OS-PG ASSIGNMENT 6
Distributed Hash Table
Deadline: 10 November 11:55 PM

A **distributed hash table** (**DHT**) is a class of a decentralized distributed system that provides a lookup service similar to a hash table: (*key*, *value*) pairs are stored in a DHT, and any participating node can efficiently retrieve the value associated with a given key.

In this assignment, you are supposed to implement DHT which supports following major operations:
1) Join a new Node in the network.
2) Set the <key,value> pair
3) Get the value of key
4) Delete a node from Network [Bonus Question]

Your design of above mentioned functions should conform to following paper.
http://research.microsoft.com/en-us/um/people/antr/PAST/pastry.pdf

In this assignment your main program should create a server which is threaded. Server itself is threaded from main program and it creates new thread for every request. Server should be able to handle multiple requests at a time.[Similar to assignment-2 but you don't need to fork here, you have to create a new thread].

You have to build a CLI which should support following commands.

1) port <x>
Listen on this port for other instances of the program over different nodes.

2)create
creates the pastry henceforth will be known by this node’s address and decided port.

3) join <x> <p>
Join the pastry with x address and port p.

4) put <key> <value>
Insert the given <key,value> pair in the pastry.

5) get <key>
It returns the value corresponding to the key, if one was previously inserted in the node.

6) lset
Prints the leafset of current node.

7) routetable:
Prints the routing table of current node.

8)nset:
Prints the neighbourhood set of current node.

[Below 2 part of bonus question]

9)quit:
Shuts down this node, not the pastry, distributing the data.

10)shutdown:
shuts down the entire pastry, no node should have any keys or pastry data, the programs at all the terminals should get closed on the notification.

Typical scenario to run your code:

1) Assign port to the process using port command like "port 3000"

2) Then run "create" command to start server in the process
Now this process is ready

3) Now open new terminal tab

4) Again run "./pastrydht" assign different port to this process e.g. "port 3001"

5) Start server using "create" command

6) Then join this new process to the process created earlier using the command "join <ur_machine's_ip> 3000"
Now these two processes are ready to communicate

Now go to any of the process and run commands like "put a 1", "put b 2", "put c 3", "put d 4" and so on...These will create new key-val pairs in pastry network

Now to get the value of key run command like “get a”

Instructions:

1) Use threads carefully, make sure you use locks wherever required.

2) You can use C++ STL.

3) Code and data will be quite big, make sure you use structs and classes to handle them. Write your code in multiple files.

4) Copying code from friends/internet/seniors will lead to straight zero. No arguments will be entertained later for copy cases .

5) **10th November 2016** is the strict deadline for this assignment.