

# Experimental Implementation of Different File Access Method

Ramanan S K, Dr.M. Sujithra M.C.A, M.Phil., Phd, Dr.A.Dchitra  
M.C.A,M.Phil,Phd.

*2<sup>nd</sup> Year, MSc. Software Systems(Integrated),Coimbatore Institute of Technology,Coimbatore  
Assistant Professor, Department of Data Science, Coimbatore Institute of Technology, Coimbatore  
Assistant Professor, Department of Data Science, Coimbatore Institute of Technology, Coimbatore*

Date of Submission: 05-11-2020

Date of Acceptance: 18-11-2020

**ABSTRACT** - Computer is an integral part of human life. Large amount of data permanently stored in computer and we need to access them as fast as we can to get a efficient system . Since hardware has limitations we need devise different methods which minimise hardware usage. In this research paper we discuss the different types of file access methods and concluded upon which access methods suites which situation

## I. INTRODUCTION

File is a logical collection of information stored on secondary storage such as hard disk. It is a collection of records. Physically, a file is smallest allotment of secondary storage device for example disk. Logically, a file is a sequence of logical records such as a sequence of bits and bytes. Files can be used to contain the data and programs (both source and object programs). Data files can be numeric, alphabetic, alphanumeric or binary. A file has various attributes like name, type, location, size, protection, timeand date of creation etc.

Computers can storeinformation in several physical forms, depending on which storage device is used. Disks and drums though are the most common devices for this purpose. Since each device has its own characteristics and physical organization, information may be stored in several ways and therefore different views of information are created. To unify all these views of information in the system, a uniform logical view of it was created. This logical view is called a file. It is the job of the OS to map this sequence of words into physical devices. The part of the OS responsible for this is the file system. It is clear that the main objective of the file systems is to free the users of the details of storing the information in the physical devices. That is, when thestorage device is changed, from disk to drum for example, the user still sees the same information as before the change. If this is

allowed in the system, then we can say that the file system is device dependent.

## II. OUTLINE OF A FILE SYSTEM

A file system is a system that manages how and where data on a storage disk, typically a hard disk drive (HDD), is stored, accessed and managed. It is a logical disk component that manages a disk's internal operations as it relates to a computer and is abstract to a human user.

In computing, a file systemcontrols how data is stored and retrieved. Without a file system, data placed in a storage medium would be one large body of data with no way to tell where one piece of data stops and the next begins. By separating the data into pieces and giving each piece a name, the data is easily isolated and identified. Taking its name from the way paper-based data management system is named, each group of data is called a "file." The structure and logic rules used to manage the groups of data and their names is called a "file system."

There are many different kinds of file systems. Each one has different structure and logic, properties of speed, flexibility, security, size and more. Some file systems have been designed to be used for specific applications.

## III. FILE ACCESS

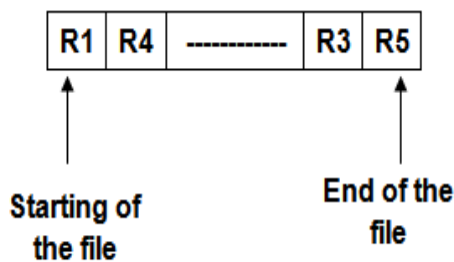
When a file is used, information is read and accessed into computer memory and there are several ways to access this information of the file. Some systems provide only one access method for files. Other systems, such as those of IBM, support many access methods, and choosing the right one for a particular application is a major design problem. In this paper we are going to see which method is suited for which application

#### IV. TYPES OF FILE ACCESS METHODS

There are five types of file access methods :-

- a. The Pile
- b. The Sequential File
- c. Indexed Sequential File
- d. Indexed File
- e. Direct/Hashed File

##### a. The Pile



**Fig1 – Pile File Representation**

Pile file contains a set of records in no particular order. The records are stored as they arrive. They do not follow any particular sequence of attribute values. This method of storing records is adopted when it is not possible to arrange the records in any logical order, when the fields of the record are not well defined and the exact usage of the file cannot be anticipated. Files are generally created in this mode by punching the documents in the order they arrive. The pile so created may be then organised into another mode.

Location of a record in a pile file can be done by sequentially searching the records till the desired value of the key attribute is reached. Addition of new records in a pile file is very simple. New records are added at the end of the file.

Changes and deletions of records in pile file stored on random access media can be done by locating the record and changing its contents or flagging the record to indicate that the record has been invalidated. The file may be reorganized periodically to remove the holes created by the deletion of records. However, updating of the pile file on sequential access media can be done only by creating a new file.

##### Pro :-

- It is useful :-
  - when it is not possible to arrange the records in any logical order
  - when the fields of the record are not well defined

- when the exact usage of the file cannot be anticipated.
- It uses space efficiently
- Insertion and deletion in pile file is very simple and consumes short amount of time

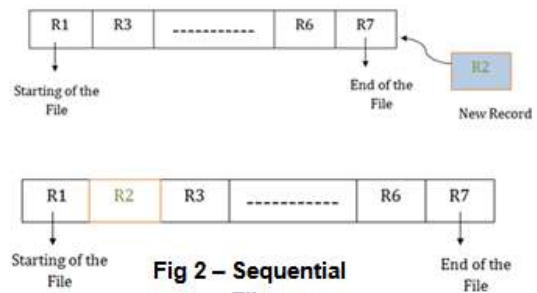
##### Cons :

- Searching a record in a particular pile file includes examining all the records in the pile till you find the record which takes a really long time and not very efficient way to access a record
- Deletion of a record leave a hole in the pile file , it has to reorganized periodically which consumes lot of time

##### Application :

Pile File method is used where the records are arranged chronologically and doesn't have proper structure like work report, the reports are just piled together in a file

##### b. The Sequential file



**Fig 2 – Sequential File Representation**

In sequential organisation, record occurrences are referenced in the order that they are stored physically. Data stored on cards and tapes are necessarily stored sequentially. Direct access devices such as disks may be, but are not necessarily referenced sequentially.

Sometimes the processing is best done through sequential access, even when direct access devices are used. Such is the case when many records of a file must be accessed to satisfy a request for data.

In a sequential file, records are maintained in the logical sequence of their primary key values. The processing of a sequential file is conceptually simple but inefficient for random access. However, if access to the file is strictly sequential, a sequential file is suitable. A sequential file could be stored on a sequential storage device such as a magnetic tape.

In a sequential file update, transaction records are in the same sequence as in the master file. Records from both files are matched on the basis of one record at a time, resulting in an updated master file

**Pros :**

- Errors in files remain localised.
- Loading a record requires only the record key.
- Simple to understand approach.
- Easy to organize, maintain and understand.
- Relatively inexpensive input/output (I/O) media and device may be used.
- Files may be relatively easy to reconstruct since a good measure of built in back up is usually available.
- Efficient and economical if the activity rate, i.e. the proportion of file records to be processed is high.

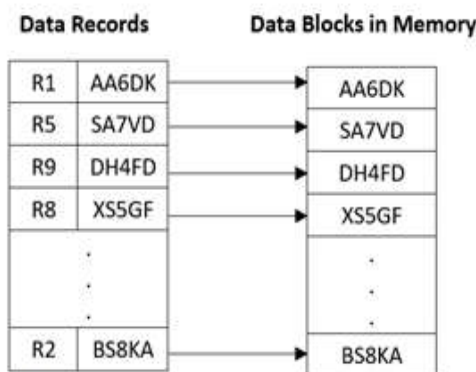
**Cons :**

- Transactions must be sorted and placed in sequence prior to processing.
- 'Data' redundancy is typically high since the same data may be stored in several files sequenced in different keys.
- Timeliness of data in file deteriorates while batches are being accumulated.
- Entire file must be processed even when the activity rate is very low.
- This method is suitable for the small records. If the record is very big, so much time is consumed in arranging the file.

**Application :**

Sequential access is used when one can only visit the values it contains in one particular order. For Example, reading data from a tape can only be done sequentially so sequential access is the only way to access the records

**c. Indexed File**



**Fig 3 – Indexed File Representation**

In the indexed file access method, one index is maintained for each key attribute of the record. The index file contains the value of the key attribute followed by the addresses of all the records in the main file with the The main file need not contain the values of those attributes for which indices have been maintained since there is already contained in the index file. The main file may also not be in any particular sequence. It can just be a pile file.

The indexed file organisation requires three kinds of files to be maintained, the main file, the Directory files and the index files. The directory file contains the value of the key attributes and the pointer to the first record in the index file where the addresses of all the records in the main file with that value of the key attribute are contained. There is a directory file for each key attribute.

Indexed file is very useful where the list of records with specified values of key attribute is required. For example, we want to have a list of all the teacher posted in Lucknow District or all those teaching physics. Addition of records in the main file can be made at the end of the file since it is not maintained in any sequence.

However, corresponding entries have to be made in all the index files for the key attributes of the record. Changes in key attributes of the record and deletions will also have to be accompanied by corresponding changes in the index files. The maintenance of index files therefore can be very time consuming.

**Pros :**

- Most flexible random access: by any one of multiple keys or RFA; key access by generic or approximate value
- Duplicate key values possible
- Automatic sort of records by primary and alternate keys; available during sequential access
- Record location is transparent to user
- Potential range of key values not physically present as in relative file organization
- Variety of data formats for keys
- Transparent data compression

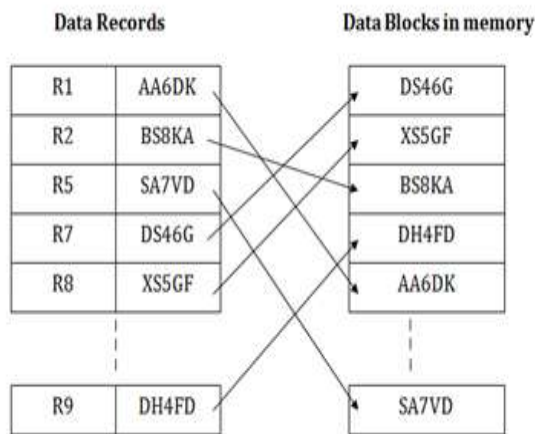
**Cons :**

- Highest overhead on disk and in memory
- Most complex programming
- Longest record access times
- Restricted to disk

**Application :**

It is suitable for the records which are well defined and has unique key for each record like an employee database, each key attribute (say department) which is chosen will have separate index file (i.e. for each department) pointing the records in the main file

**d. Indexed Sequential File**



**Fig 4 – Indexed Sequential File Representation**

The retrieval of a record from a sequential file, on average, requires access to half the records in the file, making such enquiries not only inefficient but very time consuming for large files. To improve the query response time of a sequential file, a type of indexing technique can be added.

An index is a set of < key, address > pairs. Indexing associates a set of orderable quantities, which are usually smaller in number or their properties, provide a mechanism for faster search. The purpose of indexing is to expedite the search process. Indexes created from a sequential (or sorted) set of primary keys are referred to as index sequential.

Although the indices and the Data blocks are held together physically, we distinguish between them logically. We shall use the term index file to describe the indexes and data file to refer to the data records. The index is usually small enough to be read into the processor memory.

A sequential (for sorted on primary keys) file that is indexed is called an index sequential size. The index provides for random access to records, while the sequential nature of the file provides easy access to the subsequent records as well as sequential processing. An additional feature of this file system is the overflow area. This feature provides additional space for record addition without necessitating the creation of a new file.

**Pros :**

- Permits Quick access to records in a relatively efficient way when this activity is a small fraction of the work load.
- Permits the efficient and economical use of sequential processing techniques when the activity rate is high.
- Since each record has its data block address, searching for a record in larger database is easy and quick. There is no extra effort to search records. But proper primary key has to be selected to make ISAM efficient.
- This method gives flexibility of using any column as key field and index will be generated based on that. In addition to the primary key and its index, we can have index generated for other fields too. Hence searching becomes more efficient, if there is search based on columns other than primary key.
- It supports range retrieval, partial retrieval of records. Since the index is based on the key value, we can retrieve the data for the given range of values. In the same way, when a partial key value is provided, say student names starting with 'JA' can also be searched easily

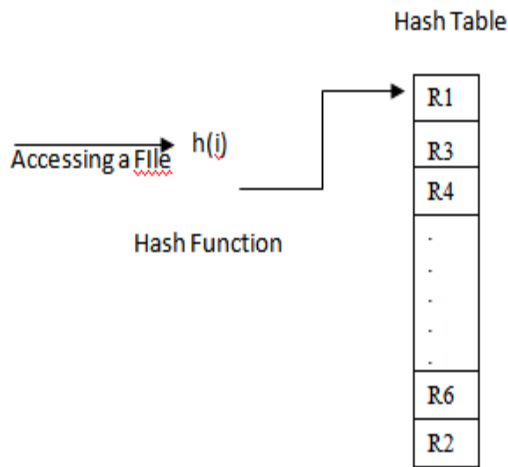
**Cons :**

- Less efficient in the use of storage space than some other alternatives.
- Relatively expensive hardware and software resources are required.
- Slow retrieval compared to random access as searching of index requires time.
- An extra cost to maintain index has to be afforded. i.e.; we need to have extra space in the disk to store this index value. When there is multiple key-index combinations, the disk space will also increase.
- Difficult to adjust, modify or update
- Due to the nature of the format, any error or adjustment to the command would require rewriting all files after the error.
- It cannot access random files

**Application :**

It is used when sequential access is required and using indexing technique will decrease the access rate.

**e. Direct/Hashed File**



**Fig 5 – Direct File Representation**

Indexes sequential access is suitable for buyer's stock data because his file must be accessed both sequentially and directly, and updating is done periodically up to the minute information is not required. Many applications on the other hand, require both on time retrieval and on both line update to maintain the most current information possible.

Stockbrokers, for example, need both rapid update and retrieval of stock price data. Airlines require rapid update and retrieval of passenger data, and automated teller machines require fast access to banking records.

Direct access to large indexed sequential access method (ISAM) files is slowed by the fact that the indexes and perhaps overflow areas must be searched before the desired record can be transferred to main memory. Even worse update of (ISAM), files is slowed by the need to update indexes and pointers and to transfer records from primary to overflow areas.

The direct access method has been devised to satisfy the need to both update and retrieve data quickly. In this approach, an algorithm is used to compute the address of a record. The primary key value is the input to the algorithm and the block address of the record is the output.

To implement the approach, a portion of the storage space is reserved for the file. This space must be large enough to hold the file plus some allowance for growth. Then an algorithm that generates the appropriate address for a given primary key is devised.

The algorithm is commonly called a hashing algorithm and the direct access method is referred to as hashed access. The process of

converting primary key values into addresses is called key-to-address transformation.

More than one logical record usually fits into a block, so we may think of the reserved storage area as being broken into record slots sequentially numbered from 1 to n. These sequential numbers are called relative record numbers, relative pointers, or relative addresses, because they indicate the position of the record relative to the beginning of the file.

**Pros :**

- Transactions need not be sorted.
- Different discs or disc units are not required for updating records as existing records may be amended by overwriting.
- It is also possible to process direct file records sequentially in a record key sequence.
- A direct file organisation is most suitable for interactive on line applications such as air line or railway reservation systems, teller facility in banking application, etc.
- Immediate access to records for updating purposes is possible.
- Random inquiries which are too frequent in business situations can be easily handled.

**Cons :**

- May be less efficient in the use of storage space than sequentially organised file.
- Expensive hardware and software resources are required.
- Relative complexity of programming.
- System design around it is complex and costly.
- Data may be accidentally erased or even written unless special precautions are taken.
- Special security measures are necessary for on line direct files that are accessible from several stations.
- File updating (addition and deletion records) is more difficult as compared to sequential files.
- Records in the on line may be expressed to the risks of loss of accuracy and a breach of security Special back up and reconstruction procedures must be established.

**Application :**

It is useful when you need to group the data like say a reservation system and you need to access all the records of a given time you directly go the index and access all the data without the need traverse through all the records

**V. IMPLEMENTATION**

To implement the different access methods a example of five records is taken as follows :-



The records come in the order 1,5,4,2,3 to the file . Now we create a file for each type and access them and compare it with each other

Since we have taken only five records the comparison between them will take only a fraction of a milli second so we increased the time taken for each comparison to 1 seconds, that way we can see the difference between each method

**Data :**

Record 1

|         |         |     |      |           |            |
|---------|---------|-----|------|-----------|------------|
| RD18001 | Ramanan | R&D | 2018 | Address-1 | 9876543210 |
|---------|---------|-----|------|-----------|------------|

Record 2

|         |           |         |      |           |            |
|---------|-----------|---------|------|-----------|------------|
| AF18001 | Kalidasan | Finance | 2018 | Address-2 | 9478493738 |
|---------|-----------|---------|------|-----------|------------|

Record 3

|         |       |     |      |           |            |
|---------|-------|-----|------|-----------|------------|
| RD19002 | Abdul | R&D | 2019 | Address-3 | 9749475938 |
|---------|-------|-----|------|-----------|------------|

Record 4

|         |         |         |      |           |            |
|---------|---------|---------|------|-----------|------------|
| AF19002 | Srinath | Finance | 2019 | Address-4 | 9348204894 |
|---------|---------|---------|------|-----------|------------|

Record 5

|         |       |     |      |           |            |
|---------|-------|-----|------|-----------|------------|
| RD20003 | Ameen | R&D | 2020 | Address-5 | 9394839028 |
|---------|-------|-----|------|-----------|------------|

**Fig 6 – Data Records for Implementation**

All five records are arranged based on each record and a file is generate for each access method

**Test Case 1 :**

```

File File Access Method :
Access Time: 5.01s
Memory space occupied = 600bytes

Sequential File Access Method :
Access Time: 3.01s
Memory space occupied = 600bytes

Indexed File Access Method :
Access Time: 3.00s
Memory space occupied = 640bytes

Indexed Sequential File Access Method :
Access Time: 2.01s
Memory space occupied = 640bytes

Direct File Access Method :
Access Time: 2.00s
Memory space occupied = 680bytes
    
```

**Fig 7– Test Case 1 Output**

Searching for the Record 4 in all the data file

**Test Case 2 :**

Searching for a record which does not exists

```

File File Access Method :
Access Time: 5.01s
Memory space occupied = 600bytes

Sequential File Access Method :
Access Time: 5.01s
Memory space occupied = 600bytes

Indexed File Access Method :
Access Time: 3.01s
Memory space occupied = 640bytes

Indexed Sequential File Access Method :
Access Time: 3.01s
Memory space occupied = 640bytes

Direct File Access Method :
Access Time: 3.00s
Memory space occupied = 680bytes
    
```

**Fig 8 – Test Case 2 Output**

**VI. CONCLUSION**

This paper discuss the need for the file system and methods to make it more efficient . It demonstrate how a file system can accessed using different access methods . Furthermore, the paper describes structure of each file access methods and explain in what way each method dominates the other methods.

**REFERENCE**

Introduction to Operating Systems By Harvey M. Deitel  
<https://www.yourarticlelibrary.com>  
<https://www.geeksforgeeks.org>  
<https://padakuu.com/article/92-file-access-methods>