PROJECT TITLE :

Napster-style Peer-To-Peer File Sharing System

User Manual

Name : Karthik Krishnamurthy

CWID : A20344464

Subj : CS-550 Advanced Operating Systems

Date : 09/21/2015

Project Title: Napster-Style Peer-to-Peer File Sharing System

USER MANUAL:

Steps to Run the Project:

1) The System should be running on a Linux Environment.

2) Java Development Kit(JDK) and JRE(Java Runtime Environment) should be installed on the System.

   The Link to Download JDK:

   http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html

   The version of JDK corresponding to the Operating system being used should be installed.

3) Ant Tool should be installed to run the Build Script.

   The Link to Download Ant:

   https://ant.apache.org/bindownload.cgi

4) Launch the Terminal application and Extract the given zip file: "PROG1_Krishnamurthy_Karthik.zip"

   Using the command: unzip PROG1_Krishnamurthy_Karthik.zip

5) Navigate into the Extracted folder > There will be a subfolder called "P2P_File_Sharing_System" which contains the CentralIndexServer and the Peer(s) Modules: Peer_1, Peer_2, Peer_3 and Performance_Evaluation

   ===================**Directory Structure**=====================

6) The **CentralIndxServer** folder contains:

   a) The Source file of the CentralIndxServer – "CentralIndxServer.java"

   b) Compiled files- ["CentralIndxServer.class", "PortListener.class" and "begin.class"]

   c) "build.xml" which is an "ant"-BuildScript to compile and build the CentalIndxServer.

7) The **Peer_1** folder contains:

a) The Source file of Peer_1 – “PeerServer.java”

b) Compiled files – [“PeerServer.class” and “PortListenerSend.class”]

c) “indxip.txt” – Which contains the IP Address of the CentralIndxServer [This text file should be edited – And the IP Address of the Machine which runs the CentralIndxServer should be entered in this file before running the Peer].

d) Few Testfiles : “my.txt”, testfile.txt, 1.txt, 2.txt and so on upto 10.txt [12 Testfiles of Peer_1 – which can be used for registration, search and download functionalities].

e) “build.xml” which is an “ant”-BuildScript to compile and build the Peer_1.

8) The **Peer_2** folder contains:

a) The Source file of Peer_2 – “PeerServer.java”

b) Compiled files – [“PeerServer.class” and “PortListenerSend.class”]

c) “indxip.txt” – Which contains the IP Address of the CentralIndxServer [This text file should be edited – And the IP Address of the Machine which runs the CentralIndxServer should be entered in this file before running the Peer].

d) Few Testfiles : 11.txt, 12.txt and so on upto 20.txt [10 Testfiles of Peer_2 – which can be used for registration, search and download functionalities].

e) “build.xml” which is an “ant”-BuildScript to compile and build the Peer_2.

9) The **Peer_3** folder contains:

a) The Source file of Peer_3 – “PeerServer.java”

b) Compiled files – [“PeerServer.class” and “PortListenerSend.class”]

c) “indxip.txt” – Which contains the IP Address of the CentralIndxServer [This text file should be edited – And the IP Address of the Machine which runs the CentralIndxServer should be entered in this file before running the Peer].

d) Few Testfiles : 21.txt, 22.txt and so on upto 30.txt [10 Testfiles of Peer_3 – which can be used for registration, search and download functionalities].

e) “build.xml” which is an “ant”-BuildScript to compile and build the Peer_3.

10) There are 2 ways to simulate the entire project:

a) Running the entire project on a Single system by running Separate processes on "localhost" or by running Multiple Virtual Machines

b) Running the project on Multiple Machines.

For **Running on Multiple machines**, refer to **SECTION B.**

Also Refer To [NOTE:] section at the end of the document before Running the Project.

SECTION A:

a) Steps for Simulating the entire project on a Single system by running separate processes on "localhost" or by running Multiple Virtual machines:

1) Launch the Terminal and Navigate into the P2P_File_Sharing_System folder from the above zip.

2) Open Multiple-tabs and navigate into CentralIndxServer folder on the First tab, into the Peer_1 folder on the second tab, into the Peer_2 folder on the third tab and into the Peer_3 folder on the fourth tab.

3) Navigate into the First tab, where CentralIndxServer files are present.

4) On the Command prompt Run the build-script using the ant command.

```
$ ant
```

5) The Build-script is executed and it cleans and compiles the files of the CentralIndxServer present in the CentralIndxServer directory. After successful build, a message is displayed on the terminal:

```
Build Successful
Total Time: 1 second
```

6) Run the CentralIndxServer by the command:

```
$ java CentralIndxServer
```

A message is displayed saying the "<Central Index Server is Up and Running>"
This runs the CentralIndxServer and it starts listening to other Peers on the Port

7) Now, Navigate into the Second tab, where Peer_1 files are present.

8) Edit the file **"indxip.txt"** present inside Peer_1 folder using "vim" or "gedit". Edit and Enter the **IP Address** of the Machine which is running the **CentralIndxServer** into this file. The IP Address of the Machine running CentralIndxServer can be obtained by running the "**ifconfig**" command on the CentralIndxServer Machine.
Example: If the IP Address of the CentralIndxServer Machine is: 10.0.0.16 or 127.0.0.1(for Localhost)

```
$vim indxip.txt
```

Inside the file-> Remove the existing IP Address and Enter:

```
10.0.0.16
```

Save the file and exit.

9) On the Command prompt Run the build-script using the ant command.

```
$ ant
```

10) The Build-script is executed and it cleans and compiles the files of the Peer_1 present in the Peer_1 directory. After successful build, a message is displayed on the terminal:

```
Build Successful
Total Time: 1 second
```

11) Run Peer_1 using the command:

```
$ java PeerServer
```

The Peer is run and a MENU is displayed on the screen with the following options for the Peer to select:

```
Enter The Option:
==================
1. Registering the File

2. Searching On CentralIndxServer

3. Downloading From Peer Server

4. Exit
```

=================**Registration Procedure**========================

12) If the Peer_1 wants to Register the file, it can choose option 1.
The Peer_1 then has to:
Enter the String in Format: 4Digit id and File Names separated by Space
Example: 3001 my.txt

Then the Peer_1 is Connected to Register on CentralIndxServer on port 2001
And the file "my.txt" is "Registered Successfully" on the CentralIndxServer.

The Peer_1 can also register multiple files by specifying the ID and filenames separated by a space
Example: 4001 A.txt B.txt C.txt D.txt……..and so on.

===================**Search Procedure**============================

13) If Peer_2 wants to Search the file “my.txt”, it can Search on the CentralIndxServer to find out which Peer has the required file.

Navigate into the Third tab, where Peer_2 files are present.

14) Edit the file **“indxip.txt”** present inside Peer_2 folder using “vim” or “gedit”. Edit and Enter the **IP Address** of the Machine which is running the **CentralIndxServer** into this file. The IP Address of the Machine running CentralIndxServer can be obtained by running the “**ifconfig**” command on the CentralIndxServer Machine.
Example: If the IP Address of the CentralIndxServer Machine is: 10.0.0.16
$vim indxip.txt
Inside the file-> Remove the existing IP Address and Enter:
10.0.0.16
Save the file and exit.

15) On the Command prompt Run the build-script using the ant command.

```
$ ant
```

16) The Build-script is executed and it cleans and compiles the files of the Peer_2 present in the Peer_2 directory. After successful build, a message is displayed on the terminal:

```
Build Successful
Total Time: 1 second
```

17) Run Peer_2 using the command:

```
$ java PeerServer
```

18) Then once it is run, select 2nd which is the Search option from the MENU to search for the desired file.
Then Peer_2 has to:
Enter the filename to search:
Example: my.txt

Then the Peer_2 is Connected to Search on CentralIndxServer on port 2002
And the Search result is returned from the CentralIndxServer:
File:'my.txt' found at peers:3001 (10.0.0.13).

Here 3001 is the Peer ID and “10.0.0.13” is the IP address of the Peer holding the File “my.txt”. In this case it is Peer_1 which has the file “my.txt”.

======================**Download Procedure**=========================

19) If Peer_3 wants to download the file "my.txt" which is registered by Peer_1, then
It can Download from Peer_1 which acts as the PeerServer.

Navigate into the Fourth tab, where the Peer_3 Files are present.

20) Edit the file **"indxip.txt"** present inside Peer_3 folder using "vim" or "gedit". Edit and Enter the **IP Address** of the Machine which is running the **CentralIndxServer** into this file. The IP Address of the Machine running CentralIndxServer can be obtained by running the "**ifconfig**" command on the CentralIndxServer Machine.
Example: If the IP Address of the CentralIndxServer Machine is: 10.0.0.16

```
$vim indxip.txt
```

Inside the file-> Remove the existing IP Address and Enter:

```
10.0.0.16
```

Save the file and exit.

21) On the Command prompt Run the build-script using the ant command.

```
$ ant
```

22) The Build-script is executed and it cleans and compiles the files of the Peer_3 present in the Peer_3 directory. After successful build, a message is displayed on the terminal:

```
Build Successful
Total Time: 1 second
```

23) Run Peer_3 using the command:

```
$ java PeerServer
```

Then once it is run, select 3rd option from the MENU which is for Downloading from the Peer_1 which acts as the PeerServer.

The Peer_3 has to then:
Enter the Peer ID: (The Peer ID of the Peer which contains the file "my.txt" has to be entered-This can be obtained by the search result)
Example: 3001

Enter the Peer IP Address to Download the file:
Example: 10.0.0.13

Enter the Filename to be downloaded:
Example: my.txt

The Peer_1 Receives a connection request for Download from Peer_3. Then the Peer_3 is Connected to PeerServer with the peerid: 3001 and IP Address (10.0.0.13)
And the requested file is downloaded.
my.txt: Downloaded.

===

SECTION B:
b) Running the project on Multiple Machines.

1) Launch the Terminal and Navigate into the P2P_File_Sharing_System folder from the above zip.

2) Let the Machine say Machine-A act as the CentralIndexServer. From the zip, Copy the CentralIndxServer folder into Machine-A.

3) Copy the Folder Peer_1 into a different Machine say Machine-B.

   Similarly, Copy the folder Peer_2 into a different Machine say Machine-C and finally Copy the folder Peer_3 into different Machine say Machine-D.

   [Here Machine-A will run the CentralIndexServer, Machine-B will run the Peer_1, Machine-C will run the Peer_2 and Machine-D will run the Peer_3].

4) In Machine-A, navigate into the CentralIndxServer folder from the Terminal.

5) On the Command prompt Run the build-script using the ant command.

```
$ ant
```

6) The Build-script is executed and it cleans and compiles the files of the CentralIndxServer present in the CentralIndxServer directory. After successful build, a message is displayed on the terminal:

```
Build Successful
Total Time: 1 second
```

7) Run the CentralIndxServer by the command:

```
$ java CentralIndxServer
```

A message is displayed saying the “<Central Index Server is Up and Running>”
This runs the CentralIndxServer and it starts listening for incoming connections from other Peers.

8) Now on Machine-B, where Peer_1 folder is copied, navigate into Peer_1 folder from the Terminal.

9) Edit the file **“indxip.txt”** present inside Peer_1 folder using “vim” or “gedit”. Edit and Enter the **IP Address** of the Machine which is running the **CentralIndxServer** into this file. The IP Address of the Machine running CentralIndxServer can be obtained by running the “**ifconfig**” command on the CentralIndxServer Machine.
Example: If the IP Address of the CentralIndxServer Machine is: 10.0.0.16

```
$vim indxip.txt
```

Inside the file-> Remove the existing IP Address and Enter:

```
10.0.0.16
```

Save the file and exit.

10) On the Command prompt Run the build-script using the ant command.

```
$ ant
```

11) The Build-script is executed and it cleans and compiles the files of the Peer_1 present in the Peer_1 directory. After successful build, a message is displayed on the terminal:

```
Build Successful
Total Time: 1 second
```

12) Run Peer_1 using the command:

```
$ java PeerServer
```

The Peer_1 is run and a MENU is displayed on the screen with the following options for the Peer_1 to select:

```
 Enter The Option:
==================
 1. Registering the File

 2. Searching On CentralIndxServer

 3. Downloading From Peer Server

 4. Exit
```

================**Registration Procedure**=======================

13)If the Peer_1 wants to Register the file, it can choose option 1.
The Peer_1 then has to:
Enter the String in Format: 4Digit id and File Names separated by Space
Example: 3001 my.txt

Then the Peer_1 is Connected to Register on CentralIndxServer on port 2001
And the file “my.txt” is “Registered Successfully” on the CentralIndxServer.
The Peer_1 can also register multiple files by specifying the ID and filenames separated by a space
Example: 4001 A.txt B.txt C.txt D.txt……..and so on

===================**Search Procedure**===========================

14) Now on Machine-C, where Peer_2 folder is copied, navigate into the Peer_2 folder from the Terminal:

15) Edit the file **“indxip.txt”** present inside Peer_2 folder using “vim” or “gedit”. Edit and Enter the **IP Address** of the Machine which is running the **CentralIndxServer** into this file. The IP Address of the Machine running CentralIndxServer can be obtained by running the “**ifconfig**” command on the CentralIndxServer Machine.
Example: If the IP Address of the CentralIndxServer Machine is: 10.0.0.16
$vim indxip.txt
Inside the file-> Remove the existing IP Address and Enter:
10.0.0.16
Save the file and exit.

16) On the Command prompt Run the build-script using the ant command.

```
$ ant
```

17) The Build-script is executed and it cleans and compiles the files of the Peer_2 present in the Peer_2 directory. After successful build, a message is displayed on the terminal:

```
Build Successful
Total Time: 1 second
```

18) Run Peer_2 using the command:

```
$ java PeerServer
```

19) The Peer_2 will have the option to Register, Search or Download the File.
If the Peer_2 wants to Search the file “my.txt”, it can Search on the CentralIndxServer to find out which Peer has the required file.

Then once Peer_2 is run, select 2 which is the Search option from the MENU to search for the desired file.
Then Peer_2 has to:
Enter the filename to search:
Example: my.txt

Then the Peer_2 is Connected to Search on CentralIndxServer on port 2002

And the Search result is returned from the CentralIndxServer:
File:'my.txt' found at peers: 3001 (10.0.0.13).

Here 3001 is the Peer ID and “10.0.0.13” is the IP address of the Peer holding the File “my.txt”. In this case it is Peer_1 which has the file “my.txt”.

=====================**Download Procedure**=======================

20) Now on Machine-D, where Peer_3 folder is copied, navigate into the Peer_3 folder from the Terminal.

21) Edit the file **“indxip.txt”** present inside Peer_3 folder using “vim” or “gedit”. Edit and Enter the **IP Address** of the Machine which is running the **CentralIndxServer** into this file. The IP Address of the Machine running CentralIndxServer can be obtained by running the “**ifconfig**” command on the CentralIndxServer Machine.
Example: If the IP Address of the CentralIndxServer Machine is: 10.0.0.16

```
$vim indxip.txt
```

Inside the file-> Remove the existing IP Address and Enter:

```
10.0.0.16
```

Save the file and exit.

22) On the Command prompt Run the build-script using the ant command.

```
$ ant
```

23) The Build-script is executed and it cleans and compiles the files of the Peer_3 present in the Peer_3 directory. After successful build, a message is displayed on the terminal:

```
Build Successful
Total Time: 1 second
```

24) Run Peer_3 using the command:

```
$ java PeerServer
```

The Peer_3 will also have the option to Register, Search or Download the File.

If Peer_3 wants to download the file “my.txt” which is registered by Peer_1, then it can first Search for the Peer which contains the File “my.txt” (in this case Peer_1). The Search result returns [“File:'my.txt' found at peers:3001(10.0.0.13)]- where (10.0.0.13)- is the IP Address of the peer which has the file “my.txt” i.e Peer_1

Then once the Peer_3 is run, select option 3 from the MENU which is for Downloading from Peer_1 which acts as the PeerServer.

The Peer_3 has to then:
Enter the Peer ID: (The Peer ID of the Peer which contains the file “my.txt” has to be entered-This can be obtained by the search result)
Example: 3001

Enter the Peer IP Address to Download the file:
Example: 10.0.0.13

Enter the Filename to be downloaded:
Example: my.txt

The Peer_1 Receives a connection request for Download from Peer_3. Then the Peer_3 is Connected to PeerServer with the peerid: 3001 and IP Address (10.0.0.13)
And the requested file is downloaded.
my.txt: Downloaded.

===

<u>Note</u>:

- In Each of the Peer Folders, there is a file “indxip.txt” – Edit this file, using “vim” or “gedit” or any other editor and replace the existing IP Address in the file with the IP Address of the CentralIndxServer.
  The IP Address of the CentralIndxServer should be entered into this file, before running the Peers.

- For Registration Process, All the files that needs to be registered should be placed in the same folder of the Peer. Example: If Peer_1 needs to register files, say “my.txt” and “testfile.txt” etc. All these files should be placed in the Peer_1 directory before Registering the files.

- Similarly, all the Downloaded files, will be available in the Peer directory of the Downloader Peer. Example: If Peer_3 wants to download a registered file “my.txt” from Peer_1, it selects 3rd option from the MENU which is to download from other Peers. Then it has to enter the Peer ID of the peer which contains the File “my.txt”, IP Address of the Peer holding the file and Filename to be downloaded. Once this is done, the file “my.txt” will be downloaded into the folder of Peer_3(the Downloader) from Peer_1.

- For Testing purposes, 10 testfiles have been placed inside each Peer Folder.
  Example: a) Peer_1 folder has 12 testfiles -> 1.txt upto 10.txt , my.txt and testfile.txt
  b) Peer_2 folder has 10 testfiles -> 11.txt upto 20.txt

c) Peer_3 folder has 10 testfiles -> 21.txt upto 30.txt

These files can be used to test for Register, Search and Download functions from different Peers. In addition, other files may also be added to the different Peer Folders and all the above scenarios can be tested.

==