# Unit VI : File Management.

(Weightage : 10 marks)

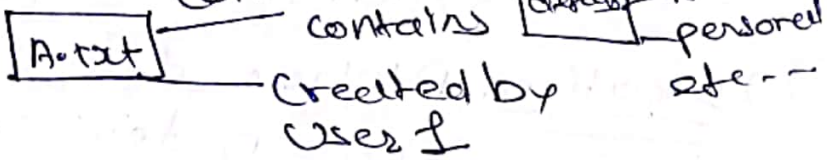
TRANJAL SAVE

## 6.1 : File - Concepts (Attributes, Operations, types and File System structure.

(2m)

### File System Management (concept)

A file is a collection of related information defined by creator.

For example, File 


In general file is a sequence of bits bytes, lines or records. [name - user]

### Types or some examples of file

1) Text file : Sequence of characters organized into files.

2) Source file : Sequence of sub-routines or functions - It is executable file design to do some specific task [a] f1() - calling f2() f2()

3) Object file : Collection of words, organized into loader record blocks.

4) Regular files : Contains data, either as text or binary.

5) Directory files : Contains references to other files or directories, helping organizing the file system.

6) Sperical files : Define system devices or temporary communication channels.

## File Attributes

Are the parameters used to keep track of file in operating system.

1) **Name**: The symbolic file name is only into. kept in human readable form.

2) **Identifier**: File system gives a unique tag or number that identifies file within file system, which is use to refer files internally.

3) **Type**: This information is need for those systems that support different types.

4) **Location**: Pointer to device and location of file on device.

5) **Size**: The current size of file (in bytes, blocks, words) ~~can~~. (1024kB)

6) **Protection**: Access control information (file = a, user - U1, RW = R)

7) **Time & Date**: Creation / modification / deletion

8) **User ID**: User identification (unique)

## Operation on files (File operation).

1) Creating a file - Space (V) Entry of new file in directory (a file name)

ii) Writing a file — system call (write(a, "ABC"))

iii) Reading a file - system call (Read a) pointer of next block to load.

iv) Repositioning a file - not involve I/O operation (file seek)

v) Deleting a file — Delete file from Entry

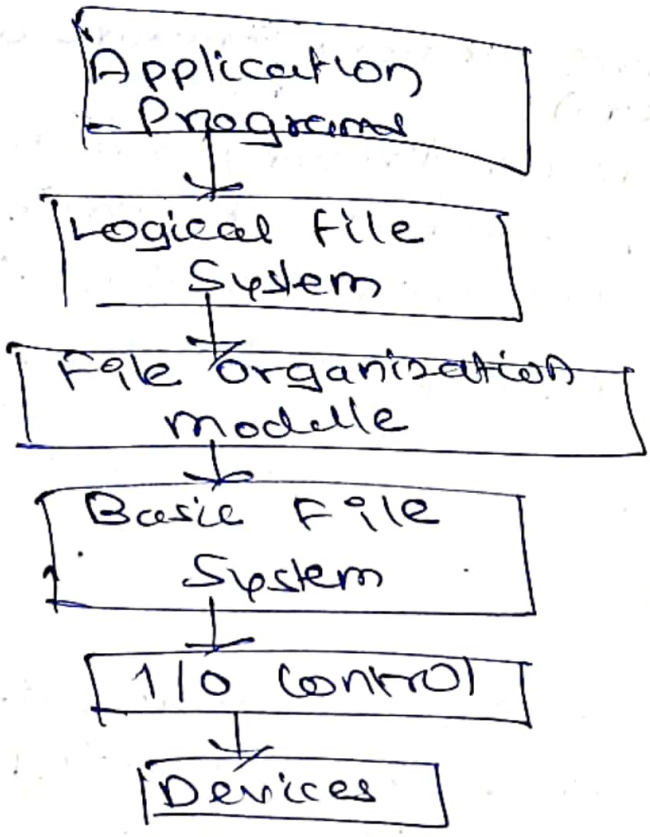
vi) Truncating a file - Delete records from file but not file only content.

# File System Architecture

Expected (2 Mark)

A file system helps manage how data is stored, located, and retrieved on a disk, make it easy to work with files.

It have several layers. as follows :




ALFBID

**Logical file System** : check if you have permission to access file. make sure file path is correct.

**File organization module** : Decide where file is stored on hard drive. keeps track of free space on disk

**Basic file systems** : Tells hard drive where to find or save file.

**I/O Control** : Uses special code (drivers) to talk to hard drive and get the file.

Unit 3

## Directory structure: Single level, two level, tree structured directory

### Questions:

Q// Explain two-level directory structure with suitable diagram. (4M)

Q// Draw and explain directory structure of a file system in terms of single level, two level and tree structure. (6M).

Q// Construct and explain directory structure of file system in terms of two-level and tree structure. (6M)

### File Directory

To organize files in computer system, in systematic manner, the operating system provide concept of directories.

A directory can be defined as way of grouping files together. Directory is itself a file that is owned by operating system.

A physical disk can be broken into multiple position or mini disks.

A directory is a container that is used to contain folders and files.

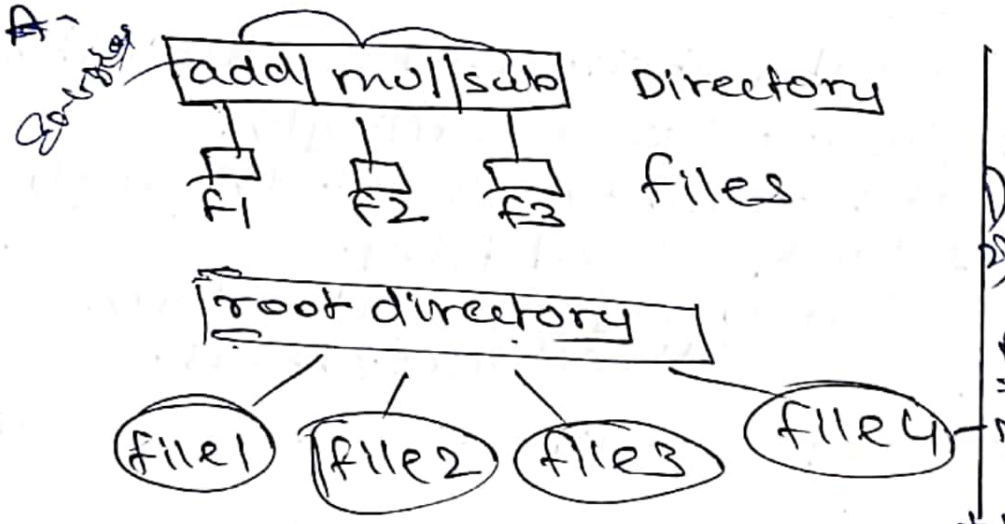
There are logical structures
- Single-level
- Two-level
- Tree structure

### Operations on directory

i) Search for file
ii) Create new file
iii) Delete a file
iv) List of directory
v) Remove a file

## ngle-Level Directory Structure

- All the files are contained in same directory.
- Each file must have unique name.
- The single-level directory is simplest directory structure.
- In this, all files are contained in same directory, which makes support and understand.
- It having one directory containing all files, so, it is called root directory.
- Simple operation like file creation, search, deletion and updating are possible.
- Two file names can't be same. In case same name previous one is overridden.




**Advantages**

1) Simple to implement
2) Easy to maintain

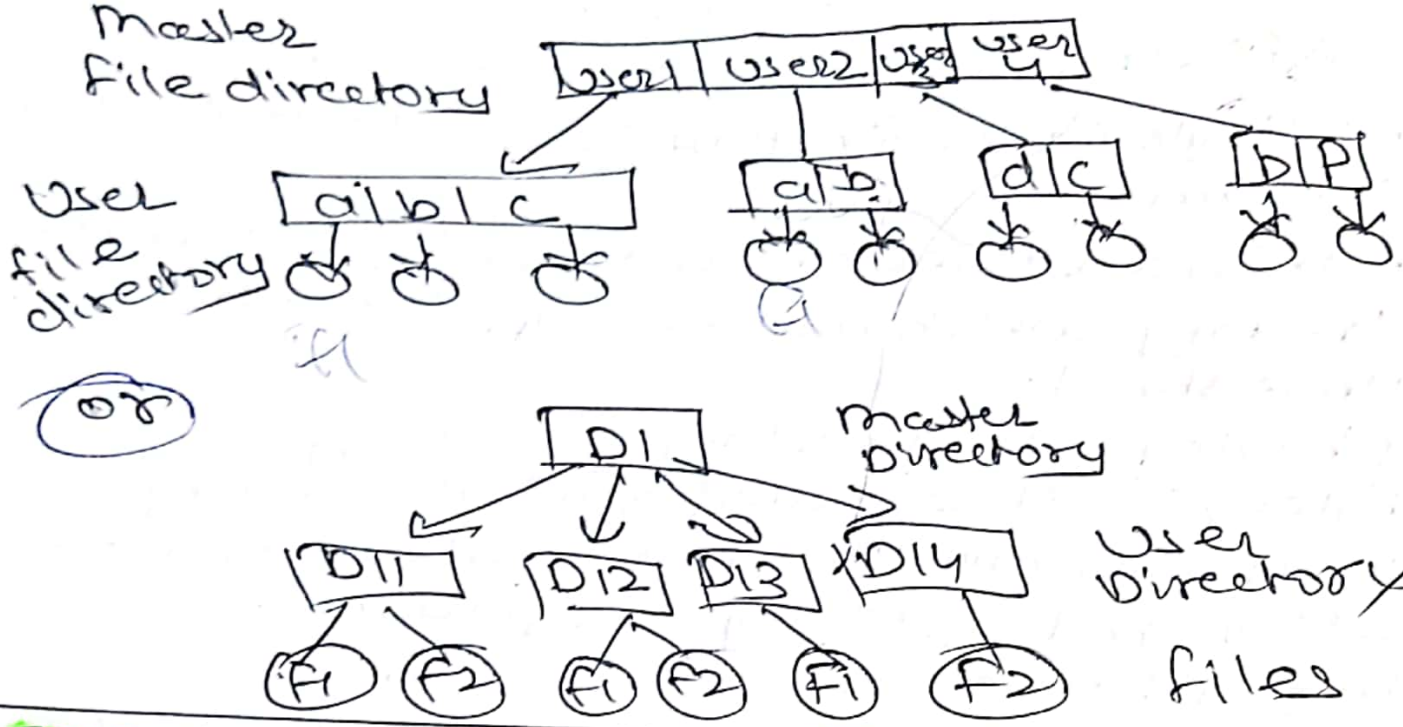
**Disadvantages**

- Not useful for multi-user system
- No. of files large searching inefficient

## Two-Level Directory Structure

- As we have seen, a single level directory often leads to confusion of file names among different users.
- In two level directory structure each user has their own user file directory (UFD).
- It have similar structure, but lists only the files of single user.
- System master file directory (MFD) is searched whenever a new user id is created.

Create directory for each user.

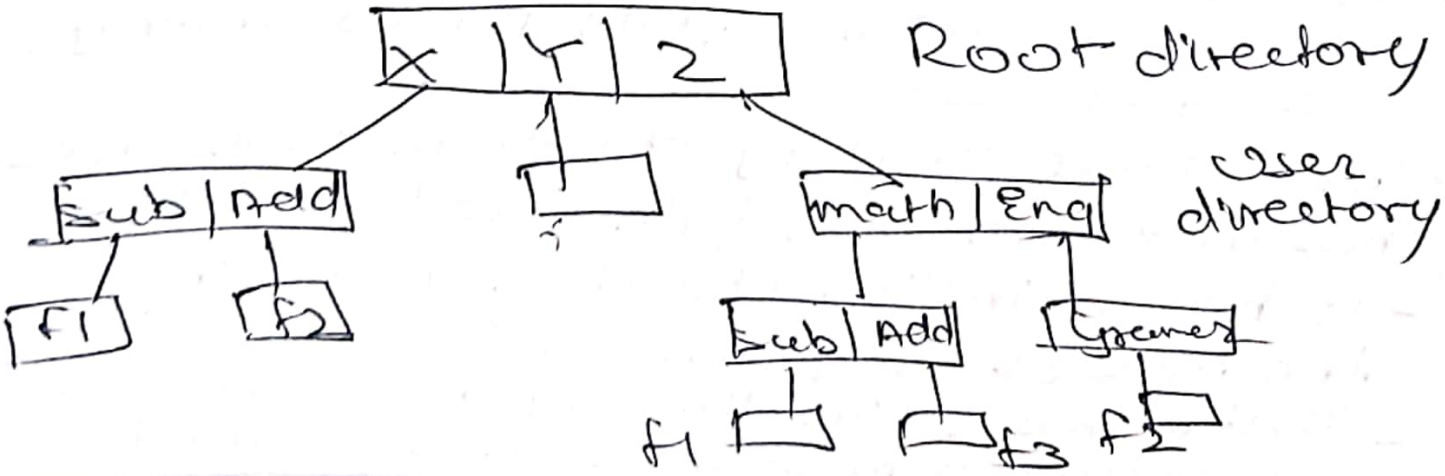



## Tree Structured Directory Structure

Allows users to create their own sub-directories and organize their files accordingly.

Path: It is route from the root through all sub directories to specified files.

- User can't access or change root directory. Or can't see users directories-others.
- User can create subdirectories to organize files.
- Grouping related files is possible & important and unimportant files are seperated.



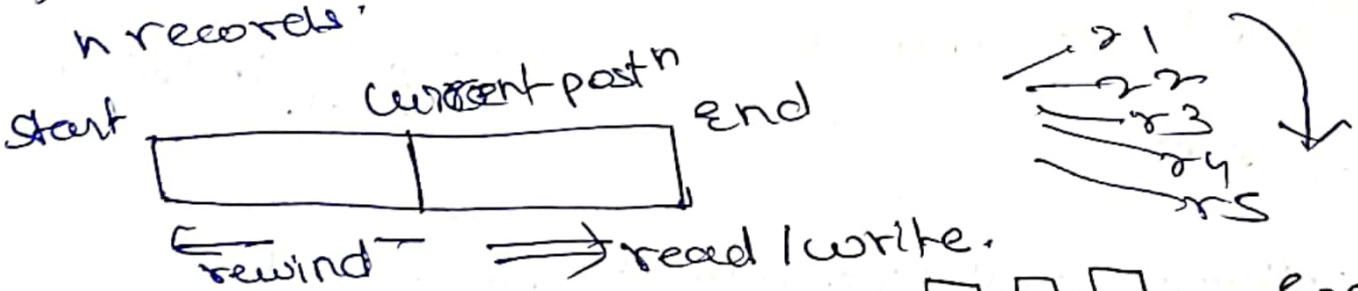
## .2 Access Methods - Sequential, Direct

### 1) Sequential Access Method.

- The simplest access method is sequential access. Early operating system provided only this kind of file access.
- In this type of file access, process reads all the records in file in order one record after other, starting at begining.
- Sometime, skipping of any record is not possible.
- Emulates magnetic tape operation.
- One record is processed after other.

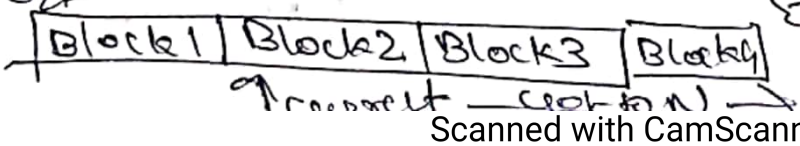
Supports following operations.

1) read next 2) write next 3) rewind 4) skip n records.




### 2) Direct Access method


- Another method is Direct access method also known as relative access method.
- It is based on disk model. It allows random access ie user can jump to any record and access that record.
- A Fixed length logical record that allows the program to read and write record rapidly.
- It allows random access to any record.
- Thus we can read block 14 then block 59 the write block 17. There is no restrictions.



## 3) Indexed (Swapping) Access method.

Index is created which contains key fi
and pointers to various blocks.
It is top of Sequential access method.
These method construct an index for file.
To find a record in file, we search the index, and then by help of pointer we access the file directory.
Pointer carries logical address.


**File Allocation methods** : Contiguous, Indexed, Linked.

### Questions

1) Enlist different file allocation methods & Explain contiguous allocation method in detail! — 6marks
2) Explain linked file allocation method.) — 4marks
3) List file allocation method and explain any one in details. — 6marks.

### File Allocation Method

Allocate space to the files so that disk space is utilized in an efficient manner.

Factors to consider: i) Processing speed < Sequential / Random.
ii) Ability to use multisector and multitrack transfer.
iii) Disk utilization
iv) main Memory Requirement.

~~Contiguous Allocation~~

From the user's point of view a file is an abstract data type. It can be created, opened, written, read, closed and deleted without any real concern for its implementation.

The implementation of file is a problem for the operating system.

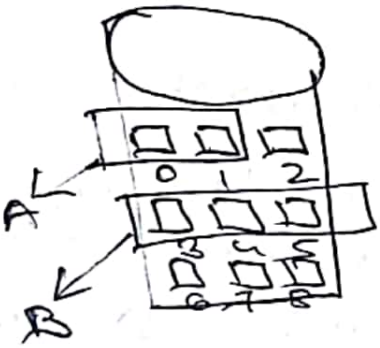
The main problem is how to allocate space to these files so that disk space is effectively utilized and files can be quickly accessed.

major methods → contiguous.
Linked.
Indexed.

---

## Contiguous Allocation

1) Each file occupies a set of contiguous address on disk.
2) Linear ordering
3) A file is defined by location of the first block and its length.
4) Both sequential and direct/random access are supported.
5) A file is defined by its starting disk address (Block b) and its length (n blocks). It occupies from b to b+n-1 blocks.




Directory

| File | start | len |
|---|---|---|
| A | 0 | 2 |
| B | 3 | 3 |

Disadvantages

Finding space for new file

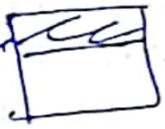
External fragmentation

### Advantages

1) Fast reading all file blocks.
2) Good overall performance.
3) Supports both sequential & direct access

Contiguous : Dynamic storage.

1) First Fit → 290K

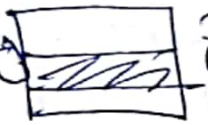



(110K wasted)

2) Best Fit → 290K


(8K wasted)

3) Worst Fit → 290K




(110K wasted)

---

## Linked Allocation

In Linked Allocation, each file is stored as a linked list of disk blocks, which do not need to be contiguous.

Each block contains pointer to next block in the chain.

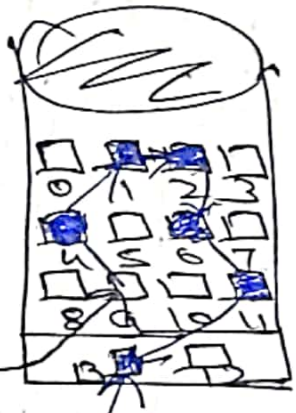
The directory entry contains a pointer to the first and last block of the file.

When file is created, the system simply removes the first free block from the free space list and links to the file.

To read file the system follows pointers from one block to next.

Directory

| File | Start | End |
|---|---|---|
| A | 9 | 13 |




**Advantages**

No External Fragmentation

Ease of growth

**Disadvantage**

No Direct Access

Sequential Access only

Pointer Errors.

## Index Allocation

Solves the Problem of Linked allocation (No Random Access).

In this all pointers are brought together into one location called index block.

Each File has its own index block.