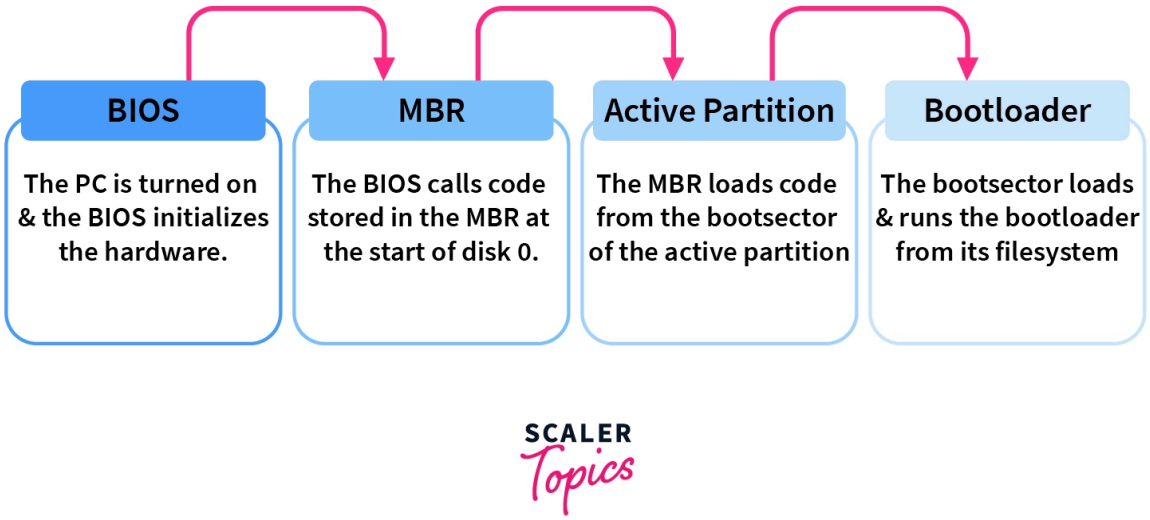
2. Booting

windows Booting process

1. **Cool booting :** Bios > MBR > active partition > BCD
2. **warn booting :** active partition > BCD



1. BIOS (Basic input/output System) :

BIOS is a firmware embedded on computer motherboard that perform essential task during booting process.

1. **Power-on-self-test (POST) :** when you turn on you pc run a series of check (POST) to insure all part like CPU, ram, SSD working properly.
2. **Boot device selection :** After POST BIOS check for bootable device USB or HDD for based on boot order setting in BIOS.
3. Finally, BIOS hand over the control to boot loader.

Feature	Legacy BIOS	UEFI
Drive Support	Up to 2.2 TB (MBR)	Over 2.2 TB (GPT)
User Interface	Text-based	Graphical and mouse support
Boot Speed	Slower boot times	Faster boot times
Security Features	no secure boot	Secure Boot, enhanced security

Feature	Legacy BIOS	UEFI
Storage	ONLY MBR	GPT + MBR

MBR (Master Boot Record) and GPT (GUID Partition Table):

Feature	MBR	GPT
Partition Limit	Up to 4 primary partitions	Up to 128 partitions (Windows)
Maximum Drive Size	2.2 TB	9.4 ZB (zettabytes)
Boot Mode	BIOS/Legacy	UEFI
Partition Table Size	512 bytes	128 bytes (per entry)
Redundancy	No redundancy	Backed up at the end of the disk
Checksums	None	Uses CRC32 checksums for integrity
Compatibility	Older systems	Modern systems (UEFI)
Dynamic Disks	Limited support	Supports dynamic disks and features

2. MBR (Master Boot Record)

The MBR is a special type of boot sector that is located at the very beginning of a storage device (like a hard drive or SSD). It plays a crucial role in the booting process of systems using BIOS firmware.

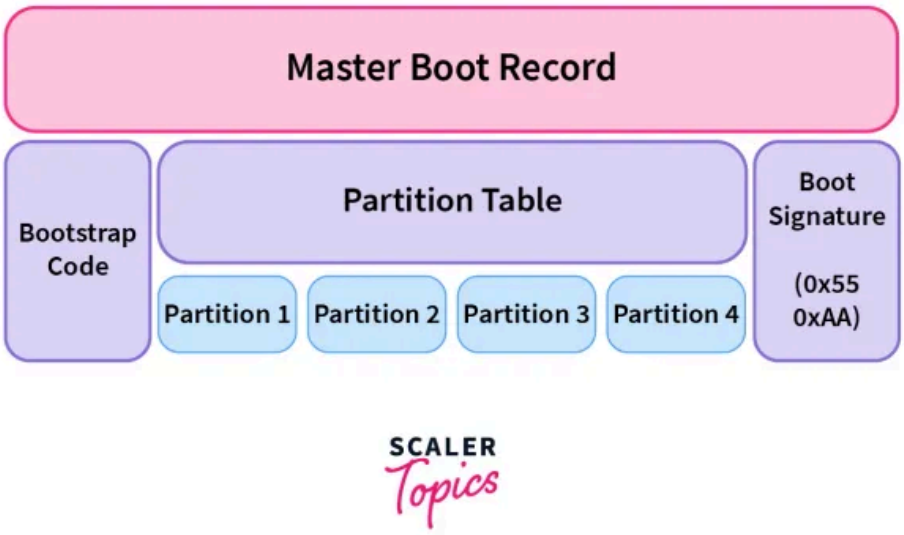
Structure of MBR

The MBR has a fixed size of **512 bytes** and consists of several key components:

1. **Boot Loader Code**:
    - The first **446 bytes** of the MBR contain the boot loader code. This is a small program that the BIOS executes after loading the MBR. Its primary purpose is to locate the active partition and load its boot sector into memory.
2. **Partition Table**:
    - The next **64 bytes** contain the partition table, which describes the partitions on the disk. The table can hold information for up to **four primary partitions**:

- Each entry includes details such as the starting sector, size of the partition, and the partition type (e.g., NTFS, FAT32).

3. **Boot Signature**:
   - The last **2 bytes** of the MBR are a boot signature, which is typically set to **0×55AA**. This signature indicates that the MBR is valid. If the signature is not present, the BIOS will not recognize the disk as bootable.



3. Active partition

The active partition is the specific partition on a hard drive that the system will attempt to boot from. It contains the necessary files (such as the boot sector) that the operating system requires to start up.

- Only one partition on a drive can be marked as active at a time. This is done to inform the boot process which partition the system should load.
- The active partition is determined from the partition table located in the MBR. When the BIOS loads the MBR, it checks for the active flag on the partitions listed.
- The active partition contains a boot sector, which is a small section of the partition that holds the boot loader code specific to the operating system. This code is executed after the MBR's boot loader code transfers control to it.

4. BCD

BCD (Boot Configuration Data) is a critical component in the boot process for Windows operating systems. It serves as a database that contains important configuration

information for the boot loader and is essential for the successful startup of Windows.

- BCD data is typically stored in a file called `BCD` located in the EFI system partition (for UEFI systems) or in the `Boot` folder of the active partition (for BIOS/Legacy systems).
- . It allows the system to manage multiple operating systems, specify boot options, and provides a structured way to configure boot settings.

The booting processes for Windows and Linux are fundamentally different, though they share some common concepts. Here's a comparison of their boot processes:

Windows Booting Process summary

1. **BIOS/UEFI Initialization**:
    - Similar to Linux, the computer starts with the BIOS or UEFI performing hardware checks (POST).
2. **MBR or GPT**:
    - BIOS reads the MBR from a selected bootable device, while UEFI reads the EFI System Partition.
3. **Active Partition**:
    - In MBR systems, the BIOS looks for the active partition to load.
4. **Boot Configuration Data (BCD)**:
    - The Windows Boot Manager (bootmgr) reads the BCD file to determine which operating system to boot and any specific boot parameters.
5. **OS Loader**:
    - The Windows OS loader (winload.exe) is executed, which loads the Windows kernel and system files.
6. **Kernel Initialization**:
    - The kernel initializes system services and drivers, and the user interface is presented.

Linux Booting Process

1. **BIOS/UEFI Initialization**:
    - Similar to Windows, it starts with BIOS/UEFI performing hardware checks.
2. **MBR or GPT**:

- The BIOS reads the MBR, or UEFI reads the EFI System Partition, just like in Windows.

3. **Boot Loader**:
   - Instead of BCD, Linux typically uses a boot loader such as GRUB (Grand Unified Bootloader) or LILO (Linux Loader). The boot loader presents a menu of installed operating systems.
4. **Loading the Kernel**:
   - The boot loader loads the Linux kernel (e.g., `vmlinuz`) and the initial RAM disk (initrd or initramfs) into memory.
5. **Kernel Initialization**:
   - The kernel initializes hardware and mounts the root filesystem, running the init process (e.g., `systemd`, `init`).
6. **User Space**:
   - The system moves into user space, launching system services and presenting the user interface.

Key Differences

- **Boot Configuration**:
  - Windows uses BCD to manage boot configurations, while Linux typically uses boot loaders like GRUB, which can manage multiple OS installations and provide a menu at startup.
- **Kernel Loading**:
  - Windows uses `winload.exe` to load the OS, while Linux directly loads the kernel file and associated initramfs.
- **User Space Initialization**:
  - The processes that follow kernel loading differ significantly in terms of system initialization and service management.