INode Management System

- **Problem Statement**:
  - Create your file system which allow the files to be stored on your disk in the form of fixedsize blocks. If you save a file that exceeds a standard block size, your computer will find the next available segment on which to store the rest of your file. Store all the address of the data blocks in master node (like I-node) so that access time can be made faster. Please refer the i-node data structure. The I-node (index node) is a fundamental concept in file systems, used to store metadata about files and directories.
    - Implement the following operations
    - 1. Insertion of the file in data blocks (in Linked List)
    - 2. Create i-nodes for files and directories with the following properties: Each i-node has a unique identifier (e.g., a numeric ID). Store metadata about the file or directory, such as its name, size, permissions, creation date, etc.
    - 3. Retrieve i-node metadata.
    - 4. Update I-node metadata.
    - 5. Delete an i-node.

- **Methodology Followed:** The methodology involves using a linked list to manage Inodes within data blocks, ensuring proper data block creation, and providing options to interact with Inodes in the system i.e., insertion, deletion, update metadata, retrieve metadata. This code allows you to simulate file management similar to how file systems use Inodes to manage files on disk. Also, we used map to make sure that two distinct files do not have identical ID's and also to track the data block allocation like in disk. **Also, the additional features that we added is to find files/Inodes by their date of creation or permission i.e., all files having read only permission or writable etc, And by their unique ID.**
  - **struct Inode**: This is linked list that is used to make I node. One I node contains Data-Block ID, ID (for file), Name of file, Permission of file, date of creation or added.
  - **map<int, Inode*> dataBlockInfo:** This map is tracking the creation of data-blocks. The key is representing the which number of data-block is that and the value according to it is representing the head pointer to the first file/I node that is stored in that particular data-block. Basically, it representing the lined list of I node in that data-block.
  - **map<int, int> is_repeated:** This map is keeping the track of Id of files. And making sure that files do not have same ID.

- **Code:**

```
#include <bits/stdc++.h>

using namespace std;

const int MAX_sizeOfDataBlock = 10;
bool flag = 0;
int sizeOfDataBlock = 0;
int curr_Datablock = 1;
int cnt = 1;
```

```
struct Inode {
  int dataBlockID;
  int id;
  string name;
  int size;
  string permissions;
  string dateOfCreation;
  Inode* next;
};

map<int, Inode*> dataBlockInfo;
map<int, int> is_repeated;

class LinkedList {
public:
  LinkedList() : temp(nullptr), headOfList(nullptr) {}

  void insertInode(int id, const string& name, int size, const string& permissions, const string& dateOfCreation) {
    ++is_repeated[id];
    Inode* newInode = new Inode;
    sizeOfDataBlock += size;
    if (sizeOfDataBlock > MAX_sizeOfDataBlock) {
      sizeOfDataBlock = size;
      ++curr_Datablock;
      temp = nullptr;
      flag = 1;
      cout << "\nThis file exceeds the size of the data-block,\nso this file will be saved in the next data-block.\n";
    }
    newInode->dataBlockID = curr_Datablock;
    newInode->id = id;
    newInode->name = name;
    newInode->size = size;
    newInode->permissions = permissions;
    newInode->dateOfCreation = dateOfCreation;

    if (temp == nullptr) {
      temp = newInode;
      headOfList = newInode;
      dataBlockInfo[curr_Datablock] = temp;
    } else {
      temp->next = newInode;
      temp = newInode;
    }

    newInode->next = nullptr;
  }

  void displayInode() {
    for (auto node : dataBlockInfo) {
      cout << "DataBlock : " << cnt << endl << endl;
      Inode* current = node.second;
      while (current != nullptr) {
        cout << "\tID: " << current->id << ", Name: " << current->name << ", Size: " << current->size
          << ", Permissions: " << current->permissions  << ", Date: " << current->dateOfCreation << endl;
```

```
            current = current->next;
        }
        cnt++;
        cout << endl;
    }
    cnt = 1;
}

void displayInode(int inodeID) {
    bool found = false;
    for (auto node : dataBlockInfo) {
        Inode* current = node.second;
        while (current != nullptr) {
            if (current->id == inodeID) {
                cout << "Data_Block: " << current->dataBlockID << ", ID: " << current->id << ", Name: " << current->name
                    << ", Size: " << current->size << ", Permissions: " << current->permissions << ", Date: "
                    << current->dateOfCreation << endl;
                found = true;
            }
            current = current->next;
        }
    }
    if (!found) {
        cout << "Inode not found with ID " << inodeID << endl;
    }
}

void displayInode(const string& dateOfCreation) {
    cout << "Inodes created on " << dateOfCreation << ":\n";
    bool found = false;
    for (auto node : dataBlockInfo) {
        Inode* current = node.second;
        while (current != nullptr) {
            if (current->dateOfCreation == dateOfCreation) {
                cout << "Data_Block: " << current->dataBlockID << ", ID: " << current->id << ", Name: " << current->name
                    << ", Size: " << current->size << ", Permissions: " << current->permissions << ", Date: "
                    << current->dateOfCreation << endl;
                found = true;
            }
            current = current->next;
        }
    }
    if (!found) {
        cout << "No Inodes found with the creation date " << dateOfCreation << endl;
    }
}

void displayInodeByPermissions(const string& permissions) {
    cout << "Inodes with permissions " << permissions << ":\n";
    bool found = false;
    for (auto node : dataBlockInfo) {
        Inode* current = node.second;
        while (current != nullptr) {
            if (current->permissions == permissions) {
```

```
            cout << "Data_Block: " << current->dataBlockID << ", ID: " << current->id << ", Name: " << current->name
                << ", Size: " << current->size << ", Permissions: " << current->permissions << ", Date: "
                << current->dateOfCreation << endl;
            found = true;
          }
          current = current->next;
        }
      }
      if (!found) {
        cout << "No Inodes found with permissions " << permissions << endl;
      }
    }

    void updatePermissions(int inodeID) {
      for (auto node : dataBlockInfo) {
        Inode* current = node.second;
        while (current != nullptr) {
          if (current->id == inodeID) {
            cout << "Enter new permissions for inode " << inodeID << ": ";
            cin >> current->permissions;
            cout << "Permissions updated." << endl;
            return;
          }
          current = current->next;
        }
      }
      cout << "Inode not found with ID " << inodeID << endl;
    }

    void deleteInode(int inodeID) {
      for (auto& node : dataBlockInfo) {
        Inode* current = node.second;
        Inode* previous = nullptr;

        while (current != nullptr) {
          if (current->id == inodeID) {
            if (previous != nullptr) {
              previous->next = current->next;
            } else {
              node.second = current->next;
            }
            delete current;
            cout << "Inode with ID " << inodeID << " has been deleted." << endl;
            return;
          }
          previous = current;
          current = current->next;
        }
      }
      cout << "Inode not found with ID " << inodeID << endl;
    }

private:
    Inode* temp;
```

```
    Inode* headOfList;
};

int main() {
    LinkedList inodeList;
    int id, size;
    string name, permissions, dateOfCreation;
    char choice;

    while (true) {
        cout << "\nInode Operations Menu:\n";
        cout << "1. Add Inode\n";
        cout << "2. Find Inode\n";
        cout << "3. Display Inodes\n";
        cout << "4. Update Permissions\n";
        cout << "5. Delete Inode\n";
        cout << "6. Exit\n";
        cout << "Enter your choice: ";
        cin >> choice;

        switch (choice) {
            case '1':
                cout << "Enter ID for the new inode: ";
                cin >> id;
                if (is_repeated[id] > 0) {
                    cout << "Enter Unique ID for the current inode: ";
                    cin >> id;
                }
                cout << "Enter name for the new inode: ";
                cin >> name;
                cout << "Enter size for the new inode: ";
                cin >> size;
                cout << "Enter permissions for the new inode: ";
                cin >> permissions;
                cout << "Enter date(DD/MM/YY) of creation of file for the new inode: ";
                cin >> dateOfCreation;
                inodeList.insertInode(id, name, size, permissions, dateOfCreation);
                break;
            case '2':
                cout << "How would you like to find the inode?\n";
                cout << "1. By ID\n";
                cout << "2. By Date\n";
                cout << "3. By Permissions\n";
                cout << "Enter your choice: ";
                cin >> choice;
                if (choice == '1') {
                    cout << "Enter the ID to find an inode: ";
                    cin >> id;
                    cout << "\nInode Details:\n";
                    inodeList.displayInode(id);
                } else if (choice == '2') {
                    cout << "Enter the date(DD/MM/YY) to find inodes created on that day: ";
                    cin >> dateOfCreation;
                    inodeList.displayInode(dateOfCreation);
```

```
            } else if (choice == '3') {
                cout << "Enter the permissions to find inodes with those permissions: ";
                cin >> permissions;
                inodeList.displayInodeByPermissions(permissions);
            } else {
                cout << "Invalid choice. Please try again." << endl;
            }
            break;
        case '3':
            cout << "\nInode List:\n\n";
            inodeList.displayInode();
            break;
        case '4':
            cout << "Enter the ID of the inode to update permissions: ";
            cin >> id;
            inodeList.updatePermissions(id);
            break;
        case '5':
            cout << "Enter the ID of the inode to delete: ";
            cin >> id;
            inodeList.deleteInode(id);
            break;
        case '6':
            cout << "Exiting the program." << endl;
            return 0;
        default:
            cout << "Invalid choice. Please try again." << endl;
            break;
    }
  }
  return 0;
}
```

- **Output:**

```
PS D:\ds assignment> g++ final.cpp
PS D:\ds assignment> .\a.exe

Inode Operations Menu:
1. Add Inode
2. Find Inode
3. Display Inodes
4. Update Permissions
5. Delete Inode
6. Exit
Enter your choice: 1
Enter ID for the new inode: 1
Enter name for the new inode: file1
Enter size for the new inode: 5
Enter permissions for the new inode: r
Enter date(DD/MM/YY) of creation of file for the new inode: 01/11/23

Inode Operations Menu:
1. Add Inode
2. Find Inode
3. Display Inodes
4. Update Permissions
5. Delete Inode
6. Exit
Enter your choice: 1
Enter ID for the new inode: 2
Enter name for the new inode: file2
Enter size for the new inode: 5
Enter permissions for the new inode: w
Enter date(DD/MM/YY) of creation of file for the new inode: 01/11/23

Inode Operations Menu:
1. Add Inode
2. Find Inode
3. Display Inodes
4. Update Permissions
5. Delete Inode
6. Exit
Enter your choice: 1
Enter ID for the new inode: 2
Enter Unique ID for the current inode: 3
Enter name for the new inode: file3
Enter size for the new inode: 10
Enter permissions for the new inode: w
Enter date(DD/MM/YY) of creation of file for the new inode: 31/10/23

This file exceeds the size of the data-block,
so this file will be saved in the next data-block.

Inode Operations Menu:
1. Add Inode
2. Find Inode
3. Display Inodes
```

```
Inode Operations Menu:
1. Add Inode
2. Find Inode
3. Display Inodes
4. Update Permissions
5. Delete Inode
6. Exit
Enter your choice: 1
Enter ID for the new inode: 4
Enter name for the new inode: file4
Enter size for the new inode: 7
Enter permissions for the new inode: r
Enter date(DD/MM/YY) of creation of file for the new inode: 31/11/23

This file exceeds the size of the data-block,
so this file will be saved in the next data-block.

Inode Operations Menu:
1. Add Inode
2. Find Inode
3. Display Inodes
4. Update Permissions
5. Delete Inode
6. Exit
Enter your choice: 1
Enter ID for the new inode: 5
Enter name for the new inode: file5
Enter size for the new inode: 3
Enter permissions for the new inode: w
Enter date(DD/MM/YY) of creation of file for the new inode: 02/11/23

Inode Operations Menu:
1. Add Inode
2. Find Inode
3. Display Inodes
4. Update Permissions
5. Delete Inode
6. Exit
Enter your choice: 2
How would you like to find the inode?
1. By ID
2. By Date
3. By Permissions
Enter your choice: 1
Enter the ID to find an inode: 5

Inode Details:
Data_Block: 3, ID: 5, Name: file5, Size: 3, Permissions: w, Date: 02/11/23
```

```
Inode Operations Menu:
1. Add Inode
2. Find Inode
3. Display Inodes
4. Update Permissions
5. Delete Inode
6. Exit
Enter your choice: 2
How would you like to find the inode?
1. By ID
2. By Date
3. By Permissions
Enter your choice: 2
Enter the date(DD/MM/YY) to find inodes created on that day: 01/11/23
Inodes created on 01/11/23:
Data_Block: 1, ID: 1, Name: file1, Size: 5, Permissions: r, Date: 01/11/23
Data_Block: 1, ID: 2, Name: file2, Size: 5, Permissions: w, Date: 01/11/23

Inode Operations Menu:
1. Add Inode
2. Find Inode
3. Display Inodes
4. Update Permissions
5. Delete Inode
6. Exit
Enter your choice: 2
How would you like to find the inode?
1. By ID
2. By Date
3. By Permissions
Enter your choice: 3
Enter the permissions to find inodes with those permissions: w
Inodes with permissions w:
Data_Block: 1, ID: 2, Name: file2, Size: 5, Permissions: w, Date: 01/11/23
Data_Block: 2, ID: 3, Name: file3, Size: 10, Permissions: w, Date: 31/10/23
Data_Block: 3, ID: 5, Name: file5, Size: 3, Permissions: w, Date: 02/11/23

Inode Operations Menu:
1. Add Inode
2. Find Inode
3. Display Inodes
4. Update Permissions
5. Delete Inode
6. Exit
Enter your choice: 3

Inode List:

DataBlock : 1

        ID: 1, Name: file1, Size: 5, Permissions: r, Date: 01/11/23
        ID: 2, Name: file2, Size: 5, Permissions: w, Date: 01/11/23
```

```
Enter your choice: 3

Inode List:

DataBlock : 1

        ID: 1, Name: file1, Size: 5, Permissions: r, Date: 01/11/23
        ID: 2, Name: file2, Size: 5, Permissions: w, Date: 01/11/23

DataBlock : 2

        ID: 3, Name: file3, Size: 10, Permissions: w, Date: 31/10/23

DataBlock : 3

        ID: 4, Name: file4, Size: 7, Permissions: r, Date: 31/11/23
        ID: 5, Name: file5, Size: 3, Permissions: w, Date: 02/11/23


Inode Operations Menu:
1. Add Inode
2. Find Inode
3. Display Inodes
4. Update Permissions
5. Delete Inode
6. Exit
Enter your choice: 4
Enter the ID of the inode to update permissions: 3
Enter new permissions for inode 3: readonly
Permissions updated.

Inode Operations Menu:
1. Add Inode
2. Find Inode
3. Display Inodes
4. Update Permissions
5. Delete Inode
6. Exit
Enter your choice: 3

Inode List:

DataBlock : 1

        ID: 1, Name: file1, Size: 5, Permissions: r, Date: 01/11/23
        ID: 2, Name: file2, Size: 5, Permissions: w, Date: 01/11/23

DataBlock : 2

        ID: 3, Name: file3, Size: 10, Permissions: readonly, Date: 31/10/23

DataBlock : 3
```

```
DataBlock : 3

        ID: 4, Name: file4, Size: 7, Permissions: r, Date: 31/11/23
        ID: 5, Name: file5, Size: 3, Permissions: w, Date: 02/11/23


Inode Operations Menu:
1. Add Inode
2. Find Inode
3. Display Inodes
4. Update Permissions
5. Delete Inode
6. Exit
Enter your choice: 5
Enter the ID of the inode to delete: 4
Inode with ID 4 has been deleted.

Inode Operations Menu:
1. Add Inode
2. Find Inode
3. Display Inodes
4. Update Permissions
5. Delete Inode
6. Exit
Enter your choice: 3

Inode List:

DataBlock : 1

        ID: 1, Name: file1, Size: 5, Permissions: r, Date: 01/11/23
        ID: 2, Name: file2, Size: 5, Permissions: w, Date: 01/11/23

DataBlock : 2

        ID: 3, Name: file3, Size: 10, Permissions: readonly, Date: 31/10/23

DataBlock : 3

        ID: 5, Name: file5, Size: 3, Permissions: w, Date: 02/11/23


Inode Operations Menu:
1. Add Inode
2. Find Inode
3. Display Inodes
4. Update Permissions
5. Delete Inode
6. Exit
Enter your choice: 5
Enter the ID of the inode to delete: 1
```

```
Enter the ID of the inode to delete: 1
Inode with ID 1 has been deleted.

Inode Operations Menu:
1. Add Inode
2. Find Inode
3. Display Inodes
4. Update Permissions
5. Delete Inode
6. Exit
Enter your choice: 3

Inode List:

DataBlock : 1

        ID: 2, Name: file2, Size: 5, Permissions: w, Date: 01/11/23

DataBlock : 2

        ID: 3, Name: file3, Size: 10, Permissions: readonly, Date: 31/10/23

DataBlock : 3

        ID: 5, Name: file5, Size: 3, Permissions: w, Date: 02/11/23


Inode Operations Menu:
1. Add Inode
2. Find Inode
3. Display Inodes
4. Update Permissions
5. Delete Inode
6. Exit
Enter your choice: 6
Exiting the program.
PS D:\ds assignment>
```