Data Structures

Distributed Hash Table System

Submitted to:

Ma'am Hira Naveed

Submitted by:

Fahad Waheed

20I-0651

BS-CSDF

Contents

Documentation

Project's program has following classes, structures, and functions:

Singly Linked List (to store actual data)

- Rule of 3 (copy constructor, copy assignment operator, destructor) applied as we're dealing with dynamic memory.
- Proper checks implemented to doesn't store the same data (duplicates).

AVL Tree (to store linked list)

- Rule of 3 (copy constructor, copy assignment operator, destructor) applied as we're dealing with dynamic memory, and it will be used in merge and split of AVL.
- Proper checks and Rotations implemented to balance the Tree every time after a node is inserted or deleted.
- AVL insertion is overloaded for both key value pair and key list pair.
- Similarly, AVL deletion is overloaded for both key value pair and key list pair.
- To search in AVL Tree a recursive function is provided.
- All the exception are properly handled.

Merge:

To merge two AVL Trees into one. Firstly, we copied first AVL into our main AVL using copy assignment operator and then one by one traversed through the nodes of second AVL and inserted every node into our main AVL.

Split:

To split an AVL Tree into two. We traversed through the nodes of main AVL and inserted every node which is less than or equal to key into first AVL and inserted every node which is greater than key into second AVL.

Routing Table (to store references of machines)

- Insert function to insert references of machines at the end of doubly linked list.
- Get machine function to get machine of a specified index.
- Display function to display the routing table of a machine (full Doubly Linked List)
- Clear function to clear all the data in the routing table.

RingDHT (A Singly Circular linked list to store all the machines)

- When machines are created automatically, they are created in such a way that the load is divided equally among the number of machines.
- When machines are created manually it is implemented with proper checks a machine with the same hashed id is not repeated.

Insertion:

All the nodes in the RingDHT are inserted in ascending order according to their hash id's.

Successor:

The function is given a hash and returns the machine that is next to machine (given hashed id). We first checked if hash is less than or equal to hashed id of machine with minimum hashed id or greater than machine with maximum hashed id, it is then returned the machine with minimum hashed id. Otherwise, we will return the first machine whose hashed id is greater than or equal to hash.

Routing:

Provides us the path that maps to the actual machine on which data is to be inserted, searched, or deleted. All the cases are handled (hash == hashed id of machine, hash < hashed id of machine, and hash > hashed id of machine).

New Machine:

When a new machine joins it divides the load of next machine (splits AVL of next machine).

Delete Machine:

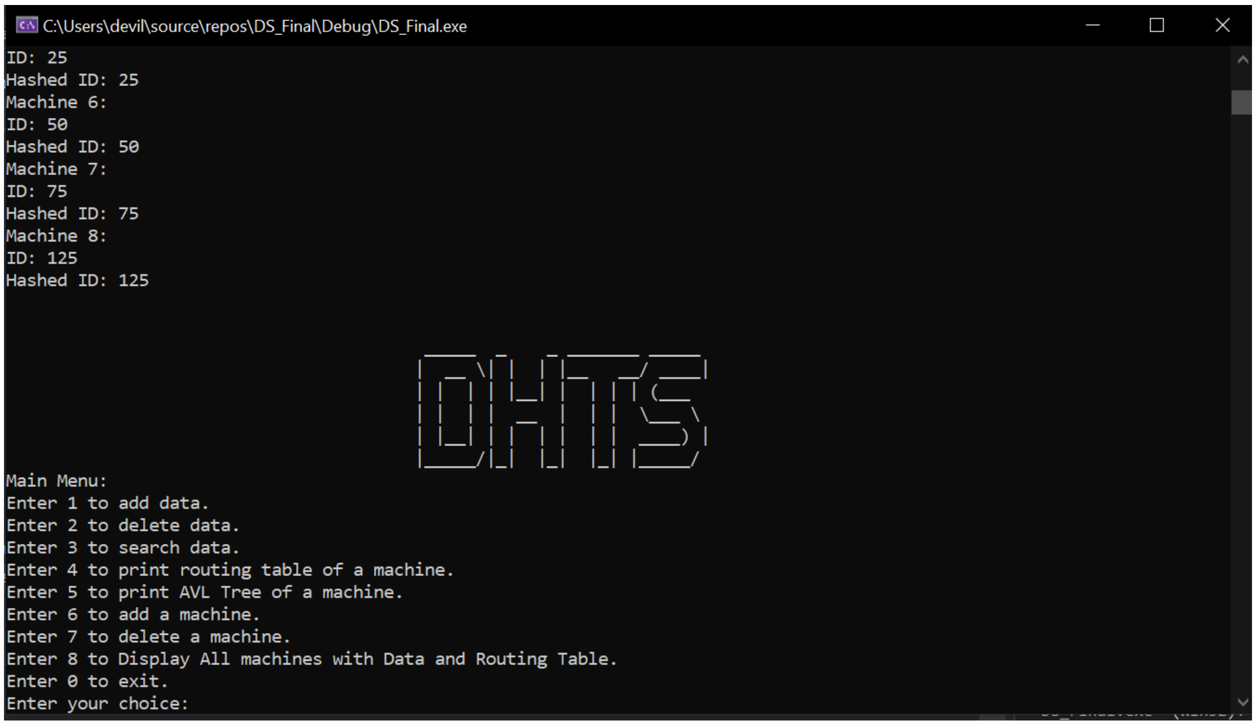
When a machine leaves it increases the load of next machine (merges AVL with AVL of next machine)

- Functions to clear and update routing table of machines are defined according to the given formula.
- Functions to insert data, search data, and remove data are implemented.
- Clear function removes all the machines from the circular linked list.

Workflow

First of all, the main function takes input for identifier space and number of machines, then calls the default constructor of the RingDHT to get the program started. Then create machines according to user choice and sets the routing table of machines. Then main function displays the menu (given below) and asks the user if he wants to add data, search data, delete data, print AVL of a machine, print Routing Table of a machine, add a new machine, delete a machine, and to display all machines along with AVL and Routing Table, and calls the function accordingly.

Snapshot of Main Menu



```
C:\Users\devil\source\repos\DS_Final\Debug\DS_Final.exe
ID: 25
Hashed ID: 25
Machine 6:
ID: 50
Hashed ID: 50
Machine 7:
ID: 75
Hashed ID: 75
Machine 8:
ID: 125
Hashed ID: 125


                                         _____  _    _ _______ _____
                                        |  __ \| |  | |__   __/ ____|
                                        | |  | | |__| |  | | | (___
                                        | |  | |  __  |  | |  \___ \
                                        | |__| | |  | |  | |  ____) |
                                        |_____/|_|  |_|  |_| |_____/

Main Menu:
Enter 1 to add data.
Enter 2 to delete data.
Enter 3 to search data.
Enter 4 to print routing table of a machine.
Enter 5 to print AVL Tree of a machine.
Enter 6 to add a machine.
Enter 7 to delete a machine.
Enter 8 to Display All machines with Data and Routing Table.
Enter 0 to exit.
Enter your choice:
```