

## Manual

### Running the program:

1. Go to project root (Assignment 1) using Terminal.
2. You can type **"ls"** command to list the files and check whether you are in the correct directory.
3. Type **"make clean"** to delete old compiled .class files.
4. Type **"make"** to compile all project files.
5. Type **"make run\_server"** to run Indexing Server program.
6. Type **"make run\_peer"** to run Peer program.
7. Type **"make run\_test"** to run Test program. (It is used for performance evaluation)

The document below explains how to use the program, what are the different input the program asks for.

### Indexing Server:

You can run the Peer program by using the makefile command **"make run\_server"**. The server runs on port 10000. The user doesn't need to do anything for this port configuration.

The Indexing Server doesn't require any kind of input from the user. It just displays various messages related to what's going on. Basically, you can treat it as logs.

### Peer:

You can run the Peer program by using the makefile command **"make run\_peer"**. The Peer program internally runs two sub-programs (threads) called Peer Client and Peer Server. Peer Client connects to the Indexing Server using port 10000 and the Peer Server works on port 20000.

### Let's run the Peer program step-by-step: (Lines marked in **RED** are console output.)

```
make run_peer
```

```
***** PEER CLIENT STARTED *****
```

```
***** PEER SERVER STARTED *****
```

**Enter Server Address:**

Enter the server IP address here. It will be the IP address of the machine/node on which you are running the Indexing Server Program. For example, in my case, it is 192.168.206.128

**Do you want your node to act as a replication node? This requires your disk space to be large. (Y/N):**

Enter Y if you want your peer node to be considered by the Indexing Server as replication node. If you reply Y, then a copy of the files of all the peers connected to the Indexing Server will be saved on your node.

Enter Y for Yes and N for No.

What do you want to do?

- 1.Register files with indexing server.
- 2.Lookup for a file at index server.
- 3.Un-register all files of this peer from the indexing server.
- 4.Print download log of this peer.
- 5.Exit.

Enter choice and press ENTER:

Enter the respective number choice (1 – 5) of whatever task you want to do. Explanation for each option is as below:

1 = Register your files which you want to share with other peers in the network with the Indexing Server.

2 = Lookup or Search a particular file and check whether it is present in the network or not and if present get its location.

3 = Unregister/stop sharing the files you have shared with other peers. At this time, the program doesn't provide the option of unregistering only one file. Selecting this option unregisters all the files of your peer/node and removes its entries from the Indexing Server.

4 = Use this option if you want to check your node's download log. It is useful if you want to check which peers downloaded/used your file.

5 = Exit/Terminate the program.

If you select option 1:

Enter path of the files (multiple files) OR full path of a file with filename (single file) to sync with indexing server:

Enter the path of the folder whose files you want register with the Indexing Server for file sharing purpose with other peers. For example, in my case, **/home/gautammishra/node1**

OR

Enter full path of the file in case you want to register only one file. For example, **/home/gautammishra/node1/beach.txt**

The Peer then informs you whether the files were successfully registered with the Indexing Server.

If you select option 2:

Enter name of the file you want to look for at indexing server:

Enter the full name with extension of the file which you would like to look for on the Indexing Server. For example, **beach.txt** OR **data comm.pdf**

If the file is found, the client responds with the Peer ID and Host (Peer) Address of all the peers/nodes where the file is present. If the file is not found, it just responds File Not Found.

If the file is found, the peer program provides you an option to download the file like –

Do you want to download this file?(Y/N):

You can enter Y (Yes) or N (No). If Y, then the file is downloaded and it gets stored in the downloads folder located in the folder where you running the program.

If the file is a text file, the program provides you the option to download or print the first 1000 characters of the file like -

Do you want to download (D) or print this file (P)? Enter (D/P):

Enter D to just download the file and P to download the file as well as print the first 1000 characters of the text file.

In case, there are multiple nodes/peers available for download for a single file, then the program asks you to enter the Peer ID of the node which you would like to use to download the file like –

Enter Peer ID from which you want to download the file:

If you select option 3:

The files (if any) which you have synced/registered with the Indexing Server will be unregistered and those files will be no longer available to other peers for download.

If you select option 4:

This option prints the download log if available. Generally, the Peer server logs various messages while transferring files to other peers as requested by them.

If you select option 5:

This option will close/terminate the application normally. You'll be no longer connected in the file transfer network. Your files are not de-registered from the Indexing Server. In case you connect again to the Indexing Server, it reminds you of all the files/locations you might have registered earlier.

**Let's run the Test program step-by-step:** (Lines marked in **RED** are console output.)

You can run the Test program by using the makefile command **"make run\_peer"**. The Test program tests two operations – Lookup and Download. Basically, it runs the selected operation 1000 times on 2-3 threads each, in parallel.

What do you want to test?

1.Lookup

2.Download

3.Exit.

Enter choice and press ENTER:

If you select option 1:

Select this option if you want to test the lookup/search operation and get the average time in seconds per request. This is useful for performance evaluation.

Enter server address and name of the file you want to search:

For example,

192.168.206.128

Beach.jpg

Enter the Indexing Server and full name of the file with extension. The test program performs 1000 lookup requests for the given file name on 3 threads. So, it performs 3000 requests and gives the average time per request in seconds

If you select option 2:

Select this option if you want to test the download/file transfer operation and get the average file transfer speed in MBps. This is useful for performance evaluation.

Enter peer address and two file names you want to download:

For example,

192.168.206.128

beach.jpg

chicago.txt

Enter the Indexing Server and full name of the file with extension of two files. We need two files because it downloads the two files 1000 times each on a different thread. So, it performs 2000 requests and gives the average file transfer speed in MBps.

If you select option 3:

This option will close/terminate the test application normally.