

File System, I/O & protection:

Everything we save on a computer is in form of files.

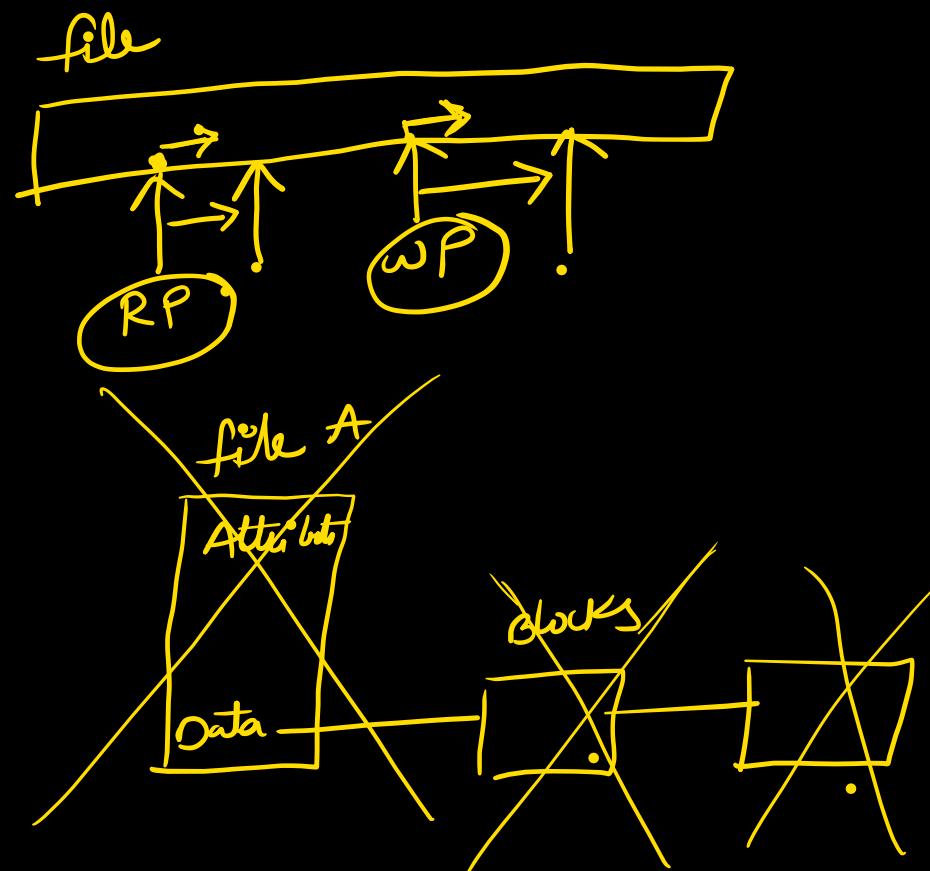
Ex: Audio, video, word doc, code

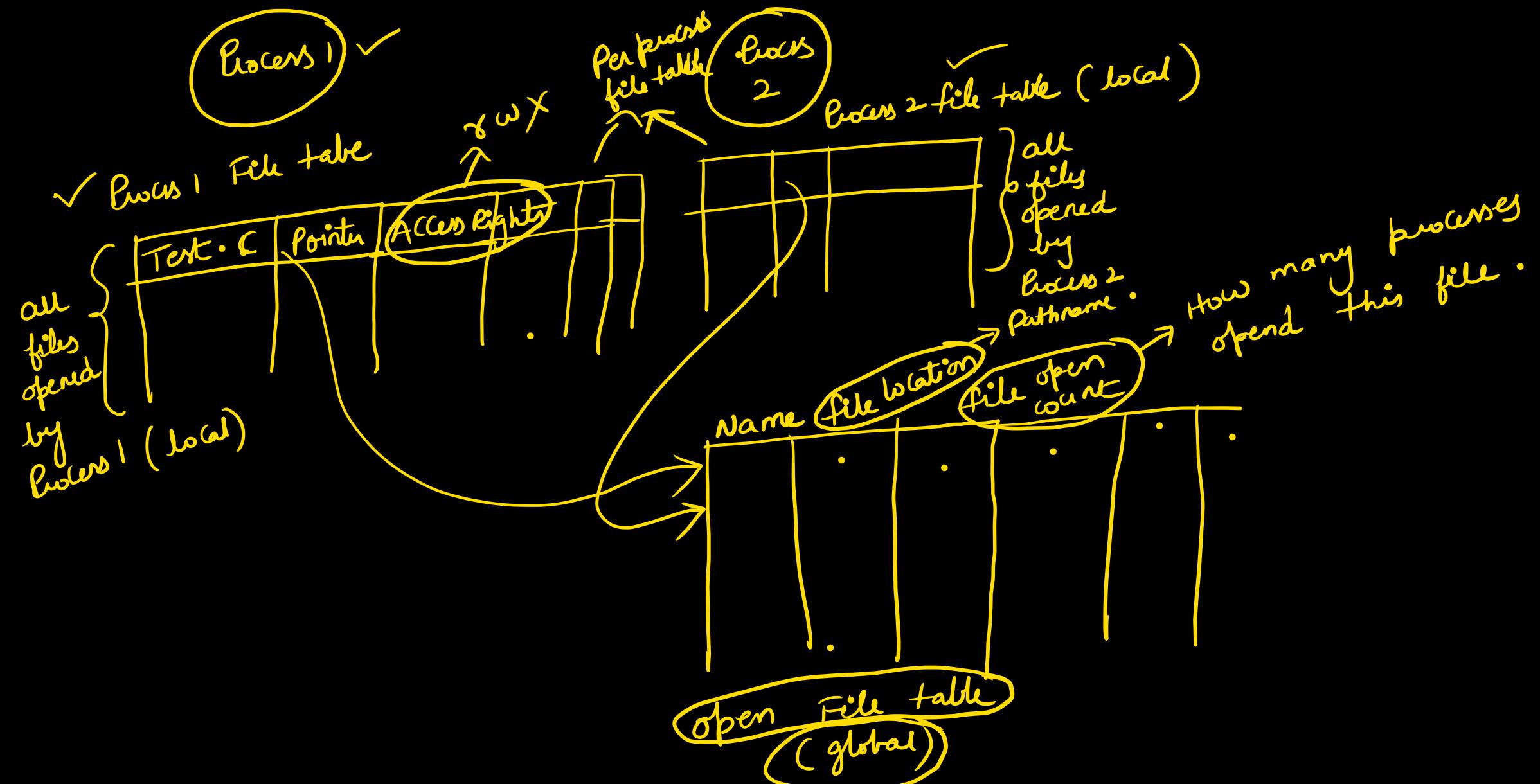
File attributes: —

- Name →
- Identifier → a number, unique, not human readable
- Type → .mp3, .mb4, .c, .java, .doc
- Location → which drive, directory (Pathname)
- Size → 1MB, 10MB
- Protection → Read, Write, Execute (permissions)
- Time and date → Created & modified (last modified)

File operations:

- Create
- Write
- Read
- Repartitioning ✓
- Delete
- Truncating





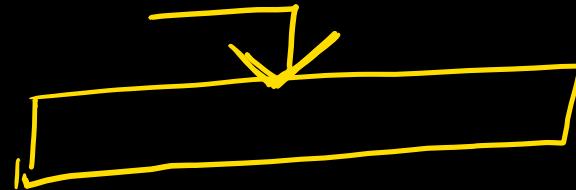
Accessing a file:

- sequential access
- direct access
- indexed access.

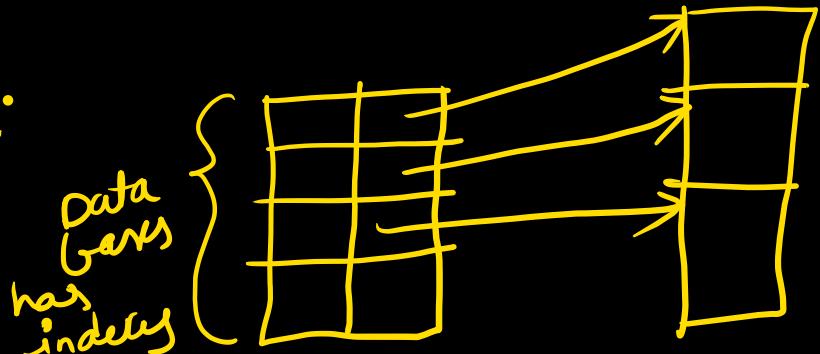
Sequential → accessing a file from beginning to end in a linear order.

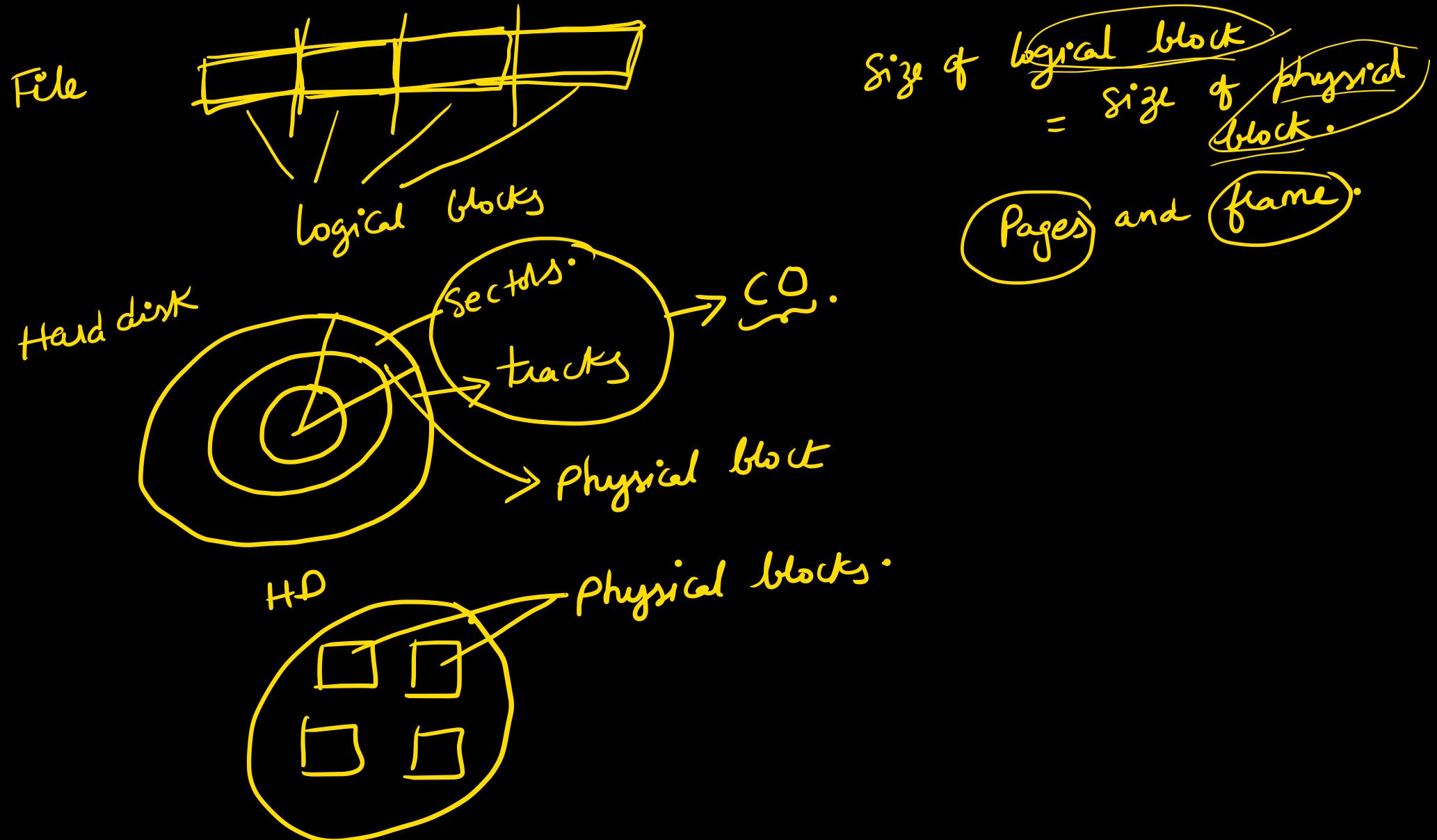


Direct access:



Indexed :





what should be size of a block?



e.g.: Block = 1024 B

file size = 1025 B

→ 1 Block + 1 Byte → 1 Block
1023B is wasted → internal fragmentation.

In Best Case - 0 wasted in worst case \cong 1 block, avg - $\frac{\text{Block size}}{2}$

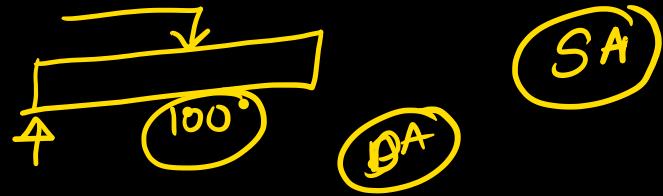
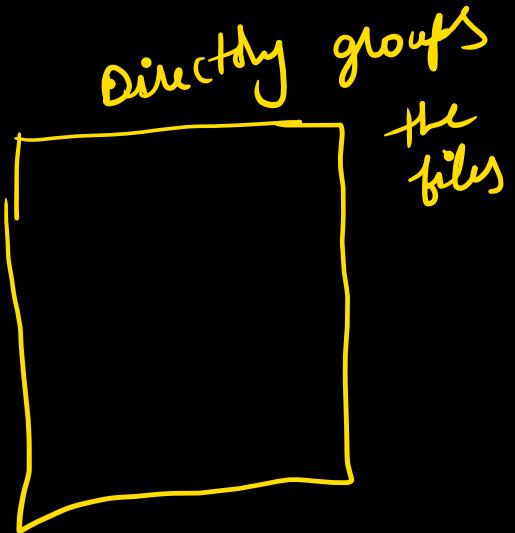
Block ↓ → Too many blocks → access will be slow.

By experimentation, it is found out that $\boxed{512 \text{ B}}$ is good size.

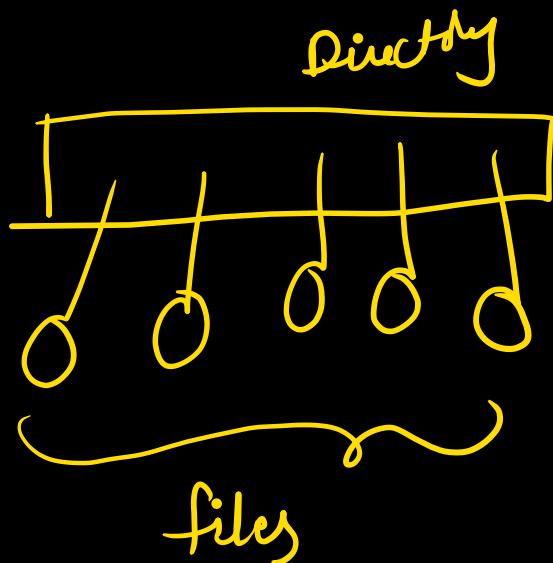
standard.

$\boxed{512 \text{ B}}, 1024 \text{ B}$
mostly

Directory:



Every filesystem will support directories
(Directory contains info about all files (meta data)
in the directory)



operations on a directory → folders:

→ search → Search for files.

→ Create → Create more files & more directories.

→ Delete → Delete a file & a directory

↓
recursive delete (* rm -r).

→ list → display all the files and directories of a directory.
↓
ls.

→ Traverse → display all files and also files in
sub directories.

→ Rename → we can rename a file & a
directory.

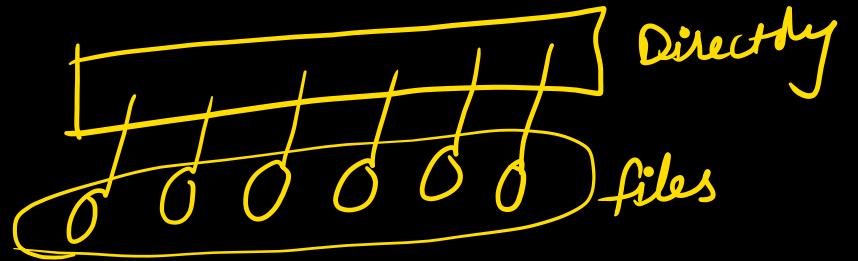


we don't discuss any particular
OS like Linux & mac os
window

General concepts.

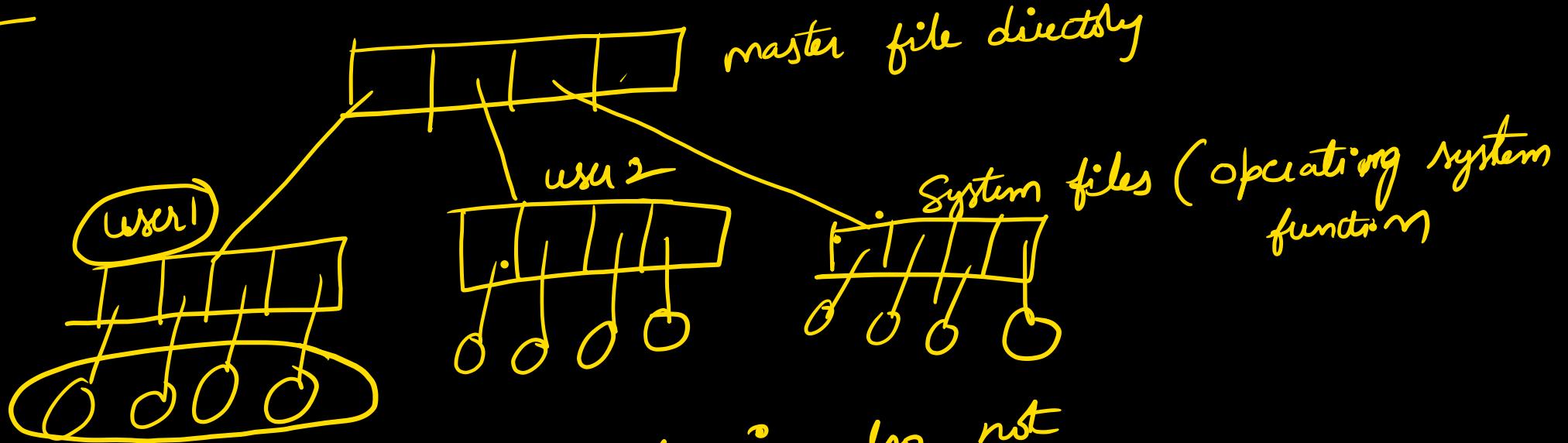
when there are too many files (like millions) then managing them in one directory will be difficult.

Single level:



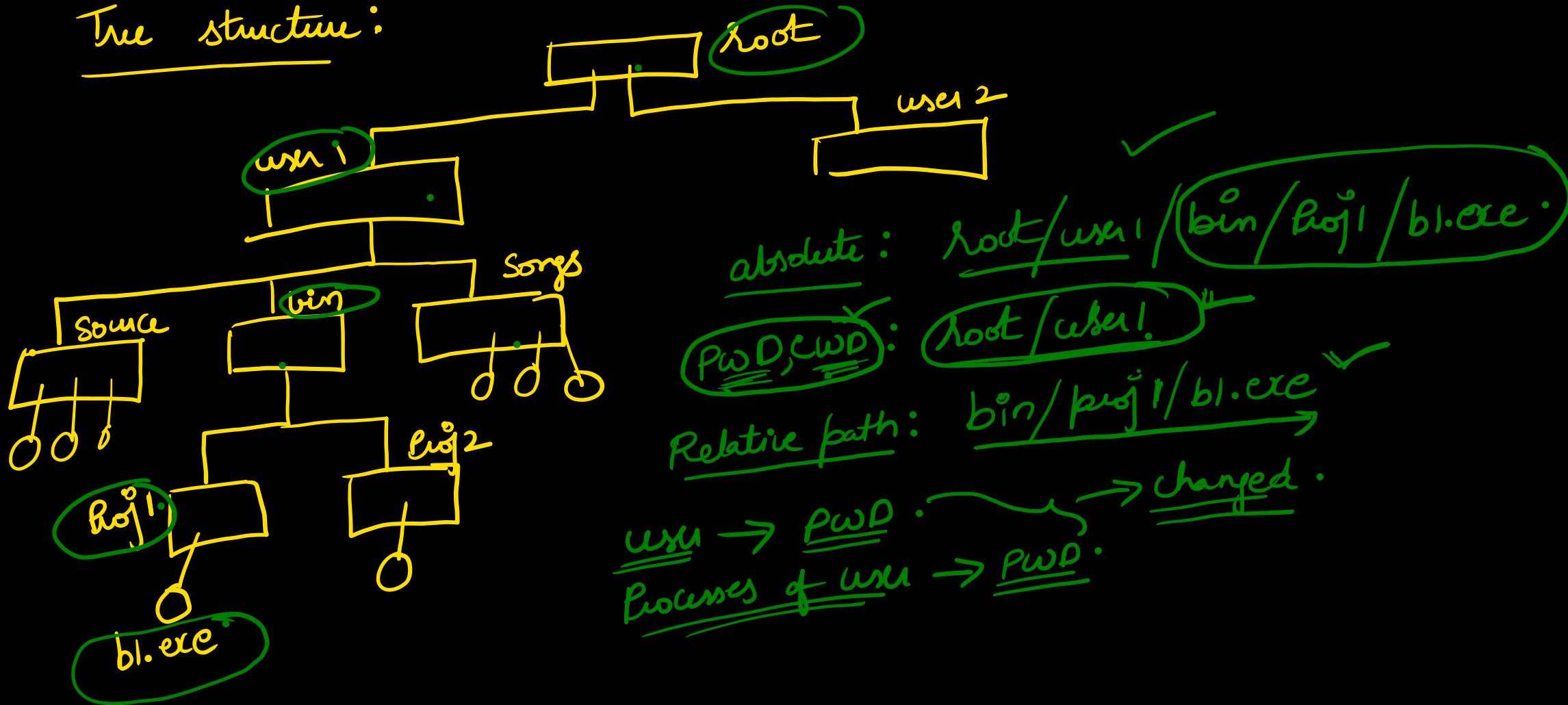
accidentally you may name two files as same
& you may delete a file.

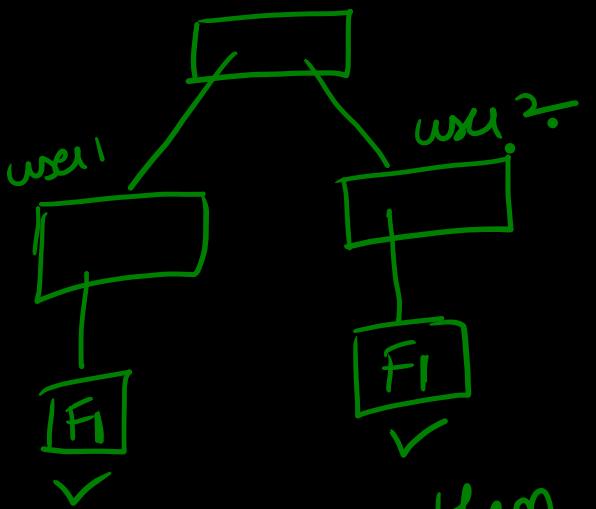
Two level:



But Two level is also not
sufficient in all cases.

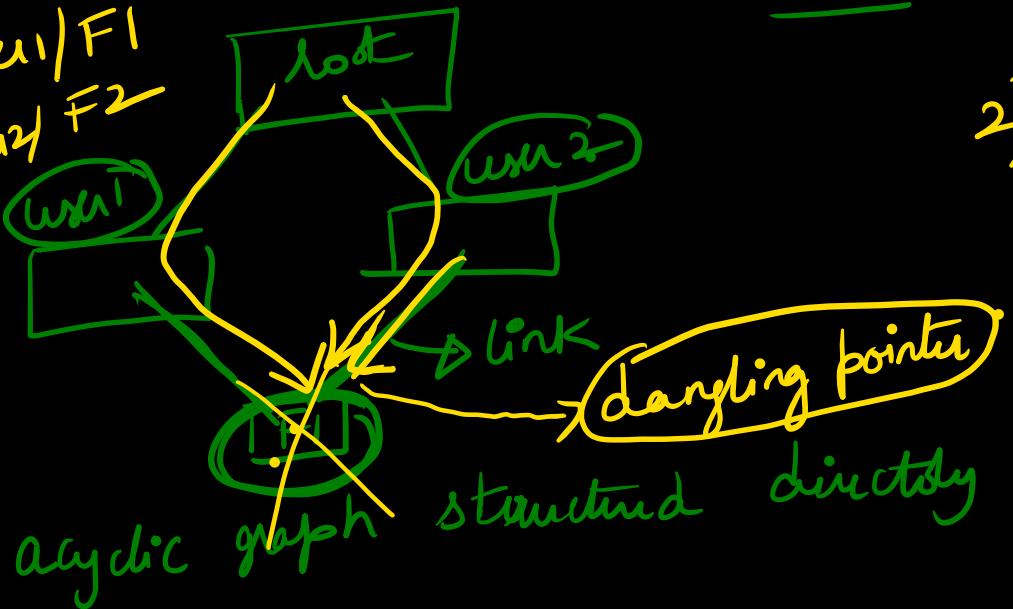
Tree structure:





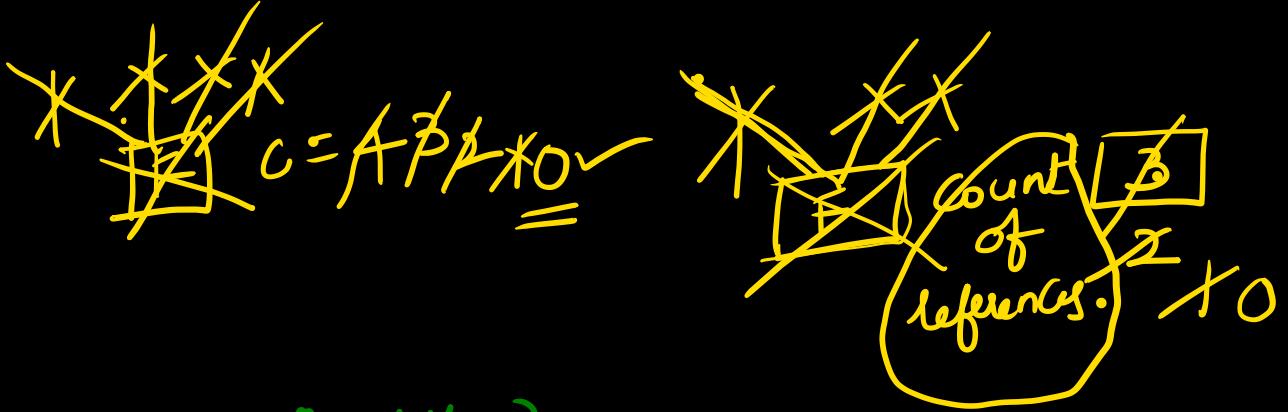
share a file , then

F1: root | user1 | F1
root | user2 | F2

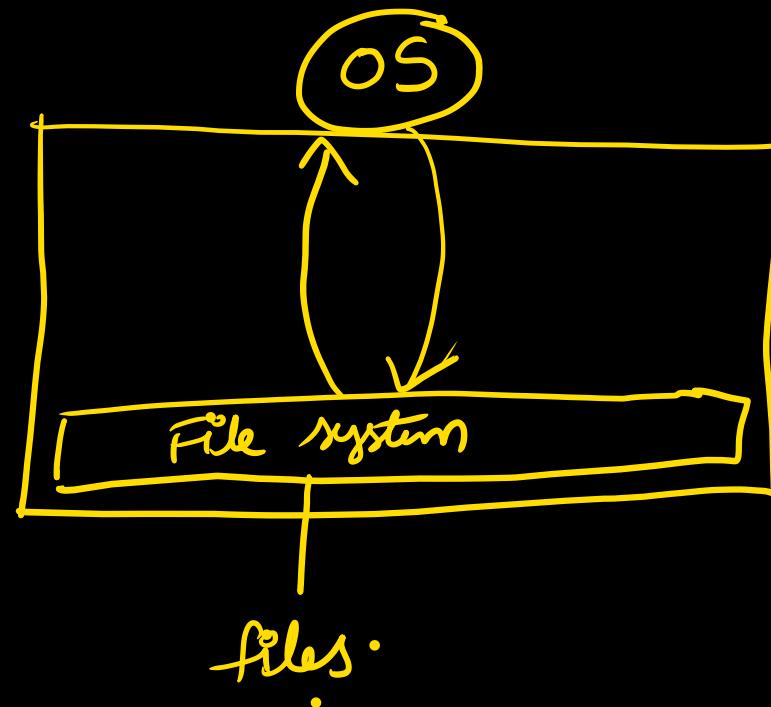


better method is follow

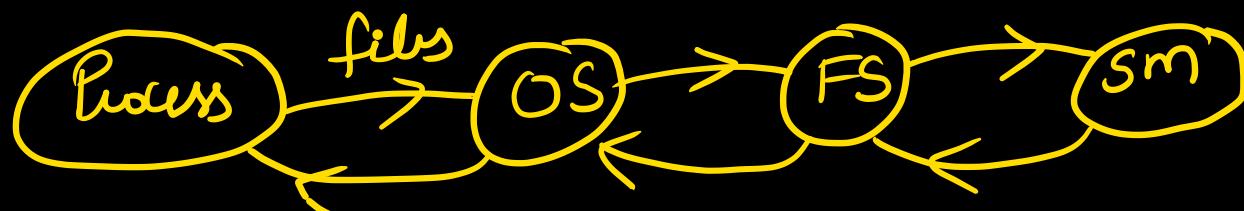
- problems:
- 1) If F1 is deleted by user 1, then there will be dangling pointer.
 - 2) Different path names → not good.



File system :



File system is a part of operating system. It is a S/w module responsible for handling files.



File system provides the mechanism for online storage and access to file contents including data and programs.

{ audio,
video
Text

online

connected

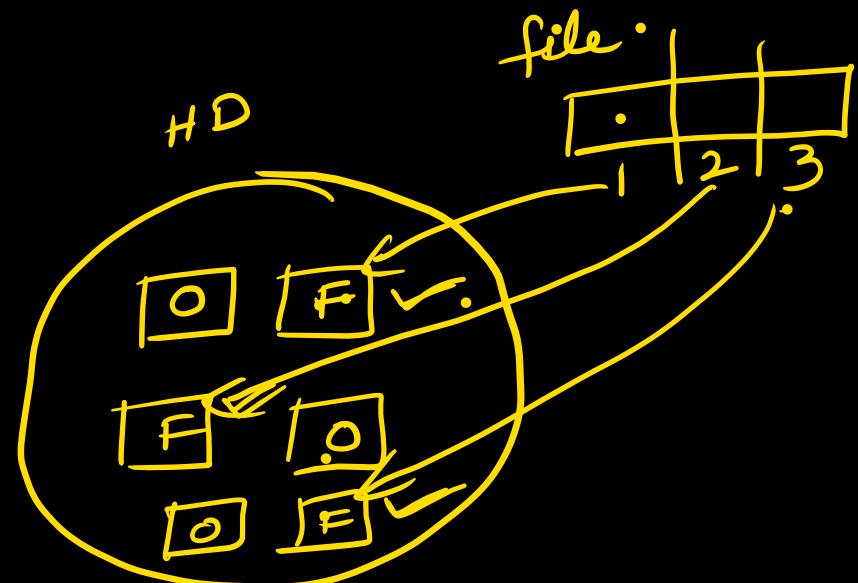
internal hard disk

of
connected devices

file is a S/w.

File system deals with following issues:

- file structure
- To allocate disk space ✓
- Recovering free space
- To track locations of data
- Interface other parts of operating system to secondary stage.



File System Structure:

File Systems provide efficient and convenient access to the disk by allowing data to be stored, located and retrieved easily.

Application programs



logical File system (meta data, directory structure)



File organization module (logical blocks → physical blocks
Free space management)



Basic File system (Commands to the I/O control buffering)



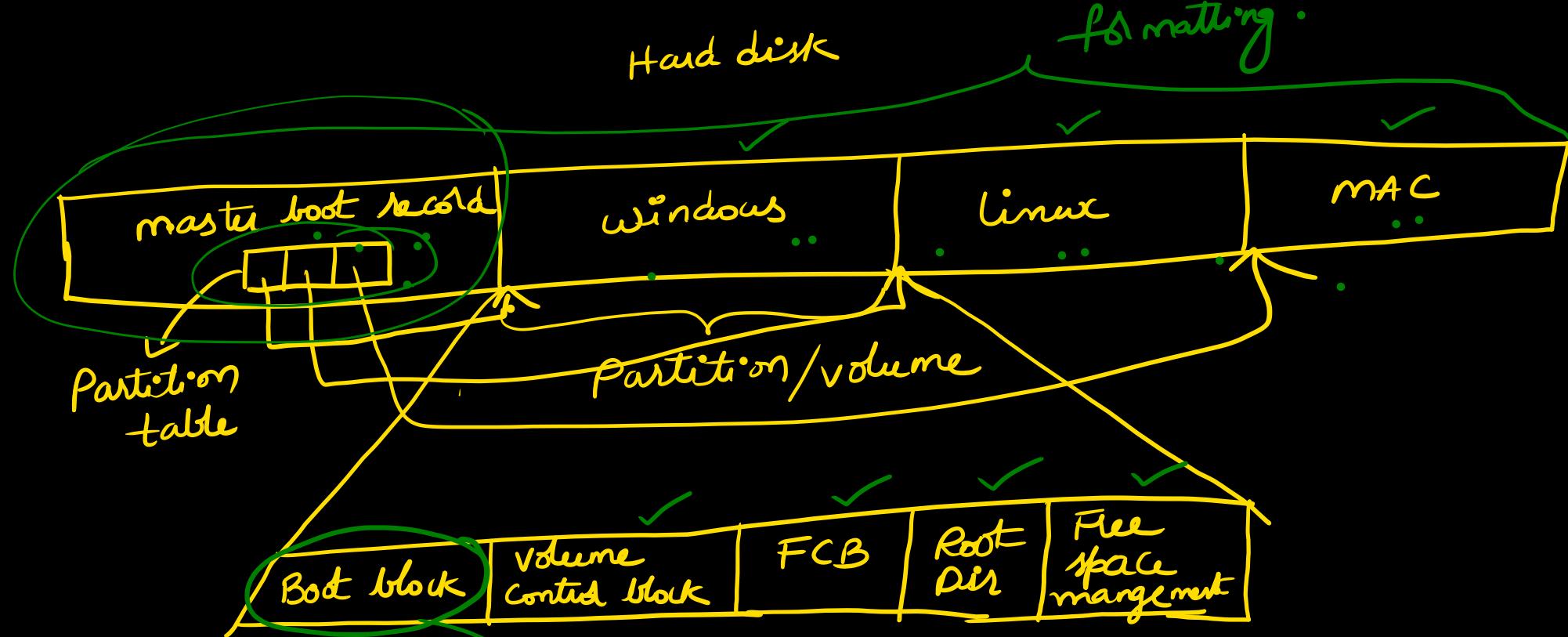
I/O control (device drivers, Interrupt handlers)



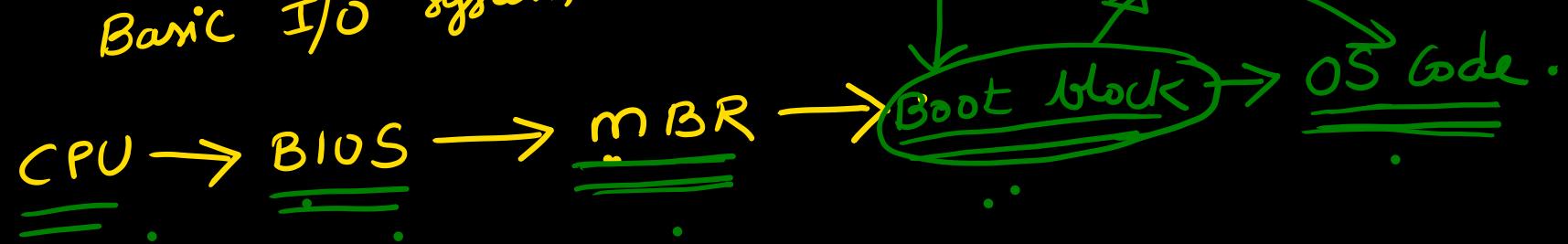
Devices

F& a:

You have windows and linux in the
same computer. Sometimes windows, sometimes linux.



Basic I/O System



on disk data structures used by File systems

several in memory (RAM) & on disk (secondary memory) structures are used to implement a file system. These structures vary depending on 'OS' and file system.