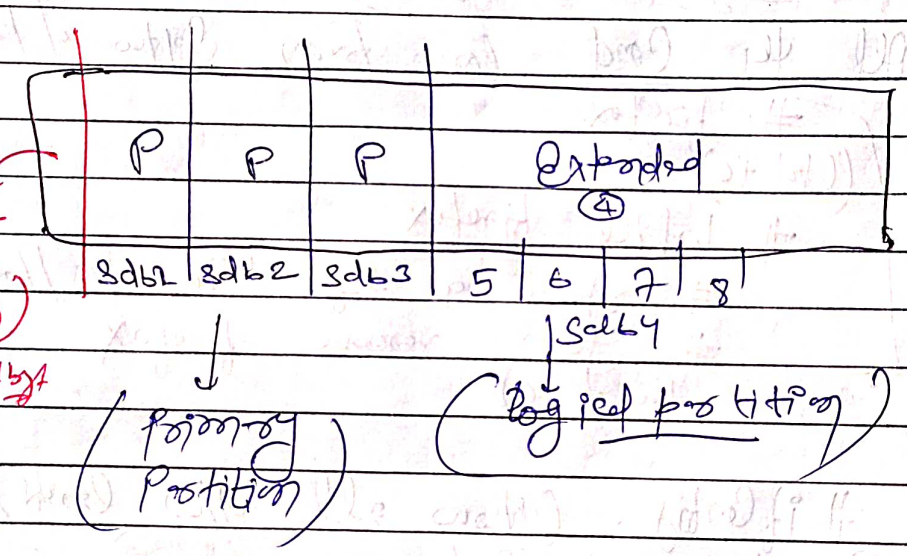
Session-33

* Partition

* Store data
  - → file
    - → folder/dir
      - → Hdd (device)
        (Complete data permanent)

* Persistent
  - → Block device
    - → Storage → HD
      - → PD
      - → Dvd

* Store data into hard disk-



* If we want to store data you need partition.
* HDD only know sectors, not bits or bytes.
  (1 Sector = 512 bytes)
* You want to create partition you have three rules-
  1. Create physical partition → P → 4
  2. Create extended partition → E → 1
  3. Create logical partition → L → ∞

* #fdisk -l (Show list of attached hdd)

Date ____
Page ____

* If you want to create partition, after partition you want to use it, create the drive (C: / D:) → create drive.

② format
③ mount

* Any world if you want to store data & create partition for it these three steps will help you.

* Today's topic → Format

→ If you have drive (:D) → if you want to store data in it you can store. But if you re format this drive what will happen.

Data [(:D) → reformat]
↓
Data will there or not.

→ If you remove your file with (Shift + del) permanently → data / file will remove permanently.

File → Data → (Shift + del)

* There is no way to remove your data permanently.

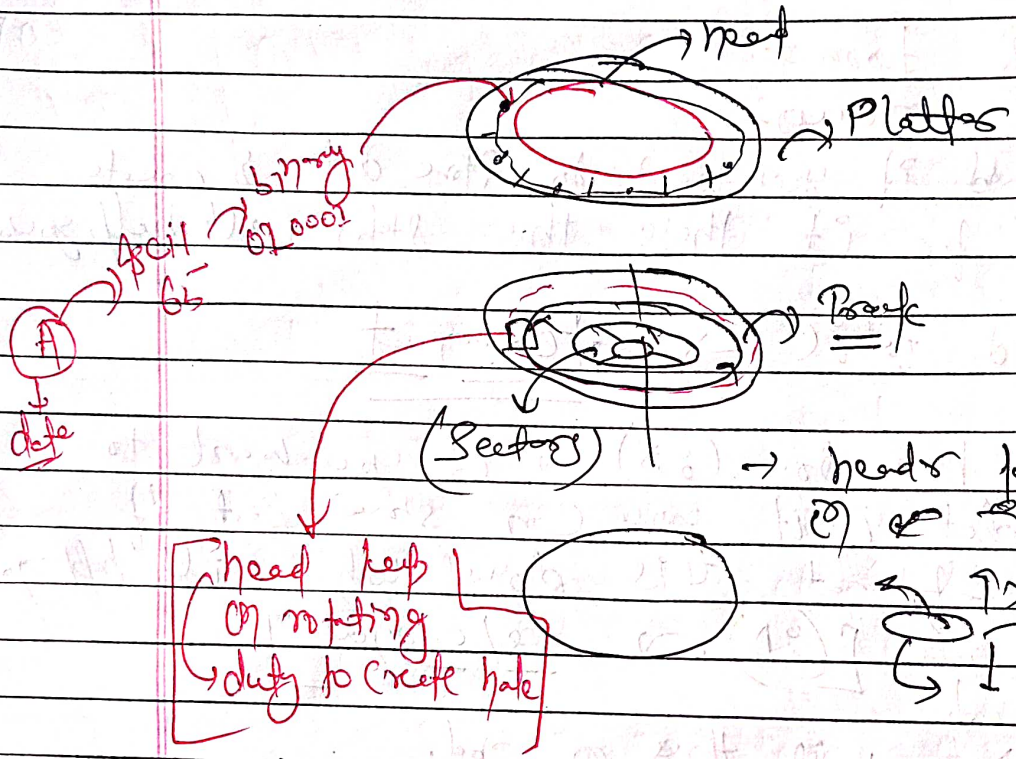
* We have electro mechanical hdd, if we format / reformat your drive the data never be deleted permanently.

* If we store data in hdd, they will create hole and there is no way to remove that hole.

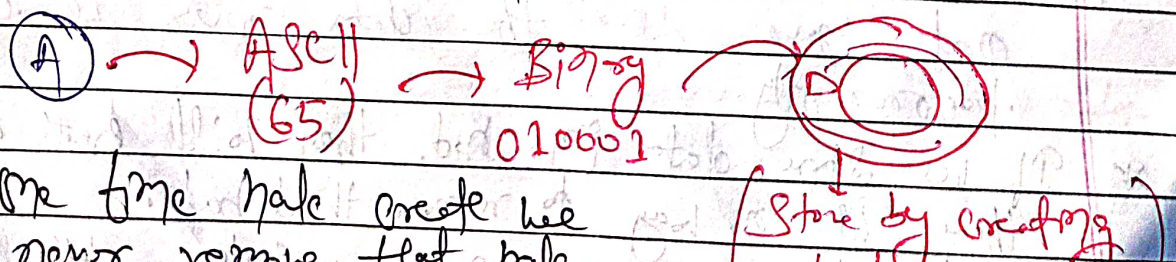
* That's why, we have recovery tools available in market.

* Lots of companies → giving guarantee → Tools → 99.99%
↓
(Shred tools)

* If we delete by (Shift + del) or from → we can recover that data.
* Internal architecture of hdd →

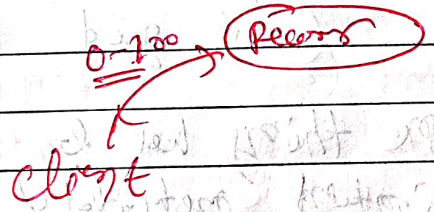


→ By hole they store data.
* If you store data into hdd they create hole, now there is no way to remove that hole.
* H.d is electro mechinical device they have different different component available.
* (Platters, head, Tracks, Sectors → rotting & create holes)
* we store data in h.d →



→ One time hole create we never remove that hole.
* Now, hdd have flame plate when we feed/store data they create hole → that hole never removes.

* If you feed the data no way to remove data physically.
* Tools (shred) → they go to that h.d & they go to all locations & changing the direction of holes → upper hole as to lower hole, lower to upper hole.
* They keep on changing direction's that's why we get data by shred tools.
* They are just filling leakage.
* If your file removed then ∴ data won't remove.
* @



* If you lose your data, without touch any data or h.d, we can recover that data → But if you create file at that position where data lost if hole created they fill/or hole changed ∴ then we can't get data.
* If hole is not override to that position.
* If you want to copy two files →
  ① file ① → 1MB → 20 sec
  ② file ② → 200MB → 100 ~~sec~~ min
* When we remove any data, technically they will take more time, but they rapidly delete that file from pc why?
→ There is not any way available that can fill & remove the hole that created. so, that's why they delete.
* If you store the data hole created.
* If you store data we can get back data if you won't override at that place, new hole not be created.
* If we delete data rapidly delete ∴ no way to fill hole.

* When we create report in School / college →

Eg:- Lab Experiment Report.page

index

(at first page)

Exp1	Exp2	Exp3	Exp4

← 2000 pages report →

→ if you give directly they will go & search the data. (1 --- 2 --- 996 page)

(not good practice)

* we can do one thing we can create one index page that contains meta-data / all the list of tables where which experiment is there.

Index Page

(Index table) →

Sl	Exp	pg no
1	Exp1	1-10
2	exp2	10-100

(This makes Searching easy)

* In future if any body want to check they first look up index & directly go to that page.

* In world if you want to make searching easier indexing is my way.

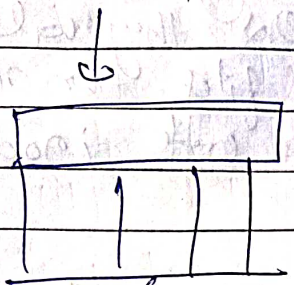
* Index table we have manage all data.

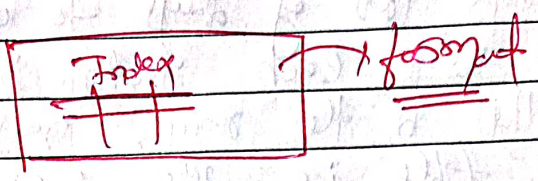
* ↳ This is formated in our way, my way to arrange.

* ↳ My way to manage / formated → to make search faster.

* When we create new partition → partition equal to report sheet.

* Partition == Report sheet.





↳ (Index → Inode Table)

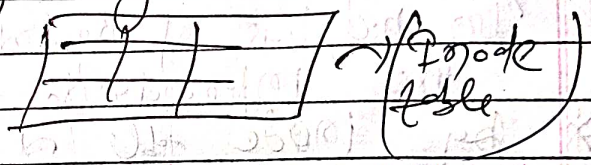
* In hard disk we store data in sectors.
* In h.d lots of sectors → in inode table they store that information.
* By inode table → data fetching easily.

→

SL	Name	Sectors
1	File 1	100..1000
2	File 2	100..2000

* When we create file → they maintain it in table.
* After creating the partition → inode table is configuring to create in any OS bcz if we have data stored of 10gb → if we want to access that data click on it & access → quickly they launch/access that data bcz the table (inode table) contains meta-data about it.
* In other hand we have, we have data of 32 GB stored if the meta-data/ information of that data is not there in table then we have to access that data it will take lot of time in fetching to each and every sectors.
* So, that is reason putting all the file name & maintaining the file in that inode table → That's why they are called file system.
* File system / format
  - Type → FAT32, NTFS

* If you won't format the drive may be it will take lots of days to open or they will go to each sectors & find files then access.
* That is the main reason to create inode table i.e., file system.
* When create partition →
  ① we format that drive / partition
  → Basically they create table



* You can think one thing when you put any movie in any drive they will suddenly show (total size & free space). How they calculate quickly?

→ Or if it is (Exact) they will quickly go to that inode table / f.s and read the last column, size & calculate it (Bcz computer is good at calculation).

(File system) → (Way to format after partition)

Inode Table

id	name	Location	size
1	f1.txt	1000 - 10240	11k
2	f2.txt	12345 - 99999	2k
3	f3.txt	100545 - 20547	1G

OS → (quickly they calculate)

→ Some thing when you remove file, they quickly → they just go to that table remove from table the data how they access.

* When we remove data → They remove info / meta-data from inode table just (so quickly they delete).

* So, that's why storing any thing takes lots of time and removing any thing just quickly.

* (Data) → (Store) → (write on h.d)
  ↓ the time to get sectors
  (put info. in inode table)

(remove) → remove info from inode table)
↳ | 3 | f1.txt | 1054 | 4k |

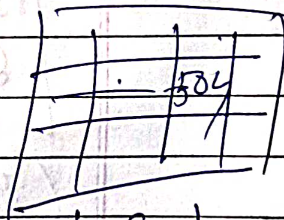
* That's why recovery tools, removed data backs.
* Some times we saw hackers will remove all the data from hd by program.
* first they detect the h.d size then they re-enter the inode table / meta-data from table manipulate that by program then they feed program & entry all all size (50 GB) then we to think our data has been Losted.

P.T / F.S / Inode Table

(Hacker) →

id	fname	location	size
1	f1.txt	400	1G
2	f2.txt	200	2G
3	f3.txt	00	47G

feeded → Reentry



→ (Look like all data losses) → (update fd)

* device level program → hacker creates.
* Inode table → way to manage the partition.

* When we reformat what happened with hd / partition?
→ When we format any partition / drive they will remove older inode table they will put new fresh page → then OS will assume all data removed & new free space comes up (changes done only in P.T)

* you have your own report sheet → ④
* (both have different way to create index page)

Index page

(Jeric) ⇒

id	file name	page	creat
1	f.name	100	9pm
2	f.name	200	

(Report → Index page)

(Vimal) →

SL	fname	page	end	tiny
1	—	—	—	—
2	—	—	—	—
3	—	—	—	—

(They creat file creation time)

(Data) → win

(Index) → format → Type → NTfs
→ FAT 32
→ Ext3 → (They won't maintain file creation time)

(Report) → Partition

* NTfs → maintain file creation time.
Ext3 → Never maintain file creation time.

↓ (in Linux)

touch z.txt

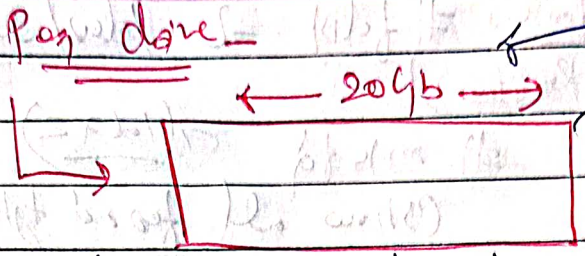
ls -l z.txt

→ (All info we get from inode table), don't modify creation time.

* format → have different → different types.
↓
Ext3 / Ext4 / Xfs / zfs

↳ main thing what for they maintain.

* In Pen drive →



→ When we format our pen drive → NTFS format they will create new Mode table stores 20 Gb.

→ When we format our pen drive → fat32 format they will create new mode table but they are restricted only in (4Gb) → my file size is 20Gb but we can able to store only 4Gb. (Restriction in file system / mode table)

* best thing is which format → partition table to use. you choose.

① fast → Create partition (allow)

② Second → format

* Why which type you have to decide, what kind of data you have to store o/c to that format it & create Mode table.

* File System → how data stored & retrieved.

* Requirement to requirement we decide format type.

* Format type decide how much data to be stored.

* One hacking tip → in Linux → Create file on loop → till fs time really at end o/c to format type then all PT will full with 0's all data done used.

* In Linux we use (Xfs) → format type.

* Let's move to the cmd →

* See the drive → # fdisk -l /dev/sdb
(show All Partition)

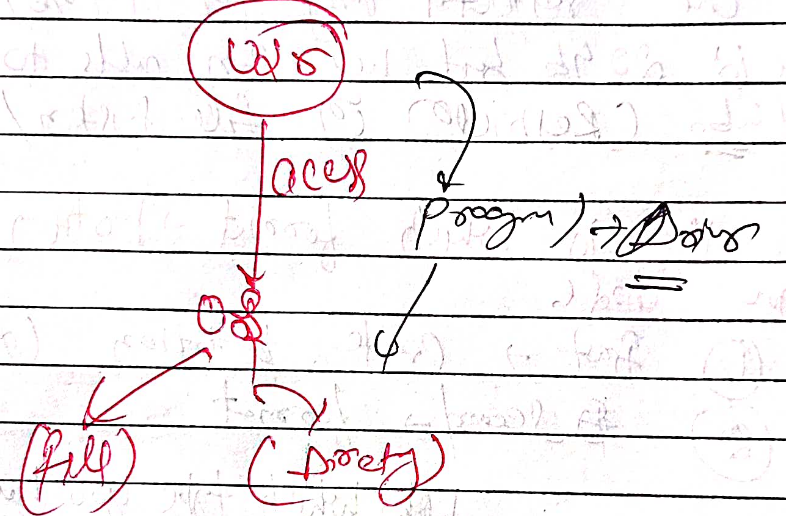
mkfs → (Tab×2)
(show all format type support)

format it

mkfs. ext4 /dev/sdb1
(This way we format)

→ After format we see inode table created.

*



* Partition is one kind of drives

/folder → /dev/sdb1 (mount)
↳ file
↳ data

(now user can access data)

User → (Can't go directly to that device)

* User Can't directly go to that device, we can like one file we can create folder & link at with /dev/sdb1.

mkdir /fold

(linked with partition) → # mount /dev/sdb1 /fold

(whatever you create in the folder they stored in partition)

cd /fold

cat > Ima.txt.

pwd

↳ /fold

df (show all the mounted folds with partition)

df -h (more info. they give)

umount /fold (unlink the mount above)

mkdir /b

mount /dev/sdb1 /b

cd /b

ls

* files stored inside (/dev/sdb1) are in /b.

→ (we just go inside # cd /b
Finally you landed to /dev/sdb1)

(in order to see the folds
it done
(in order that folds
it mount point)

* This way we found the partition & mount it to mount point.