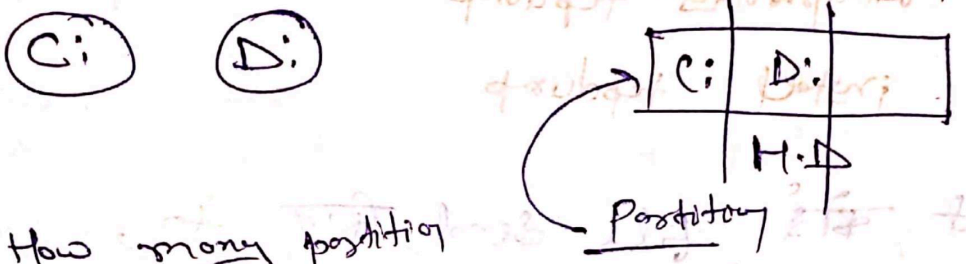
Linux Partition

* Data → Storage → H.D → SATA
  → Storage
  Sector = 512 bytes

* Hardware Partition

* In windows we have partitions –



(1) How many partition we can create?

(2) Partn → Drive → reformat → Data → Lose?

(3) What is min size of partition created in win / Linux?

(4) In 1Mb how much kb we have?

① < 999 kb
② 1000 kb
③ 1024 kb
④ None of these

✗ → 1024 kb

✓ B → 1000 kb

* 1Mb = 1 kb
* 1Mb = 1000 kb
- Human being calculate unit 10 to power.
- Computer calculate unit 2 to power.
* Unit → HB → 10 (to the power)
→ Computer → 2 (to the power) These are units for
* 1 kb = 10^3
km
kHz
kb
→ Human being (Human Readable)
* for Computer they are kib, Gib, Mib.
1MB = 10^3 kB = 1000
1MiB = 2^{10} k.B = 1024 k.B → Actual units
→ Binary
* RHCSA → (RHCSA) → most specialize
↓
442 (Training module) (Advance performance tuning)
(one dedicated chapter about unit)
* In 1Mb we loss 24 bytes loss.
* जब हम 2 TB की hd लिए, 2tb
Company selling, in every 1 kb we loss our 24 bytes data, so जो 2Tb की space कम होता है। we won't get complete space. 2Tb.
→ का 2^{10} = 1024
2TB
→ 1.8TB (we get)

* Company are selling $2^{10} = 1024$ to us but internally they follow decimal 10^3.

$1MB = 2^{10} KB =$ ~~1000~~ → X

↓

1000 KiB

* We have some cmd's to see this →

lvdisplay (This cmd give real knowledge about partition in H.D. in GiB)

Partition

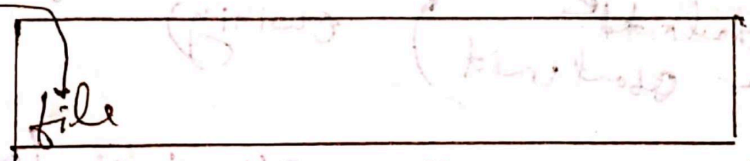
* Data → Persistant

↳ H.D / Flopy / Pen / DVD

↳ Storage

HD = Storage

Data



→ Without creating a file we can't store any data.

→ For creating file we need Directory

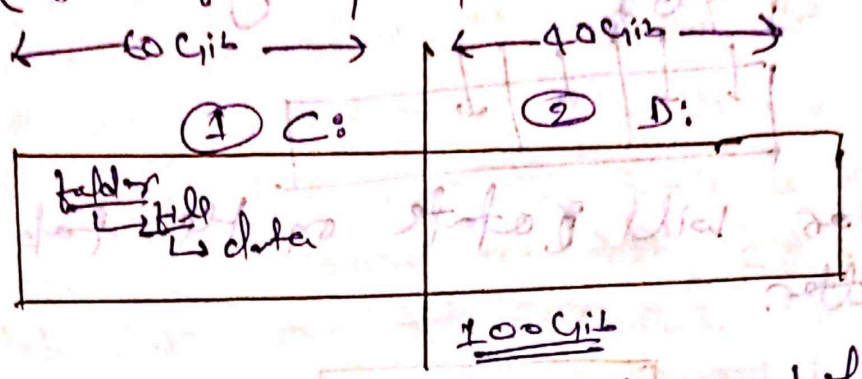
→ For creating directory we need some Partition / drive.

Data

→ (Partition / drive)

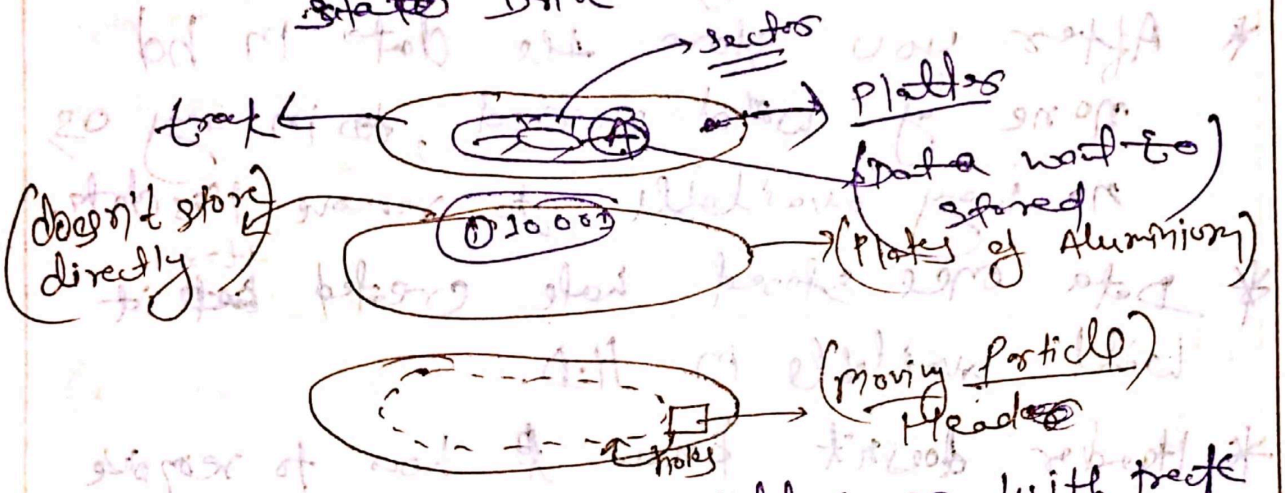
(GiB)

* If you have H.D (100 GB), first you create partition, then create folder, then create file, then put the data.



* Why we need multiple partition? What is actual need of partition.

* H.D → Electro mechanical H.D.
Hard Drive



→ DVD kind of plate is platter, with track & all track divided into sector.

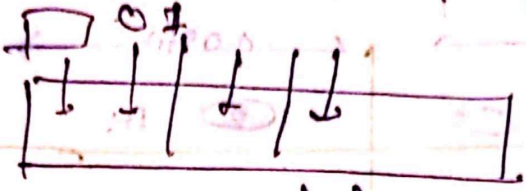
→ Data store per sector wise, Data store in sector.

→ They convert into ASCII then binary.

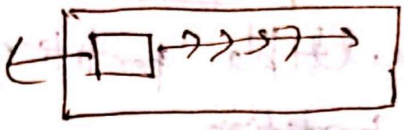
A → 65 (ASCII) → 01000001 (binary)
store

* Binary doesn't store data directly.

* When want to store data head, head will create a hole & store the data in bit by bit.



* Header will rotate on the tab by platters.



* After the hole created, we can't create it again.
* After you store the data in hd none of hard method, in any OS no way available to remove this data.
* Data once stored hole created then it will available in H.D.
* Header doesn't know how to remove the hole
* If data remove permanently, from OS, then by recovery tool we can get back our data. Bcz hole created data stored in H.D, Permanently, we can get it back.
* Shred remove file → Tool to get back data from H.D. (Data H.D)

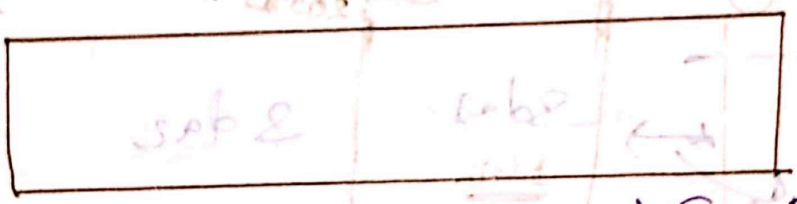
* अगर मेरे Sata Hard disk में कोई directory से file delete हो गया O.S में, then we can get it back, bcz it creates some holes in H.D, we we store the data. So, by some shared/recovery tools we get data back but agar वो same location में data वहां store कर देंगे, matlab data override कर देंगे तो वो data पर override हो जायेगा. we can't get it back.

---

* Lets know detail about H.D →

sda/hda/vda → hard disk name
sdb/hdb/vdb

← 100 GiB →



(mb/Gib → O.S)

1 sector = 512 bytes
= $\frac{1}{2}$ KB

* Hard disk only understand sectors.

#fdisk -l (to know how many hard disk we have)

Disk /dev/sda: 100 GiB → 209 715 200 sectors (This can't know right units)
Units: sectors of 1 * 512 = 512 bytes

H.D know only sectors.

* Dev is the folder it will manage all the device folder.

* In our 100 GiB we have 209715200 sectors are available we see in GiB this sectors

lc

→ calculator option

512 * 209715200

107374182400

(512 * 209715200)/1024

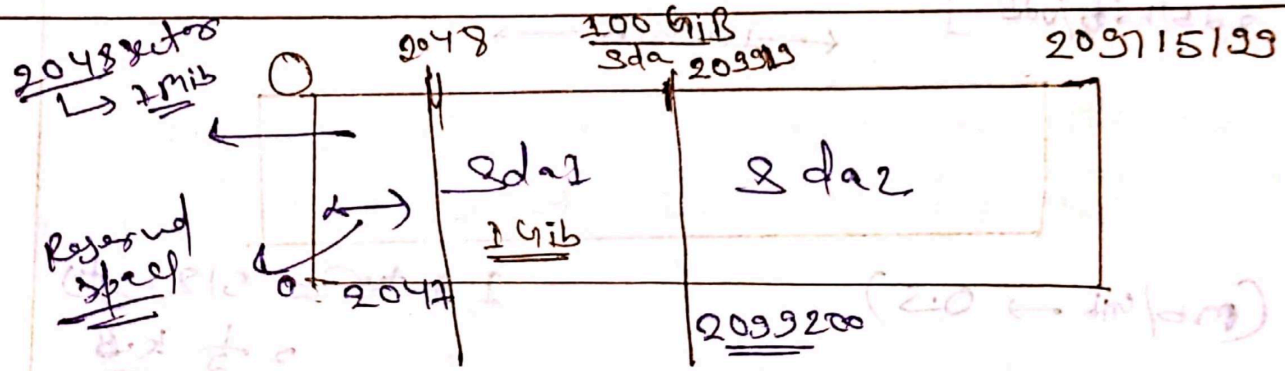
104857600 (bytes)

((512 * 209715200)/1024)/1024

102400 (mib)

(((512 * 209715200)/1024)/1024)/1024

100 GiB



* A/c to my H.D, when we install the os it will create 2 partition

* sda1 = 209919 − 2048
=

* for new partition sda3, we need unallocated space/Sector in H.D.

* If sda2 full we can't no install/store data.

* New Partition → New H.D ⇄ go to market buy

OR

→ virtualized

H.D → virtual Hard Disk

* In virtulizat^n we have technique we won't go to market & purchase new H.D. you can create own hard disk by decide size of space/size.

* we go to the s/w ware add virtual hard disk of 10 gib.

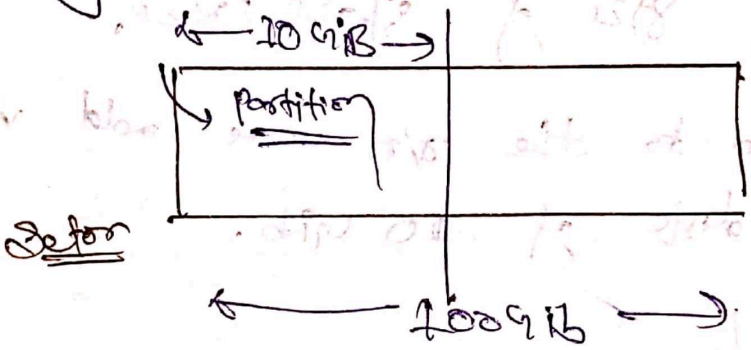
fdisk -l

→ we take two hard disk now.

* Data → Storage → H.D (SATA)

Sector = 512 byte

* If you want storage from hard disk, at first you have to create a partition. You have to tell hard disk how many sector you want.



* #fdisk -l (to know how many hd you have)

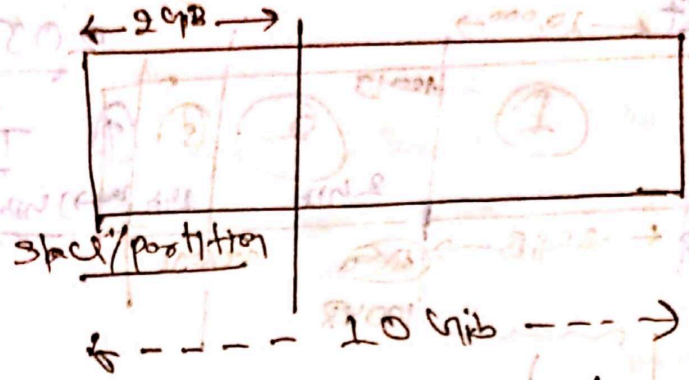
R. drive I
dev 2

fdisk -l /dev/sda
(To know a particular hard disk)

* Data → (Store data permanent) → Storage
↳ 2 GiB → need space → (Partition)

* Why we need partition?

→ we have 2 gib data we want to store permanent, we need storage, How till the storage we need 2 gib space to store my data is partition.



→ If you want some space from storage.
- P.D
- H.D
- cloud storage

*for new → Partition/space:

Steps (3 process)

How to create a partition →

(i) you have to finilize how much space you need. Go to partition ~~create~~ Hard disk create a physical partition.

(ii) After create the partition you have to format it.

(iii) After you ~~have to~~ formate you have to mount it.

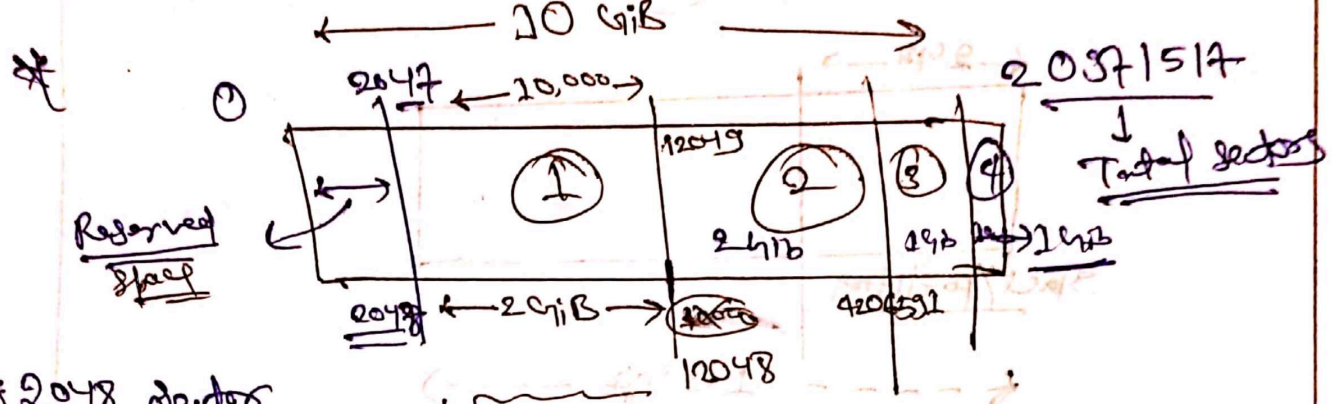
→ If you do same these 3 step process, finally your drive will be ready.

Data → Drive C:

* Step (1) create a partition →

fdisk -l /dev/sdb

Disk /dev/sdb: 10 GiB, 10737418240 bytes, (new drive we 20971520 sectors attached of 10 GiB)



* 2048 sector
↓
Reserved space
↳ 2048 * 512
= 1048576 / 1024
= 1024 / 1024
= 1 MiB

↓
12048 - 2048
⇒ 10,000 created

HDD → 1 GB
1 GiB 2 GB 1 GB 1 GB

⇒ Now we create a partition of 2 GiB.

fdisk /dev/sdb (जो 10 GiB में से अंदर गई)

cmd : p
cmd : n

p → primary
e → extended.
↳ enter.

Partition no. (1-4, default 1):
First sector (2048 - 20971519, default 2048):
↓ ↓
We have space from one to 2099617. 2048 This is reserved space.

where you want to start partition 2048.
↳ Enter
Last sector, : +10000 → (Type size in sectors)
created new partition 1 of type 'Linux' &
size of 4.9 MiB.
Cmd : p

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sdb1		2048	12048	10001	4.9M	83	Linux

Cmd;
Cmd: n (Here create new partition again)
Partition Type → (Starts 12049, 4206591)
p → primary
e → extended
Select (default p): (↳ Enter)
Last Sector, : +2G → (Size in MB, GB)
created a new partition 2 of type 'Linux' & size of 2 GiB.
Cmd; p

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sdb1		2048	12048	10001	4.9M	83	Linux
/dev/sdb2		12049	4206591	4194543	2G	8e	Linux

*Now creat third partition →
Cmd : n
Select (default p) : ↳ Enter
Last sector : +1G

Cmd; p

Device
/dev/sdb1
/dev/sdb2
/dev/sdb3

* Again more.

Cmd: n

Select (default e): p

Last sector - - - : +1G

Cmd : p

Device	Size
/dev/sdb1	4.3 M
/dev/sdb2	2G
/dev/sdb3	1G
/dev/sdb4	1G

* In one harddisk you can created only 4 partition. (win/Linux/mac).

* Any Company H.D, you can create, only 4 physically partition.

* Now we have 10 Gib, we cred 4 partition of 1Gib, 1Gib, 2Gib, 4Gib., so, we have 6Gib more unlocated space left.

* If we create more, it will fail.

→ unallocated / sector space.

* we can't create 5th partition physically we won't able to use 6 Gib space.

* /dev/sdb1 2048 4MB

(Information of partition)

→ In above partition, this is data stored the meta data of partition.

→ If storing one string of meta data is 16 bytes.

→ These information stored inside the table. (Partition Table)

→ 64 bytes total information (meta-data)

→ $\frac{64}{16} = 4$ → max 4 partition in drive

→ Partition stored in reserved store.

→ In our case 1 MiB → store 64 bytes meta data information

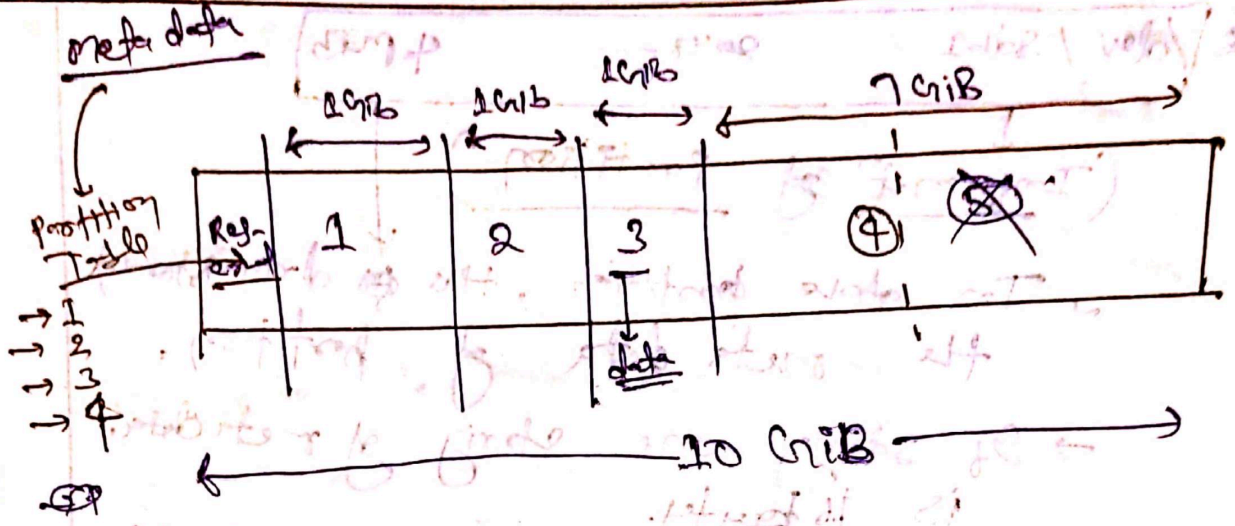
→ If storing one information of 1 partition is taking 16 byte space. This information stored inside the Partition table. Size of partition table 64 bytes in size (4 partitions).

This is reason we create 4 partion only.

These information stored in reserved space.

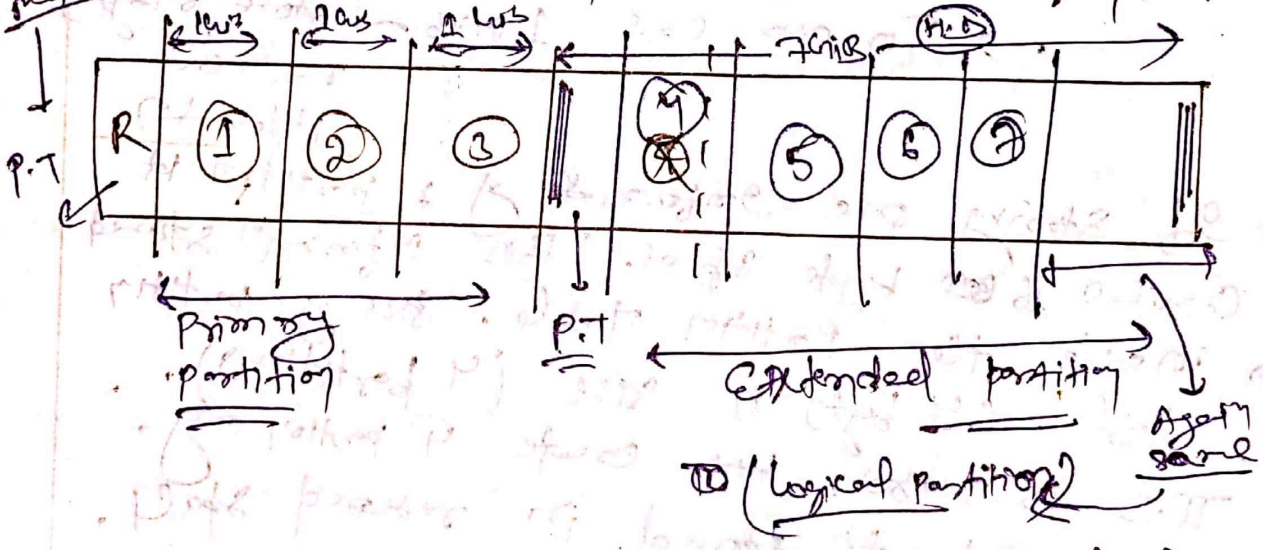
---

* we see many place there are more than 4-partition available. Here, we do some manupilation.



→ partition is created to use the data.
→ In Partition Table we can create only ④ partition.
→ Partition Table is type of Program.

*. we can't able to create 5th partition.
Now we can do some technique/trick---



* यहा हमारे 3 - partition create कर चुके है, Now, हमारे पास → 1 GiB, 1 GiB, 1 GiB space है, 7 GiB unallocated space है। so, we can treat 7 GiB as new Hard disk then now. here we have new partition Table & Reserved space. Also now can create maxm 4- partition.

(20GiB)

* Primary partition → P1, 1, 2, 3. (3GiB)
* Extended partition → P1, 1, 2, 3 (3GiB)
* Now you want to use more 4GiB.
* Logical-partition → P1, 1, 2, 3.
→ These are three time partition.

* जब पहले partition create करते time P को
  था primary partition था।
* इसके बाद वाला partition would be
  New HD (unallocated space) → Extended partition.

* for deleting partition.

: n
: d
: 4 → (4th partition removed)
→ Now we have only 3 left. (1 - partition free/left)

* Now we create Extended partition → (Treat as New H.D)

Cmd : n
select : e
(↳ Enter)
→ Give all remaining size
(↳ Enter)
End : p
Drive
① ② ③ ④ ⟶ 7 GiB (All size give)
Extended partition

* 4-partition (7 GB) is Extended partition. In future when we need we can extend more. (Type → Extended)

* We can't store data in Extended partition.

* n → new partition
All primary used ⇧
Adding logical partition 5th.

Last Sector : +1G

Cmd : p
(↳ Enter)

* We can create extended partition, then store the data. (logical partition)

* Partition → Physical
  - primary
  - Logical

* In Extended total 60 Logical partition we can create.

* 3P + 1E + 60L = 64 partition
63 partion → usable to store the data

* minimum size of partition is 1 sector, 512 bytes, we can create.
* 1 minor partition = 1 device.
* In our program, developer set the limits.
* /dev/sdb → /dev/sdb1
  ↓ (new device come)
  main drive — driver
→ OS don't know how to connect with hardware, there driver help to connect to program.
* #lblk → (List of block devices)
  (List of disk & how many partition, size we have)
* Block Device → where we store our data. we can store data here.
  → usable.
* when we create new partition we have to load the driver. of the partition.
  # partprobe /dev/sdb
  It command work in red hat - 7
  # Udevam settle → (red hat - 8)
  (load the driver)

Session-28

* 1 kb = 1024 byte

Mib → 1024 kb → 2^{10}

Mb → 1000 → 10^3

1

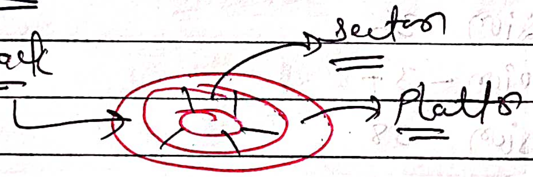
4Tb → 1024 × Every kb they

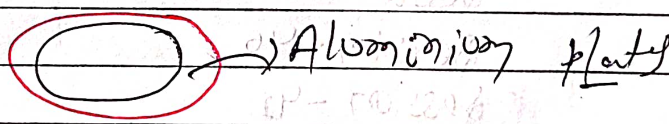
↓ L, 1000 have byte loss

3.7 we get

* #lvdisplay → (All details & Gib format)

* Hard disk -

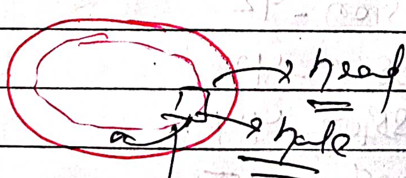




A → 65 → 01000001

↓

(hole)

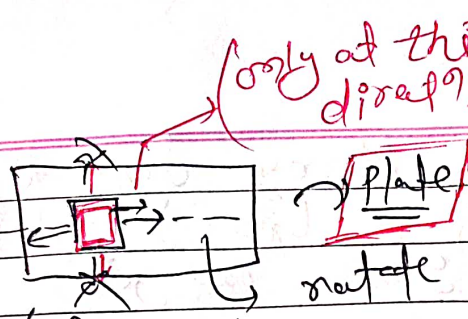


hole keeps on moving

→ head will do hole in plate & data finally store in hole.

→ If you want to read data from hdd they store data in holes. To read we need drivers.

→ from hole we can not read directly.



* After you store data in h.d no way to remove data in h.d bcz they create holes.
* We can override or replace data from hdd, we can't remove data permanently.
* Shred tools : Recover your data from hdd.
* H.d don't know about Mib/Gib/Mb/Gb → only for Computer & human understandings.
* H.D → only know about Sectors = 512 byte
  See how many hd you have → ⇒ 1/2 kib =
  # lsblk -2 (list of attached disk)
  (Show size in Gib)
* (Disk → /dev/sda → 100 Gib) → one disk
* [sda / hda / vda]
  ↓
  (first hd)

→ All devices managed inside /dev/ → folder.
→ If you multiply no. of sectors * bytes (512) = get Gib ⇒ # bc (bmg cal)
  # 512 * 20971520 ↵
  # 10737418240
  # [(512 * 20971520)/1024] ↵ no. of byte (Byte)
  # 10485760 ↵ (kib)
  # ((512 * 20971520)/1024)/1024
  → 10240 (Mib)
  # (((512 * 20971520)/1024)/1024)/1024
  (↓ 100 Gib)

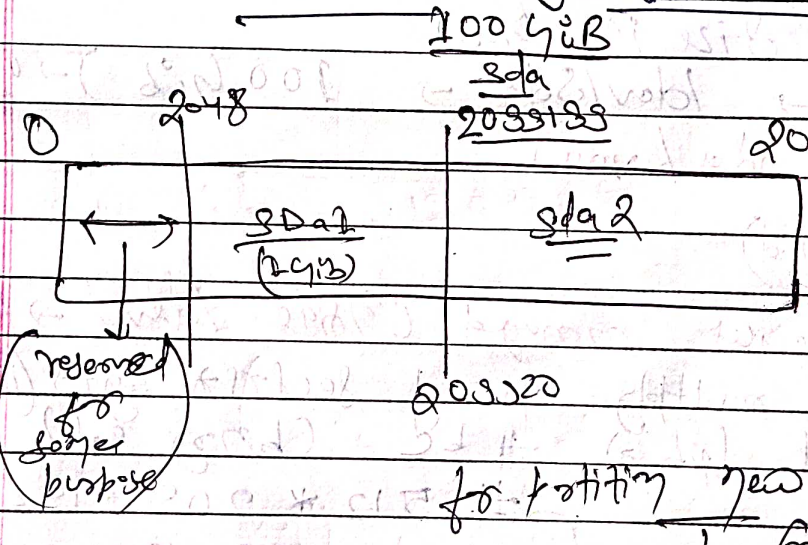
* Companies selling storage devices in GB but they actually in GiB so byte to →
  20 GB → 9 % (we get)

* We create partition to store the data → directly we can't store.

* When you installed OS the two partitions automatic created.

→ for free space we need unallocated space → free space. (sectors/space) → we need

* Architecture of current H.D -



for partition new
↳ (need unallocated sectors/space.

* creating new partition
↳ use new H.D
① market go & get
② OS (virtualization)
↓
(virtual H.D)

(we can create) →

* # init 0 (Shut down OS)

* Manually in VM → add new virtual hd (10 GiB)
# fdisk -l (Check all the drive attached)
(→ sda → sdb)

* Now, create partition →

Session-29

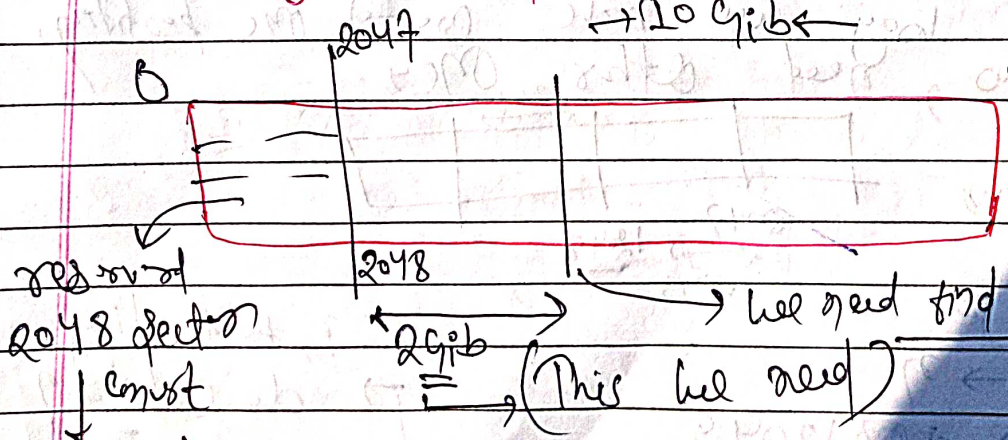
* # fdisk -l (All the hd list)
# fdisk -l /dev/sdb (Show particular hd list)

* need to store 2 GiB data → create partition
create partition (3) Steps →

Step (1) → decide how much space needed.
Step (2) → format
Step (3) → Mount / activate.

Store data → need drive → for drive three Steps

* last class attached → physical / virtual hd obj.
* Currently we have → 10 GiB

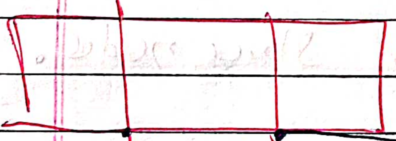


* For this we need to go inside hd & create partitions.

fdisk /dev/sdb (Take inside hdd)
They have own cmd.

→ P (detailed about hdd)
→ n (create new partition)
(p, e) → extended
↓
primary
→ (default p) ; hit enter.
Now, they tell sectors that are free →
(2048 – 20971519)
→ hit Enter
(2048) + 10000
↳ (last till 10k they have)

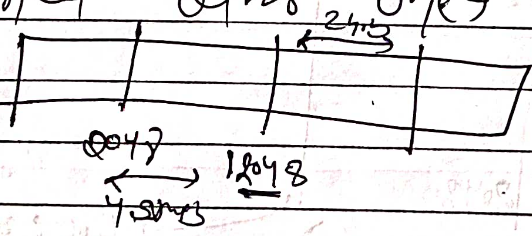
→ Till now we created partition



2048 (+10000) 12048
↳ 10000 – 2048
→ 7952 (no. of sector created)
↓
4.3 mib created after convert.

* This way we have created one partition
→ Now, need other one →

Again same process



→ ① → P
* → n
→ 1012049
→ +2G (we can say mib by 4)
created.

Internally they do calculation

* This way we have created partition →
→ (Now we can create 4-5 partitions)

* In one hdd → we can create only 4 physical partitions.
* If you have or not have unallocated space max they can create only 4-partitions. (They won't allow)
* How to create more than 4-partitions ?

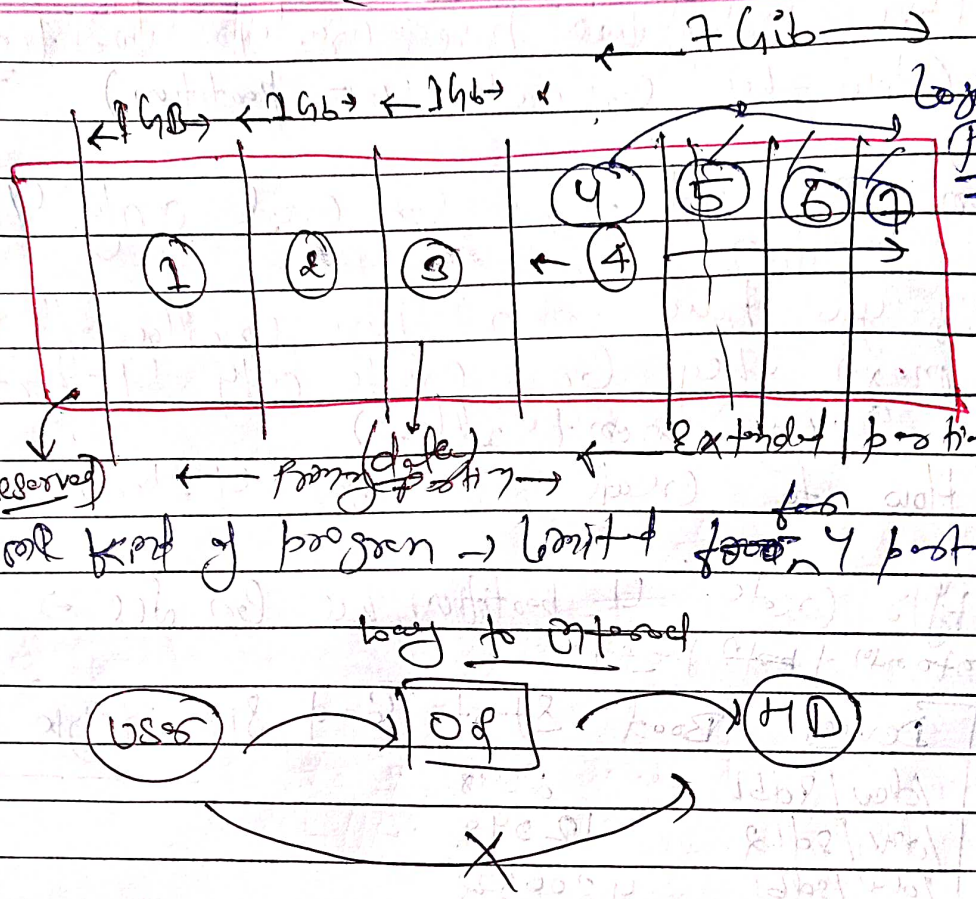
* After create 4-partition we can see →
→ Information of hdd partition

Devices	Boot	Start	End	Size	id	Type
/dev/sdb1		2048	--			
/dev/sdb2		12049	--			
/dev/sdb3		420652				
/dev/sdb4		6303744	-			

one partition info 16 byte space take

↳ This contains all the information about our hdd partitions
→ This is meta data of partition.
* One information of partition from partition table take → 16 bytes of space.
* 64 bytes space allocated to partition Table.
$\left(\frac{64}{16}\right) = 4$ partition we can create.
* Those reserved space in hdd (1MiB) → In that 64 bytes they take to store information in P.T.
* Now, how can we create more than 4-partitions ? → we can do many ways.

(10 GiB)



P.T → old kind of program → limited 4 partition

→ we can say → use 7 GiB as new h.d

7 GiB → new h.d

P.T

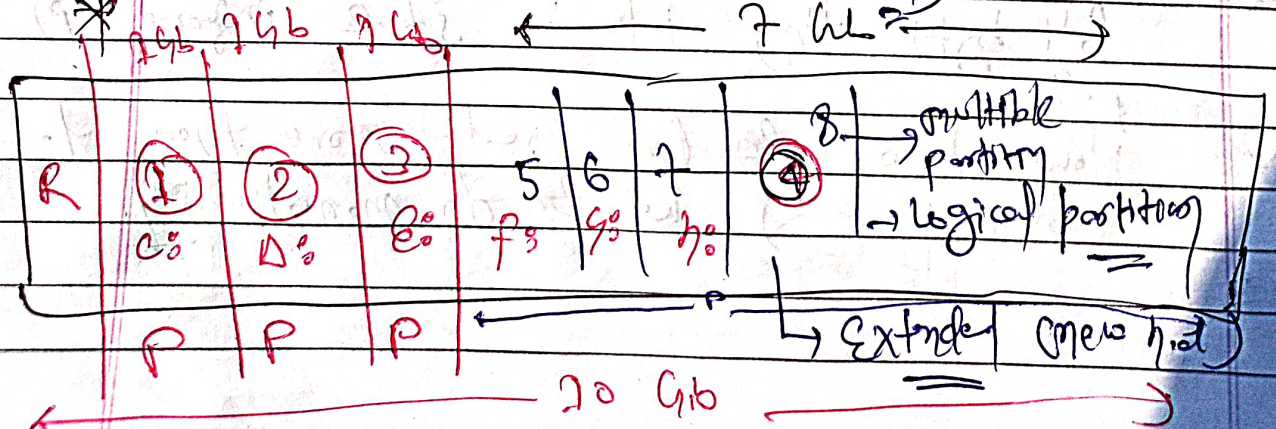
1
2
3
4

1 2 3 → Primary Partition

4 5 6 7 → Extended partition

(Information in main table & rem data)

(Extending the capability of partition)



* Always recommended → create last p all space as extended partition.
* We can't store data in extended partition → from that create new partition & store data.

:n
Adding logical :5
: +1 (starting sector for 1 GiB)
The more created inside 4th partition → logical partition.

→ Now, we can create multiple extended till the space ends.

* Store data → Partition
↓ Primary Partition → Logical Partition (for this create Logical partition)

* Inside extended → 60 logical partition create.

3 Primary + 1 Extended + 60 Logical = 64 partition

* (Total no. of Extended → logical partition created) in RHEL-8.

* Min size of partition we can create is (1 sector = 512 bytes)
: n
(12613632 – 12613632) +4
: p
(512 bl)

* 1 min partition = 1 sector.
: q ↳ Quit.
: w ↳ Save → (At last save all partition)

* /dev/sdb → /dev/sdb1
↳ new device

* need driver to connect to OS & device.

(List of block device) → #lsblk (tell all the partition)
or
#fdisk -l /dev/sdb (list out)

* Store data → Block device / Storage

#partprobe /dev/sdb
(Go & check new partition created they will update device drivers
↳ then we can store data
(Till version 7 we used #probe
#partprobe will work)

#udevadm settle (They will update)