1. BIOS/UEFI

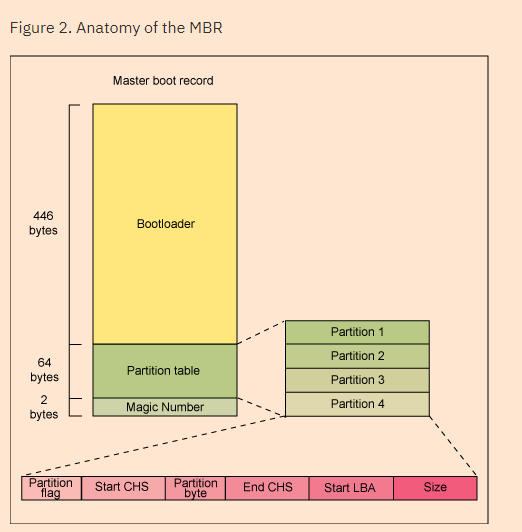
* effectue le POST (Power On Self Test : test et initialisation des composants)

UEFI remplaçant du BIOS mais exige GPT.

* Cherche un périphérique bootable
* Passe la main au premier secteur du périphérique bootable, le MBR.

1. MBR /GPT

* Démarre au premier secteur et fait 512 octets
  + Le MBR : master boot record divisé en 3 parties (Cf. image)
    - 4 Partitions primaires maxi
    - 2.2 To maxi pour une partition
    - Impossible de démarrer un MBR avec un système UEFI
  + Ou le GPT : GUID partition table



1ère partie : GRUB typiquement.

2ème partie : Table des partitions sur ce périphérique

3ème partie : Check de validation du MBR (erreurs)

* Passe la main au bootloader

1. Boot Loader

* GRUB/LILO va chercher dans la table des partitions une partition bootable
  + Charge le kernel (image kernel **/boot/vmlinuz**) en RAM
  + Extrait initrd/initramfs en RAM  via **/boot/initramfs** : FS temporaire qui va permettre de monter le vrai FS : Il doit être monté très tôt dans le processus de boot afin d’exécuter des programme dans l’espace utilisateur (et non kernel) avant le montage du vrai FS.

Exemple : Le FS se situe le réseau et demande une authentification qui demande un service user.

* **/boot/system.map** : symboles du noyau
* Passe la main au kernel

1. Kernel et initrd/initramfs

* Kernel :
  + Gère la mémoire, les périphériques I/O (early boot process)
  + Utilise initramFS pour :
    - Monter 1 root filesystem au format CPIO qui servira à monter le vrai FS
    - Obtient une @IP via DHCP
    - Drivers pour le montage du FS principal
    - Lancer un mini-Shell : le dash
    - Klibc et kinit
  + Charge le premier processus Init/Systemd

1. Init/Systemd

* PID 1 ne possède pas de père et père de tous les processus
* Amène le système à un .target défini (run-level) en lançant tous les services nécessaire