LDRP Institute of Technology and Research, Gandhinagar

CT303-N
Data Structures and Algorithms

Concepts of Fields, Records and Files

- **Field**
  - It is defined as a unit of meaningful information about an entity like date of flight, name of passenger, address etc.
  - The smallest piece of information that can be referenced in programming language.
  - A **field** defines the individual elements of a record.
- **Record**
  - It is a collection of units of information about a particular entity. Passenger of an airplane, an employee of an organization, or an article sold from a store.
  - A record generally holds information about some real-world entity – the financial transactions of a single customer, the fee payments or grades of a single student, the diagnoses and prescriptions of a single patient, and so on.
  - A record is a collection of *fields*, possibly of different data types, typically in a fixed number and sequence.

Concepts of Fields, Records and Files

- **File**
  - A collection of records involving a set of entities with certain aspects in common and organized for some particular purpose is called a file. For example collection of records of all passengers.
  - A file is a large list that is stored in the external memory of computer.
  - A file may be used as a repository for list items commonly called records.
  - A file is a collection of data stored in one unit, identified by a filename. It can be a document, picture, audio or video stream, data library, application, or other collection of data. The following is a brief description of each file type.
  - Files can be opened, saved, deleted, and moved to different folders.
  - A computer *file* is a computer resource for recording data discretely in a computer storage device.
  - A *file* is an object on a computer that stores data, information, settings, or commands used with a computer program.

Various File Organizations

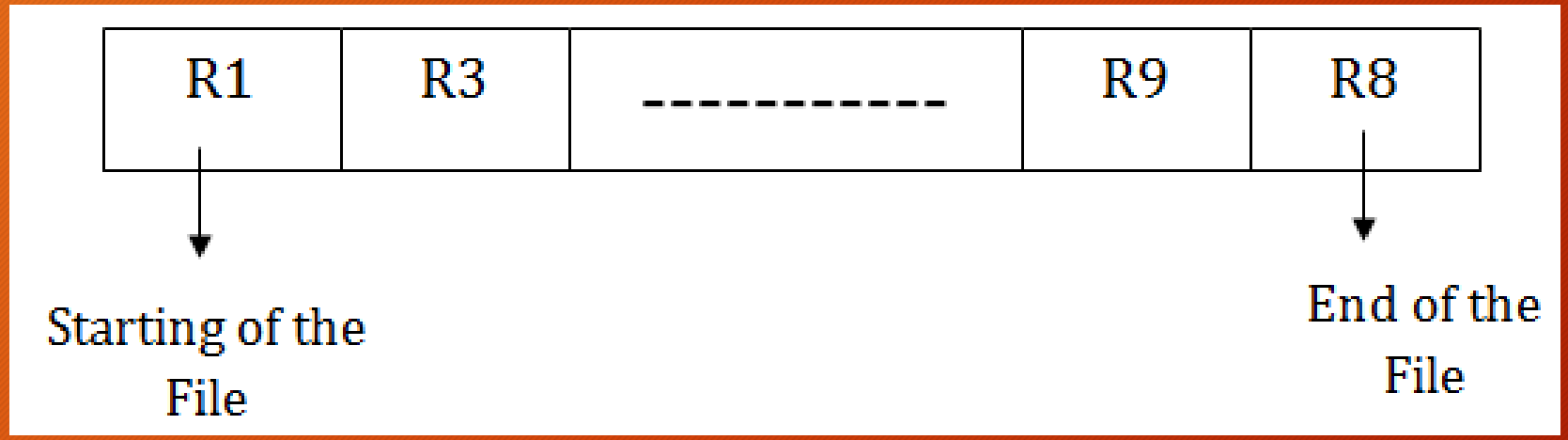
- *File organization* refers to the way records are physically arranged on a storage device.
- The way a file is organized does not necessarily imply the way in which the records within that file will be accessed.
- **File organization** determines the **method** of access, efficiency, flexibility and storage devices to be used.
- There are three file organizations to understand the relationship between Fields, Records and Files- sequential, Indexed sequential and relative.

1. **Sequential File**
2. **Indexed Sequential File**
3. **Direct Access / Random Access**

Sequential File

- In a sequential file, records are stored one after another on a storage device such that accessing any record can be done only after accessing all records stored before it.
- A sequential file is physically placed on a storage device by storing the sequence of records in adjacent locations on a track. If the file is stored in a direct access storage device like drum and the file is larger than the amount of space on a track, then the records are placed on adjacent tracks. The basic operations to be performed on a sequential file are read, write and update.
- Some operating systems provide file-accessing facilities which allow a file to be extended by writing records after the last stored record. It also allows forward and backward movement of pointer that points to the current record of the file without reading or writing.

Sequential File



Sequential File

Advantages of sequential file

- It is simple to program and easy to design.
- Sequential file is best use if storage space.

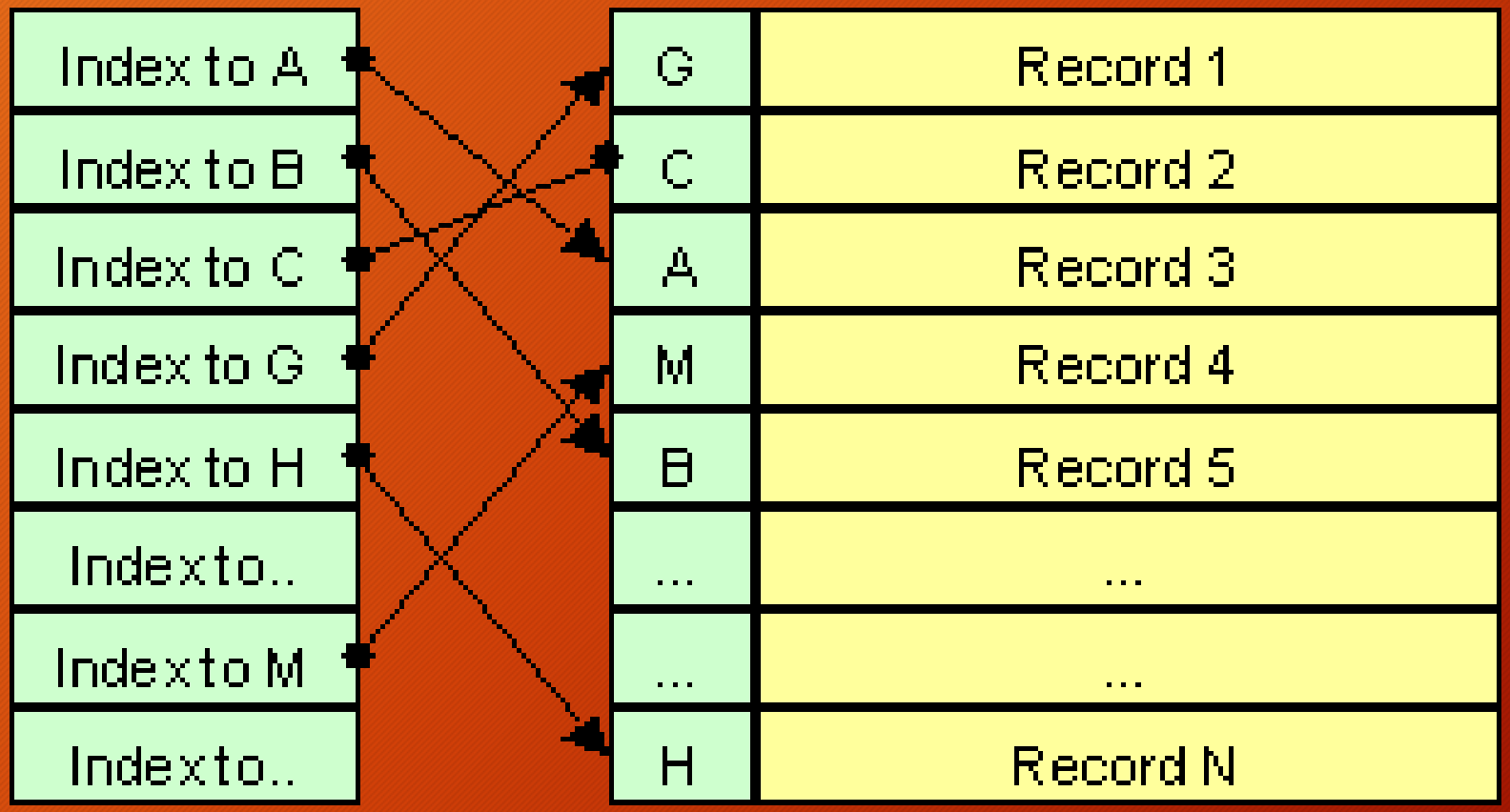
Disadvantages of sequential file

- Sequential file is time consuming process.
- It has high data redundancy.
- Random searching is not possible.

Indexed Sequential File

- A file organization where the records can be accessed directly as well as sequentially is called indexed sequential file. The capability of directly accessing a record based on a key can only be achieved if the external storage device supports this type of access i.e. magnetic drums and disks.
- No attempt is made to store the records in any particular order.
- An index is held giving the address of every record in the file. To find a record, you first consult the index.
- Index is a sequential file which contains index as its record. Index files contain two fields for each file i.e. a Key field and a pointer pointing to some record in the main file. To find a specific field in main file, index is searched for the key value required. The pointer related to key field searches the record at the location it points to.

Indexed Sequential File



Indexed Sequential File

Advantages of Indexed sequential access file organization

- In indexed sequential access file, sequential file and random file access is possible.
- It accesses the records very fast if the index table is properly organized.
- The records can be inserted in the middle of the file.
- It provides quick access for sequential and direct processing.
- It reduces the degree of the sequential search.

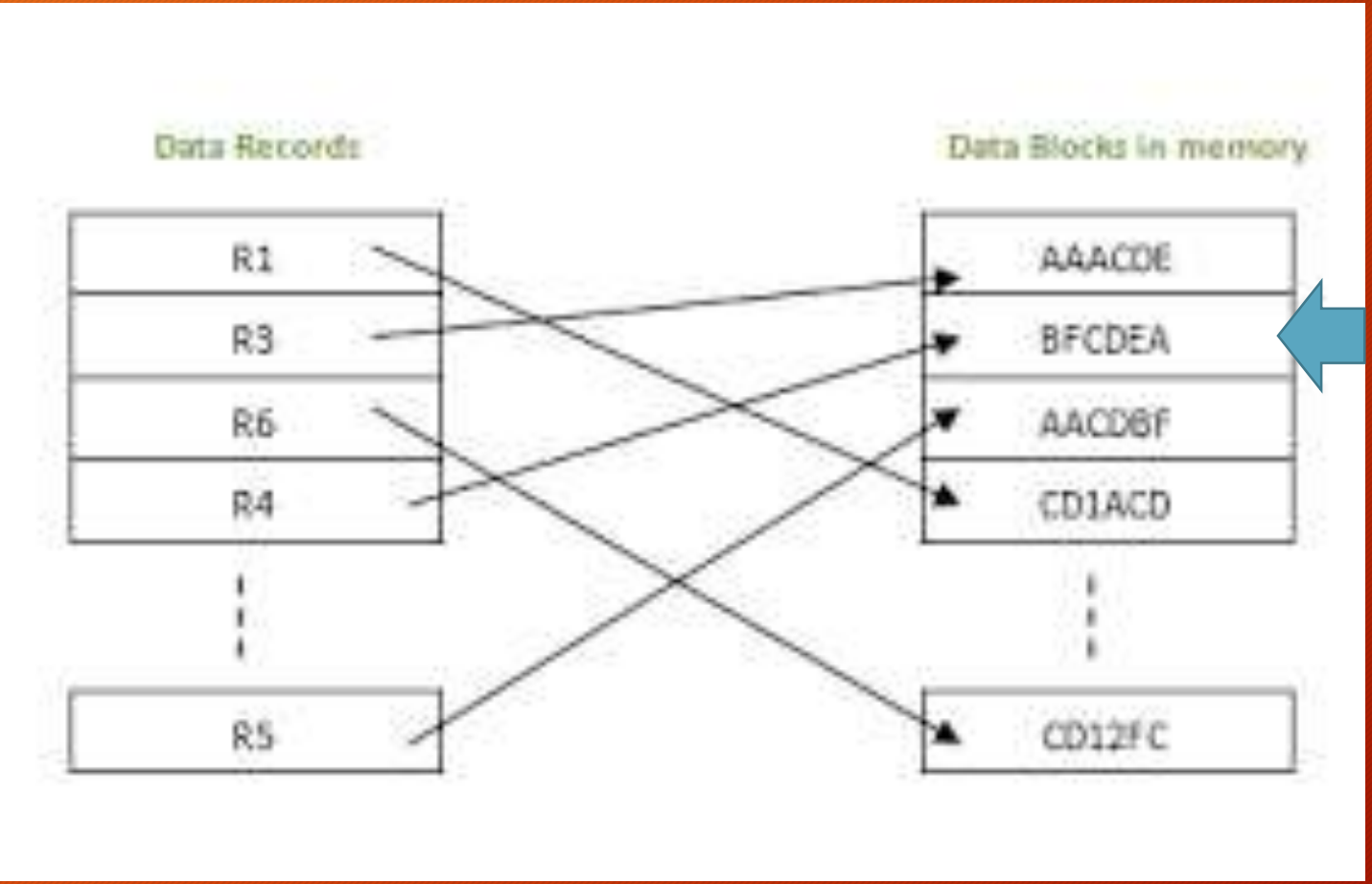
Disadvantages of Indexed sequential access file organization

- Indexed sequential access file requires unique keys and periodic reorganization.
- Indexed sequential access file takes longer time to search the index for the data access or retrieval.
- It requires more storage space.
- It is expensive because it requires special software.
- It is less efficient in the use of storage space as compared to other file organizations.

Direct Access File

- Direct access file is also known as random access or relative file organization.
- This is also known as "self-indexing" organisation.
- 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 records do 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.
- The total storage space is divided into areas called buckets, each one large enough to hold a number of records; the buckets are numbered from zero upwards.
- An algorithm, known as a "hashing" or "key-
- transformation" algorithm, converts a record-key to a number, and the record is stored in the bucket with that number.
- Only one disk access is required for any record, so random access is good, which is what this system was designed for.

Direct Access File



Address Calculated according to the Key Value.
The Hash value.

Direct Access

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.

Multi-key File Organization and Access Methods

- The ability to search on many keys is enabled by building multiple index files (multikey file organization) “on top of” the data file.
- The physical database then consists of one or more data files and many index files.
- Each index file supports access by a particular field or group of fields.
- There are mainly two approaches for providing additional access paths into a file of data records.
  - Multilist file organisation
  - Inverted file organization

Multilist File Organisation

- The linkage between an index and the file of data records is called multilist organisation.
- A multilist file maintains an index for each secondary key.
- The index for secondary key contains, instead of a list of primary keys related to that secondary key, only one primary key value related to that secondary key.
- That record will be linked to other records containing the same secondary key in the data file.

Multilist File Organisation

Book_Id	*Book_title*	*Category*	*Page_count*	
354	Ransack	Novel	200	Record 1
556	Differential Calculus	Textbook	450	Record 2
489	C++	Textbook	800	Record 3
678	Call Away	Novel	200	Record 4
456	Introduction to German Language	Language Book	200	Record 5
887	Learning French Language	Language Book	500	Record 6

BookId	Length	Pointer
500	3	1
700	2	2
900	1	6

Category	Length	Pointer
Language Book	2	5
Novel	2	1
TextBook	2	2

A Multilist on BookId And Category.

Inverted File Organization

- In inverted file organisation, a linkage is provided between an index and thefile of data records.
- A key's inverted index contains all of the values that the key presently has in the records of the data file. Each key-value entry in the inverted index points to all of the data records that have the corresponding value.
- Inverted files represent one extreme of file organisation in which only the index structures are important.
- The records themselves may be stored in any way.

Inverted File Organization

Book_Id	Book_title	Category	Page_count	
354	Ransack	Novel	200	Record 1
556	Differential Calculus	Textbook	450	Record 2
489	C++	Textbook	800	Record 3
678	Call Away	Novel	200	Record 4
456	Introduction to German Language	Language Book	200	Record 5
887	Learning French Language	Language Book	500	Record 6

Category	Pointer
Language Book	Rec_5, Rec_6
Novel	Rec_1, Rec_4
Textbook	Rec_2, Rec_3

Book_Id	Pointer
500	Rec_1, Rec_3, Rec_5
700	Rec_2, Rec_4
900	Rec_6

An inverted list index for BookId And Category.