# FILE SYSTEM

***We have three essential requirements for long-term information storage:***

1. It must be possible to store a very large amount of information.

2. The information must survive the termination of the process using it.

3. Multiple processes must be able to access the information at once.

# File Naming

Some OS distinguish between upper- and lowercase letters: UNIX
But on MS-DOS it doesn't sens.

**File name:** maria, Maria, and MARIA

File systems for old OS are **FAT-16, FAT32, NTFS**(nativ),

Second file system **ReFS**(Resilient)

# File extentions

- **File extentions** *t*wo-part file names:
  In MS-DOS 1 to 3 characters
  In UNIX up to user, two or more extentions as *homepage.html.zip*

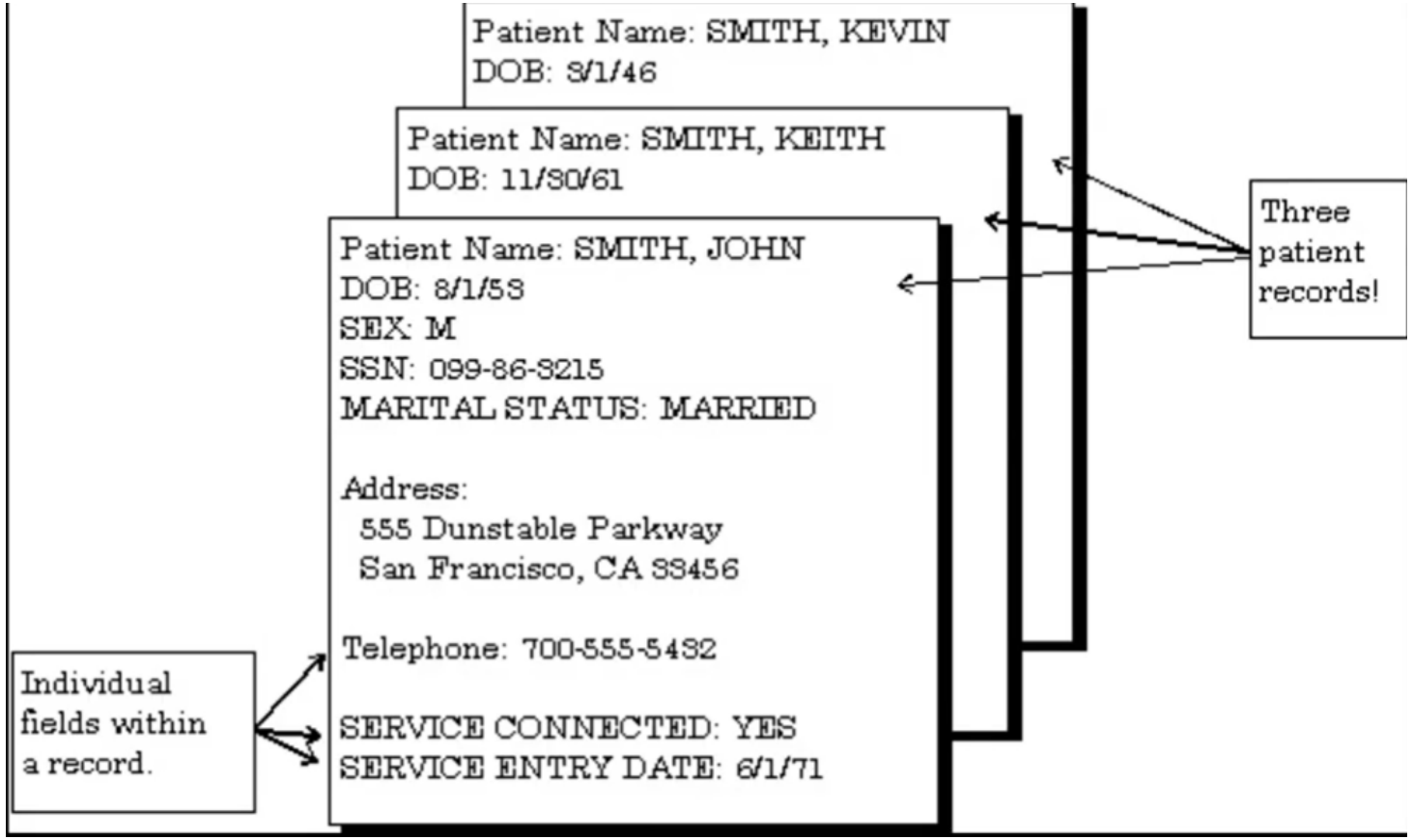
| Extension | Meaning |
|---|---|
| .bak | Backup file |
| .c | C source program |
| .gif | Compuserve Graphical Interchange Format image |
| .hlp | Help file |
| .html | World Wide Web HyperText Markup Language document |
| .jpg | Still picture encoded with the JPEG standard |
| .mp3 | Music encoded in MPEG layer 3 audio format |
| .mpg | Movie encoded with the MPEG standard |
| .o | Object file (compiler output, not yet linked) |
| .pdf | Portable Document Format file |
| .ps | PostScript file |
| .tex | Input for the TEX formatting program |
| .txt | General text file |
| .zip | Compressed archive |

- **Create:** A new file is defined and positioned within the structure of files.
- **Delete:** A file is removed from the file structure and subsequently destroyed.
- **Open:** An existing file is declared to be "opened" by a process, allowing the process to perform functions on the file.
- **Close:** The file is closed with respect to a process, so the process no longer may perform functions on the file, until the process opens the file again.
- **Read:** A process reads all or a portion of the data in a file.
- **Write:** A process updates a file, either by adding new data that expands the size of the file, or by changing the values of existing data items in the file.

# File System managament of the OS

# File Structure

- • Field
  • Record
  • File
  • Database



# Users and applications wish to make use of files. Typical operations that must be supported include the following

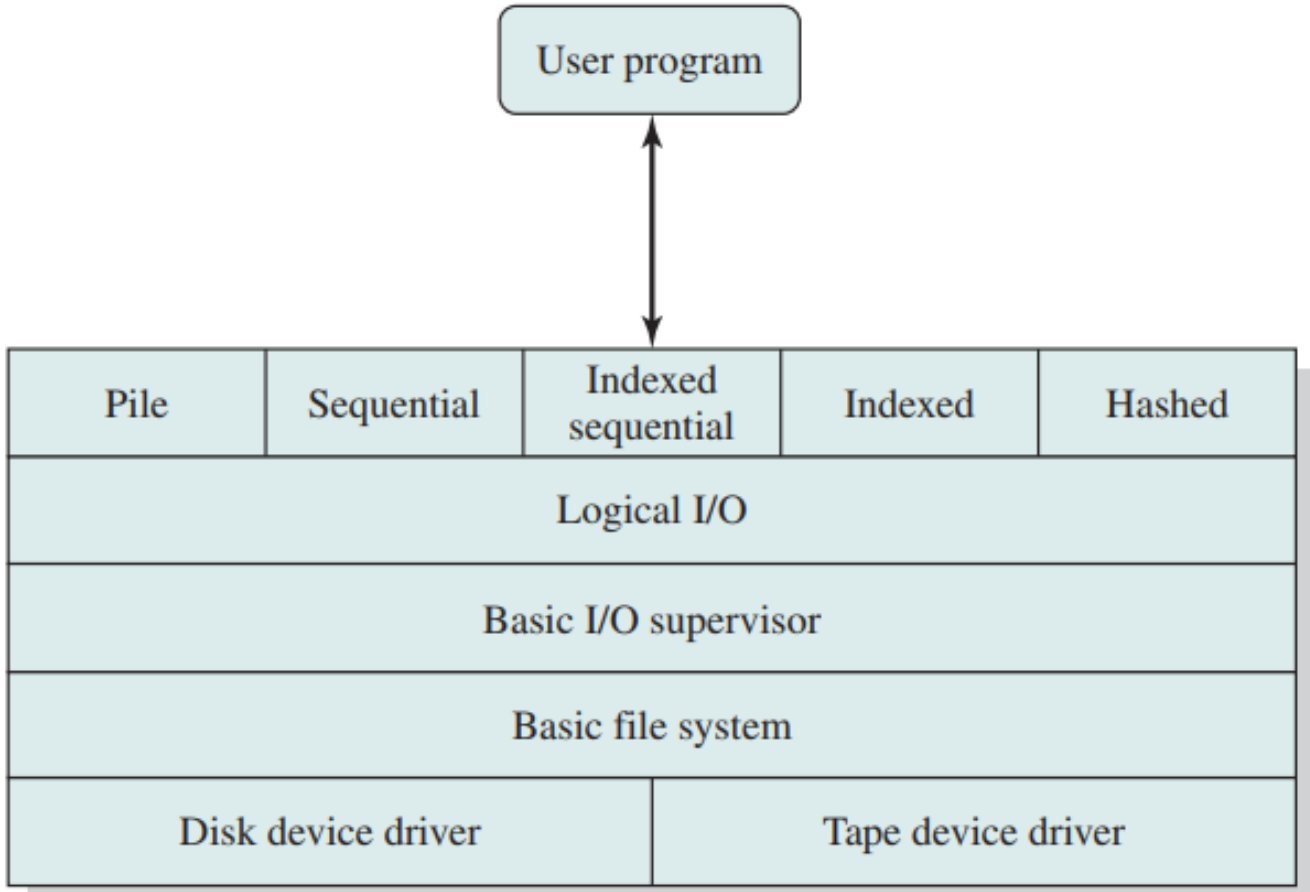
- Retrieve_All:
- Retrieve_One:
- Retrieve_Next:
- Retrieve_Previous: (Similar to Retrieve_Next,)
- Insert_One:
- Delete_One:
- Update_One:
- Retrieve_Few:

- **On UNIX and UNIX-like systems, the basic file structure is just a stream of bytes.**

# File Management Systems

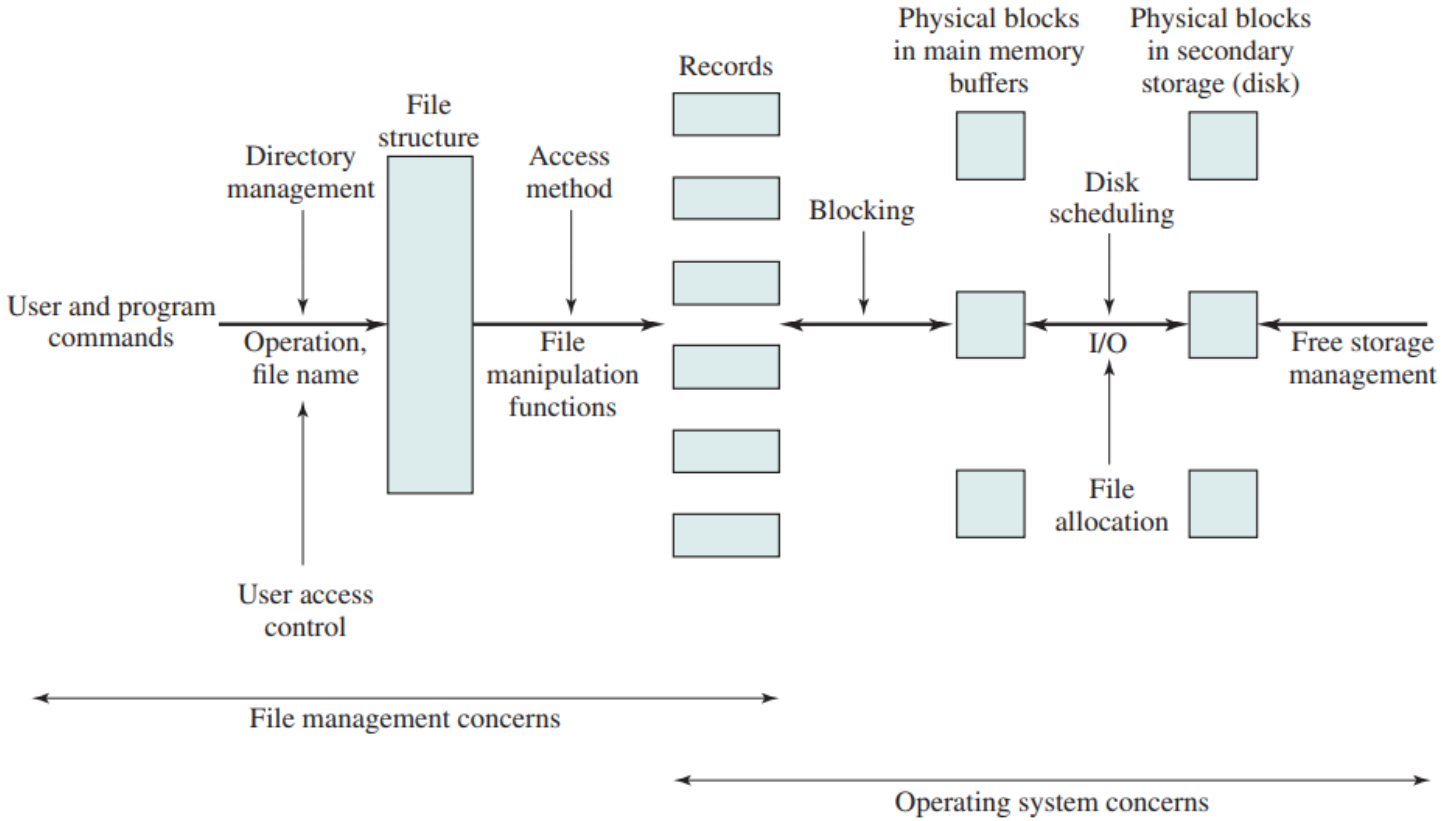
1. Each user should be able to create, delete, read, write, and modify files.
2. Each user may have controlled access to other users' files.
3. Each user may control what types of accesses are allowed to the user's files.
4. Each user should be able to move data between files.
5. Each user should be able to back up and recover the user's files in case of damage.
6. Each user should be able to access his or her files by name rather than by numeric identifier

# File System Software Architecture



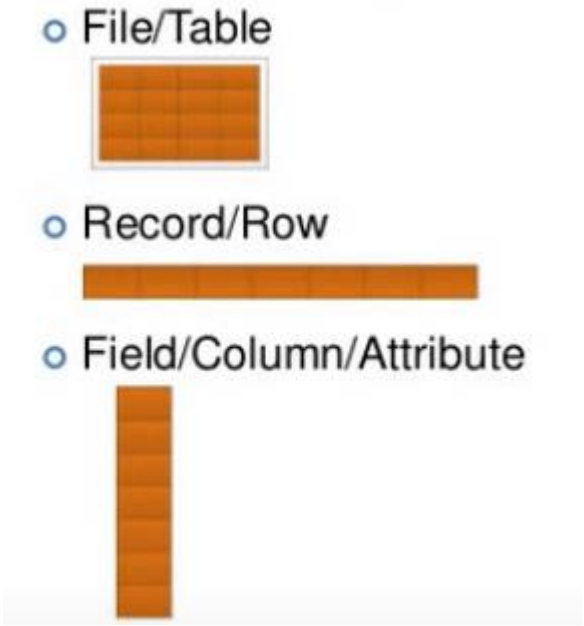

**File System Software Architecture**

# Elements of File Management




**Elements of File Management**

# What is File? What is File Organization?




**File:**

- File is a collection of records related to each other.
- The file size is limited by the size of memory and storage medium.

**File organization:**

- File organization refers to the way data is tored in a file.

**Objectives of File Organization:**

- It contains an optimal selection of records, i.e., records can be selected as fast as possible.
- To perform insert, delete or update transaction on the records should be quick and esasy.
- The dublicate records cannot be induced as a result of insert, update or delete.
- For the minimum cost of storage records should be stored efficiently.

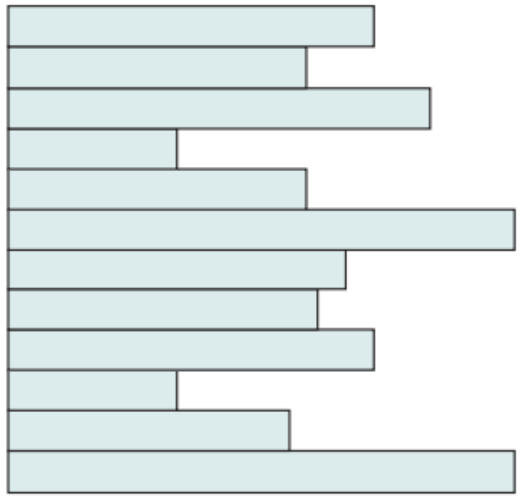
# FILE ORGANIZATION AND ACCESS

**In choosing a file organization, several criteria are important:**

- Short access time
- Ease of update
- Economy of storage
- Simple maintenance
- Reliability

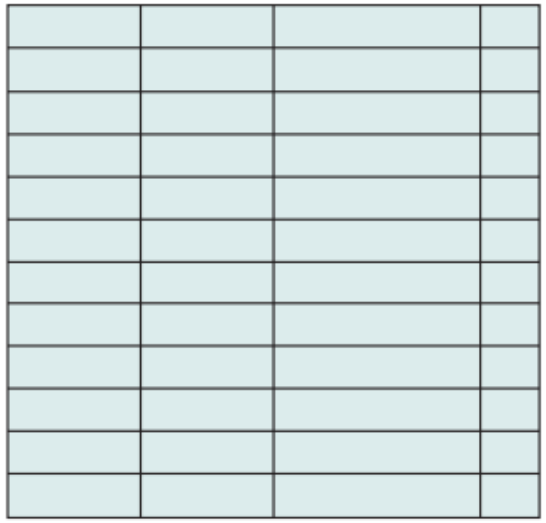
# File organization structures & Data structure

1. The pile
2. The sequential file
3. The indexed sequential file
4. The indexed file
5. The direct, or hashed, file



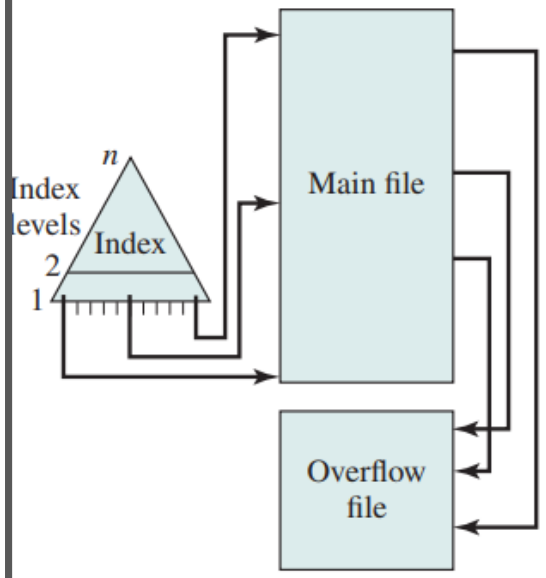
Variable-length records
Variable set of fields
Chronological order

(a) Pile file

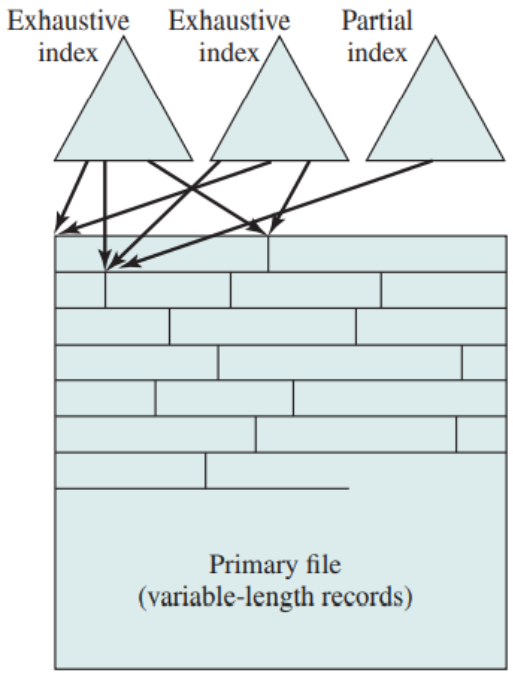


Fixed-length records
Fixed set of fields in fixed order
Sequential order based on key field

(b) Sequential file




(c) Indexed sequential file

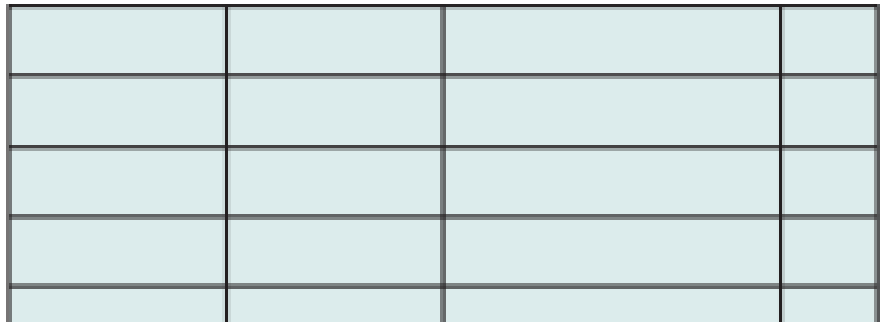


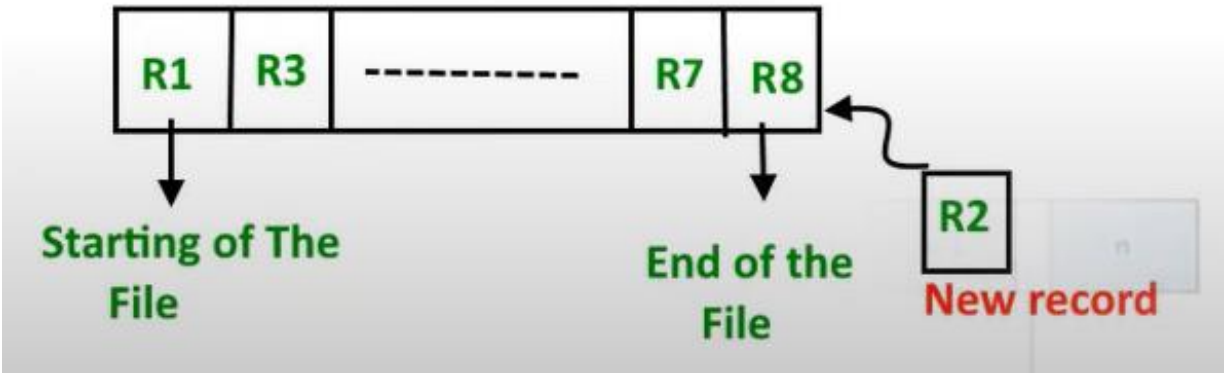

(d) Indexed file

# File organization structures

# The sequential file

- The easiest method for file Organization is Sequential method.
- In this method the file are stored one after another in a sequential manner.
- The records are arranged on the ascending or descending order of a key field.
- Sequential file search stars form the beginning of the file and the records can be added at the end of the file.
- In sequential file, it isnot possible to add a records in the middle of the file without rewriting  the file.



# Advantags & Disadvantage sequential file

**Advantags of Sequential File**

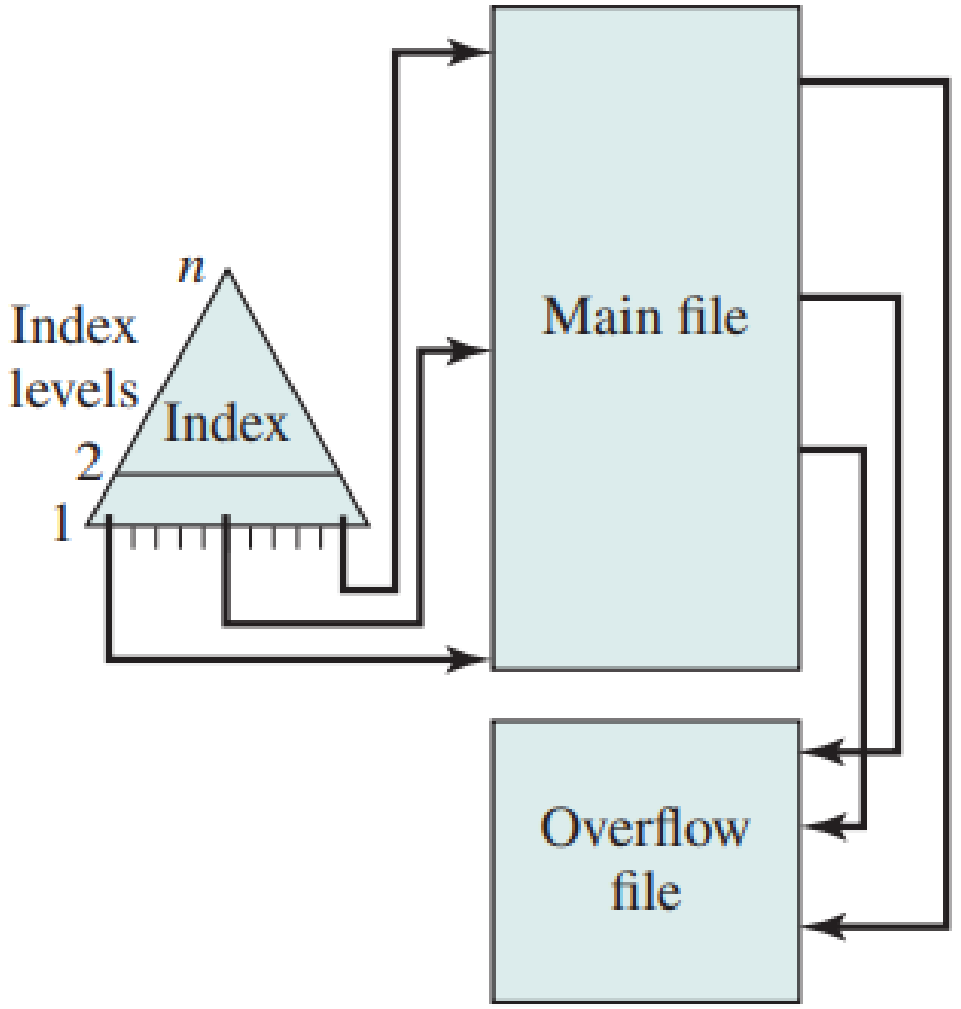
- It is simple in design.
- It require no much efford to store the data.
- Fast and efficient method for huge amount of data.
- File can be easity stored in magnetic tapes i.e cheaper storage mechanism

**Disadvantage of Sequential File**

- Sequential file is time consuming process.
- Random search is not possible.
- Stored file method is inefficient as it takes time and space for sorting records
- Sequential records cannot support modern technologies that require fast access to stores records.

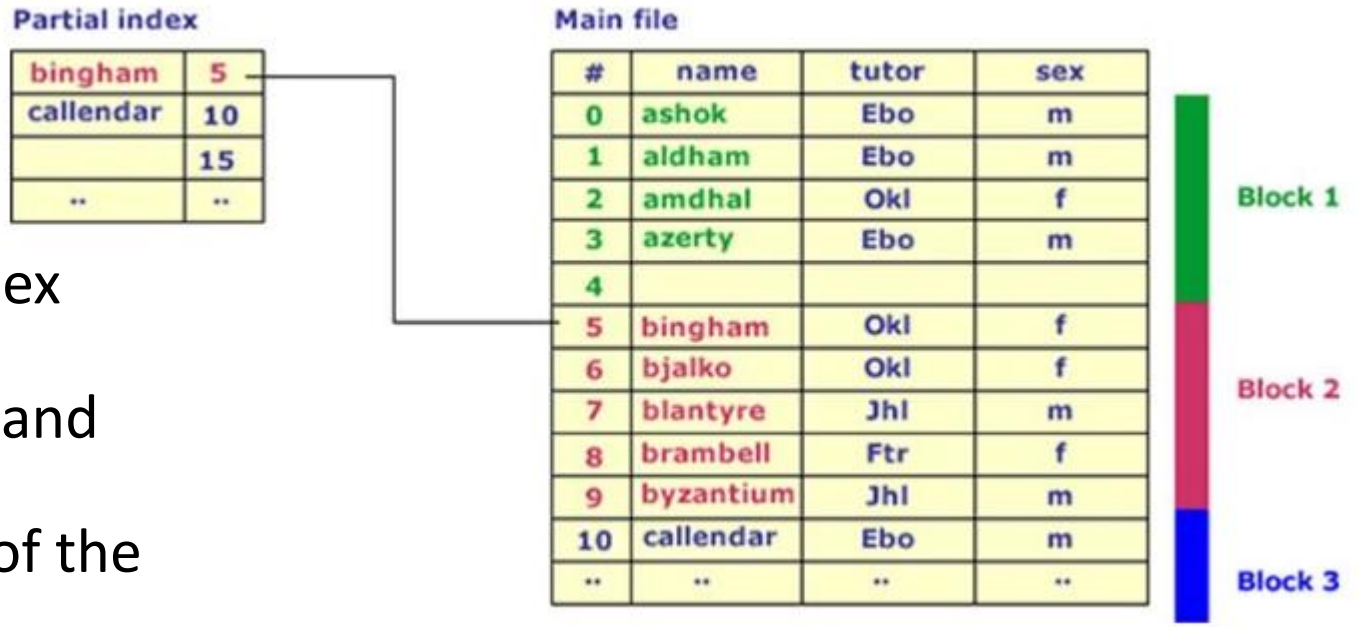
# The indexed sequential file

- ISAM method is an advanced sequential file organization.
- In this method, records are stored in the file using the primary key.
- An index value is generated for each primary key and mapped with the record.
- Index contains the address of the record in the file.
- The data can be access either sequentially or randomly using the index.
- The index is sored in a file and read into memory when the file is opened.
- If any record has to be retrieved based on its index value, then the address of the data block is fetched and the record is retrieved from the momory.




(c) Indexed sequential file

# The indexed sequential file




**Advantags of Index Sequential File**

It accesses the records very fast if the index table is properly organized.

Search a record in huge database is quick and easy.

The record can be inserted in the middle of the file.

**Disadvantage of Index Sequential File**

- Index sequential access file requires unique key and periodic reorganization.
- It requires more storage space
- When new records are inserted, the these files have to be reconstructed to maintain the sequence.
- When the record is deleted then the space used by it needs released. Otherwise, the performance of the database will slow down