***02 Block Driver***

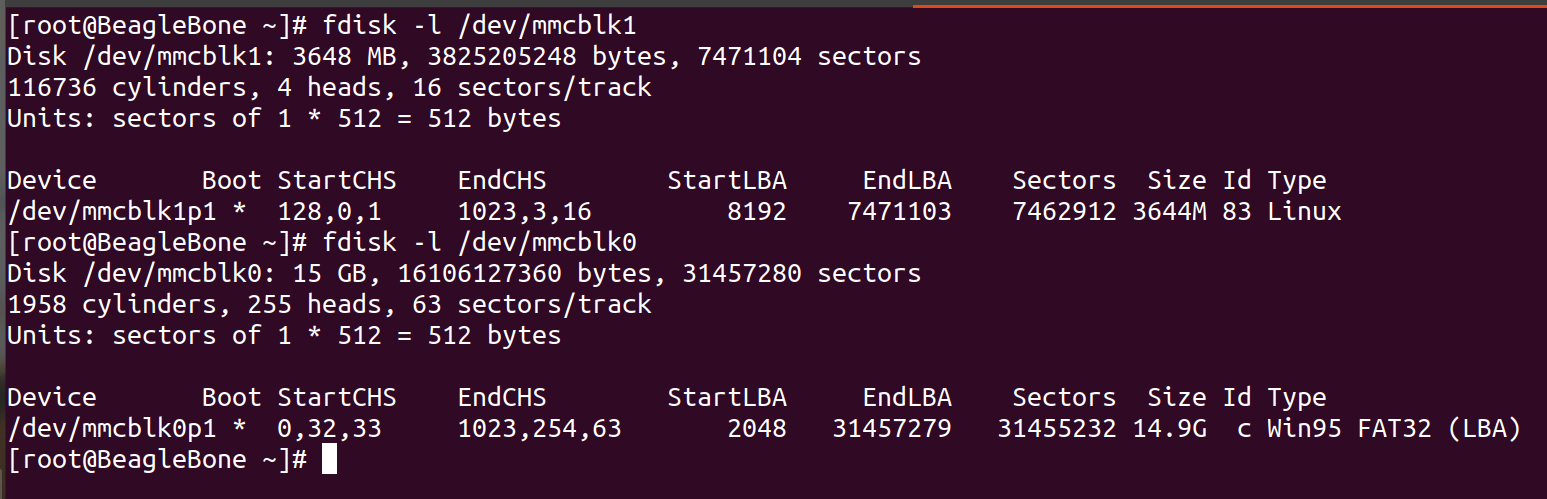
What to Expect

● Why the need for the Block Layer?

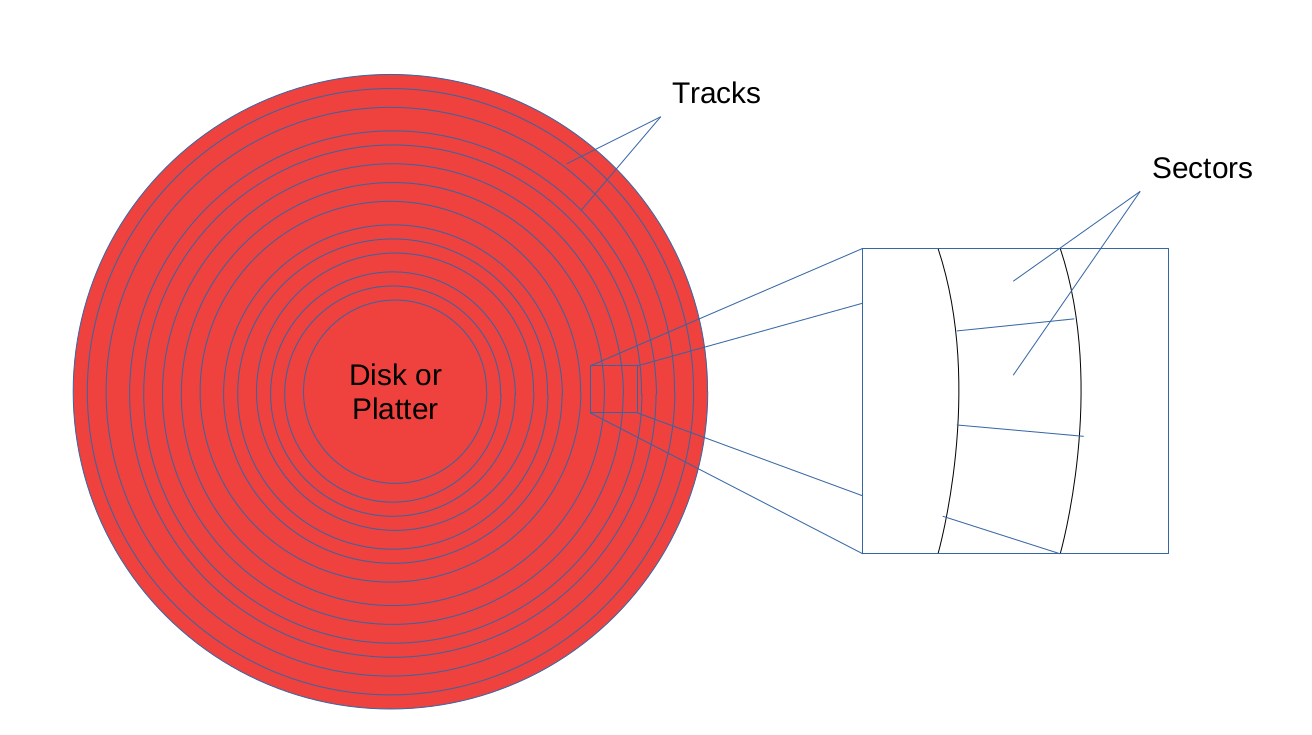
● Decoding a Block Device in Linux

● Role of Block Drivers

● Writing a Block Driver



**The Generic Hard Disk**



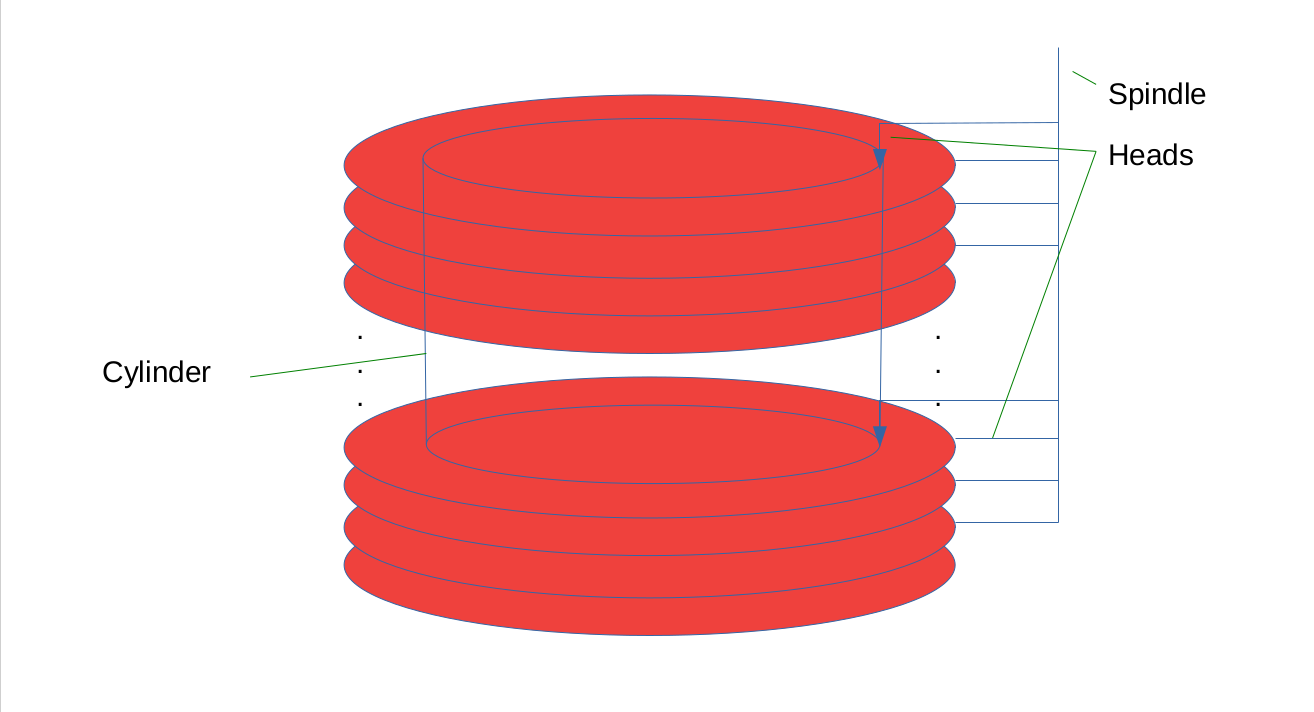
Consenctic rings they are basically track. These circular tracts basically devide into sectors.

Every track having something sectors.

Size of sector -: 512 bytes

You will find upto 63 sectors.

The General Hard Disc



**Computing a Generic Hard Disk**

**● Example (Hard Disk)**

**● Heads (or Platters): 0 – 9**

**● Tracks (or Cylinders): 0 – 24**

**● Sectors: 1 – 64**

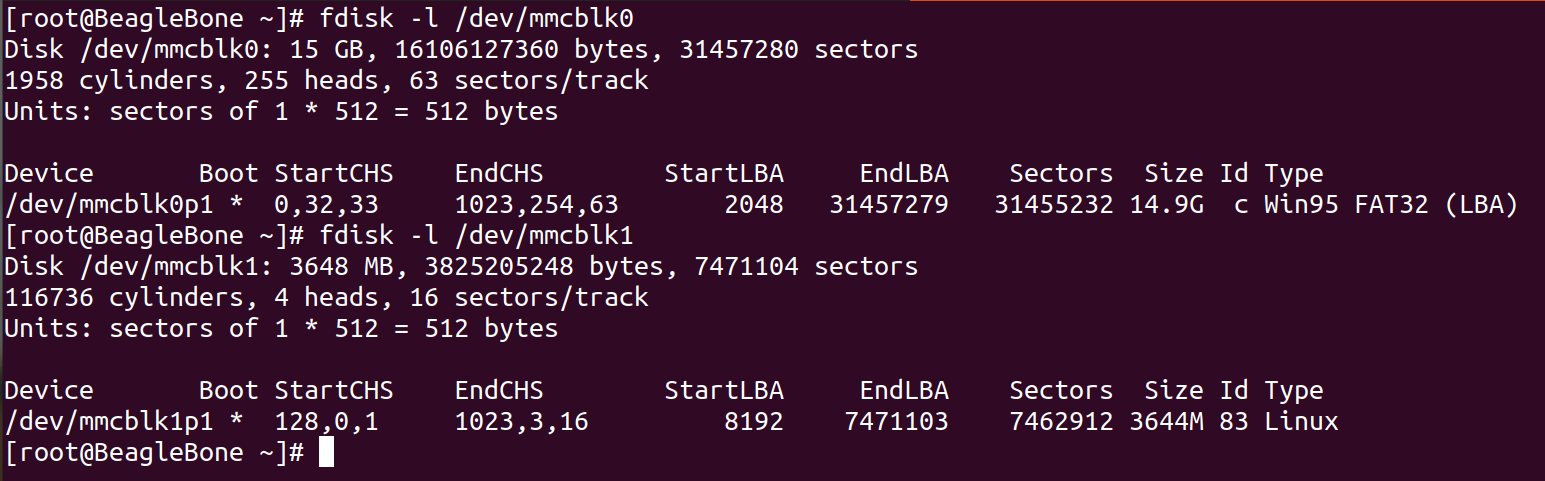
**● Size of the Hard Disk**

**● 10 x 25 x 64 x 512 bytes =81920000 8000KiB**

**● Device independent numbering**

**● (h, t, s) → 64 \* (10 \* t + h) + s → (1 – 16000)**

**How many byte per cylinder?**

****

**1 sector = 512 bytes**

**Number of sector x Size of sector**

**63 x 512**

63 x 512 x 255(disc) = 8225280

Partition & Partion Table

===================

* Divides the hdd into one or more logical disks
* called partitions
* Helpful in organizing different types of data

– Different operating systems data

– User data

– Temporary data

– ...

* Logical division & so need to be maintained by
* metadata – Partition table

Excersise-:

=======

Assignment #01

a. Use dd to create a 1MiB file named disk.

(Hint: You may use /dev/zero or /dev/urandom)

b. Use fdisk to set the heads to 4, sectors to 32, and cylinders to the appropriate number, for disk.

c. Use fdisk to create more than 4 primary partitions, or more than 2 extended partitions.

d. Use fdisk to create the following partitions using the complete disk without gaps:

+ 1 P (FAT32 (LBA)), 1 E, 1 P (Linux)

+ 2 L (Linux swap, Linux)

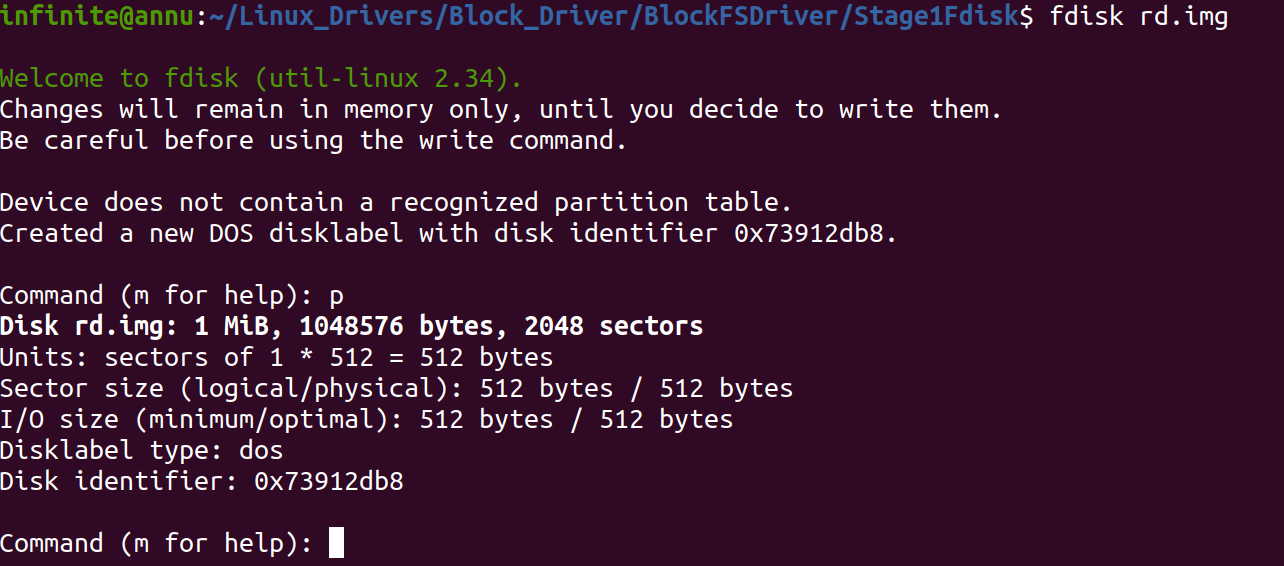
e. Display the disk partitions using p of "fdisk disk" in both sectors & cylinders (unit) display mode (use u).

Heads Up: Avoid using sudo in any of the above experiments, as you really do not need it.

1**. dd if=/dev/zero of=rd.img bs=512 count=2048**

#P|L | L|P

**2. fdisk rd.img**

****

disk\_size = nhds \* secs \* cyl \* size of sector

1Mib = 4 \*32 (512)\* cycl

1Mib = 65536 \* cyc

1024/65536 = cyl

cyl =16

1. Use dd to create a 1MiB file named disk.

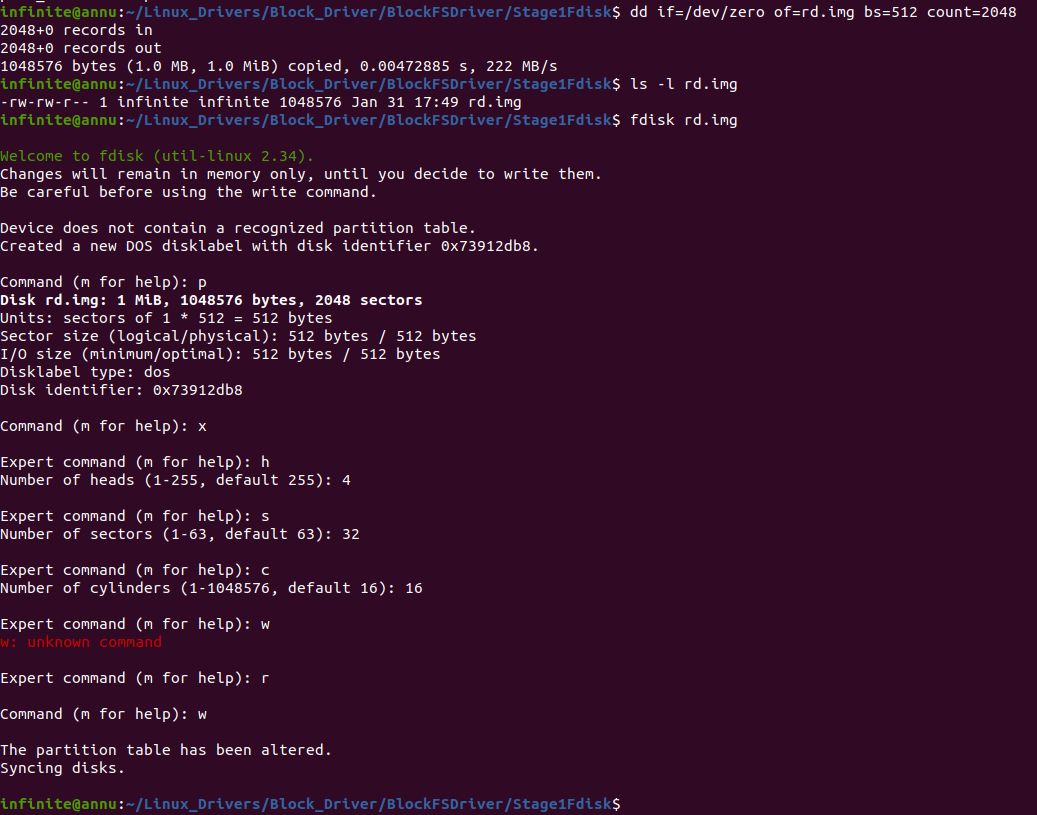
(Hint: You may use /dev/zero or /dev/urandom)

infinite@annu:~/Linux\_Drivers/Block\_Driver/BlockFSDriver/Stage1Fdisk$ **dd if=/dev/zero of=rd.img bs=512 count=2048**

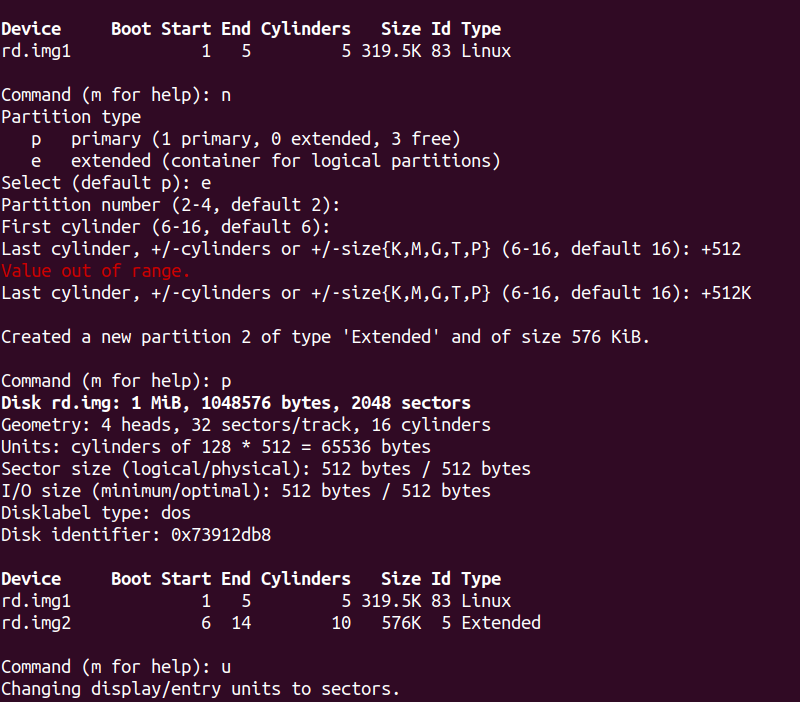
2048+0 records in

2048+0 records out

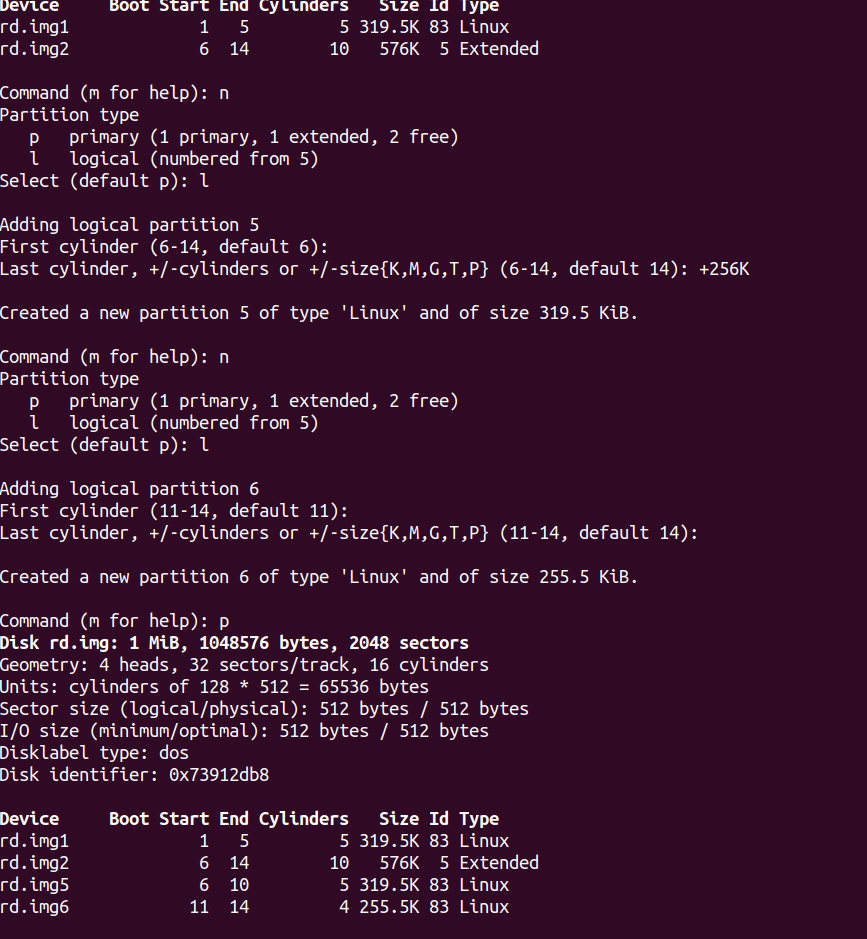
1048576 bytes (1.0 MB, 1.0 MiB) copied, 0.00472885 s, 222 MB/s

****

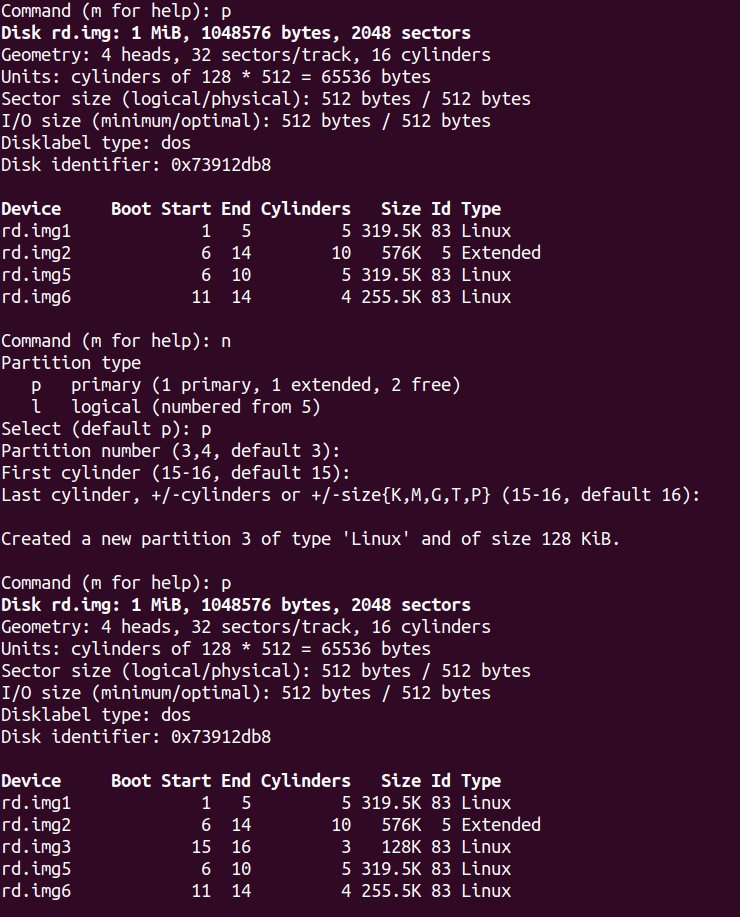
**Create 2nd Partition Extended**

****

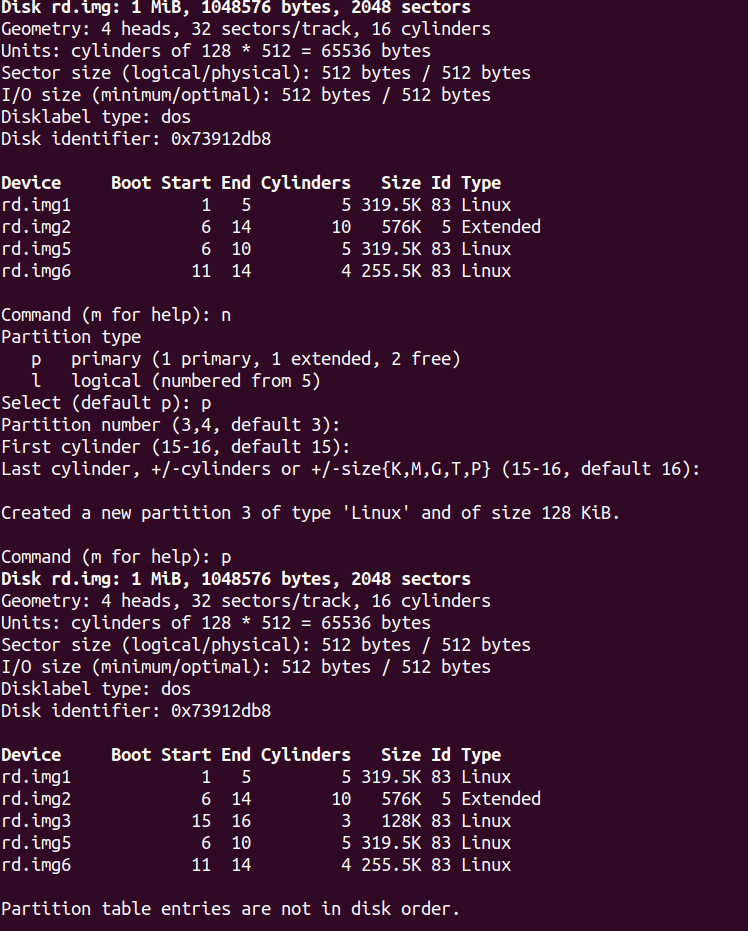
**3. Logical Partition (2 Logical Partition)**

****

**4. Now Primary Partiotion**

****

**Last Primary Partition -: w for save and sync**

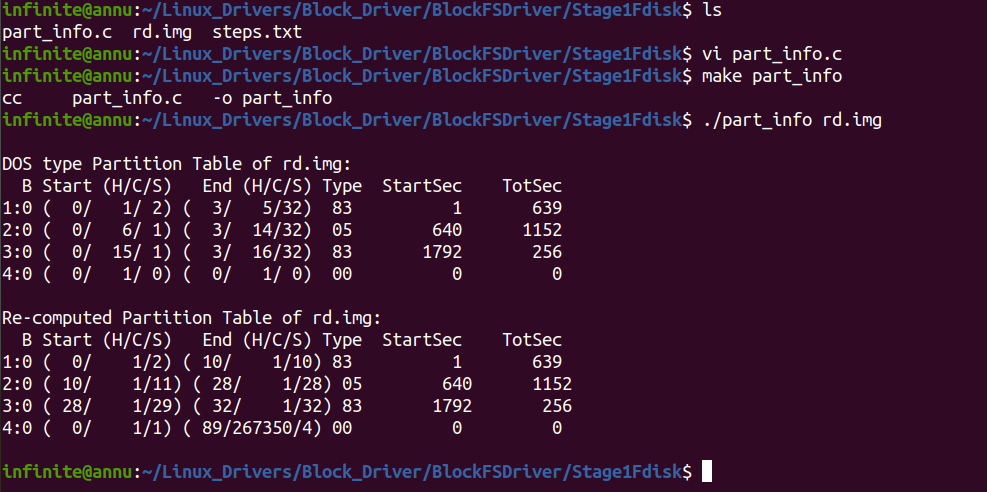
****

How to check h,s,c wise like Linux?

Ans -:

make part\_info

**./part\_info rd.img**

****

we need to creat device file to support like

we dont have here seprate for rd.img

ls /*dev /* rd.img1

ls /*dev /* rd.img2

ls /*dev /* rd.img3

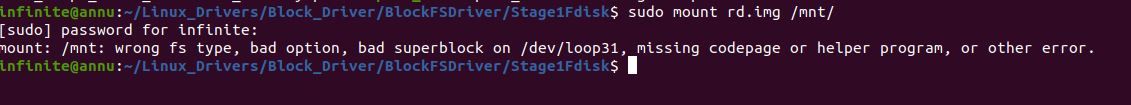
ls /*dev /* rd.img4

fdisk able to read rd.img

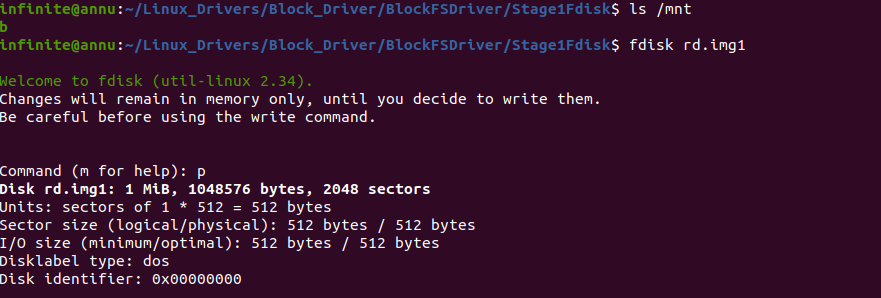
Not easy to creat file system. Like to store file in that partition we not able to mount.

fdisk is able to read it but filesystem not able to use it

We wants to use a driver so that this driver can show saperate device file to us so we are able to mount and use it .



Why my partition gone?



Check file History

**0002\_Steps\_Why\_Partition\_Gone\_After\_fdisk.patch**

Partition read it fdisk but when I do mkfs i wants something like rd1.img.

So i need saperate file so i can create file syatem. The moment i created

We need basically rd1.img , rd2.img like that. On which i creat file syatem. But it was only one partition

File system create data structure data structure and inode table and all info.

Whne i do mkfs so it lost mbr and data structure.

**So mkfs overwrote evrything.**

**Need -:**

**support for block driver which create one saperate create device file on which i can create file system. So with the help of that we can store the data into partition. Thats why Block IO layer comes into picture.**

**How we can creat rmdisk ;**

perticular portion of RAM will creat as disk. We will operate on it. And block driver we called block driver of ramdisk.

**# 1Mib (disk) – Ramdisk Block Driver**

**Excersise -: End Goal**

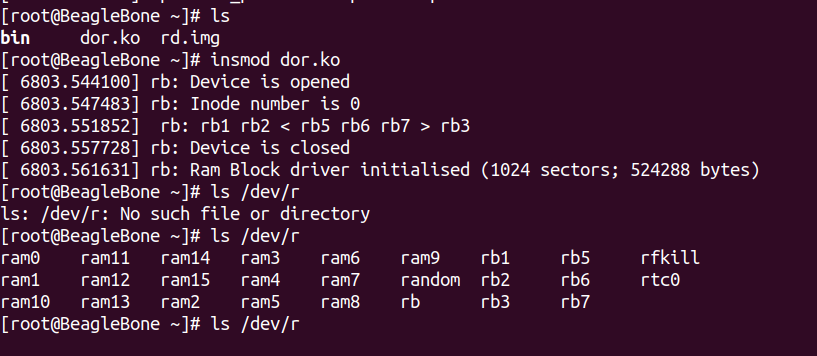
**===============**

**Block\_Driver/BlockFSDriver/Stage1Fdisk**

file – dor.ko

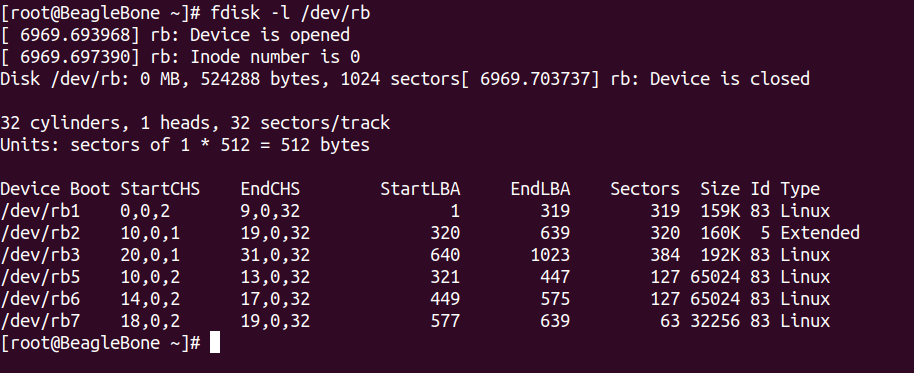
file -: rd.img

Transfer into board

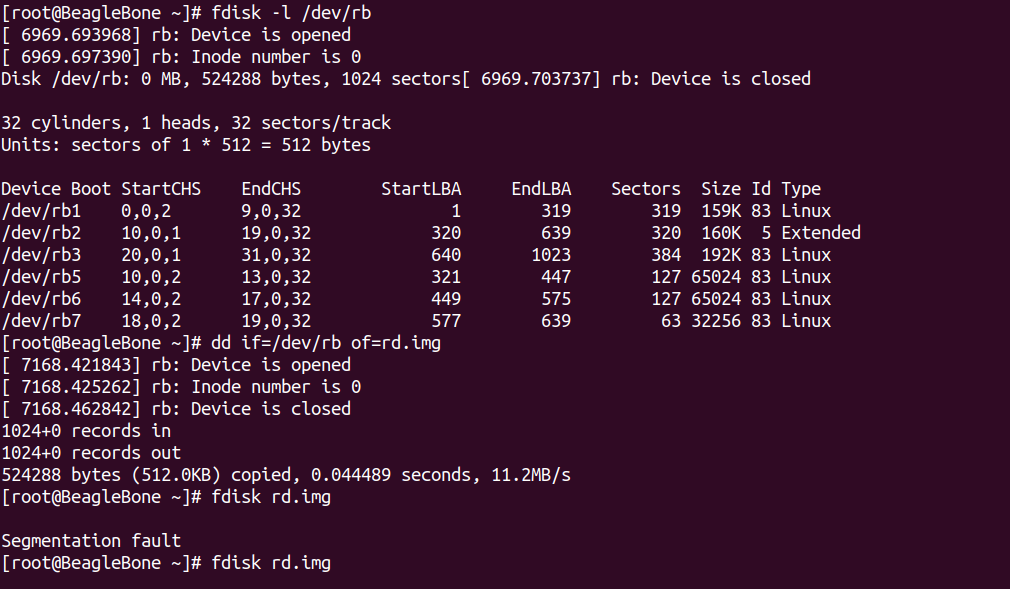


We wants to achieve this type of driver at the end.

Partttion on board -:

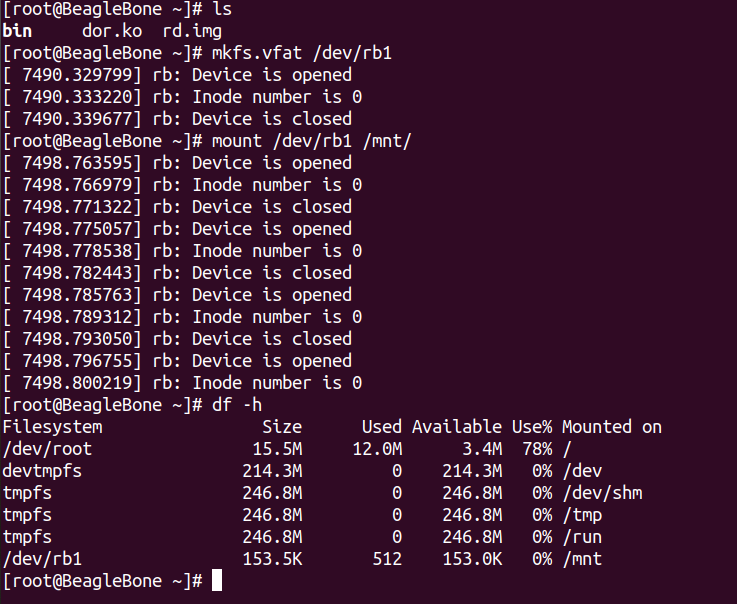


**320** its end but why its starting from 321 -: store LBR

****

**mkfs.vfat /*dev/*rb1**

**mount /*dev/*rb1 /dev**

****

#disk on ram -> loaded driver -> disk created on the ram -> unloaded -> ram was released

#loaded – malloc(512 KB) ------>> read/write on ram

path -: BlockFSDriver/Stage2Ramdisk

apis.txt

Makefile

partition.c -: partition layout

partition.h

ram\_block.c -: Block driver (Verticals)

ram\_device.c -:

ram\_device.h steps.txt

files -:

ram\_block.c

Todo 1 -:

in char driver

char = regter char driver + register file operation + create device file

block = allocate gendisk + add disk

Having device file without file opertaion no need

block = **allocate gendisk+ rgister fops + add disk**

file -: **include/linux/blkdev.h**

struct block\_device\_operations {

int (\*open) (struct block\_device \*, fmode\_t);

void (\*release) (struct gendisk \*, fmode\_t);

int (\*rw\_page)(struct block\_device \*, sector\_t, struct page \*, unsigned int);

int (\*ioctl) (struct block\_device \*, fmode\_t, unsigned, unsigned long);

int (\*compat\_ioctl) (struct block\_device \*, fmode\_t, unsigned, unsigned long);

unsigned int (\*check\_events) (struct gendisk \*disk,

unsigned int clearing);

/\* ->media\_changed() is DEPRECATED, use ->check\_events() instead \*/

int (\*media\_changed) (struct gendisk \*);

void (\*unlock\_native\_capacity) (struct gendisk \*);

int (\*revalidate\_disk) (struct gendisk \*);

int (\*getgeo)(struct block\_device \*, struct hd\_geometry \*);

/\* this callback is with swap\_lock and sometimes page table lock held \*/

void (\*swap\_slot\_free\_notify) (struct block\_device \*, unsigned long);

struct module \*owner;

const struct pr\_ops \*pr\_ops;

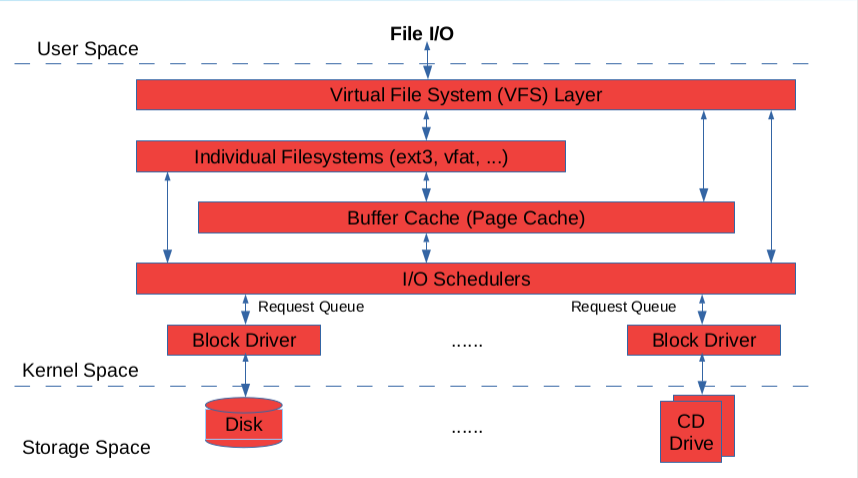
};

In files there are not read or write operation.

From userspace we are not exposing read/write system call.

Why?

**Bock input /output**



input /output

VFS = system driver +block driver

VFS knows which file sytem driver works

when we do read/ write operation it will store in some memory (Buffer cache) for frequesntly data.

When unmount or save file. Like all operation flush on the disk. So Buffer cache send request to Disk. So there are some request.

In char driver like direct read or write.

So i need a queue , so the block driver has queue. So block driver serve keep on processing request que.

bloack driver doesnt have directly interaction with disk(userspace). No direct dealing with read/write operation. W.r.t char driver buffer cache + additional layer not be there

Lets Assume -> request comes

# sector 2 to sector N

# Sector 10---->>> Sector 5

So 1st serve request no 10 and then request Sector no 5 -: This is inproper

Another senariao

Sector 10 -->> Sector 9-->> Sector 7 -->>Sector 22 -->>Sector 4-->>Sector 8 -->>

So we need like elevator . Select one which you wants.

So that layer we called based on the request come store in I/O Schedulers.

So

**block = allocate gendisk + rgister fops + Intialize Queue + add disk**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***