Operating System

File Management

- → Why do we need second storage?
- → File
- → Attributes of file
- → Operations of file
- → Directories & their levels
- → Disk architecture
- → File allocation methods

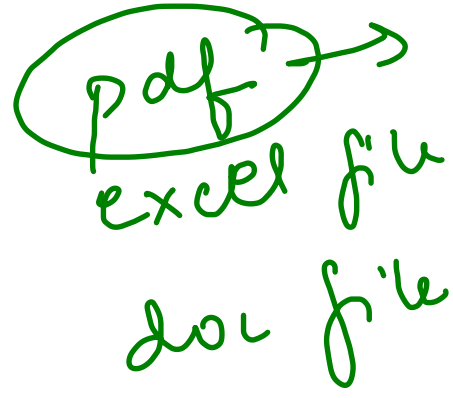
Secondary Memory Vs Primary memory

(disk)	(Main memory)
cheap	expensive
abundance	limited
permanent	volatile
slow	fast

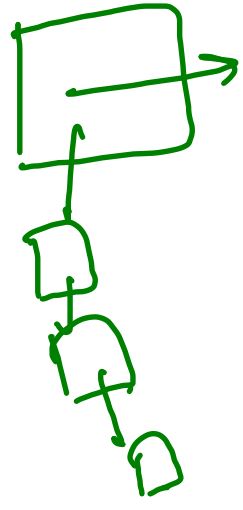
File :-

Named collection of related information that is recorded on secondary storage.

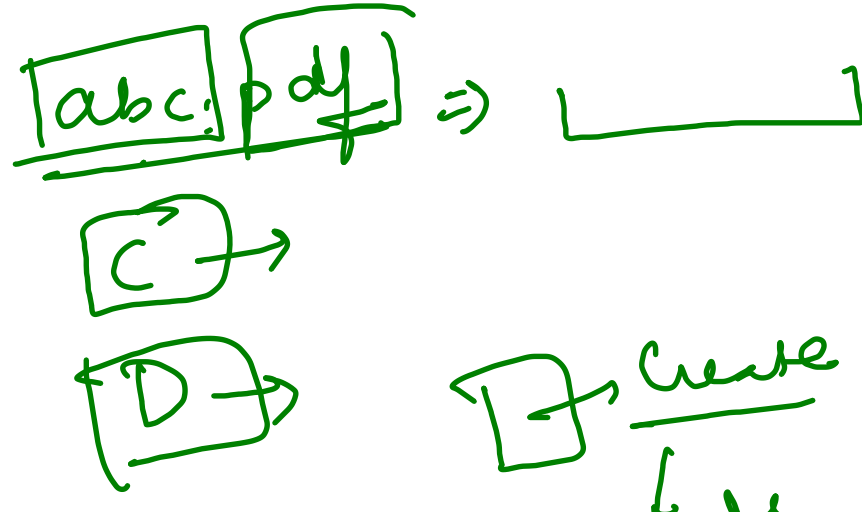
→ Sequence of records
→ stored in a file system.
→ can be data or programs.



Attributes of file

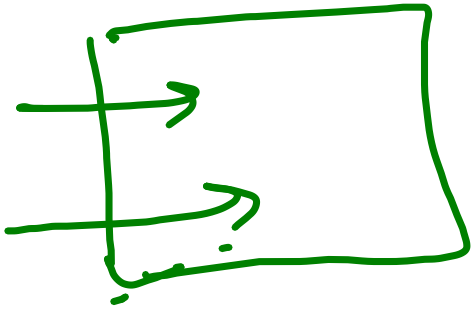


1. Filename
2. File type
3. Location →
4. Size
5. Protection/permission (if any)
6. Time stamp
7. Folder (Directory info)

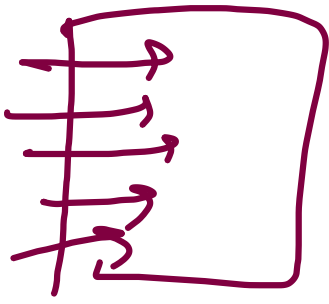


Operations on file

1. Create →
2. Write
3. Read
4. Delete



Directories / Folder

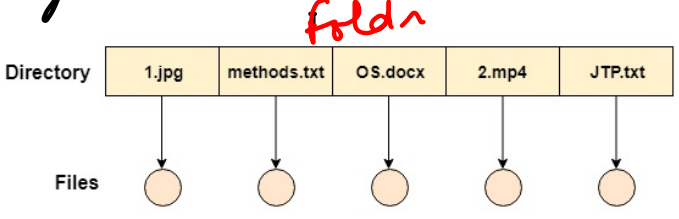


→ Can be defined as list of related files on the disk

→ can be organized with diff levels of complexity, efficiency, flexibility.

Single level directory

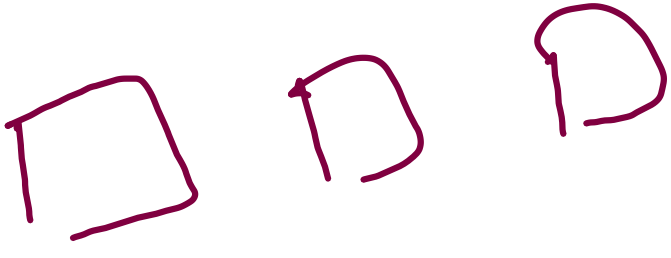
System contains one directory which mentions all the files present in the system.



Single Level Directory

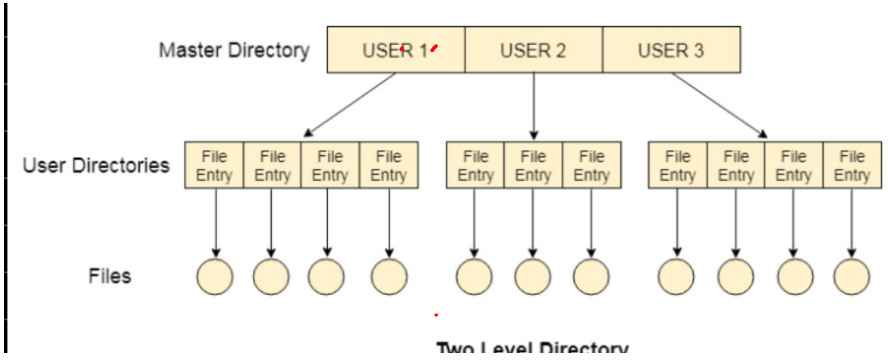
Adv ⇒ 1. Easy and simple to implement

Disadv
1. Cannot have two files with the same name.
2. Searching takes time.
3. Grouping problem.

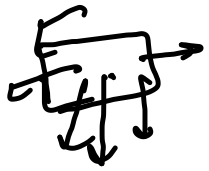


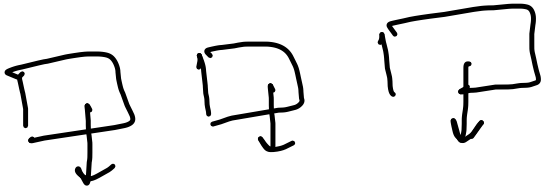
Two level directory

→ each user has their own UFD (user file directory)
→ one master directory contains separate directories dedicated to each user.



Two Level Directory

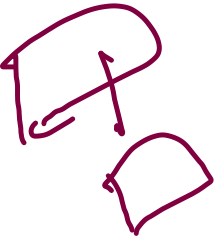


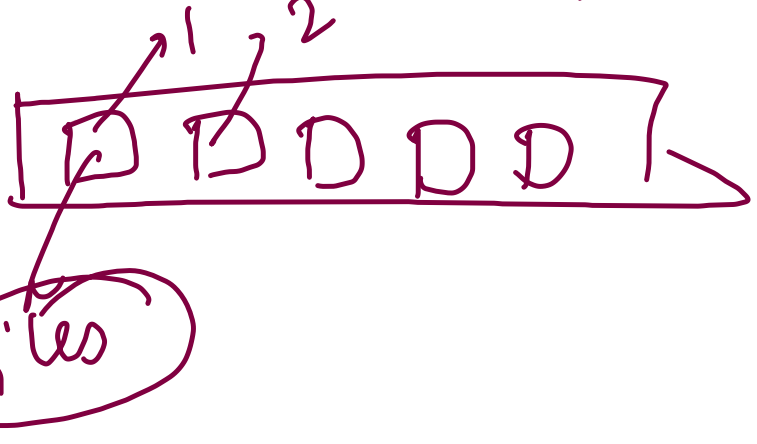


Adv - 1) Searching is efficient
2) Diff. users can have same file name
3) Files have path name.

Dis. Grouping problem

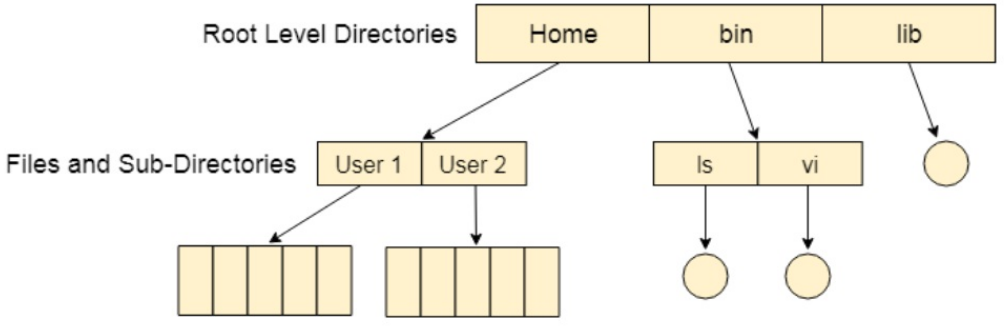
MF → 2 →



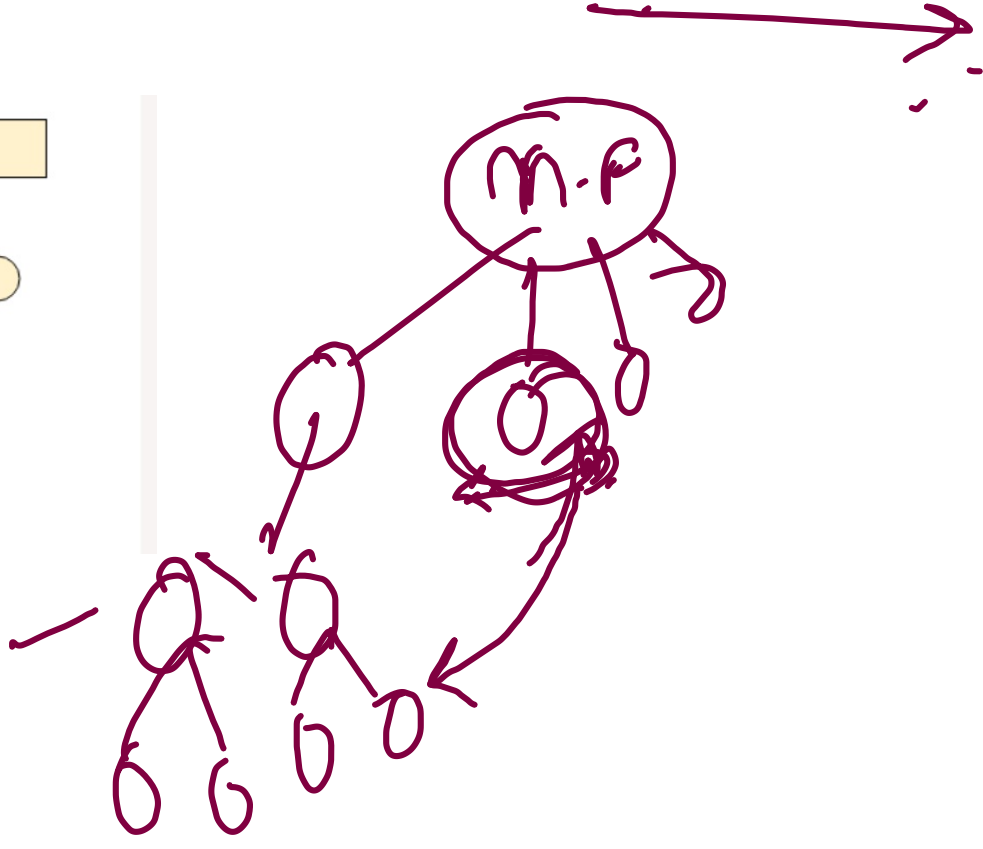


Tree structured directory

→ Allows users to create their own subdirectories and organize files accordingly.



The Structured Directory System

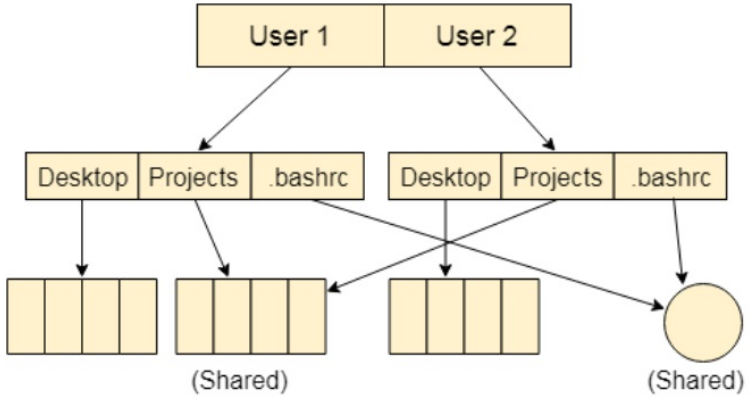


1) Searching is more efficient
2) File cannot be shared.
3) Grouping capability is present

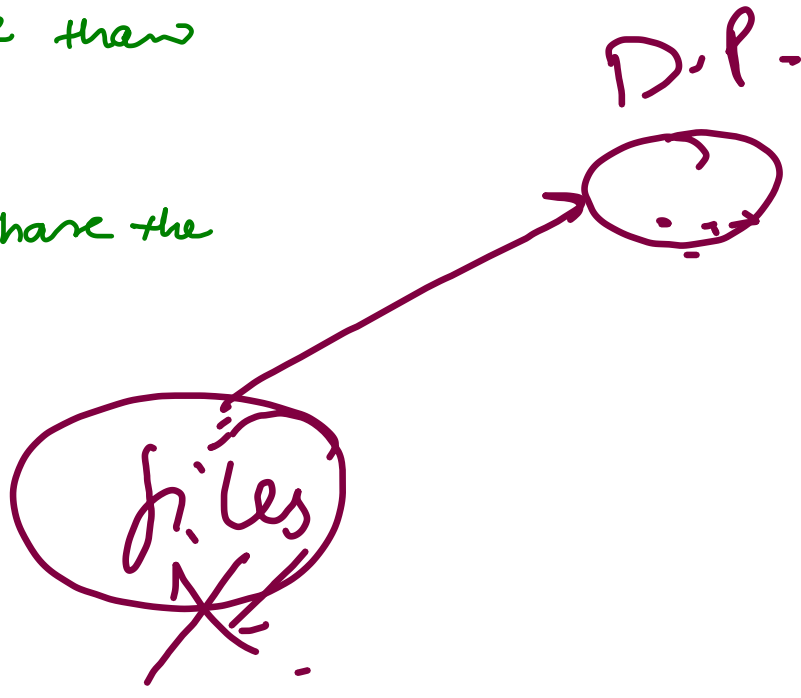
Acyclic Graph Directories

→ Allows the same file to exist in more than one directory.

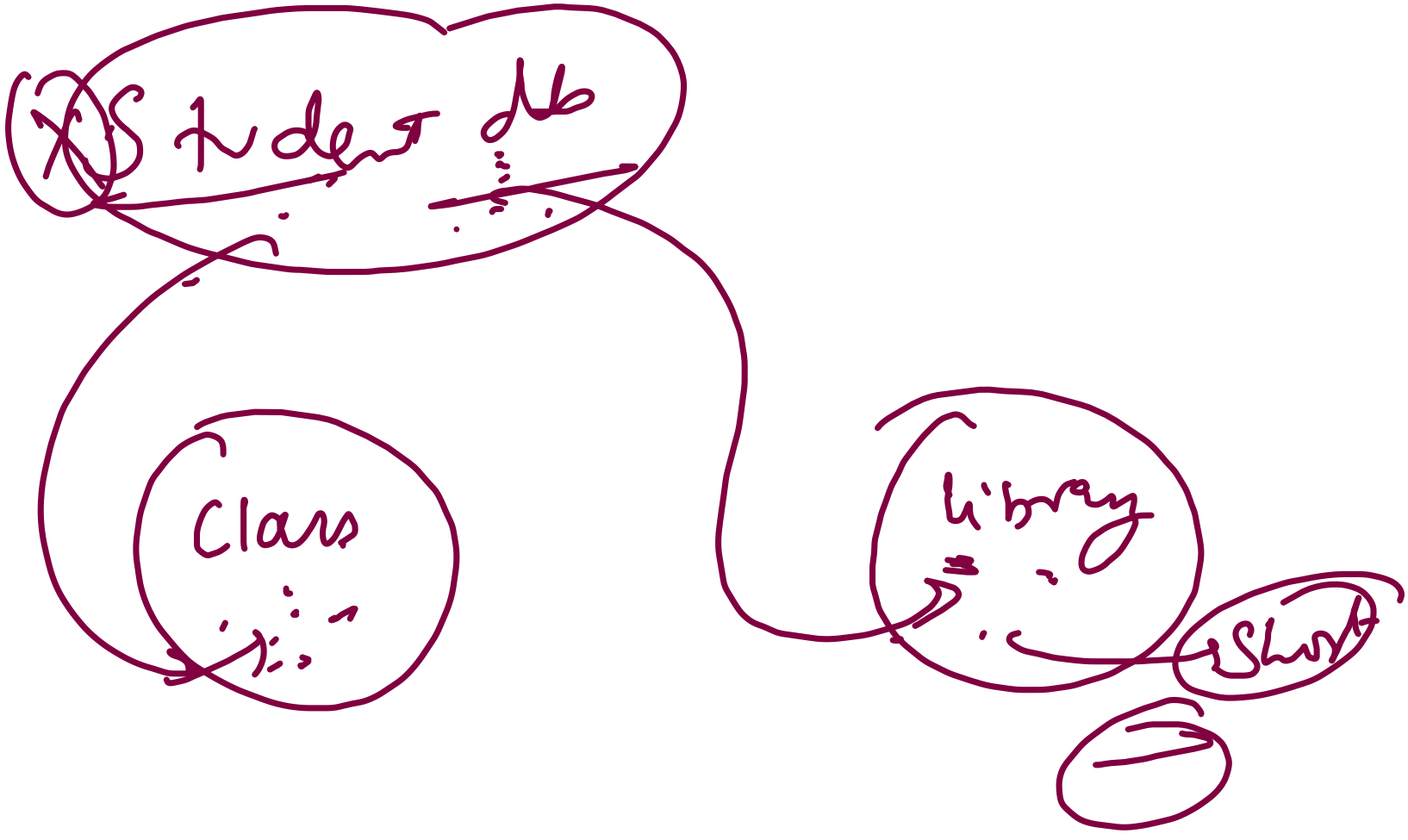
→ Two or more directories can point or share the same file



Acyclic-Graph Structured Directory System

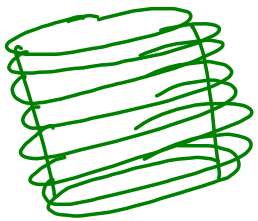


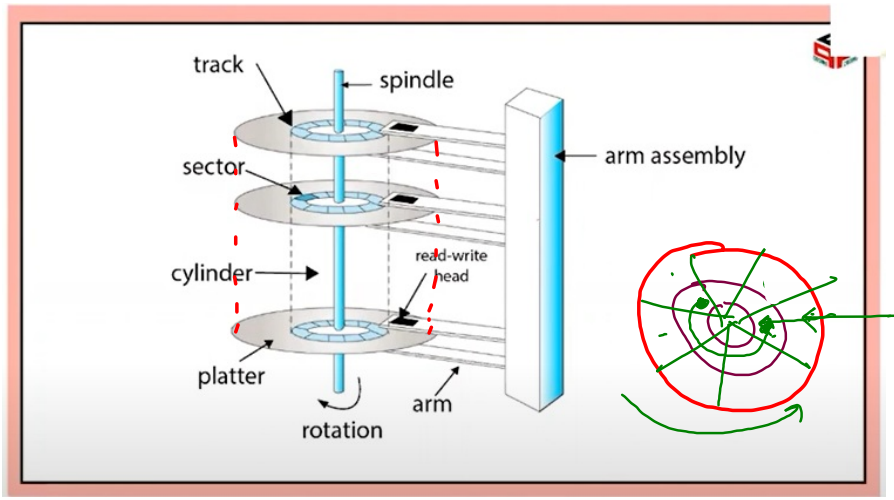
Dis → Dangling pointer.

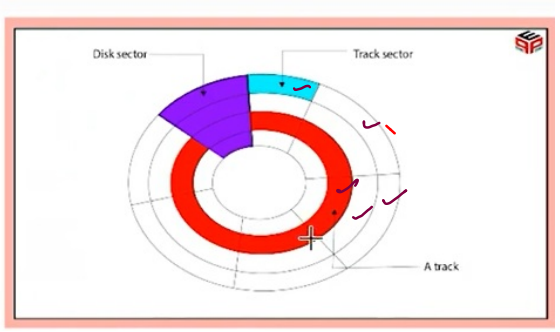

Student db
Class
Library

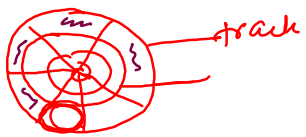
Disk architecture

CD







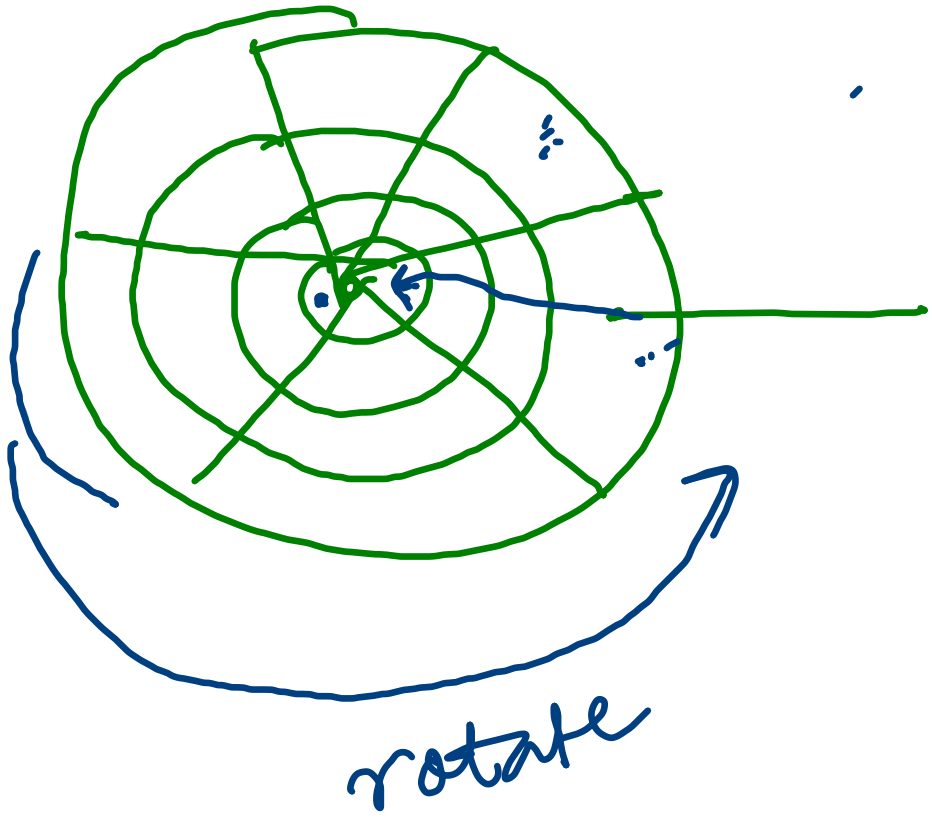


Platter — surface → top → track
bottom

R/w

read

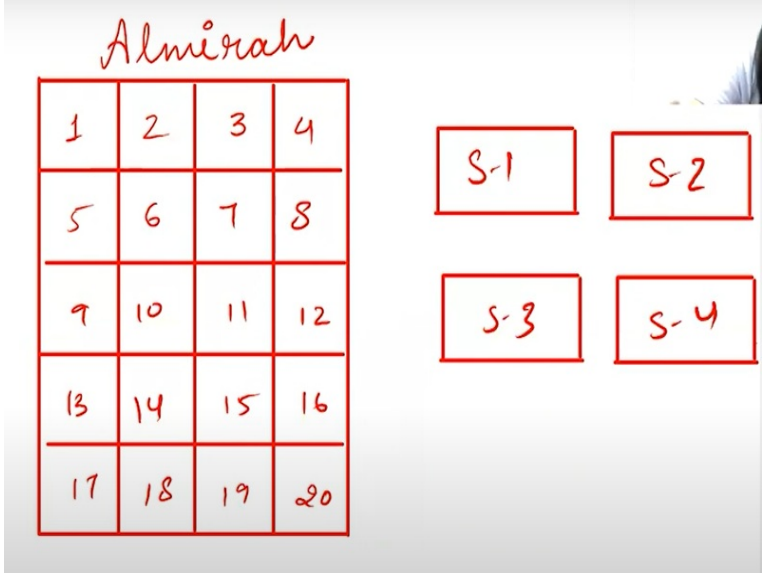
write

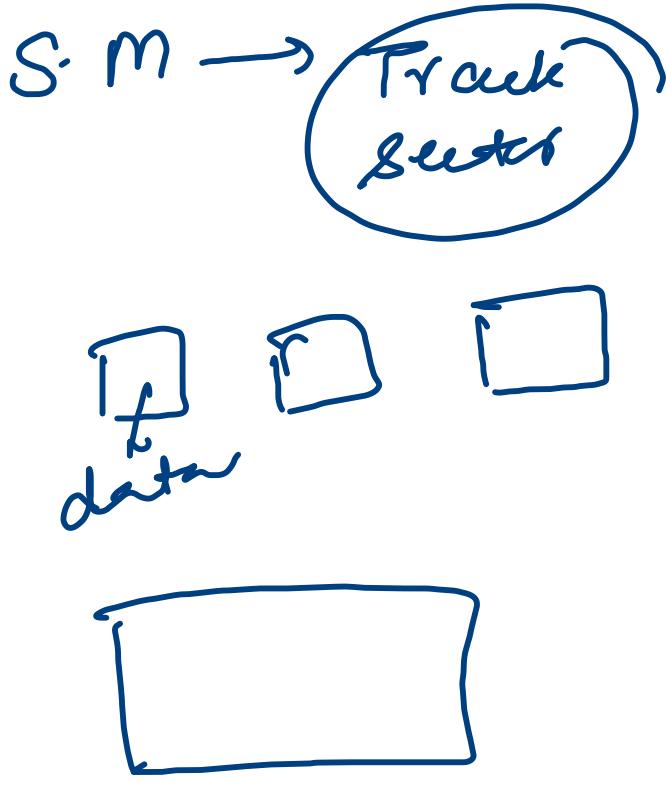


File allocation methods

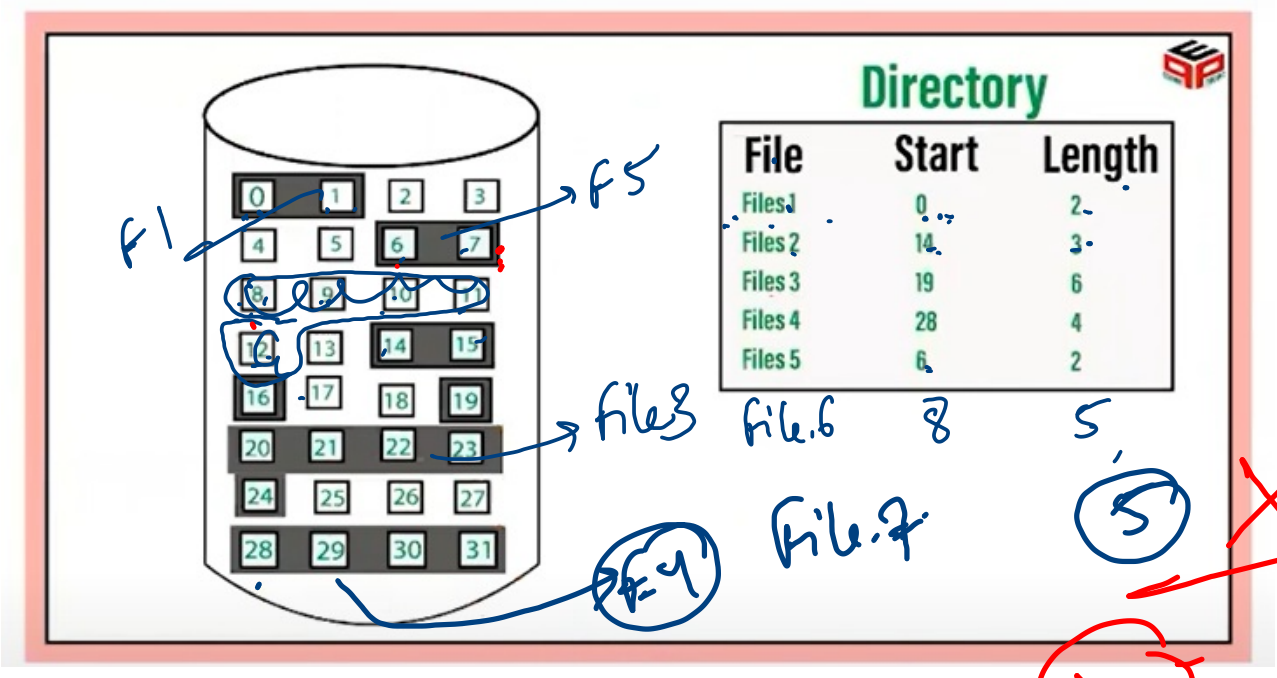
- Contiguous
- non-contiguous
  - Linked list
  - FAT
  - Indexed

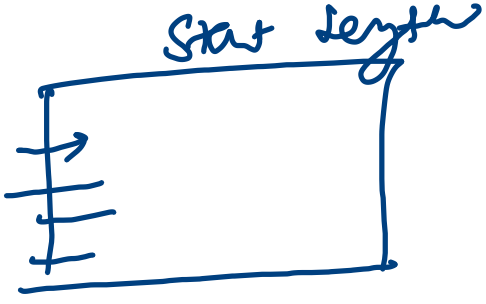
1) Max. Utilization of disk
2) Ease of Access





1. Contiguous file allocation method





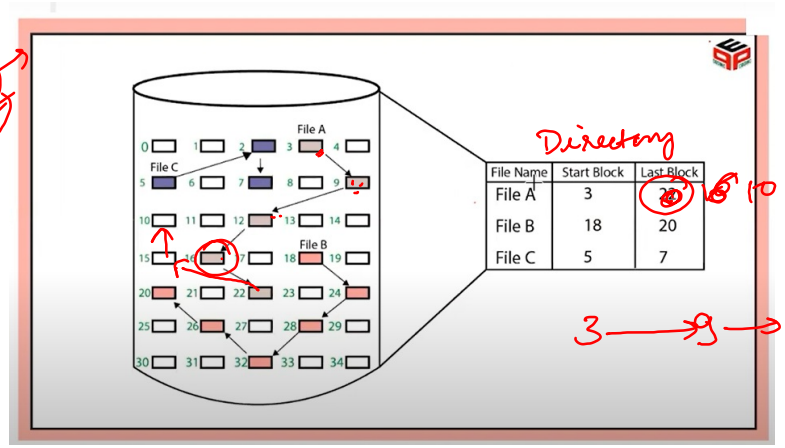
1) fragmentati
2) Inc.

Adv - 1) Simple
2) Easy access

Dis - External fragmentation
- Diff. to grow a file

Non- contiguous

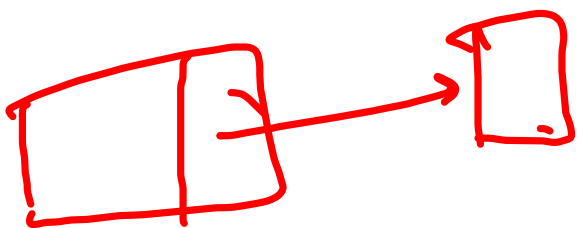
1. Linked List Allocation



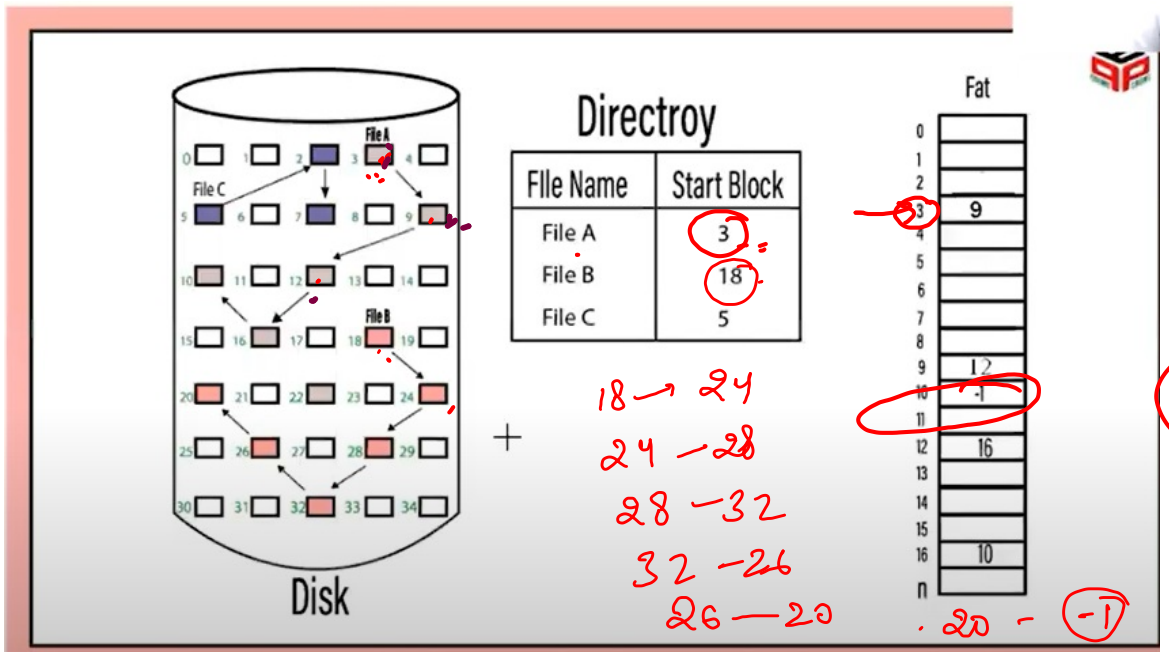
Adv :- 1) No fragmentation
2) Easy to expand or grow a file

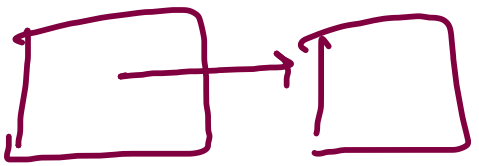
Dis :- 1) Slow because of seq. access
2) Extra space req to store pointers

End of allocation



3. File Allocation Table





3 → 9

-1

Adv :- 1) Random access
2) Effective utilization

Dis. :- Each block needs a fat entry

M.M

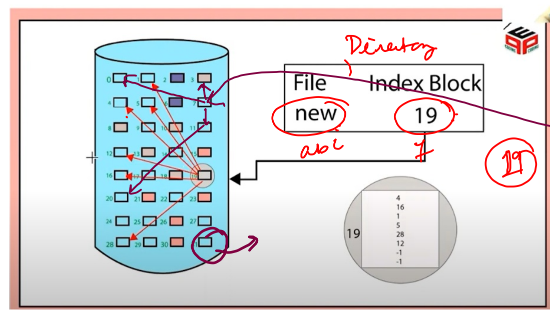
Pages → frame

P.T.

Pages → ~~bloc~~ data → blocks

Index

Indexed Allocation



file →

index

10KB

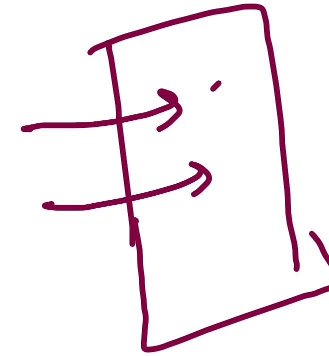
Adv:- 1) Supports random access

2) No fragmentation

(Unix)

Dis: 1) Even small files require 2 data blocks

2) Size of file is limited.



Contiguous

Ease of Access ✗

Max. uti ✗

N.C

Linked li — Eq A ✗

Max uti

FAT

W.

E.A ✓

M.U ✓

Index

Unix

E.A ✓

M.U ✓

1. A disk is having 16 platters, 2 surfaces, each surface is divided into 1K tracks & each track is having 512 sectors, each secter can store 2KB data.

Calculate capacity of disk.

$16 \times 2 \times$