

Programming Assignment #2

COEN241 Introduction to Cloud Computing
Department of Computer Engineering
Santa Clara University

Dr. Ming-Hwa Wang
Phone: (408) 525-2564
Course website:
Office Hours:

Summer Quarter 2014
Email address: m1wang@scu.edu
<http://www.cse.scu.edu/~mwang2/cloud/>
Tuesday & Thursday 9:00-9:30pm

Due date: July 20, 2014

Node Lookup in P2P Networks (200 points)

Please implement a centralized version of the distributed node lookup program in peer-to-peer (p2p) file sharing network using the Chord algorithm.

Your program should be able to find any file at any remote computer. Each computer (or node) has a unique name (a structure/record contains its IP address and port id), and each item has a name in ASCII and unique in the node it resides. Your program should support the following operations: insert an item to a node, delete an item from a node, add a node to the p2p network, remove a node from the p2p network, and print the information. To simplify your work, you can use the following hash function instead of SHA-1, omit the port id, and you can simply use the 4-byte IP address as the node hashing.

The input to the hash function is a text string, and the output of the hash function is an IP address (or equivalently, a 32-bit unsigned integer.) The input string is divided into 4-byte chunks (fill with zeros if the last chunk is not full), reverse the bits for all odd numbered chunks, and return the exclusive OR of all those chunks.

For example, the input is "Listen to the music", which is divided into 5 chunks: "List", "en t", "o th", "e mu", and "sic". Those 5 chunks are represented in hex as 0x4C697374, 0x656E2074, 0x6F207468, 0x65206D75, and 0x73696300. Then reverse the bits of all odd chunks and return the exclusive OR of all chunks: 0x2ECE9632 ^ 0x656E2074 ^ 0x162E04F6 ^ 0x65206D75 ^ 0x00C696CE.

The syntax of the commands are:

```
addNode(<nodeIP>)  
removeNode(<nodeIP>)
```

```
insertItem(<myNodeIP>, <itemName>)  
deleteItem(<myNodeIP>, <itemName>)  
find(<myNodeIP>, <itemName>)
```

To make Autotest working, your program reads in commands (any of the above function calls) and executes the commands one after another. For addNode, removeNode, insertItem, and deleteItem, display success for fail; and for find, display the sequence of the IP addresses it traversed.

Student Name:

SSN/ID:

Score:

Correctness and boundary condition (60%):

Whitespace and free format compliance (5%):

Compiling without warning/error (5%):

Error Handling (5%):

Modular design, file/directory organizing, showing input, documentation, coding standards (20%):

Automation (5%):

Subtotal:

Late penalty (20% per day):

Special service penalty (5%):

Total score: