**Authors:**

\* CSci5105 Spring 2015

\* Assignment# 7

\* name: <Ravali Kandur>, <Charandeep Parisineti>

\* student id: <5084769>, <5103173>

\* x500 id: <kandu009>, <paris102>

\* CSELABS machine: In README

**Design Description:**

1. ChordInterface.java -> Interface which will denote the prototype for each node in the CHORD ring. Instance of this should be exported to rmi.
2. Node.java -> Implementation of ChordInterface.java which will have all the functionality of a node in CHORD ring.
3. Client.java -> Client side of the implementation which has an interface to interact and perform some operations (mentioned below) on the server.
4. NodeInfo.java -> Internal class which holds the identification details of a node like url, id and number (which is user friendly name of the node).
5. Utils.java -> Utility class which has helper methods for hashing using SHA1.
6. FindNodeResponse.java -> Class which represents the typical response structure when a find\_node query is made by the client.
7. JoinResponse.java -> Class which represents the typical response structure received from a server when a join request is made by a new node.
8. ClientLogger.java -> Helper class to initialize and handle the client log file
9. ServerLogger.java -> Helper class to initialize and handle the server log file

**Compilation Files:**

1. Use Makefile : go to directory assignment7 and type ‘make’ and ‘enter’
2. Use direct compilation : go to directory assignment7 and type ‘javac \*.java’

**How to start the nodes:**

1. **Same Host:**

This sample case uses 5 nodes. This folder contains a sample script with name **script\_same\_hosts.txt** for more instructions.

1. **Different Host**:

This sample case uses 5 nodes. This folder contains a sample script with name **script\_different\_hosts.txt** for more instructions.

**Expected Log Files:**

1. Client Logs -> clientLogfile
2. Server logs -> serverLogfile<nodeId> as mentioned in the URL

**Test Cases verified:**

1. Lookup for a word which does not exist in DHT
2. Lookup a word which exists in the DHT
3. Insert a word which is not already existing
4. Insert a word which already exists. This replaces the word meaning with new one
5. Lookup for a word which will be redistributed to a new node after new servers join.
6. When one node is joining the ring, others will get a message saying server is busy
7. No other node other than node0 will serve join/find\_node/join\_done kind of requests
8. Tested cases with and without log Trace flag set while sending requests to server
9. Tested with as low as one node in the ring to as big as 8 nodes in the ring.
10. Tested #9 on same hosts and different hosts
11. Tested logging functionality

**Client Operations:**

1. Print ring structure: prints current ring structure

2. Print ring structure without fingers: since I noticed that checking tree structure with finger table printed is difficult, I have added this. This prints the same as 1, except the finger table

3. Lookup word without log trace: looks up a word in the DHT and does not give you a trace

4. Lookup with log trace: looks up a word in the DHT and also gives you a trace

5. Insert with log trace: inserts a word in the DHT and also gives you a trace

6. Insert without log trace: inserts a word in the DHT and does not give you a trace

7. Exit: exit the client

**Things to keep in mind before running:**

1. Intentionally did this for easy testing, should give all rmi ports. I chose 50000 as example.

2. Node URL should be given as hostname/node\_number. Implementation relies on the hostname provided here in the URL

3. Client file will be named clientLogfile

4. Server files will be named as serverLogfile<node\_number>. For easy to read purpose, I am expecting all the nodes to have different names even if they are on different hosts

5. I intentionally chose a combination of same hosts with different node\_number and different hosts with different node\_number. We need to have a different node\_number even if we run on different hosts as we want the log files to be easy to interpret

6. In case you want to end a certain server, make sure you restart all the other servers accordingly as in this assignment we are not dealing with a node leaving the ring

7. In case you an exception as shown below. This means, rmiregistry has not been cleaned up yet. Please do all the steps to initate the server again and this time make sure rmiregistry is cleaned up and started afresh

java.rmi.ServerException: RemoteException occurred in server thread; nested exception is:

java.rmi.UnmarshalException: error unmarshalling arguments; nested exception is:

java.lang.ClassNotFoundException: ChordInterface