National University
Of Computer and Emerging Sciences

DS Project Report

In partial fulfillment
of the requirement for the course of

CS2001-Data Structures

By

I22-0942	**Muhammad Umar Hassan**	**CS-H**
I22-1234	**Muhammad Raza**	**CS-H**
I22-1220	**Muhammad Huzaifa**	**CS-H**

Table of Contents

Contents

Enhanced IPFS Configuration and Management System

- **Flexible Configuration:** Tailor the IPFS to your needs by specifying the number of machines, the B-tree order, and the size for an identifier space.

- **Customized ID Assignment:** Take control of the ID assignment process by choosing to assign IDs manually or automatically to the machines according to your preferences.

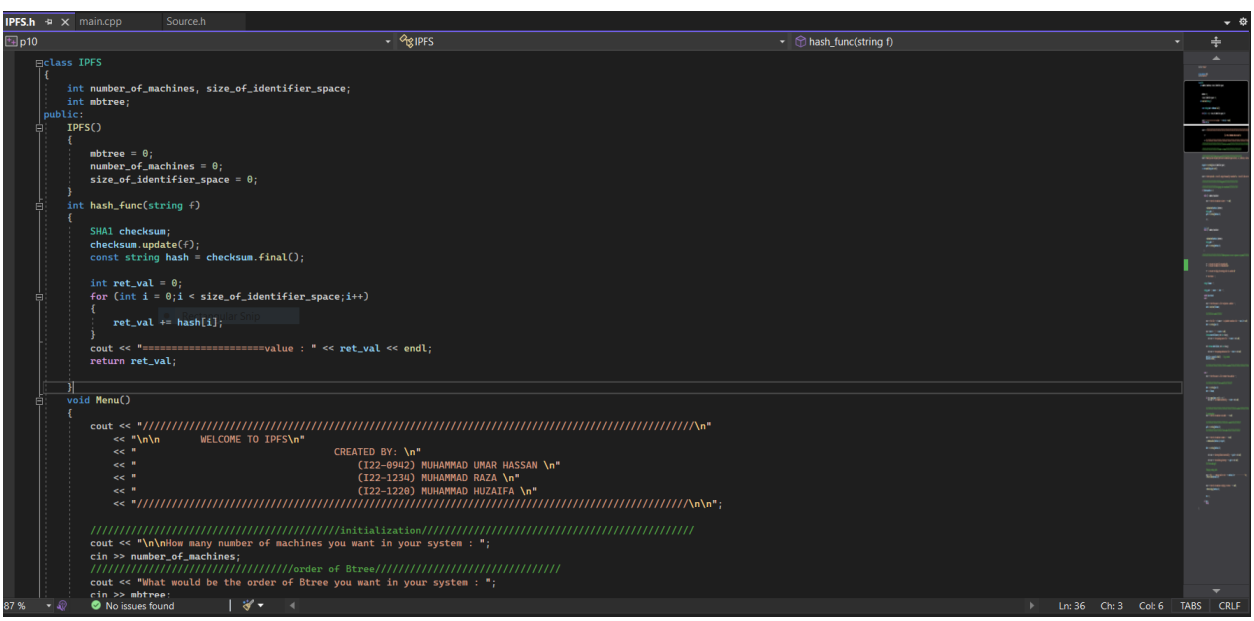
- **User-Friendly Interface:** Once the machines are set up, access a user-friendly menu to seamlessly perform various operations, including file uploading, file removal, addition of a new active machine, removal of a machine, as well as viewing the routing table and B-tree of a machine.

Class Structure for Enhanced IPFS

1. `IPFS`:

Its member functions includes

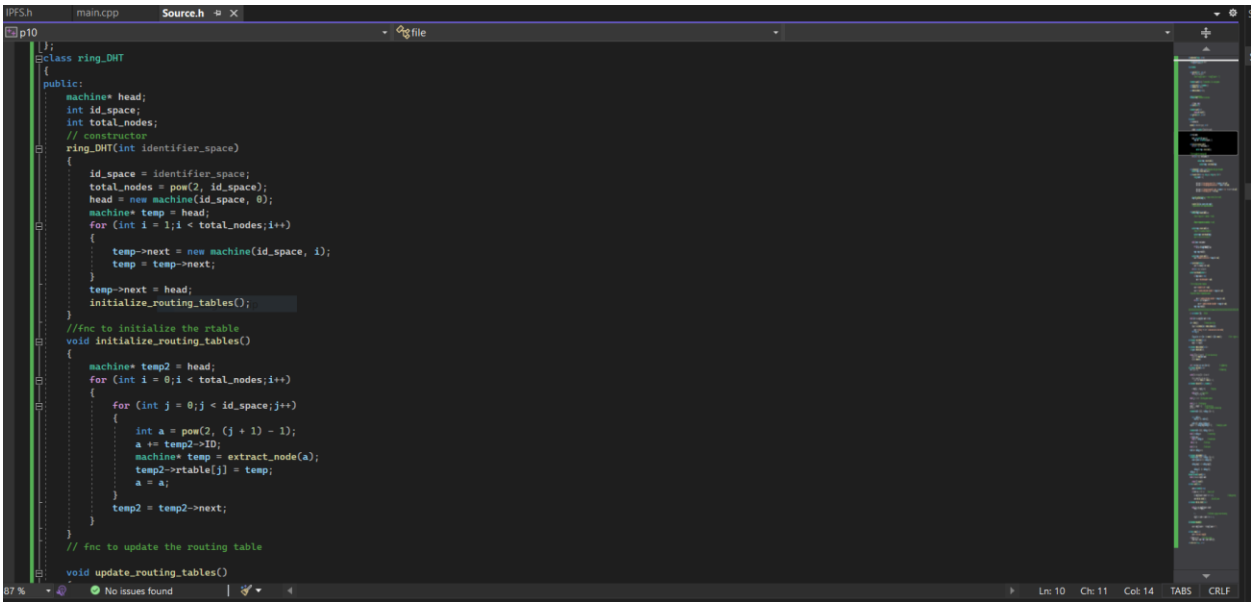
- The default address for IPFS id "C:/IPFS_of_bit_(identifier_size)".
- `int hash_func(string f) : Utilizes the filename to generate a SHA-1 hash value, which is then converted to an integer hash based on the identifier size`
- `void Menu() : Encompassing the IPFS menu, this feature empowers users to define the identifier space size, B-tree order, number of machines, and various other configuration options. It additionally manages the creation of directories for all machines.`



2. `ring_DHT`**:**

Its member functions includes

- ring_DHT(int identifier_space) : `A constructor for 'ring_DHT' that initializes all its members, including the 'identifier_space'.`
- `void initialize_routing_tables() : where the routing table for all machines is initialized`
- `void update_routing_tables() : Where the routing tables of machines are updated after addition and removal of a machine.`
- `void add_machine(int number, int m) : that adds the new active machine in the 'ring_DHT' and then calls update_routing_tables().`
- `void transfer_files(int size, string src, string dest, file* f) : a helper function for file handling when the machine is removed from 'ring_DHT'.`
- `void remove_machine(int number, string de_path) : that removes the active machine in the 'ring_DHT' and transfer the Btree of that machine to next available machine. Then it calls update_routing_tables().`
- `int insert_file(file f, int hash) : which is responsible for adding a file in the B-tree of files on a machine.`
- `int delete_file(int hash) : which is responsible for removing a file from the B-tree of files on a machine.`
- `void Print_rtable(int ID) : Which is responsible for printing the routing table of machine.`
- `void btree_display(int ID) : Which is responsible for printing the B-tree-traversal of machine.`
- and some helper functions……



```
};
class ring_DHT
{
public:
    machine* head;
    int id_space;
    int total_nodes;
    // constructor
    ring_DHT(int identifier_space)
    {
        id_space = identifier_space;
        total_nodes = pow(2, id_space);
        head = new machine(id_space, 0);
        machine* temp = head;
        for (int i = 1;i < total_nodes;i++)
        {
            temp->next = new machine(id_space, i);
            temp = temp->next;
        }
        temp->next = head;
        initialize_routing_tables();
    }
    //fnc to initialize the rtable
    void initialize_routing_tables()
    {
        machine* temp2 = head;
        for (int i = 0;i < total_nodes;i++)
        {
            for (int j = 0;j < id_space;j++)
            {
                int a = pow(2, (j + 1) - 1);
                a += temp2->ID;
                machine* temp = extract_node(a);
                temp2->rtable[j] = temp;
                a = a;
            }
            temp2 = temp2->next;
        }
    }
    // fnc to update the routing table

    void update_routing_tables()
```

3. `machine:`

Its members are

- `int ID;  // ID of machine`
- `machine* next;  // next responsible machine`
- `int Successor_ID;  // successor's ID`
- `machine** rtable;  // Routing table`
- `Btree b;  // Its B-tree`
- `bool status;  // Active or inactive status`
- `int id_space;  // identifier_space`

```
IPFS.h    main.cpp    Source.h
p10                                   file

class machine
{
public:
    int ID;
    machine* next;
    int Successor_ID;
    machine** rtable;
    Btree b;
    bool status;
    int id_space;

    machine(int identifier_space, int id)
    {
        id_space = identifier_space;
        status = false;
        ID = id;
        next = NULL;
        Successor_ID = -1;
        rtable = new machine * [identifier_space];
    }
};
class ring_DHT
{
```

4. `Btree:`

Its member functions are

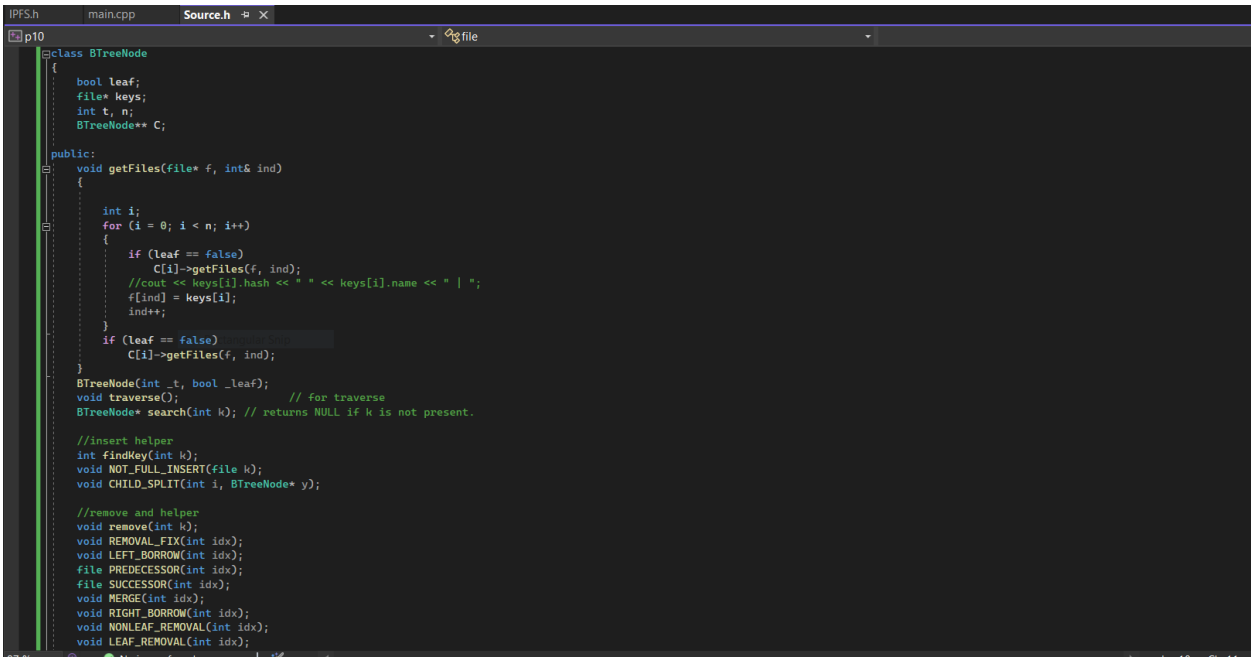
- `void traverse() : Which is responsible for B-tree-traversal.`
- `void insert(file k) : Which is responsible for B-tree insertion of file.`
- `void remove(int k) : Which is responsible for B-tree deletion of file.`
- and some helper functions

```
IPFS.h    main.cpp    Source.h
p10                                   file

class Btree
{
public:
    int t; // Minimum degree
    BTreeNode* root; // Pointer to root node
    int total_files;

    Btree()
    {
        root = NULL;
        t = -1;
        total_files = 0;
    }
    void traverse()
    {
        if (root != NULL)
            root->traverse();
    }
    void insert(file k);
    Btree(int _t)
    {
        t = _t;
        root = NULL;
        total_files = 0;
    }
    BTreeNode* search(int k)
    {
        if (root == NULL)
            return NULL;
        else
            return root->search(k);
    }
    void remove(int k);

    void getFiles(file* f, int ind)
    {
        if (root != nullptr)
            root->getFiles(f, ind);
    }

};
```

5. BTreeNode:

Its member functions are

- `void traverse() : for traverse of keys inside Node.`
- `void NOT_FULL_INSERT(file k) : for insertion when array of keys is not full.`
- `void remove(int k) : for removing keys from its array and calling its helper functions based on B-tree cases.`
- And some helper functions of insert and delete of file.



```
IPFS.h    main.cpp    Source.h
p10                   file
class BTreeNode
{
    bool leaf;
    file* keys;
    int t, n;
    BTreeNode** C;

public:
    void getFiles(file* f, int& ind)
    {

        int i;
        for (i = 0; i < n; i++)
        {
            if (leaf == false)
                C[i]->getFiles(f, ind);
            //cout << keys[i].hash << " " << keys[i].name << " | ";
            f[ind] = keys[i];
            ind++;
        }
        if (leaf == false)
            C[i]->getFiles(f, ind);
    }
    BTreeNode(int _t, bool _leaf);
    void traverse();                // for traverse
    BTreeNode* search(int k); // returns NULL if k is not present.

    //insert helper
    int findKey(int k);
    void NOT_FULL_INSERT(file k);
    void CHILD_SPLIT(int i, BTreeNode* y);

    //remove and helper
    void remove(int k);
    void REMOVAL_FIX(int idx);
    void LEFT_BORROW(int idx);
    file PREDECESSOR(int idx);
    file SUCCESSOR(int idx);
    void MERGE(int idx);
    void RIGHT_BORROW(int idx);
    void NONLEAF_REMOVAL(int idx);
    void LEAF_REMOVAL(int idx);
```

6. File:

Its members are

- `string name;`
- `int hash;`

IPFS.h main.cpp Source.h

p10 file

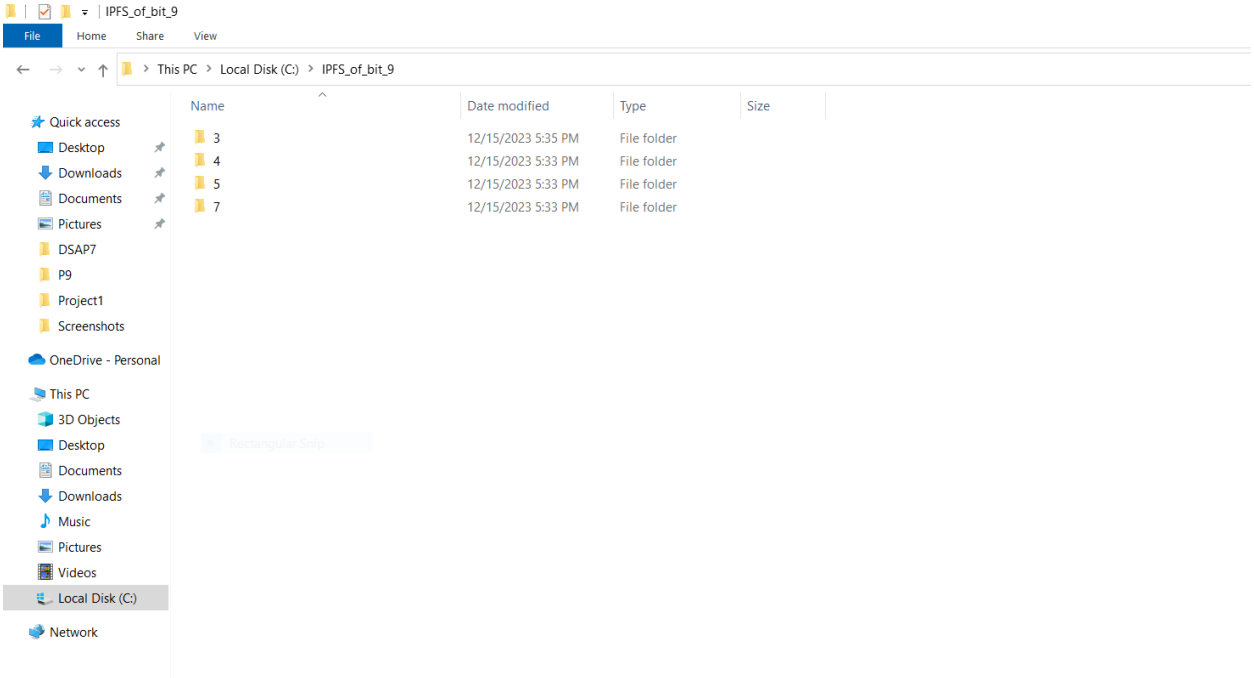
```
#pragma once


class machine;
int hashFunction(string s, int m);
class file
{
public:
    string name;
    int hash;
    file() { name = " ";hash = 0; }
    file& operator=(const file& other)
    {
        if (this != &other) {
            hash = other.hash;
            name = other.name;
        }
        return *this;
    }
};
//Btree Node
```

USER Interface

```
C:\Users\Umar Hassan\Desktop\VISUAL STUDIO\p10\x64\Debug\p10.exe
////////////////////////////////////////////////////////////////////////////////////////////////////


       WELCOME TO IPFS
                                CREATED BY:
                                    (I22-0942) MUHAMMAD UMAR HASSAN
                                    (I22-1234) MUHAMMAD RAZA
                                    (I22-1220) MUHAMMAD HUZAIFA
////////////////////////////////////////////////////////////////////////////////////////////////////



How many number of machines you want in your system : 4
What would be the order of Btree you want in your system : 5
Would you like to specify the size of identifier space in bits, i.e., 160 bits, 4 bits etc : 9
Select option
 1. You will assign IDS manually to machine
  2. You will allow us to assign IDS automatically to machines
 Your choice :  1
Enter ID for machine to insert :
5
Enter ID for machine to insert :
4
Enter ID for machine to insert :
3
Enter ID for machine to insert :
7


Select option
1. You want to insert file to machine
2. You want to remove file from machine
3. You want to add a machine
4. You want to remove a machine
5. You want to display the routing table of a machine
6. You want to display the Btree of a machine
0. Exit


Your choice : 1
Enter the name of a file to upload on a machine : abcd
====================value : 721
The file: abcd is uploaded to machine of ID: 3
C:/IPFS_of_bit_9/3/abcd.txt |  abcd.txt
File copied successfully from abcd.txt to C:/IPFS_of_bit_9/3/abcd.txt


Select option
1. You want to insert file to machine
2. You want to remove file from machine
```

IPFS Folders and Files

1. Folders of machines:



2. Files on machine 3: