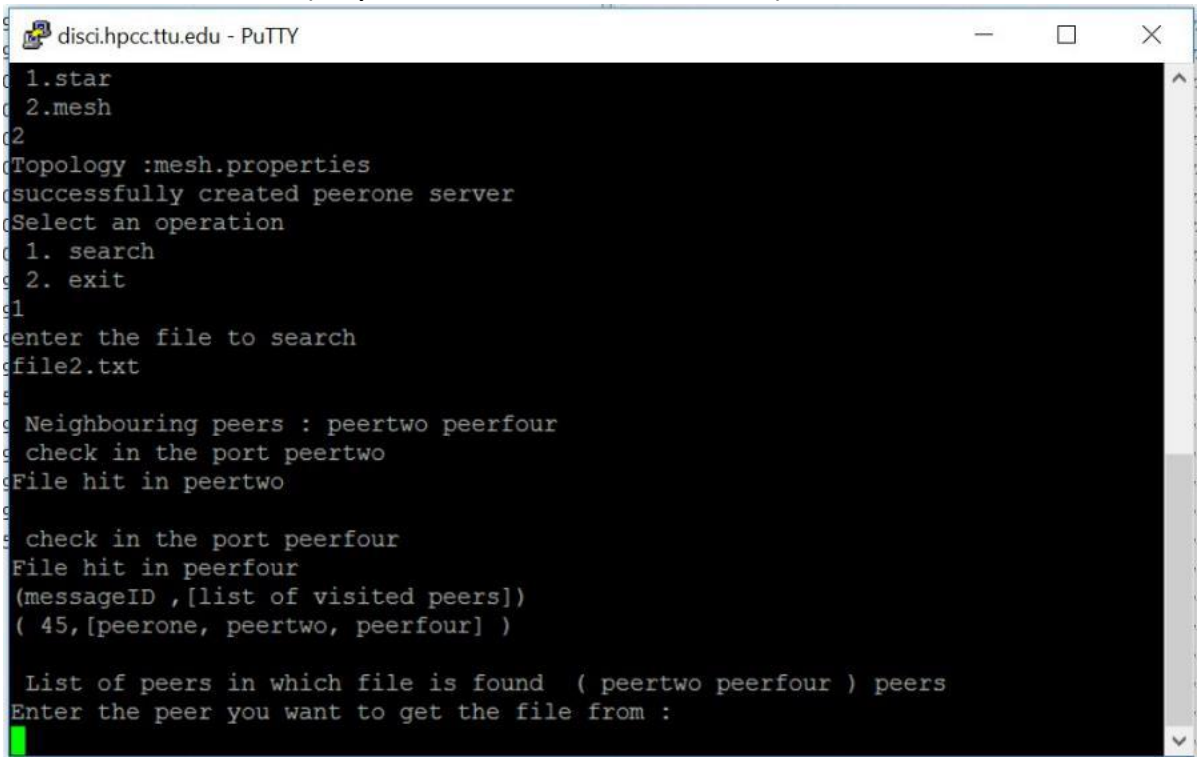
Navigate to any of the peer client terminal
In this example I have navigated to **peerone** client to perform search operation

Command:

Select - 1(Search Operation)



Command : file2.txt(Any other file u wish to search):



So the given file is found in the immediate neighbors peertwo and peerfour

Alternate cases:

- file is not found in immediate levels so it searches in next levels
- Time to live value is expired before the file is found (overcome this by searching from other peers)

Check files in peer1 before transfer
Command:

- cd Peer_1
- ls

```
If you have questions concerning this notification screen, please
contact IT Help Central at 742-HELP (4357).
-bash-4.1$ cd javaapplication12
-bash-4.1$ cd Peer_1
-bash-4.1$ ls
file11.txt  file14.txt  file1.txt  file6.txt  file8.txt
file12.txt  file16.txt  file3.txt  file7.txt  file9.txt
-bash-4.1$
```

File2.txt is not found before transfer

Transferring File

Command:
Select-Peerfour(or peertwo)

disci.hpcc.ttu.edu - PuTTY

```
1
enter the file to search
file2.txt

 Neighbouring peers : peertwo peerfour
 check in the port peertwo
File hit in peertwo

 check in the port peerfour
File hit in peerfour
(messageID ,[list of visited peers])
( 45,[peerone, peertwo, peerfour] )

 List of peers in which file is found  ( peertwo peerfour ) peers
Enter the peer you want to get the file from :
peerfour
connect to peerfour server
Waiting for download...
File has been transferred successfully
Select an operation
 1. search
 2. exit
File: file2.txt action: add
```

As it is a push based approach there will be a file monitoring thread which constantly checks for the changes happening in the peer server. Hence it returns action: add because a new file is added into the peerone server

Open peer_1 directory to check if file has been transferred

Command:

- cd peer_1
- ls

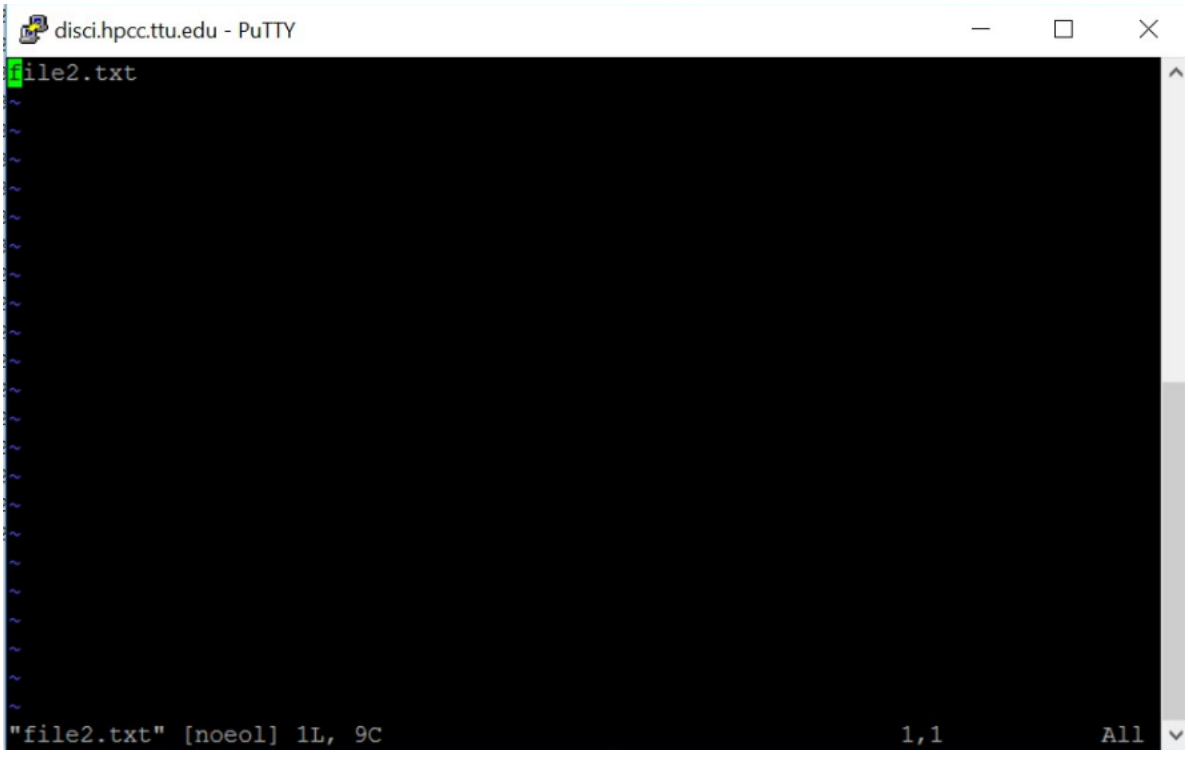
```
If you have questions concerning this notification screen, please
contact IT Help Central at 742-HELP (4357).
-bash-4.1$ cd javaapplication12
-bash-4.1$ cd Peer_1
-bash-4.1$ ls
file11.txt  file14.txt  file1.txt  file6.txt  file8.txt
file12.txt  file16.txt  file3.txt  file7.txt  file9.txt
-bash-4.1$ ls
file11.txt  file14.txt  file1.txt  file3.txt  file7.txt  file9.txt
file12.txt  file16.txt  file2.txt  file6.txt  file8.txt
-bash-4.1$
```

File2.txt has been transferred in peer_1 directory

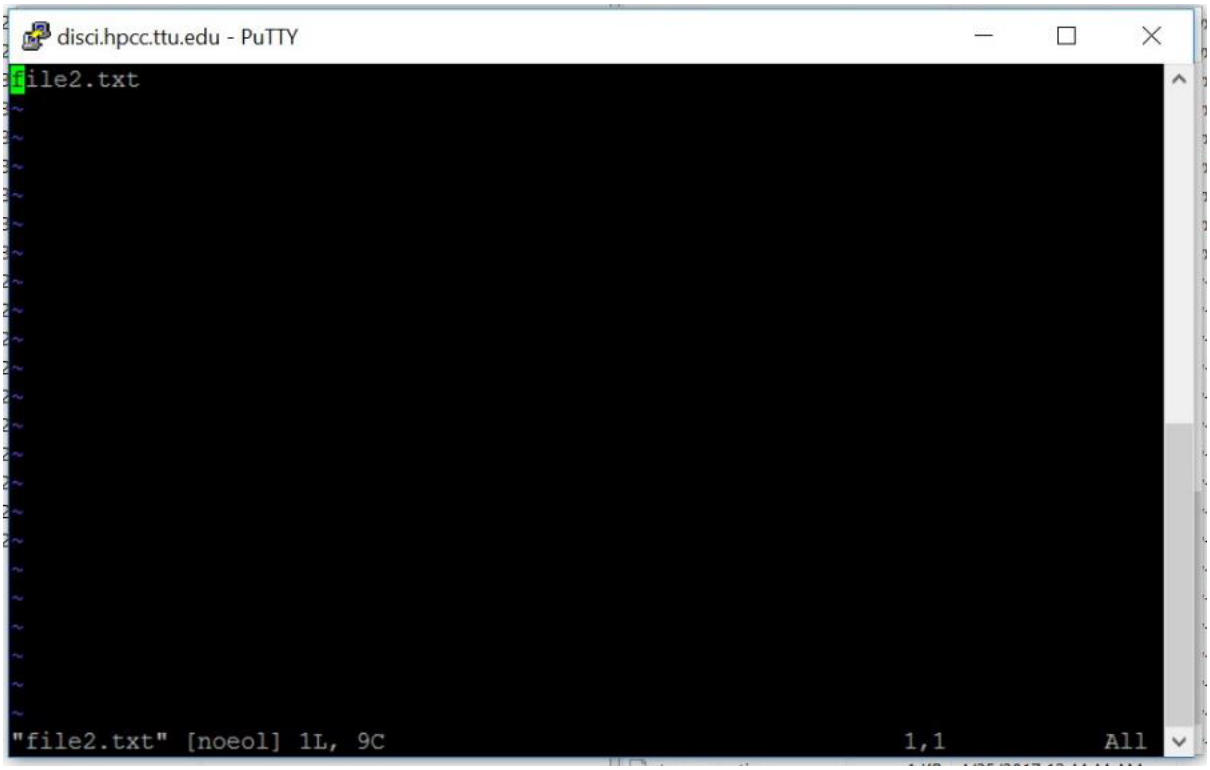
Modifying the contents of file

Contents of file in both the peers before modification:

Peer_4:



Peer_1:

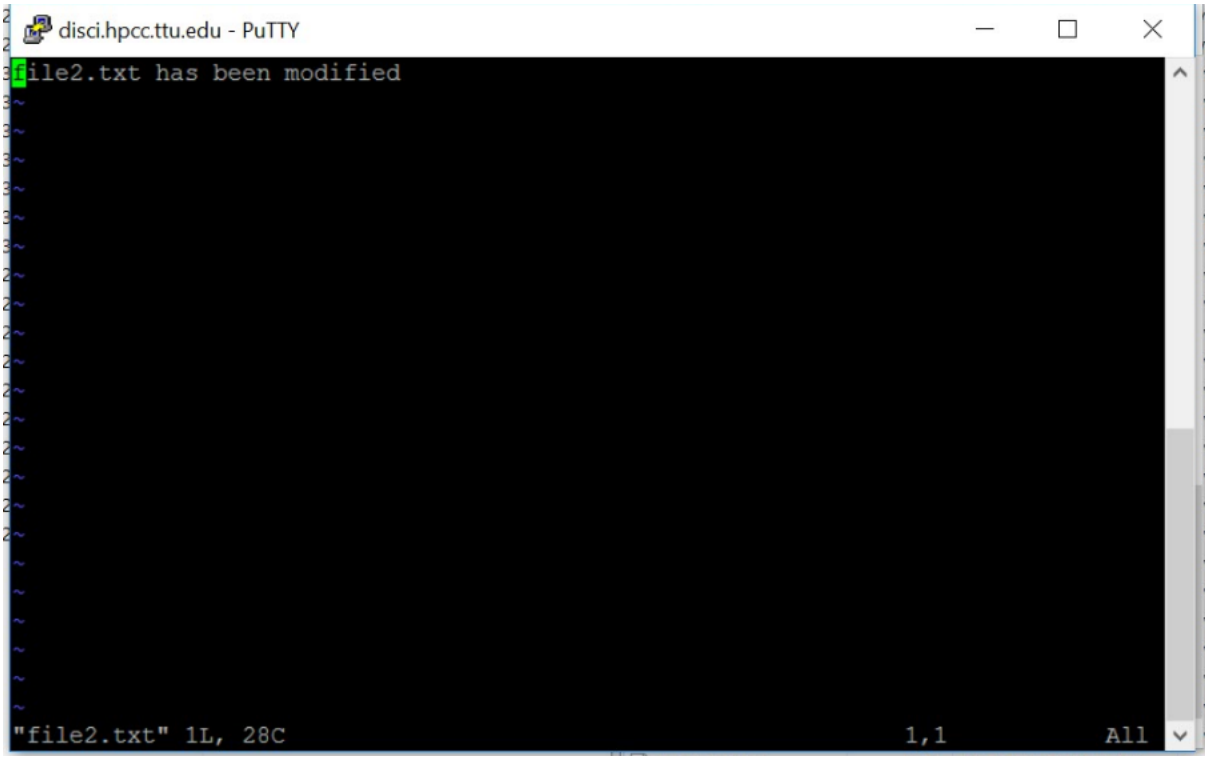


Modifying contents:

Commands:

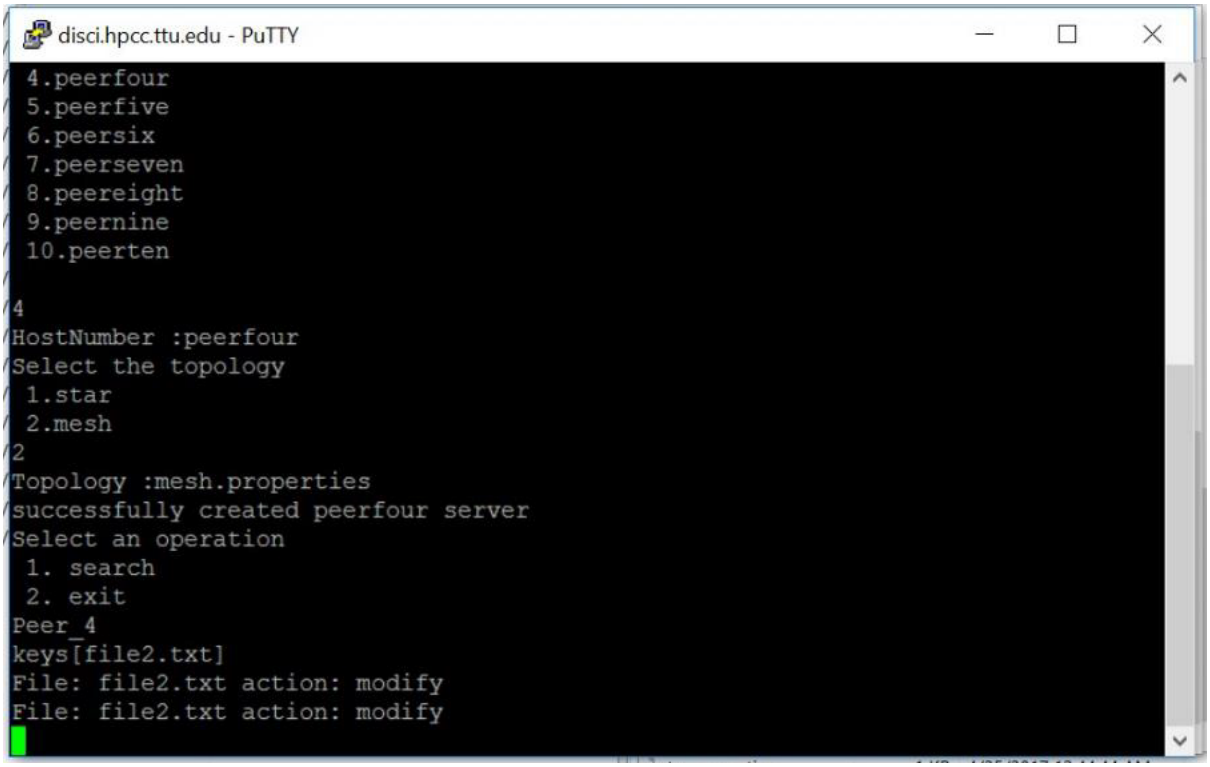
• Open file2.txt(Master file) in Peer_4 directory : cd Peer_4, vi file2.txt

Change the contents of file and **save** (esc+:wq) the file

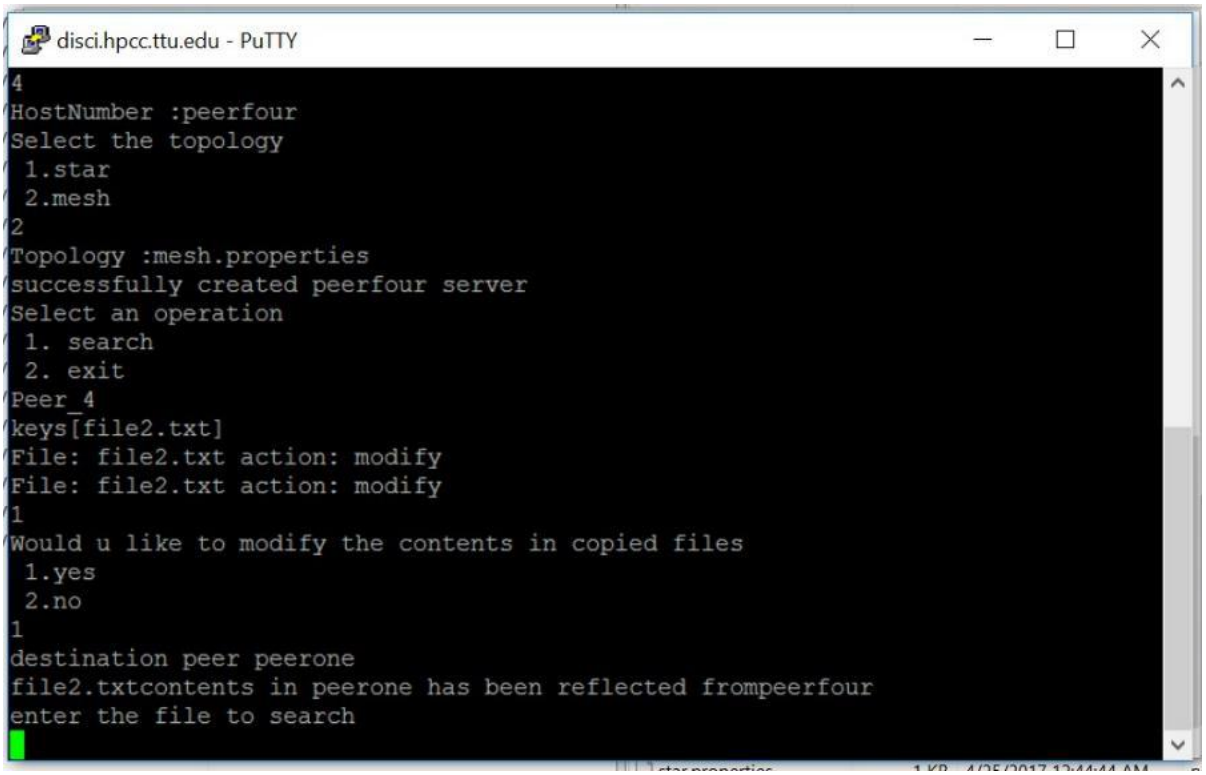


Open terminal of peerfour clients:

As it is a push based approach the thread for file monitoring detects the file changes and displays file has been modified on **peerfour terminal**

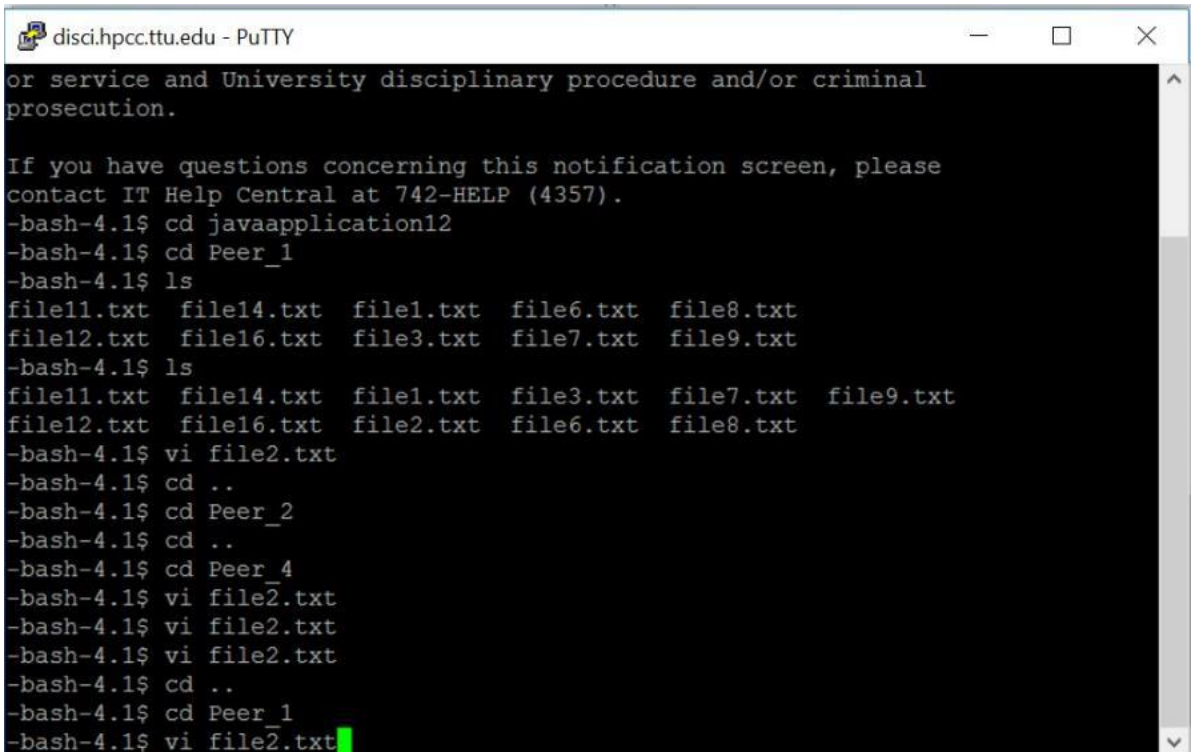


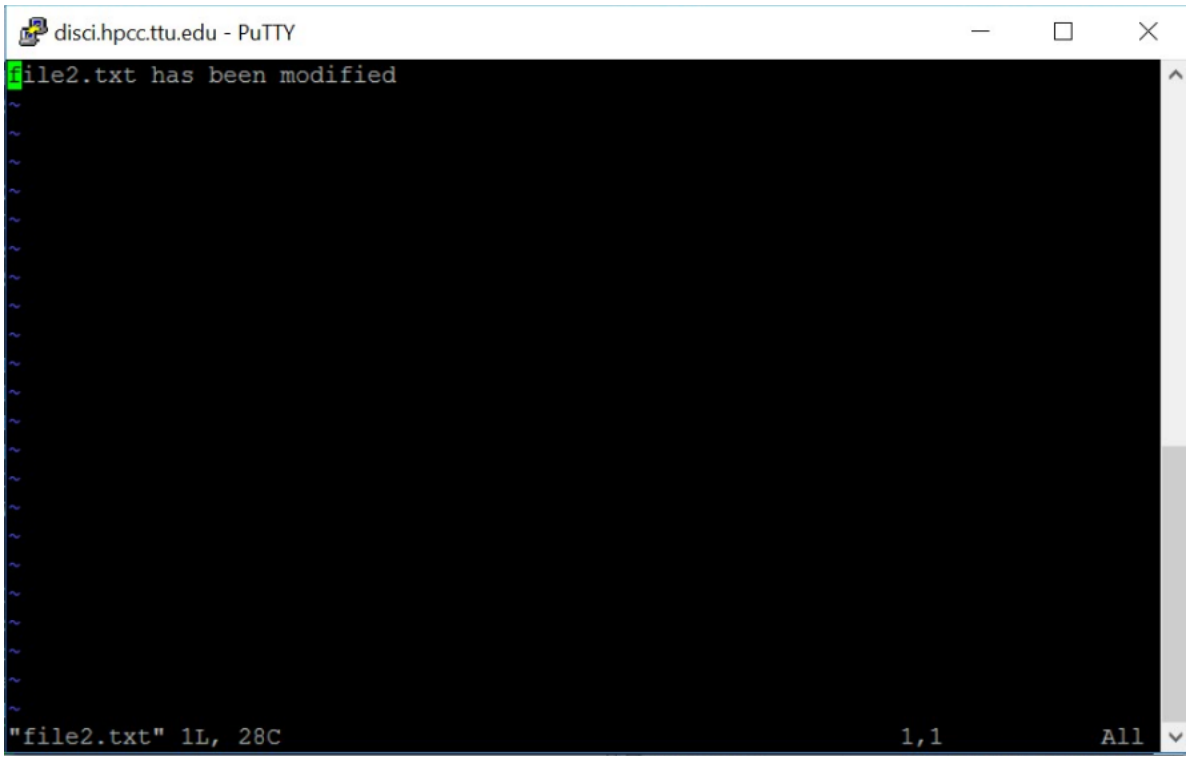
Now Continue executing peer4 client
Command : Select-1 (search operation)
Now it shows a prompt asking if you want to reflect the modified changes in cached file



Open file2.txt (Cached file) in Peer_1 directory to see if the changes are reflected
Commands :

- cd Peer_1
- vi file2.txt





Changes have been reflected in file2.txt of Peer_1 server