Adrian Birylo
Daniel Hitchings
CS 451 Project 3
Due Date: 4-8-2014

Manual

Preface

Below are the manual test ran on the program to test the validity of the peer to peer program.

Tests

1. Start client before server running.
   Expected
   Error - Index Server not bound. Please start index server before launching client
   Result
   Error - Index Server not bound. Please start index server before launching client
   Pass: True

2. Start server and check if running.
   Expected
   Server is running.
   To exit type 'exit'.
   Result
   Server is running.
   To exit type 'exit'.
   Pass: True

3. Start another instance of the server if one is already running
   Expected
   Error -Server is already bound.
   Result
   Error -Server is already bound.
   Pass: True

4. Start client with no id as argument
   Expected
   Please enter a peer id as a command line argument
   Result
   Please enter a peer id as a command line argument
   Pass: True

5. Start client with a unique id of 1 after server is running.
   Expected
   Client running... PeerID = 1
   Options:
   1 - Search for filename
   2 - Obtain filename from peer

3 - List files in shared directory
4 - Update registry
5 - Exit
Result
Client running... PeerID = 1
Options:
1 - Search for filename
2 - Obtain filename from peer
3 - List files in shared directory
4 - Update registry
5 - Exit
Pass: True

6. Start another client with an id of 1 after another client of id 1 is running.
   Expected
   Error = Peer 1 is already bound. Please choose a different peer id or restart rmiregistry.
   Result
   Error = Peer 1 is already bound. Please choose a different peer id or restart rmiregistry.
   Pass: True

7. Check that client registers file with server. Using files Text1.txt, Text2.txt, and Text2.txt in folder Client1. Only client 1 is running. Then run command 1 and search for Text1.txt, Text2.txt, Text3.txt, and Text4.txt.
   Expected
   For Text1.txt, Text2.txt and Text3.txt the result should be:
   The clients that have the file Text#.txt are:
   Client 1
   For Text4.txt the result should be:
   There are no peers sharing Text4.txt.
   Result
   For Text1.txt, Text2.txt and Text3.txt the result should be:
   The clients that have the file Text#.txt are:
   Client 1
   For Text4.txt the result should be:
   Pass: True

8. Check that if two clients have the same file that both show up. With client 1 running as in the above test case run client 2 with Text1.txt in the folder Client2. Then run the search command for Text1.txt
   Expected
   The clients that have the file Text1.txt are:
   Client 1
   Client 2
   Result
   The clients that have the file Text1.txt are:
   Client 1
   Client 2
   Pass: True

9. Attempt to get a file that does not exist.

Expected
Enter filename: Text11.txt
Enter peer. If no preference enter 0.
0
The file Text11.txt does not exist.
Result
Enter filename: Text11.txt
Enter peer. If no preference enter 0.
0
The file Text11.txt does not exist.
Pass: True

10. Obtain file and check if exists. Client1 directory has Text1.txt and Client2 directory does not have Text1.txt. Then in client 2 run the command to get Text1.txt.
Expected
Enter filename: Text1.txt
Enter peer. If no preference enter 0.
0
Then Client2 directory has Text1.txt.
Result
Enter filename: Text1.txt
Enter peer. If no preference enter 0.
0
Then Client2 directory has Text1.txt.
Pass: True

11. Add new files to the shared directory for the client and run update registry command and check if registry is update. **Expected**
Run client.
Add new files to client.
Update registry
Search for file to see if the registry update.
Result
Run client.
Add new files to client.
Update registry
Search for file to see if the registry update.
Pass: True