AISSMS

INSTITUTE OF INFORMATION TECHNOLOGY

ADDING VALUE TO ENGINEERING

Department Of Computer Engineering

Data Structure And Algorithms Lab

Group-F

SUBMITTED TO THE DEPARTMENT OF COMPUTER ENGINEERING
AISSMS IOIT

SE COMPTER ENGINEERING

SUBMITTED BY

Kaustubh S Kabra
ERP No.- 34
Teams No.-20

2020 -2021

Experiment Number:-11

Aim:-Department maintains a student information. The file contains roll number, name, division and address. Allow user to add, delete information of student. Display information of particular employee. If record of student does not exist an appropriate message is displayed. If it is, then the system displays the student details. Use sequential file to main the data.

Objective:-

1) To understand concept of file organization in data structure.
2) To understand concept & features of sequential file organization

Theory:-

File organization refers to the relationship of the key of the record to the physical location of that record in the computer file. File organization may be either physical file or a logical file. A physical file is a physical unit, such as magnetic tape or a disk. A logical file on the other hand is a complete set of records for a specific application or purpose. A logical file may occupy a part of physical file or may extend over more than one physical file.

There are various methods of file organizations. These methods may be efficient for certain types of access/selection meanwhile it will turn inefficient for other selections. Hence it is up to the programmer to decide the best suited file organization method depending on his requirement.

Some of the file organizations are:

1. Sequential File Organization
2. Heap File Organization
3. 3. Hash/Direct File Organization
4. 4. Indexed Sequential Access Method
5. 5. B+ Tree File Organization

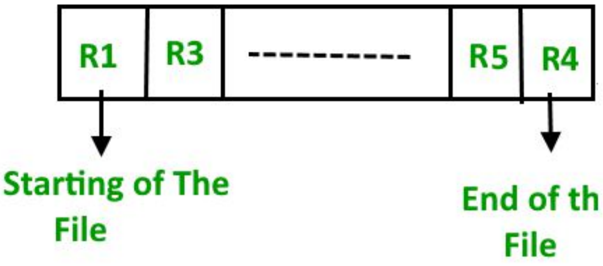
6. 6. Cluster File Organization

Sequential File Organization:

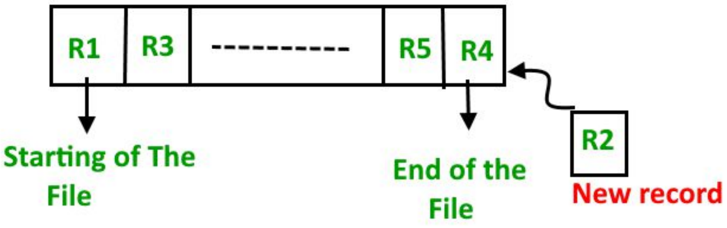
Sequential File Organization –

The easiest method for file Organization is Sequential method. In this method the file are stored one after another in a sequential manner. There are two ways to implement this method:

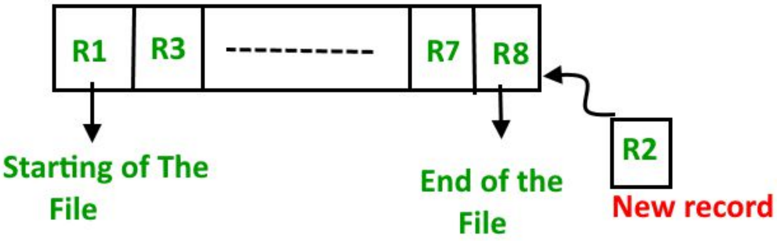
1. ***Pile File Method –*** *This method is quite simple, in which we store the records in a sequence i.e one after other in the order in which they are inserted into the tables.*



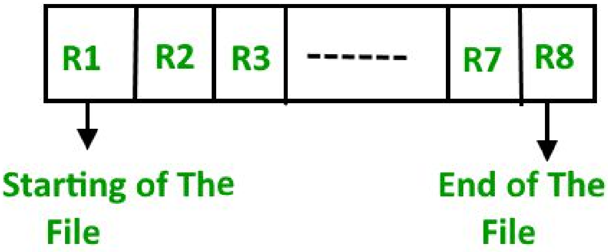
Insertion of new record –
Let the R1, R3 and so on upto R5 and R4 be four records in the sequence. Here, records are nothing but a row in any table. Suppose a new record R2 has to be inserted in the sequence, then it is simply placed at the end of the file.



2. ***Sorted File Method –****In this method, As the name itself suggest whenever a new record has to be inserted, it is always inserted in a sorted (ascending or descending) manner. Sorting of records may be based on any primary key or any other key.*



Insertion of new record –
Let us assume that there is a preexisting sorted sequence of four records R1, R3, and so on upto R7 and R8. Suppose a new record R2 has to be inserted in the sequence, then it will be inserted at the end of the file and then it will sort the sequence .



Pros and Cons of Sequential File Organization –

Pros –

- *Fast and efficient method for huge amount of data.*
- *Simple design.*
- *Files can be easily stored in magnetic tapes i.e cheaper storage mechanism.*

Cons –

- *Time wastage as we cannot jump on a particular record that is required, but we have to move in a sequential manner which takes our time.*
- *Sorted file method is inefficient as it takes time and space for sorting records.*

Algorithm:-

1) Declare a class with the variables to store name, roll number, division and address of n students.
2) Open a file with .dat as extension to store the data of students.
3) Accept records of n students and store it in the dat file sequentially.
4) If user wants to display the file, then traverse the dat file and display each record along with all the details.
5) If user wants to delete a record from the file, then accept the roll number of the student from the user.
6) If the given roll number is present, then delete the record by initializing NULL values to all the fields of the particular record.
7) If the roll number is not present then display record not found.
8) If user wants to search a particular record from the file, then accept the roll number from the user.
9) Traverse the file and search for the given roll number.
10) If roll number is present, then display found, else display roll number not present.
11) STOP

Program:-

```
#include<iostream>

#include<iomanip>

#include<fstream>

#include<cstring>

#include<stdlib.h>

using namespace std;


class STUDENT_CLASS

{
        typedef struct STUDENT
        {
                char name[10];
                int roll_no;
                char division;
                char address[50];
        }Rec;

        Rec Records;
        public:
                void Create();
                void Display();
                void Delete();
                int Search();
};

void STUDENT_CLASS::Create()
{
        char ch='y';
        fstream seqfile;
        seqfile.open("STUDENT.DAT",ios::app|ios::in|ios::out|ios::binary);
        do
        {
```

```
        cout<<"\n Enter Name: ";
        cin>>Records.name;
        cout<<"\n Enter roll no: ";
        cin>>Records.roll_no;
        cout<<"\n Enter Division: ";
        cin>>Records.division;
        cout<<"\n Enter address: ";
        cin>>Records.address;

        seqfile.write((char *)&Records,sizeof(Records));
        cout<<"\nDo you want to add more records?";
        cin>>ch;
    }while(ch=='y' || ch=='Y');
    seqfile.close();
}

void STUDENT_CLASS::Display()
{
    fstream seqfile;
    int n;
    seqfile.open("STUDENT.DAT",ios::in|ios::out|ios::binary);

    seqfile.seekg(0,ios::beg);
```

```
        cout<<"\n The Contents of file are ..."<<endl;

        while(seqfile.read((char *)&Records,sizeof(Records)))
        {
                if(Records.roll_no!=-1)
                {
                        cout<<"\n Nmae: "<<Records.name;
                        cout<<"\n Roll no: "<<Records.roll_no;
                        cout<<"\n Division: "<<Records.division;
                        cout<<"\n Address: "<<Records.address;
                        cout<<"\n";
                }
        }
        int last_rec=seqfile.tellg();
        n=last_rec/sizeof(Rec);
        seqfile.close();
}

void STUDENT_CLASS::Delete()
{
        int pos;
        cout<<"\n For deletion";
        fstream seqfile;
```

```
    seqfile.open("STUDENT.DAT",ios::in|ios::out|ios::binary);
    seqfile.seekg(0,ios::beg);
    pos=Search();
    if(pos==-1)
    {
        cout<<"\n The record is not present in the file";
        return;
    }

    int offset=pos*sizeof(Rec);
    seqfile.seekp(offset);
    strcpy(Records.name,"");
    Records.roll_no=-1;
    Records.division='N';
    strcpy(Records.address,"");
    seqfile.write((char *)&Records,sizeof(Records))<<flush;
    seqfile.seekg(0);
    seqfile.close();
    cout<<"\n The record is Deleted !!!";
}

int STUDENT_CLASS::Search()
{
    fstream seqfile;
    int id,pos;
    cout<<"\n Enter the roll_no for searching the record: ";
    cin>>id;
    seqfile.open("STUDENT.DAT",ios::ate|ios::in|ios::out|ios::binary);
    seqfile.seekg(0);
    pos=-1;
    int i=0;
    while(seqfile.read((char *)&Records,sizeof(Records)))
    {
        if(id==Records.roll_no)
        {
                pos=i;
                break;
        }
        i++;
    }
    return pos;
}

int main()
{
    STUDENT_CLASS List;
```

```
char ans='y';
int choice,key;

do
{
    cout<<"\n********Main Menu********"<<endl;
    cout<<"\n 1.Create";
    cout<<"\n 2.Display";
    cout<<"\n 3.Delete";
    cout<<"\n 4.Search";
    cout<<"\n 5.Exit";
    cout<<"\n Enter your choice: ";
    cin>>choice;
    switch(choice)
    {
        case 1:List.Create();
            break;

        case 2:List.Display();
            break;

        case 3:List.Delete();
            break;
            case 4:key=List.Search();
            if(key<0)
                cout<<"\n Record is not present in the file";
            else
                cout<<"\n Record is present in the file";
            break;

            case 7:exit(0);
        }
        cout<<"\n\t Do you want to go back to Main Menu?";
        cin>>ans;
    }while(ans=='y'||ans=='Y');
    return 0;
}
```

Output:-

```
********Main Menu********


 1.Create
 2.Display
 3.Delete
 4.Search
 5.Exit
 Enter your choice: 1

 Enter Name: Harsh

 Enter roll no: 59

 Enter Division: A

 Enter address: Pune

Do you want to add more records?y

 Enter Name: Kaustubh

 Enter roll no: 34

 Enter Division: A

 Enter address: Jalgoan

Do you want to add more records?y
```

```
 Enter Name: Onasvee

 Enter roll no: 16

 Enter Division: A

 Enter address: Pune

Do you want to add more records?n

     Do you want to go back to Main Menu?y

********Main Menu********

 1.Create
 2.Display
 3.Delete
 4.Search
 5.Exit
 Enter your choice: 2

 The Contents of file are ...
```

```
Nmae: Harsh

Roll no: 59

Division: A

Address: Pune


Nmae: Kaustubh

Roll no: 34

Division: A

Address: Jalgoan


Nmae: Onasvee

Roll no: 16

Division: A

Address: Pune


    Do you want to go back to Main Menu?y


********Main Menu********


1.Create

2.Display

3.Delete

4.Search
```

```
5.Exit
Enter your choice: 4


Enter the roll_no for searching the record: 59


Record is present in the file
        Do you want to go back to Main Menu?y


********Main Menu********


1.Create
2.Display
3.Delete
4.Search
5.Exit
Enter your choice: 3


For deletion
Enter the roll_no for searching the record: 59


The record is Deleted !!!
        Do you want to go back to Main Menu?y
```

```
********Main Menu********

1.Create

2.Display

3.Delete

4.Search

5.Exit

Enter your choice: 2


The Contents of file are ...


Nmae: Kaustubh

Roll no: 34

Division: A

Address: Jalgoan


Nmae: Onasvee

Roll no: 16

Division: A

Address: Pune


        Do you want to go back to Main Menu?n
```

Analysis:-

Time Complexity:

Creation, deletion, display and searching in the file is completed in O(n) time.

Conclusion:– *Thus, we have learned and implemented sequential file organization. We have created a file using sequential organization to store the details of students.*