

Join	message
4	18
8	30
10	32
20	34
30	44

find	message
4	50
8	70
10	72
20	80
30	85

Description of basic process and algorithm:

The way we implemented the chord was through message passing among different threads. Each node on the chord has a separate thread. Whenever it joins, it calls function RPC to inform other nodes which operation needs to be executed. The synchronization is guaranteed by using conditional lock to let the caller thread wait until other threads finish. In order to implement join, we ask the node 0 to find the closest predecessor of the joining node and then update the successor and predecessor. Then other nodes' finger tables are updated. Then the key are updated. To remove one node, we essentially reverse the joining process. In order to find one node, we traverse the chord using fingers in finger table to find the node closest to the wanting node to make the running time to be $\log(n)$

Instruction for compiling and running the code:

Compile: use the make file in the folder to compile

Running: Type in the command join p, find p k, leave p, show p and show all according to the specification to run different instructions.