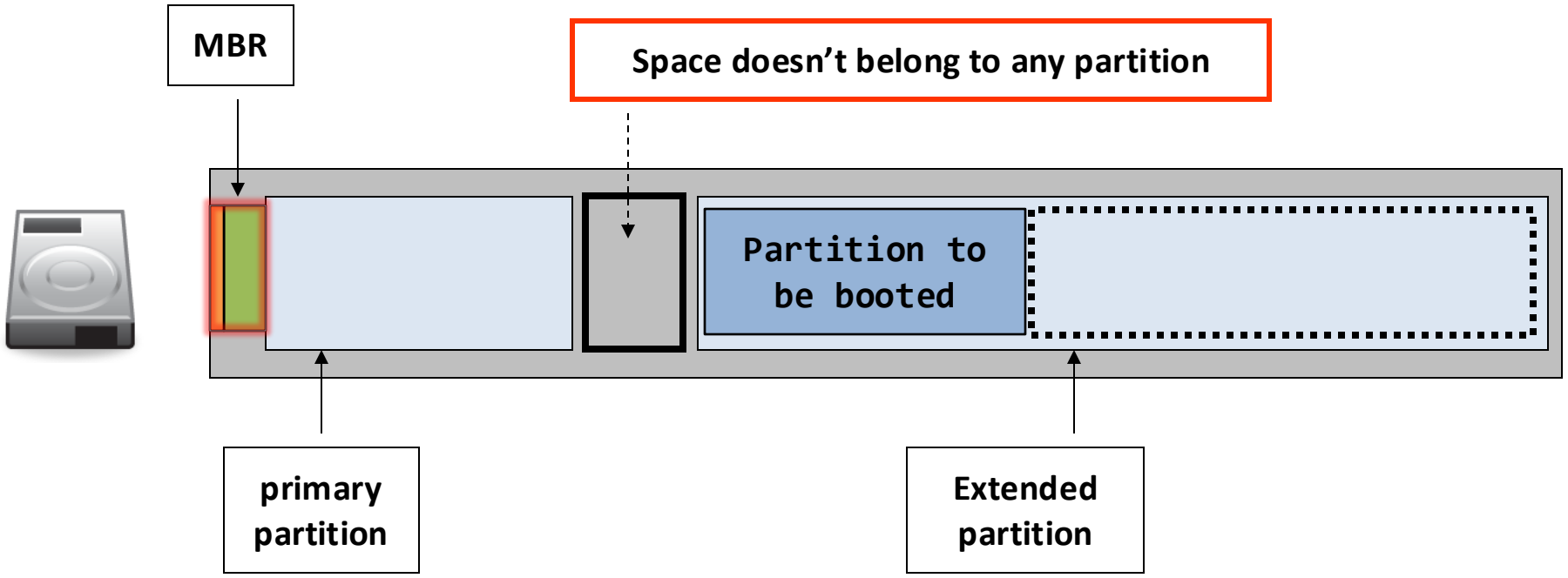
Operating Systems

Eric Lo

11 Disk and Booting

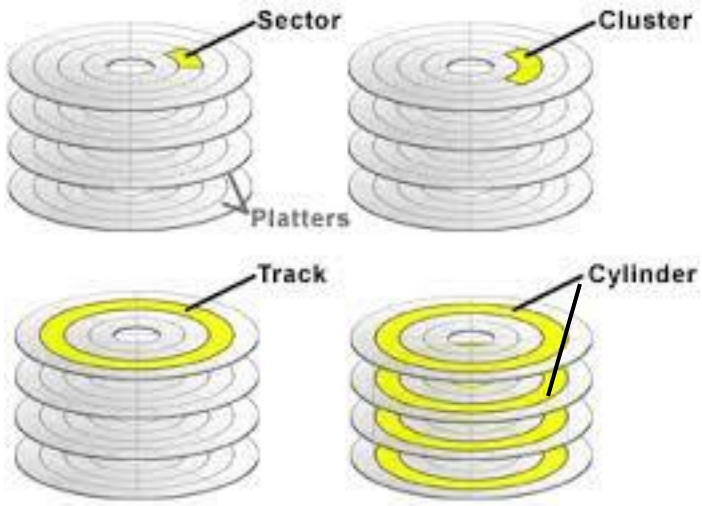
Disk Partitions

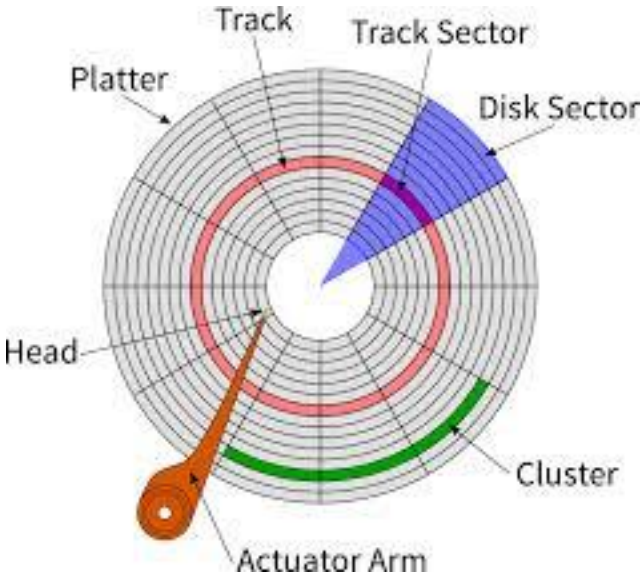


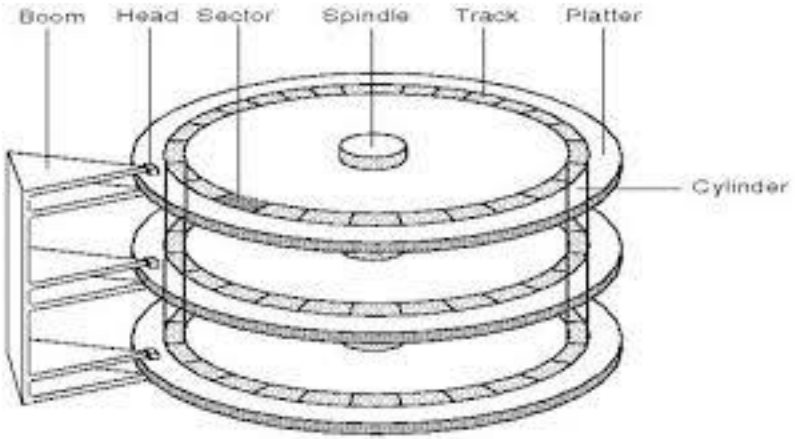
Disk Partitions

- Why do we need to have partitions?
  - **Multi-booting**
    - You can have a Windows XP + Linux + Mac installed on a single hard disk.
  - **Data management**
    - You can have one logical drive to store movies, one logical drive to store the OS-related files, etc.
  - **Backup and Maintenance**
    - Partitions are independent and can support different file systems (crash of one unlikely hurts the others)

HDD





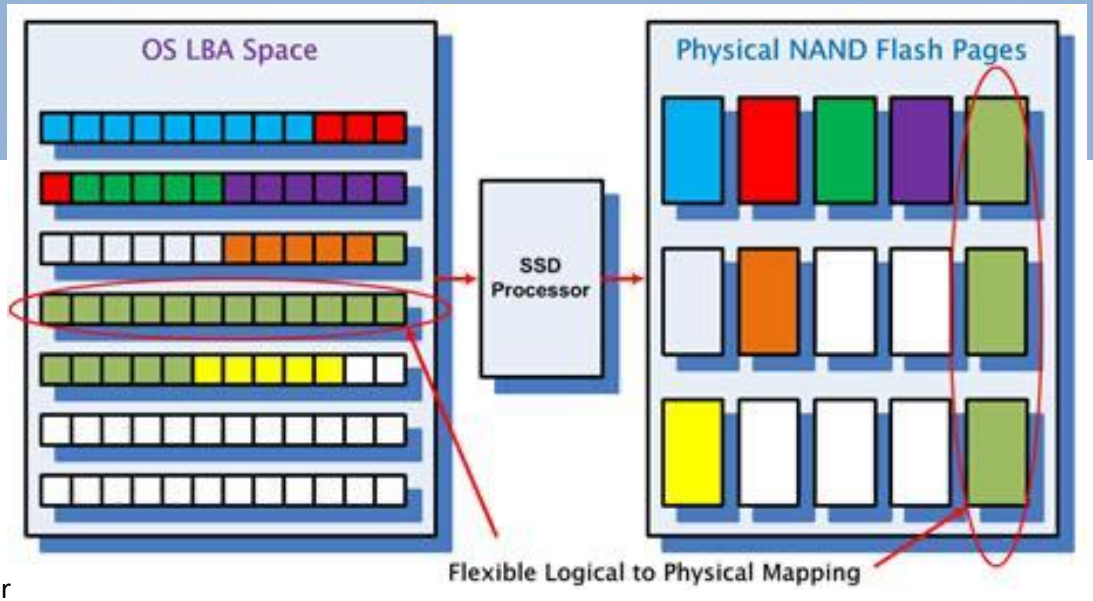


Some good movies

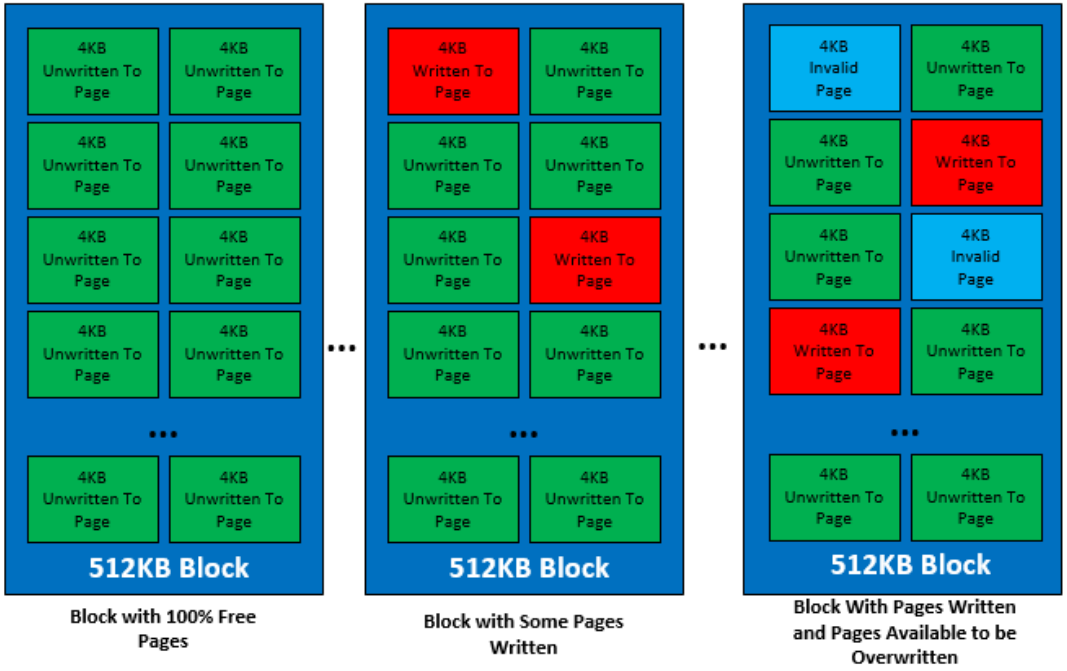
http://www.youtube.com/watch?v=9eMWG3fwiEU

http://www.youtube.com/watch?v=L0nbo1VOF4M

SSD



Ref: ELinfor



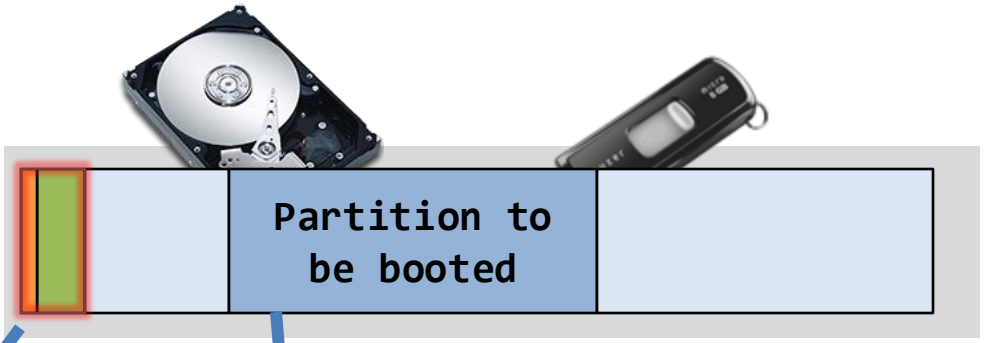
Ref: The IT Hollow

Booting

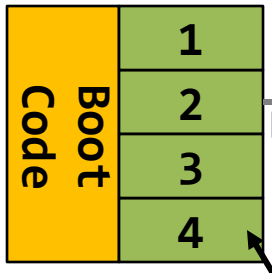
BIOS (Basic Input/Output System)*

EPROM (erasable programmable read-only memory)

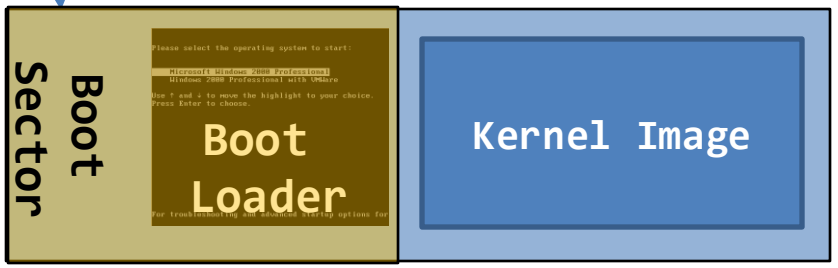
Step 1. BIOS locates the ***first*** bootable device



Step 2.
Executes its **boot code**



Locate the bootable partition



Partition table

First sector (512 bytes)
Master Boot Record

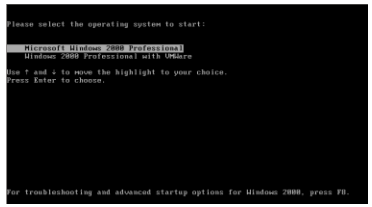
Step 3.
Boot code executes the fatter boot loader

UEFI has replaced BIOS:
https://www.howtogeek.com/56958/htg-explains-how-uefi-will-replace-the-bios/

MBR & Boot loader

- **Master boot record** (MBR) stores two things:
  - Boot code; and
  - Partition table.

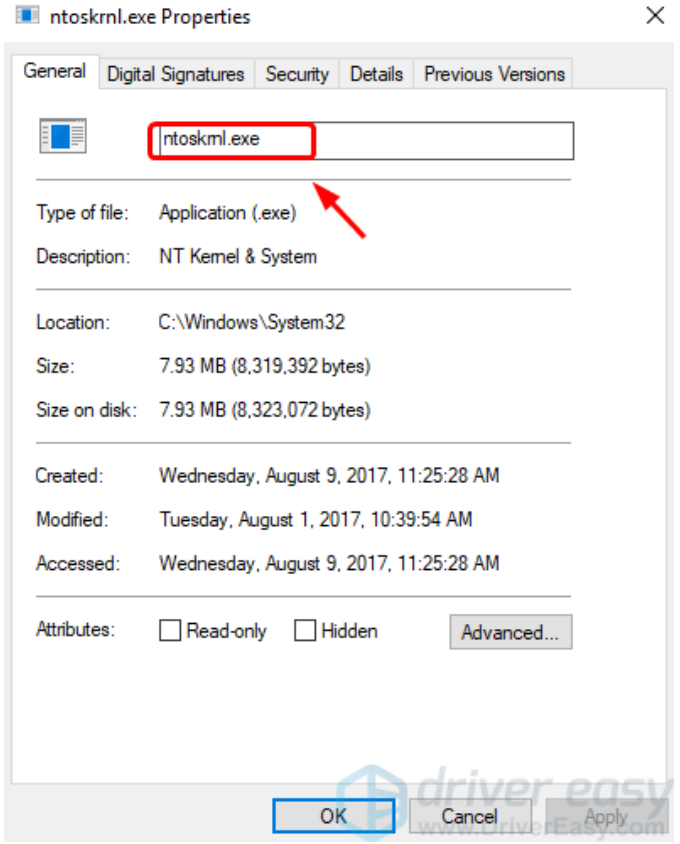
- The job of the boot code (program) is to execute a boot loader in a *bootable partition*.
  - Linux: **GRUB – GRand Unified Bootloader**;
  - Windows: **C:\boot.ini**.

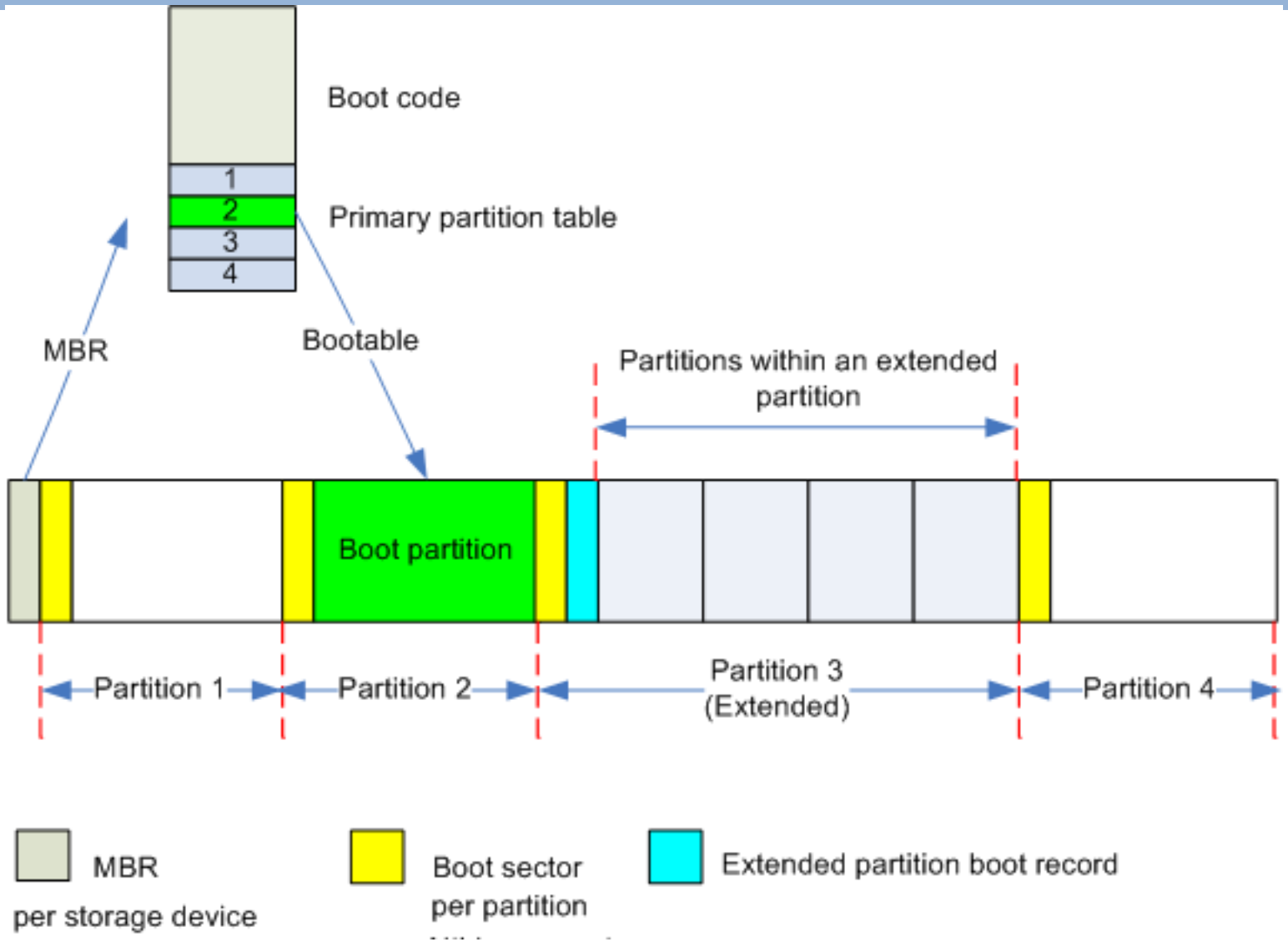


- The job of the boot loader is to locate **one kernel image** and **boot (bring it to memory & execute) it**.

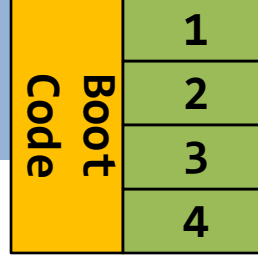
MBR & Boot loader

- The kernel image is just a **file**
  - Linux: **`/boot/vmlinuz`**;
    - Compressed
      - including a self-decompressor
  - Win 11:
    - C:\Windows\System32\ntoskrnl.exe
  - When the kernel image is found, the kernel starts.
  - It initializes all kernel subsystems.
    - E.g., initialize memory layout, initialize drivers, etc.

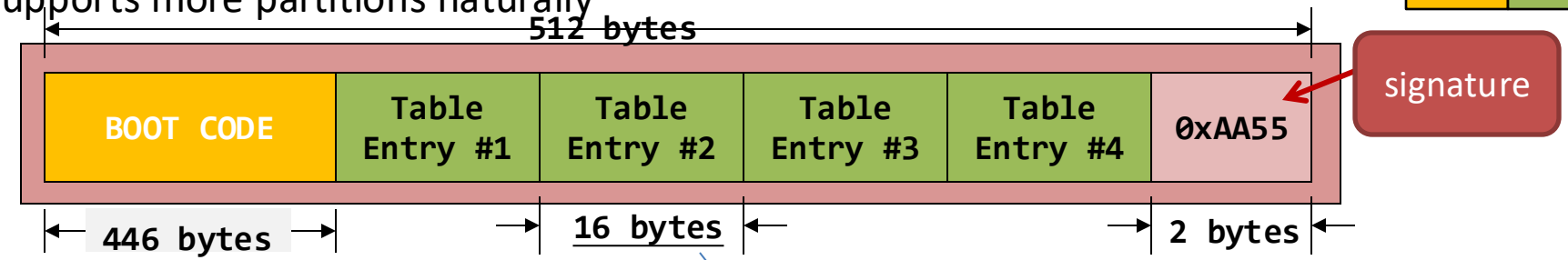



Boot code
1
2
3
4
Primary partition table
MBR
Bootable
Partitions within an extended partition
Boot partition
Partition 1
Partition 2
Partition 3 (Extended)
Partition 4
MBR per storage device
Boot sector per partition
Extended partition boot record

MBR in great detail



Nowadays, MBR is replaced by GUID Partition Table (GPT) that supports more partitions naturally



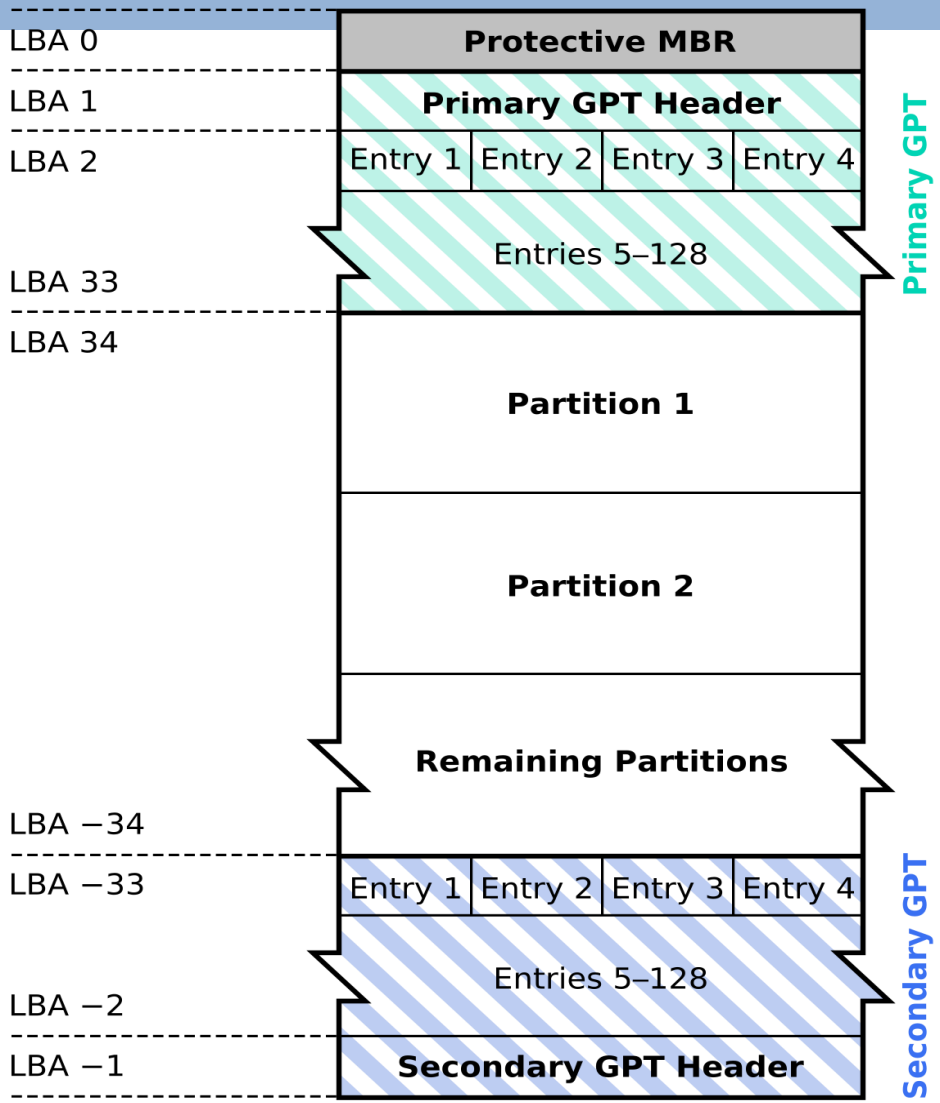
The range of a partition of is described by the:
(offset, length) tuple.

Partition Table Entry* (16 bytes)	
Bytes	**Description**
0-0	Bootable flag; 0x80 means bootable.
1-3	Starting Cylinder-Head-Sector address
4-4	Partition type http://www.datarecovery.com/hexcodes.asp
5-7	Ending CHS address
8-11	Starting Logical Block Address
12-15	Sizes in sectors

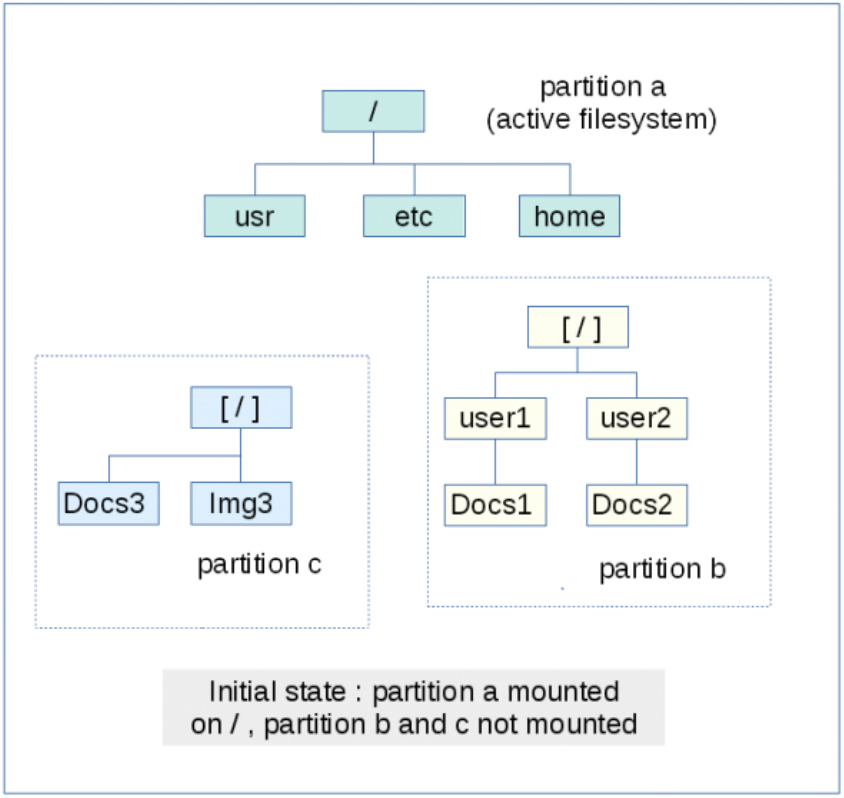
```
[examples@3150] sudo ./read_part /dev/sda2
```

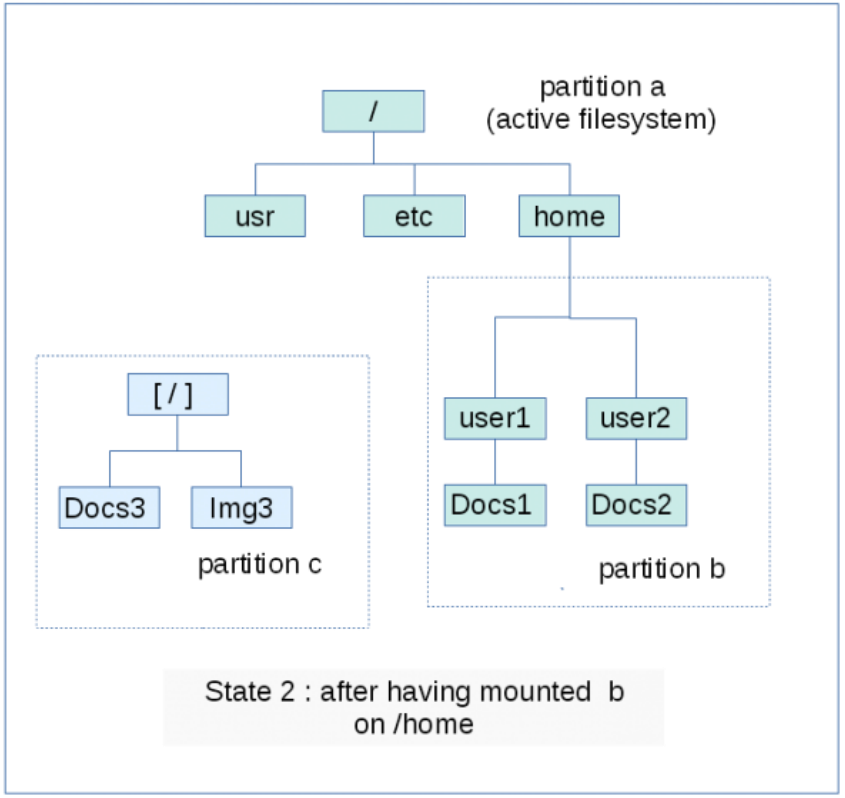
Linux command: sudo fdisk -l ; lsblk

GUID Partition Table Scheme



Mounting





https://en.opensuse.org/SDB:Basics_of_partitions,_filesystems,_mount_points