ASSINGMENT NO.	**12**
TITLE	**Direct Access File**
PROBLEM STATEMENT /DEFINITION	Implementation of a direct access file -Insertion and deletion of a record from a Direct Access File.
OBJECTIVE	To understand practical implementation and usage of Direct Access File for solving the problems.
OUTCOME	After successful completion of this assignment, students will be able to implement and use **Direct Access File** for efficient solution to problems.
S/W PACKAGES AND HARDWARE APPARATUS USED	• (64-bit)64-BIT Fedora 17 or latest 64-bitupdate of equivalent Open source OS • Programming Tools (64-bit) Latest Open source update of Eclipse Programming frame work
REFERENCES	1. E. Horowitz S. Sahani, D. Mehata, "Fundamentals of data structures in C++", Galgotia Book Source, New Delhi, 1995, ISBN: 1678298 2. SartajSahani, ―Data Structures, Algorithms andApplications in C++‖, Second Edition, University Press, ISBN:81-7371522 X.
INSTRUCTIONS FOR WRITING JOURNAL	1. Date 2. Assignment no. and title 3. Problem definition 4. Learning Objective 5. Learning Outcome 6. Software / Hardware requirement 7. Concepts related Theory 8. Algorithms/ Pseudo Code 9. Class ADT 10. Test cases 11. Conclusion/Analysis

Prerequisites:

- Basic knowledge of File Data Structure.
- Its operations – Create, Open, Write and Read etc.

Learning Objectives:

- To understand practical implementation and usage of Direct Access File for solving the problems.

Learning Outcomes:

- After successful completion of this assignment, students will be able to implement and use Direct Access File for efficient solution to problems.

Concepts related Theory:

File is a collection of records.

File Organization: File organization ensures that records are available for processing. It is used to determine an efficient file organization for each base relation.

Types of File Organization

1. Sequential access file organization
2. Direct access file organization
3. Indexed sequential access files organization

Direct access file organization

Direct access file is also known as Random access or relative file organization. In direct access file, all records are stored in direct access storage device, such as hard disk. The records are randomly placed throughout the file. The record does not need to be in sequence because they are updated directly and rewritten back in the same location.

This file organization is useful for immediate access to large amount of information. It is used in accessing large databases. It is also called as hashing.

Advantages of direct access file organization

- Direct access file helps in online transaction processing system (OLTP) like online railway reservation system.
- In direct access file, sorting of the records are not required.
- It accesses the desired records immediately.
- It updates several files quickly.
- It has better control over record allocation.

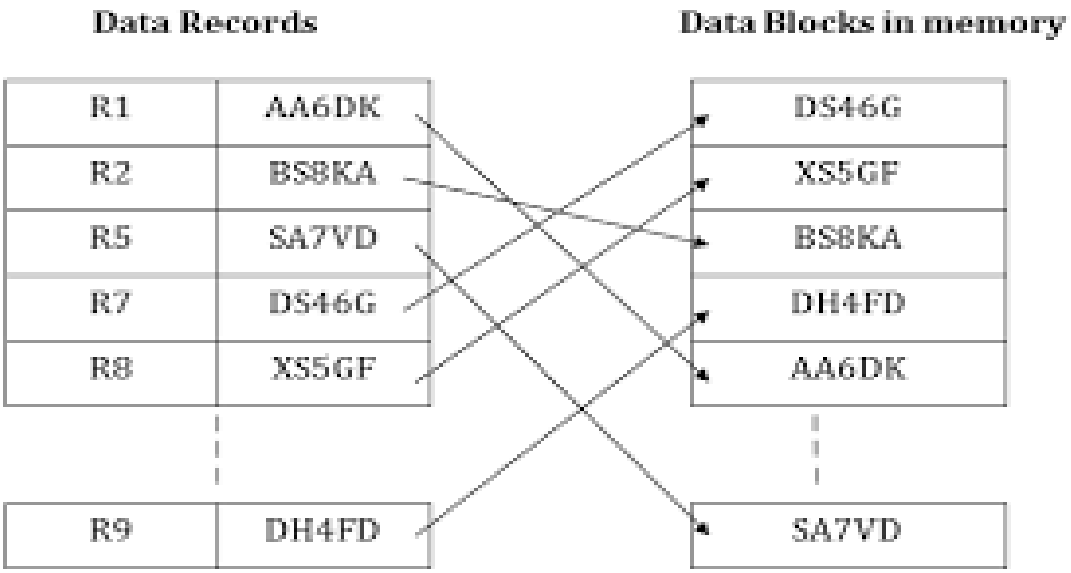
Disadvantages of direct access file organization

- Direct access file does not provide back-up facility.
- It is expensive.

- It has less storage space as compared to sequential file.

Representation:

Following is a pictorial representation of Direct Access File –



Basic Operations:

Following are the basic operations of Sequential File −

- **Creating a file** − Creation of a file is also defined as the loading of the file.
- **Opening a file** − Before Read/Write, that file must be opened.

  (Name of a file and the access mode (read or write).
- **Writing Records** − A write operation to write Records into the file.
- **Reading Records** – To read Records from file sequentially.

Record:- is a collection of fields related to particular entity, such as Student, Employee etc.

```
//Header Files
#include<iostream>
#include<fstream>
```

a) **Creating an Object of fstream**
   ofstream fout;

b) To open file for Writing

```
fout.open("One.txt" , ios::out);
```

c) Writing into the Direct Access file

```
//Record-01 : 1011 Akshay  BE-Comp  Nashik
cout<<"\n Location: "<<fout.tellp();
fout<<"1011 Akshay  BE-Comp  Nashik";
//Record-02 : 1012 Akanksha  BE-Comp  Pune
cout<<"\n Location: "<<fout.tellp();
fout<<"1012 Akanksha  BE-Comp  Pune ";
//Record-03 : 1013 Tejal  BE-Comp  Mumbai
cout<<"\n Location: "<<fout.tellp();
fout<<"1013 Tejal  BE-Comp  Mumbai";
cout<<"\n Location: "<<fout.tellp()<<endl;
```

d) Close the file.

```
fout.close();
```

e) Create Object to read from file.

```
ifstream fin;
```

f) Open File for reading.

```
fin.open("One.txt" , ios::in);
```

g) <u>Reading from the file....</u>

```
//to know current location.....
cout<<"\n Location: "<<fin.tellg();
//Shift control to other location
fin.seekg(64 , ios::beg);
//display the other location
cout<<"\n Location: "<<fin.tellg();
```

```
//Now Reading....
char c;
while(!fin.eof())
{
      fin.get(c);
      cout<<c;
}           // We have Directly Accessed the Record-03 of the file.
```

h) Close the file.

```
fin.close();
```

Algorithms and Pseudo code:

Step1: START

//.....for Writing

Step2: Create the Object/handle to write into Direct Access File(DAF).

Step3: Open the DAF in Write Mode.

Step4: Write into the DAF using the Object/Handle created.

Step5: Repeat Step-4 to write number of lines.

Step6: Close the File.

//.....for Reading

Step7: Open the DAF in Read Mode.

Step8: Read the Lines/Records written in File in any order. (Non-Sequentially)

Step9: Close the File.

//.....for Deletion

Step10: Delete a line/record from file and try to read it.

Step11: STOP.

Conclusion:

In this way the Direct Access File is implemented to write records sequentially and read records randomly using the objects of ofstream and ifstream classes to obtain desired results.

Sample Questions:

1. What is the Direct Access file?

2. Compare the features of sequential file, index sequential file & direct access file.

3. What is a Direct Access Storage devices? Give examples.

4. How to add and read records in Direct Access file?

5. What are primitive operations on Direct Access file?

6. Explain the different types of external storage devices?

7. With the prototype and example, explain following functions:

    i) seekg()

    ii) tellg()

    iii) seekp()

    iv) tellp()