Unit 6
File Organization

Definitions

- FILE
  - Collection of related records
- RECORD
  - Collection of related fields (e.G. Name, age)
- DATABASE
  - Collection of related files

Definition of File Organization

- File organization means the way data is stored so that it can be retrieved when needed.
- It includes the physical order and layout of records on Storage devices
- The techniques used to find and retrieve stored records are called **access methods**

Logical vs. Physical Organization of Data

- Logical organization
  - The abstract way that the computer program is able to access the data
  - Use of logical structures (e.G. Linked lists)
- Physical organization
  - The actual physical structure of data in memory
  - I.E. What the sequence of bits look like in memory

Basics (General Idea)

- In case of file organization, database is stored in form of collection of files.
- Each file is organized logically as a sequence of multiple records.
- A record is sequence of fields in a relation.
- Records are mapped onto disk blocks for storage.
- Size of such records on file system may vary.
- One approach to mapping database to files is to store records of one length in a given file called as fixed length records.
- An alternative approach is variable length records

Records in files: Fixed Length Record

- Let us consider following example

```
    Type student=record
          Sname : char(20);
         Sid : char(4);
        Fees : real;
      End
```

- If each character occupies one byte, an integer occupies 4 bytes, real occupies 8 bytes then student record is 32 bytes long

Disadvantage

- Block size should be multiple of 32 .
- It would then require two block accesses to read or write a record which is more than size 32.
- It is difficult to delete a record from such fix structure.

Variable length records

- Variable length records arise in database systems in several ways:

  1. Storage of multiple record types in a file.
  2. Record types that allow variable lengths for one or more fields.
  3. Record types that allow repeating fields

Variable length records

- *Type student=record*

```
        Class _name : char(20);
        Student_info : array [1..∞ ] Of record;
                        Sid : char(4);
                        Fees : real;
                        End
        End
```

- We define student-info as an array with an arbitrary number of elements ,so that there is no limit on how large a record can be.

Types of File Organization

- Sequential file organization
- Indexed sequential file organization
- Direct or random file organization

Sequential File Organization

- Records are arranged in physical sequence by the value of some field called the sequence field.
- The field chosen is the key field, one unique values that are used to identify records.
- The records are arranged on the storage devices ,often magnetic tapes in increasing and decreasing order by the value of the sequence field.
- Access to records in a sequential file is serial. To reach a particular record, all the preceding records must be read.

Sequential File Organization

- It is the oldest method of file organization
- This organization is simple
- Easy to understand and easy to manage.
- It is best suited for sequential access retrieving records one after the another in the same order in which they are stored.
- With this type of organization , insertion, updation and deletion are done by rewriting the entire file.
- Suitable for applications such as payroll system.

Sequential File Organization

- Records in a file are stored sequentially (in order) by some key field

  2480 bob

  2569 alice

  3020 paul

- Originally designed to operate on magnetic tapes

Key	Record Address
a	4
b	7
c	5
d	3
e	12
m	9
n	10
p	2
s	11
t	6
z	1

File Operations

- Most common operations on files
    - Read
    - Write

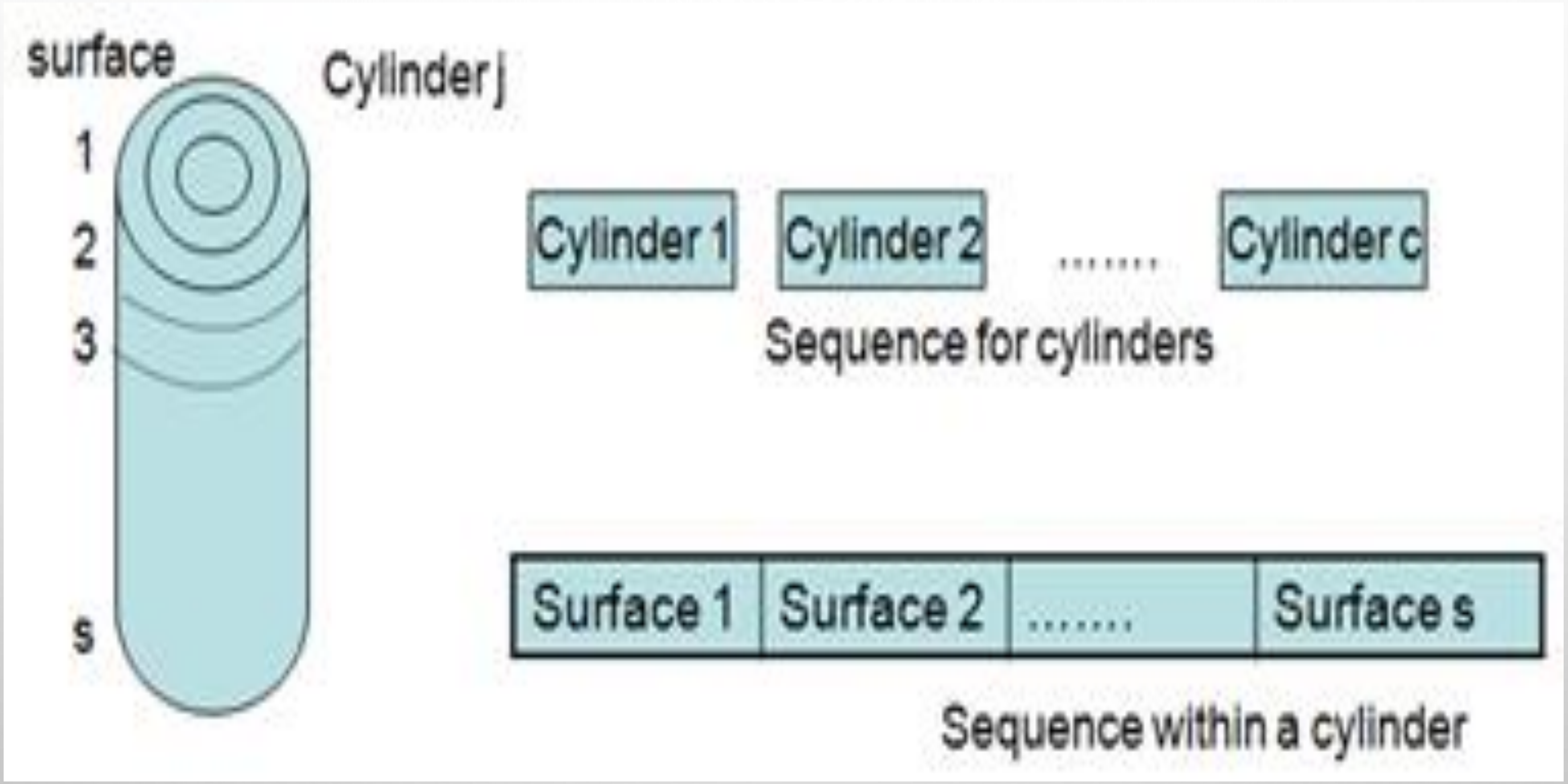
- File pointer

- Whenever a file is opened for read or write operation a file pointer is maintained to keep track of the current position in the file

File Operations

- **READ OPERATION**
    - Reads the next portion of the file
    - Automatically advances file pointer
- **WRITE OPERATION**
    - Appends data to the end of the file
    - Advances pointer to the new end of file
- **RESET**
    - Reset to beginning

Sequential File Organization



Applications

- PAYROLL OF EMPLOYEES
- STUDENT DATA PROCESSING

Advantages and Disadvantages

- ADVANTAGES:
  - Simplicity
  - Less overheads
  - Sequential file is best use if storage space.

- DISADVANTAGES:
  - Difficulty in searching
  - Problem with record deletion for queries.
  - Sequential file is time consuming process.
  - It has high data redundancy

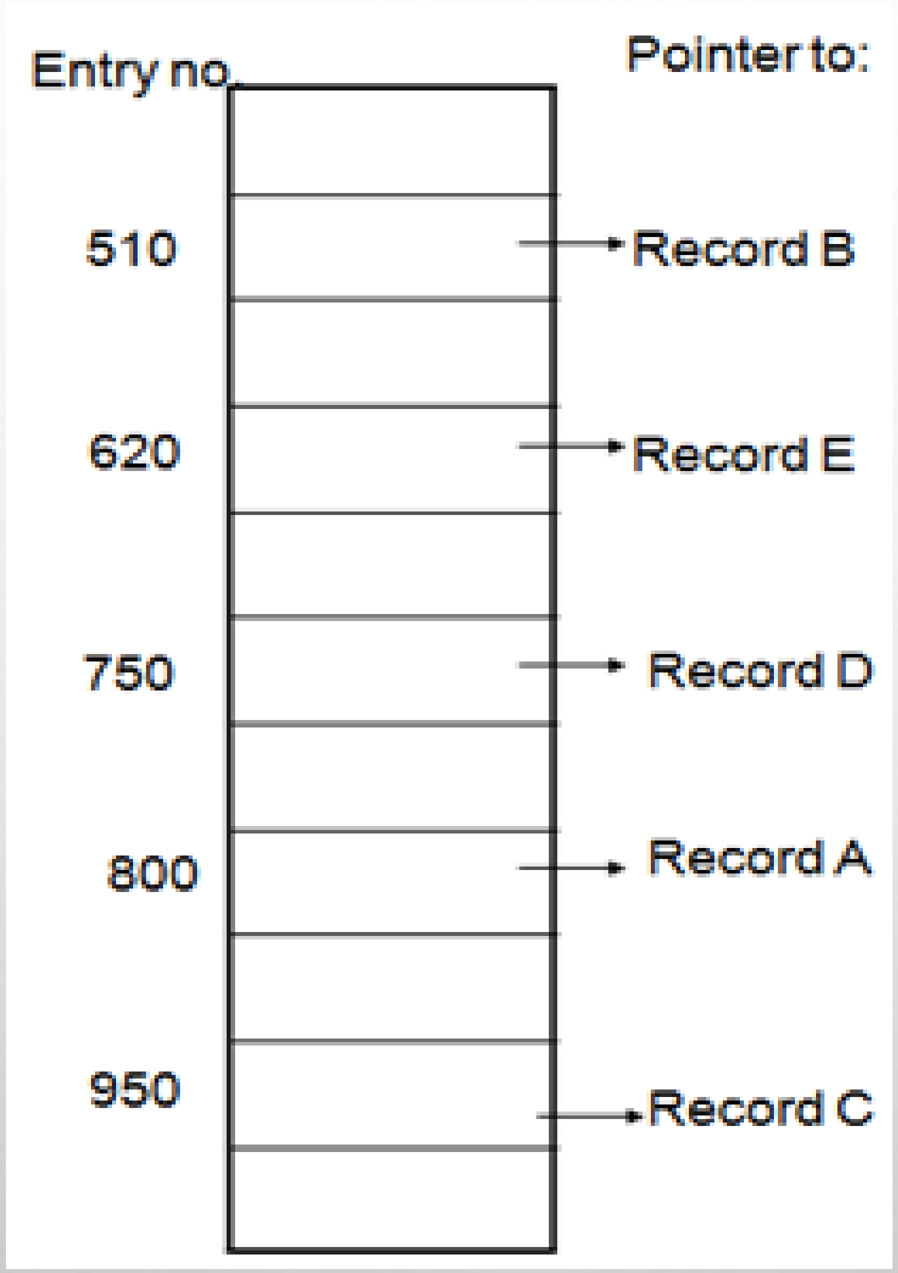
DIRECT FILE ORGANIZATION

- 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 (dasd), such as hard disk. The records are randomly placed throughout the file.
- The records does not need to be in sequence because they are updated directly and rewritten back in the same location.
- The records are stored at known address. Address is calculated by applying a mathematical function to the key field.

DIRECT FILE ORGANIZATION

- 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.
- **Example :** any information retrieval system. Eg train timetable system.

DIRECT FILE ORGANIZATION



Advantages and Disadvantages

ADVANTAGES:

- In direct access file, sorting of the records are not required.
- It accesses the desired records immediately.
- It updates several files quickly.

DISADVANTAGES:

- Direct access file does not provide back up facility.
- It is expensive.
- It has less storage space as compared to sequential file.

INDEXED SEQUENTIAL ACCESS METHOD (ISAM)

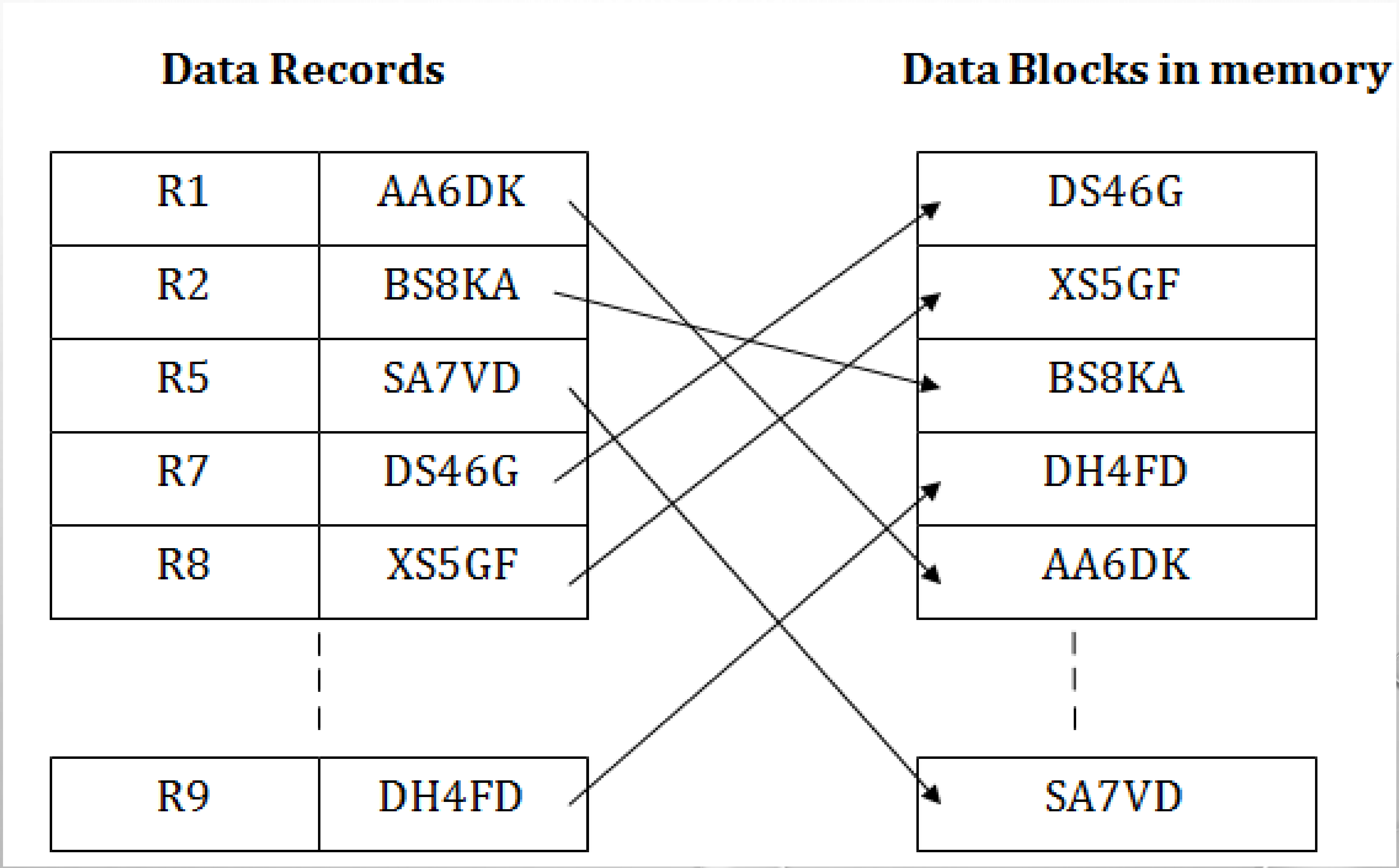
- Indexed sequential access file combines both sequential file and direct access file organization.
- The records in this type of file are organized in sequence and an index table is used to speed up
- Access to the records without requiring a search of the entire file.
- The records of the file can be stored in random sequence but the index table is in stored sequence on the key value.
- File can be both randomly as well as sequentially accessed.

INDEXED SEQUENTIAL ACCESS METHOD (ISAM)

- Records can be updated deleted and inserted in indexed file organization.
- 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.

Implementation

- To implement the concept of indexed sequential file organizations, we consider an approach in which the index part and data part reside on a separate file.
- The index file has a tree structure and data file has a sequential structure. Since the data file is sequenced, it is not necessary for the index to have an entry for each record
- When the new records are inserted in the data file, the sequence of records need to be preserved and also the index is accordingly updated.
- Two approaches used to implement indexes are static indexes and dynamic indexes.


Data Records
Data Blocks in memory
R1
AA6DK
R2
BS8KA
R5
SA7VD
R7
DS46G
R8
XS5GF
R9
DH4FD
DS46G
XS5GF
BS8KA
DH4FD
AA6DK
SA7VD

Advantages and Disadvantages

ADVANTAGES:

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.

Advantages and Disadvantages

DISADVANTAGES:

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.

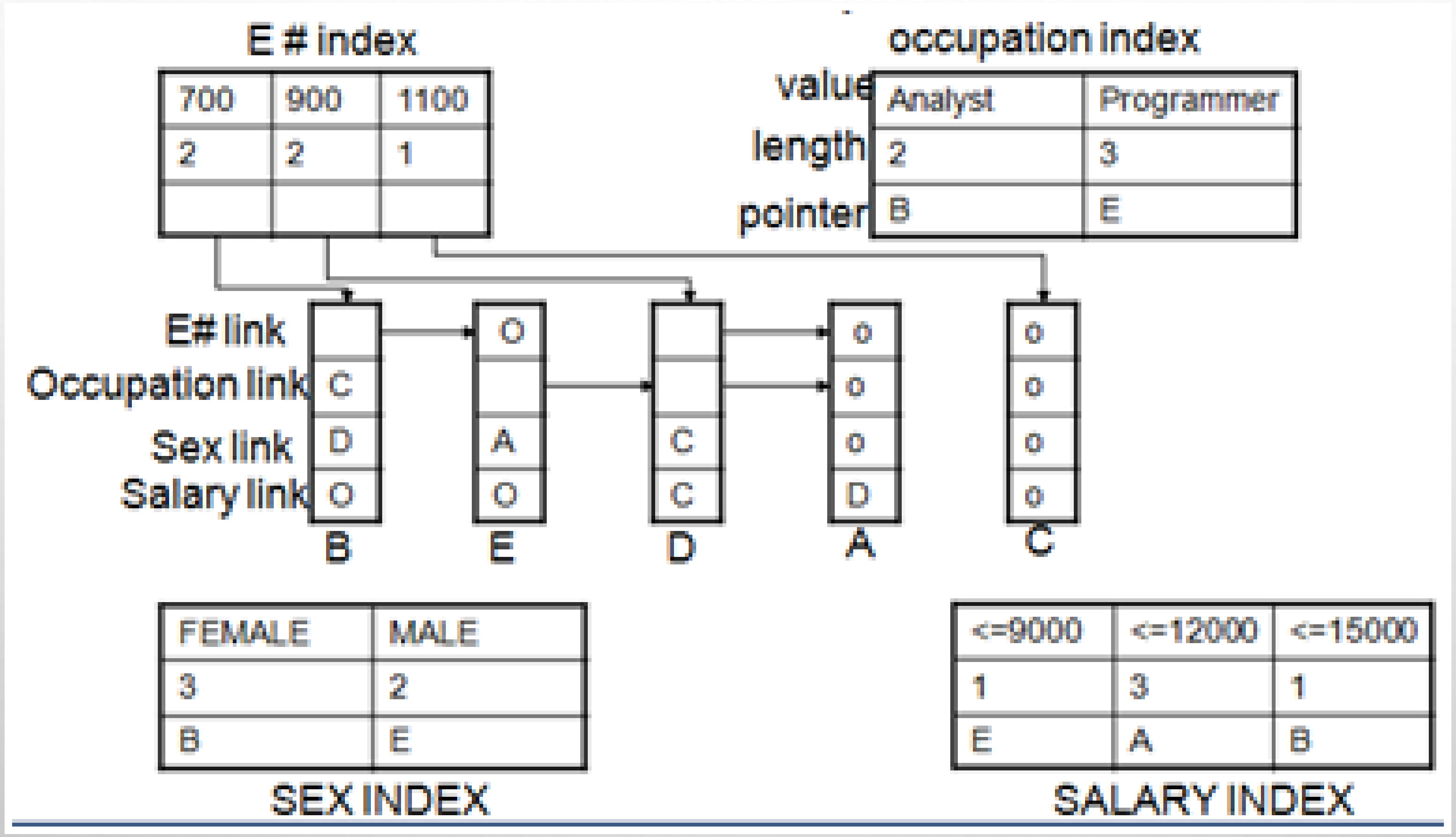
It is less efficient in the use of storage

Space as compared to other file Organizations.

Linked organization

- In linked organization, the physical sequence of records is different from the logical sequence of records
- The next logical record is obtained by following a link value from the present record
  - ✓ Multilist Files
  - ✓ Coral Rings
  - ✓ Inverted Files
  - ✓ Cellular Partitions

Linked organization



E # index

700	900	1100
2	2	1

occupation index

value	Analyst	Programmer
length	2	3
pointer	B	E

	B	E	D	A	C
E# link		O		o	o
Occupation link	C			o	o
Sex link	D	A	C	o	o
Salary link	O	O	C	D	o

SEX INDEX

FEMALE	MALE
3	2
B	E

SALARY INDEX

<=9000	<=12000	<=15000
1	3	1
E	A	B

Linked organization

- Two approaches for providing additional access paths into a file of data records.
  - ✓ Multilist file organisation
  - ✓ Inverted file organisation

Multilist File Organisation

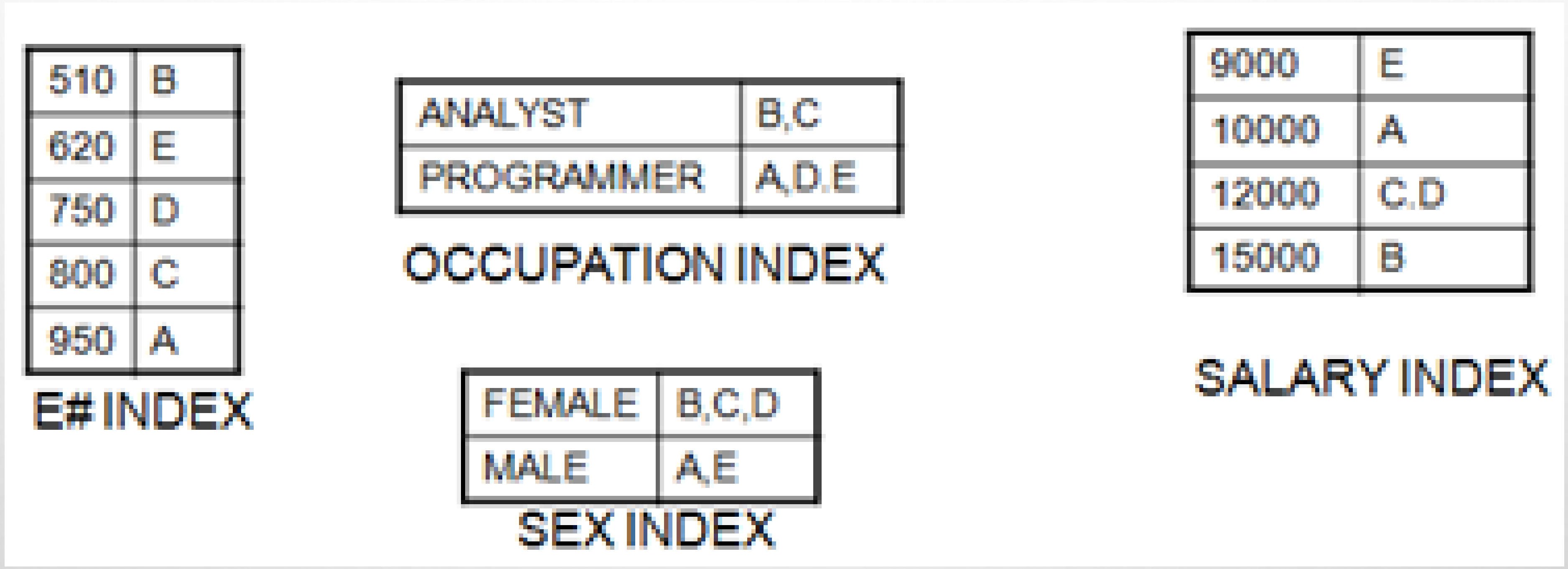
- The basic approach to providing the linkage between an index and the file of data records is called multilist organisation.
- A multilist file maintains an index for each secondary key.
- The index for secondary key contains, only one primary key value related to that secondary key.
- That record will be linked to other records containing the same secondary key in the data file.
- The multi-list organization differs from inverted file. While the entry in the inverted file index for a key value has a pointer to each data record with that key value, the entry in the multi-list index for a key value has just one pointer to the first data record with that key value.

Multilist File Organisation

- Linking records together in order of increasing primary key value facilitates easy insertion and deletion once the place at which the insertion or deletion to be made is known.
- Searching for a record with a given primary key value is difficult when no index is available, since the only search possible is a sequential search.

Inverted files

- Inverted files are similar to multilists. Multilists records with the same key value are linked together with link information being kept in individual record. In case of inverted files the link information is kept in index itself.



510	B
620	E
750	D
800	C
950	A

E# INDEX

ANALYST	B,C
PROGRAMMER	A,D.E

OCCUPATION INDEX

FEMALE	B,C,D
MALE	A,E

SEX INDEX

9000	E
10000	A
12000	C.D
15000	B

SALARY INDEX

Inverted files

- The retrieval works in two steps. In the first step, the indexes are processed to obtain a list of records satisfying the query and in the second, these records are retrieved using the list.
- The no. of disk accesses needed is equal to the no. of records being retrieved + the no. to process the indexes.
- Inverted files represent one extreme of file organization in which only the index structures are important. The records themselves can be stored in any way.
- Inverted files may also result in space saving compared with other file structures when record retrieval doesn't require retrieval of key fields. In this case key fields may be deleted from the records unlike multilist structures.

Inverted files and Multilist files

- Both inverted files and multilist files have:
  - ✓ An index for each secondary key.
  - ✓ An index entry for each distinct value of the secondary key.
  - ✓ The index may be tabular or tree-structured.
  - ✓ The entries in an index may or may not be sorted.
  - ✓ The pointers to data records may be direct or indirect.
- The indexes differ in that
  - ✓ An entry in an inverted index has a pointer to each data record with that value.
  - ✓ An entry in a multilist index has a pointer to the first data record with that value.

Cellular partitions

- To reduce the file search times, the storage media may be divided into cells. A cell may be an entire disk pack or it may simply be a cylinder. Lists are localized to lie within a cell.
- Thus if we had a multilist organization in which the list for key1=prog list included records on several different cylinders then we could break the list into several smaller lists where each prog list included only those records in the same cylinder.
- The index entry for prog will now contain several entries of the type (addr, length) where addr is a pointer to start of a list of records with key1=prog and length is the no. of records on the list.
- By doing this all records of the same cell may be accessed without moving the read/write heads.

Coral Rings

- The coral ring structure is an adaptation of the doubly linked multilist structure.
- Each list is structured as a circular list with a head node.
- The head node for the list for key value Ki = X will have an information field with value X.
- The field for key Ki, is replaced by a link field.
- Thus,associated with each record. Y, and key, K, in a coral ring there are two link fields: ALINK(Y,i) and BLINK (Y,i).
- The ALINK field is used to link together all records with the same value for key Ki.
- The ALINKS form a circular list with a head node whose information field retains the value of Ki for the records in this ring.
- The BLINK field for some records is a back pointer and for others it is a pointer to the head node.
- To distinguish between these two cases another field FLAG(Y, i) is used. FLAG(Y,i) = 1 if BLINK(Y,i) is a back pointer and FLAG(Y,i) = 0 othcrwise.

Coral Rings

- In practice the FLAG and BLINK fields may be combined with BLINK(Y,i) 0 when it is a back pointer and 0 when it is a pointer to the head node.
- When the BLINK field of a record BLINK(Y,i) is used as a back pointer, it points to nearest record. Z, preceding it in is circular list for K, having BLINK(Z,i) also a back pointer.
- In any given circular list, all records with back pointers form another circular list in the reverse direction. The presence of these back pointers makes it possible to carry out a deletion without having to start at the front of each list containing the record being deleted in order to determine the preceding records in these lists.
- Since these BLINK fields will usually be smaller than the original key fields they replace, an overall saving in space will ensue.
- This is, however, obtained at the expense of increased retrieval time.
- Indexes are maintained as for multilists. Index entries now point to head nodes. As in the case of multilists, an individual node may be a member of several rings on different keys.

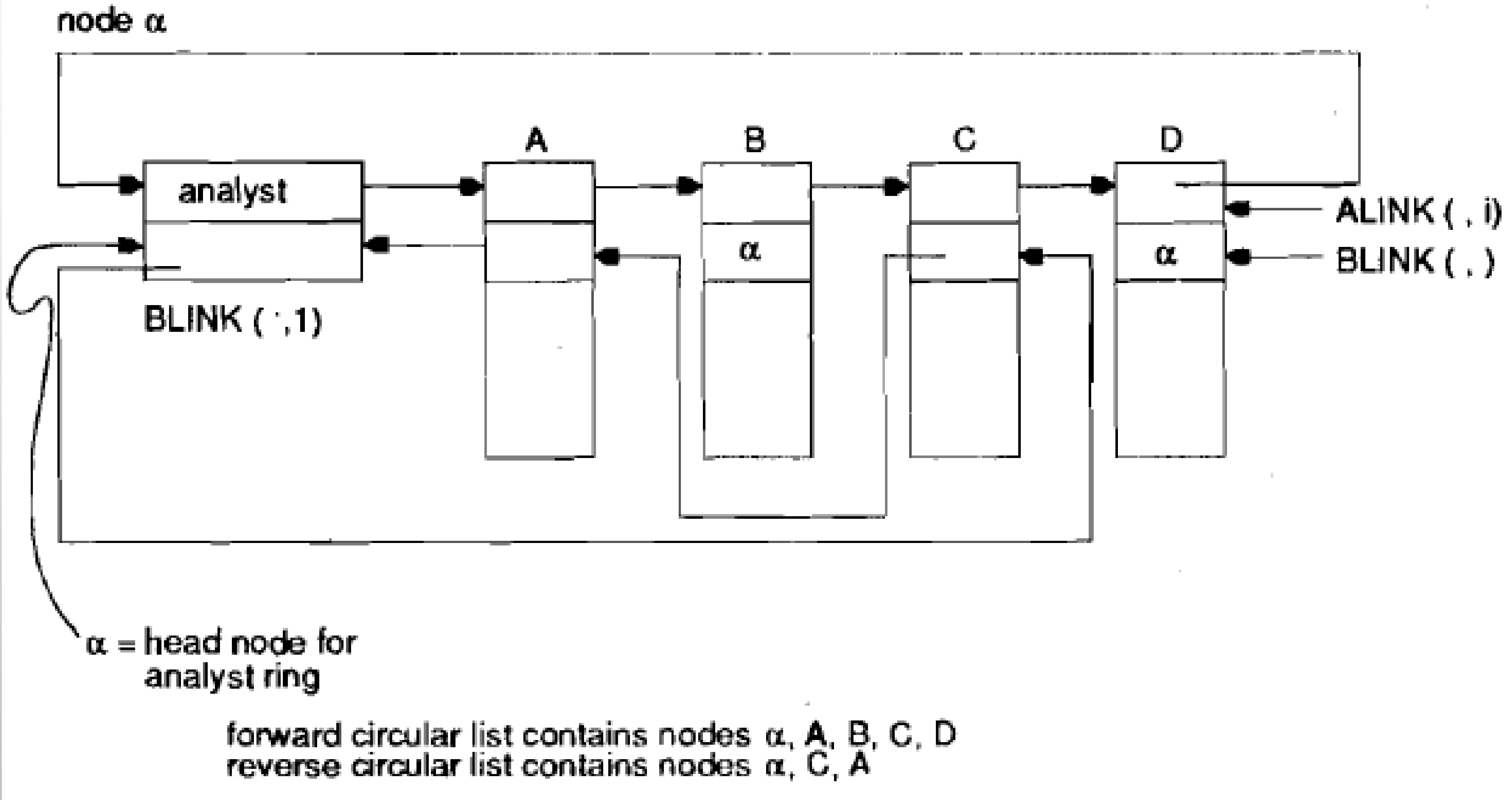

node α
analyst
A
B
C
D
α
α
ALINK (, i)
BLINK (,)
BLINK (·,1)
α = head node for analyst ring
forward circular list contains nodes α, A, B, C, D
reverse circular list contains nodes α, C, A

Figure 26 : Coral rings for analysts in a hypothetical file

Reference links

- http://www.egyankosh.ac.in/bitstream/123456789/26258/1/Unit-2.pdf
- http://egyankosh.ac.in/bitstream/123456789/12255/1/Unit-3.pdf