

Le disque dur

Principales notions utiles pour gérer de manière avertie ses partitions lors de l'installation d'un système d'exploitation Gnu/Linux sur son ordinateur.

Définition [wikipedia]

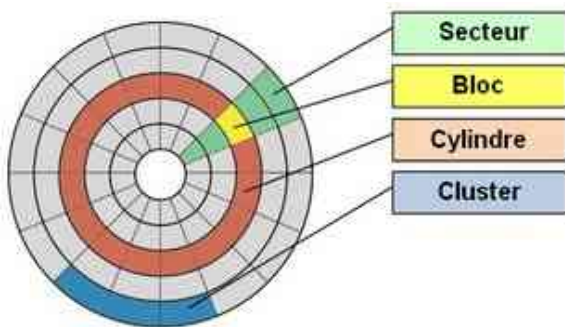
DD

Un **disque dur**, parfois abrégé **DD**, en anglais **HD** ou **HDD**, est une [mémoire de masse](#) à disque tournant [magnétique](#) utilisée principalement dans les [ordinateurs](#), mais également dans des [baladeurs numériques](#), des [caméscopes](#), des [lecteurs/enregistreurs de DVD](#) de salon, des [consoles de jeux vidéo](#),

SSD

Un **solid-state drive**, abrégé en **SSD**, appelé parfois **disque SSD** en [France](#) ou **disque électronique** ou **disque à semi-conducteur**, au [Québec](#), est un [matériel informatique](#) permettant le [stockage de données](#) sur de la [mémoire flash](#). (clé usb par exemple)

Blocs et Secteurs (facultatif)



Je ne vais pas rentrer dans les détails. Retenez juste que le disque est considéré comme un espace linéaire constitué de blocs de données, appelés secteurs, de 512 octets chacun.

Démarrage d'un pc :

Quand vous pressez le bouton pour allumer votre pc , c'est la carte mère qui démarre en premier .

Elle va détecter tout le matériel présent et notamment les disques durs.

Une puce (bios ou UEFI) va aller chercher des infos dans le premier disque dur (par défaut celui qui est branché sur le port 0 de la carte mère).

Cette puce va lire le disque en commençant par le début du disque c'est-à-dire le MBR (voir plus bas) les informations trouvées vont définir ce qui va être démarré (ou pas).

Notions de base

Unités de mesures (facultatif)

Le bit est la plus petite unité. Physiquement c'est une matière qui se charge électriquement en positif ou négatif. On utilise généralement 0 ou 1 (ou encore vrais ou faux), c'est donc une écriture binaire. (mathématiquement on y compte donc en base 2)

0 = 0	1 = 1	2 = 10	3 = 11	
4 = 100	5 = 101	6 = 110	7 = 111	
8 = 1000	9 = 1001	10 = 1010		255 = 11111111 (8bit = 1 octet)

Les caractères sont donc codés en binaire.

Il existe différents type de codage (ASCII, Unicode, UTF-8, ...etc...).

Par exemple en UTF-8, le caractère A s'écrit 01000001 .

Dans le cas de l'UTF-8, un octet = 1 caractère. Ce qui n'est pas le cas en Unicode par exemple !!

l'octet	Le byte
c'est une unité de mesure qui vaut exactement 8bits	c'est une unité d'adressage donc la taille peut varier, même si normalisée à 8bits depuis quelque temps.
Unité française	Unité anglaise

1 Ko (kilo-octet) = 1000 octets c'est 10^3 :on compte en base 10 (système décimal)

1 Kio (kibi-octet) = 1024 octets c'est 2^{10} :on compte en base 2 (système binaire)

La confusion est possible car :

linux compte et affiche en base 2. (en kibi)

windows compte en base 2 et affiche en base 10. (mixte) **confusion !!**

mac (à partir d'osX) compte et affiche en base 10. (en kilo)

Partitionnement (important)

Le partitionnement est simplement le fait de fractionner son disque dur en plusieurs parties.

sda désigne le disque dur branché sur le port 0 de la carte mère (sdb le suivant, sdc ...etc...)

sda1 sera la première partition de sda . Sda2 la suivante, sda3 ...etc...

Exemple de partitionnement pour un disque (sda) de 300Go en msdos (limité à 4 partitions primaires)

-sda1 (windows 8) 100 Go (Ntfs)

-sda2 (linux swap) 8Go (Swap)

-sda3 partition étendue 192 Go (un conteneur de partitions secondaires)

-sda4 (système linux « / ») 50 Go (Ext4)

-sda5 (documents linux « /home ») 100 Go (Ext4)

-partition sauvegarde 42 Go (Ntfs)

La table de partition (très important)

Une table de partition indique où commence et où finit chaque partition. Sans elle, les partitions ne seraient pas accessibles.

Bios, partitions msdos et MBR:

Avec un pc muni de BIOS, lors de la création de la table de partition, on utilise généralement le type « msdos ». Créer une nouvelle table supprime toutes les anciennes partitions de ce disque. (Dans ce nouvel espace vierge, vous pourrez y créer vos différentes partitions.)

Cette table de partition est contenue dans le mbr.

Le MBR (mode bios/msdos) en detail :

Le **master boot record** ou **MBR** (parfois aussi appelé « **zone amorce** ») est le nom donné au premier secteur adressable d'un disque dur (ou d'une clé usb ou d'un autre média amovible.)

Sa taille est de 512 octets (0 à 511). Il contient 3 choses :

- 446octet (0 à 445) le chargeur d'amorçage (*boot loader*), c'est une routine d'amorçage dont le but est de charger le système d'exploitation
- 64octet (446 à 509) la table de partitionnement
- 2octet (510 à 511) une signature

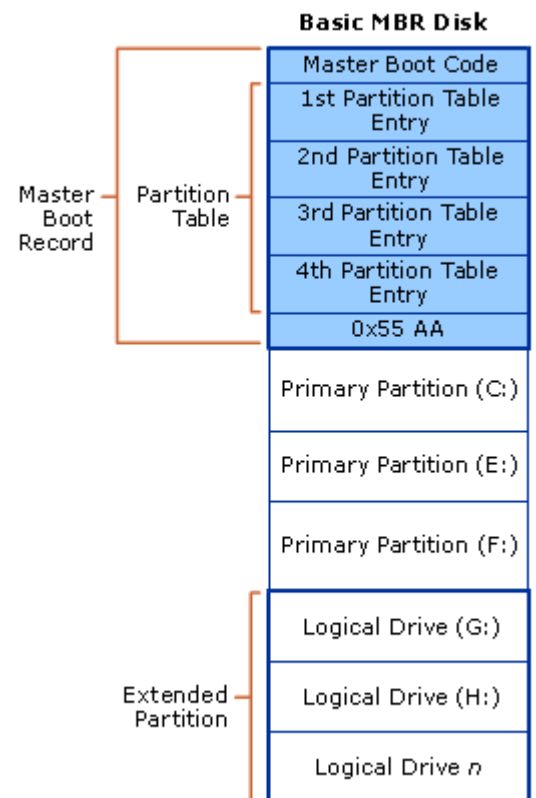
Ce fonctionnement à quelques limitations :

- 4 partitions primaires maximum (après on utilise les partitions secondaires appelées aussi logiques)
- taille des partitions limitée à 2,2 To

Pour créer des partitions secondaires, on va créer une partition étendue, c'est une partition primaire qui va servir de conteneur pour les partitions secondaires. (une seconde table de partition secondaire est créée dans l'EBR de la partition étendue. (l'EBR est une sorte de MBR secondaire juste pour cette partition))

Cette manipulation permet de s'affranchir de la limite de 4 partitions du Bios. (fdisk en gère jusqu'à 60)

Ces limites ont poussé les fabricants à se tourner vers une nouvelle technologie : l'UEFI (remplaçant du Bios) et sa table GPT (remplaçant de msdos).



UEFI et partitions gpt :

Si votre pc est doté d'un UEFI (remplaçant du bios), celui-ci exige une table de type « GPT » .

avantages :

- prise en charge des tailles de partitions > **2.2 To**
- Capacité de 128 partitions par défaut
- Table de partitions en 2 exemplaires: une au début du disque, une à la fin. (plus robuste)
- Toutes les partitions sont de type primaires.

Le MBR existe toujours mais n'a plus la même fonction. C'est devenu une structure qui protège les disques GPT des écritures provenant d'*utilitaires disques* qui ne (re)connaissent pas les informations de GPT. C'est la raison qui empêche un BIOS « classic » d'utiliser les disques en GPT.

Le GPT en détails :

GPT : signifie **T**able de **P**artitionnement **G**UID, et désigne l'ensemble formé de l'*entête* et du *tableau des partitions*

Structure globale du disque :

- MBR protecteur (début du disque dur)
- GPT primaire : entête
- GPT primaire : tableau de partitions
 - Partitions : début de la partition 1
 - ...
 - Partitions : fin de la partition x
- GPT secondaire : tableau de partitions
- GPT secondaire : entête (fin du disque dur)

Un pc doté d'un UEFI peut (toujours ?) permettre d'utiliser un disque en msdos en choisissant un mode « legacy » ou « csm » dans la configuration de l'UEFI

attention : les constructeurs continuent souvent à utiliser le nom « bios » pour qualifier l'UEFI !!!

Systeme de fichier (important)

Ntfs , Fat32, Ext4, c'est quoi ?

C'est une manière (un code, une langue) pour organiser l'espace d'une partition pour gérer au mieux nos fichiers.

Lorsque l'on crée une partition, il est nous est obligatoirement demandé de choisir un « système de fichier ». Ce tableau vous aidera à prendre une décision correcte :

format	avantage	inconvenient	usage
ntfs	Pour windows	Pas pour linux	Partition windows
Ext4	Peu de fragmentation	Partition non accessible depuis windows sans logiciel approprié	Partition linux
fat32	Reconnu partout fragmentation	Taille des fichier limité à 4Go (voire 2Go)	Clé usb
linux-swap			Extension de mémoire vive (pous linux) Sa taille est généralement égale à la quantité de ram installée sur son pc.

D'autre format existe comme le btrfs , le reiserfs , le jfs, le xfs , ...etc ... ils sont moins utilisés mais peuvent être de bons choix (pas toujours). Certains sont robustes, d'autres rapides, ou sécurisés, vieux ou trop récents .. Dans un souci de simplicité de ce document, je ne les aborderai pas. Google Qwant est votre ami ;-)

ps : ce tableau est volontairement simpliste (voire caricatural), c'est juste une première approche !!

Précision :

Le terme « formater » n'est pas le fait d'effacer les données, c'est plutôt changer la « langue » à utiliser et réinitialiser l'indexation des données. Ce qui rend les données introuvables par le nouveau format. Mais en réalité même si les données ne sont plus accessibles (visibles), elles sont toujours bel et bien écrites sur le disque. Ce qui permet d'envisager de récupérer d'anciennes données.

Pour réellement effacer des données, il est nécessaire de réécrire (des 0 ou des 1) par-dessus, ce qui peut être très long. (idéalement plusieurs fois de suite pour rendre quasi-impossible toute récupération).

Lorsque l'on jette un disque dur à la poubelle, on ne sait pas où il va, je vous conseille **VIVEMENT** d'effacer réellement vos données avant !!!! ou mieux encore d'y percer un trou avec une perceuse !!! Ou alors des gros coups de marteau :-)

(Voir en Annexe les commandes linux)

Il existe des programmes pour effacer les données je n'ai plus le nom en tête. Un truc du genre easeser

le chargeur d'amorçage

Nous voilà dans la pratique. Nous sommes en train d'installer un OS (Linux, Windows ou autres) nous avons choisi de gérer nos partitions manuellement afin de choisir exactement ce que l'on veut. Et une question des plus importantes est où placer notre chargeur d'amorçage ou boot-loader (du nom de grub sous linux) ?

Exemple avec un seul disque dur : (Quiz?)

Exemple 1 :

Vous n'avez qu'un dd, et l'ordinateur ne contiendra qu'un seul OS . → choisissez sda

Bios: une routine d'amorçage sera dans le mbr et lancera votre OS

UEFI : une routine d'amorçage sera gérer par gpt (où ? Comment ?) et lancera votre OS

exemple 2 :

Vous n'avez qu'un dd, un windows est déjà présent , vous mettez linux à coté → choisissez encore sda en bios comme en UEFI le loader linux remplacera le loader windows et vous proposera un choix au démarrage

exemple 3 :

vous avez déjà Linux et vous installez Windows. Aucun choix, le loader Windows écrasera le précédent et vous n'aurez plus accès à Linux.

Solution avec Bios : sauvegardez votre mbr (la partie loader) avant, et restaurez-le après l'install de Windows. Un «sudo update-grub » sous Linux trouvera Windows et vous proposera le choix au prochain démarrage

solution avec UEFI : ? **à faire**

autre solution : créez une entrée dans le loader Windows avec easyBCD à installer sous Windows

exemple 4 :

Vous avez déjà Linux et vous installez un second Linux. La question est « par lequel de mes 2 Linux je souhaite que le démarrage soit géré ? » il est possible de ne pas installer de chargeur. Un «sudo update-grub » sous Linux le premier Linux trouvera tous les OS et proposera un choix au prochain démarrage. Un « sudo grub-install /dev/sda » sur le second Linux permettra de changer d'avis.

Exemple avec un deux disques durs : (Quiz ?)

exemple 5 :

j'ai 2 dd. Je veux un OS sur chaque disque dur.

1/ je peux, comme si je n'avais qu'un dd, avoir mon loader (de mon choix) sur sda (ou sdb).

2/ je peux avoir 2 loader un sur sda, un sur sdb. MAIS, il faudra configurer mon bios (ou UEFI) pour lancer le disque 1 ou le disque 2 par défaut au démarrage.

Par exemple, si vous avez Windows sur votre sda, et décidez d'installer Linux sur sdb. et vous souhaitez garder le mbr de sda (qui contient le loader Windows), vous pouvez alors choisir d'installer le loader sur sdb

le mbr de sda (contenant le chargeur Windows) ne reconnaîtra pas Linux et donc ne proposera pas son démarrage. (possibilité d'y créer une entrée Linux via easyBCD (logiciel Windows))

Le mbr de sdb (contenant le chargeur linux) lui, par défaut, reconnaîtra parfaitement Windows et proposera son démarrage

Autres exemples (avancé)

mais il est possible de le mettre ailleurs :

→ un autre exemple : vous préférez avoir votre chargeur de démarrage sur une clé usb

si votre clé est sdc, choisissez-la. Votre système d'exploitation ne démarrera uniquement que quand la clé usb sera branchée (et si votre bios ou uefi sont configurés pour booter cette clé)

Mais pourquoi faire ? :

pas envie que quelqu'un démarre votre pc quand vous n'êtes pas là ? Ne pas toucher à votre mbr Windows ? Pour tester, bidouiller, dupliquer votre mbr ? Un Linux par utilisateur et une clé par utilisateur ? Bref c'est un usage assez spécifique...

→ ou encore l'installer sur une partition (sda1) et chaîner les loader. Je vous laisse vous renseigner :)

sauvegarder son mbr :

pourquoi sauvegarder son mbr (inutile en UEFI ?)

- Si vous installez Linux après Windows, le loader de Linux remplacera (si vous en faites le choix) celui de Windows. Même si Linux détectera Windows et vous le proposera au démarrage, vous pourriez avoir envie de garder une copie de l'ancien mbr au cas où ...

- Si vous installez Windows après Linux, le loader de Windows remplacera celui de Linux (sans aucune autre possibilité) et ne vous permettra plus de démarrer votre linux !! la solution consiste à sauvegarder votre mbr avant et à le restaurer après. Puis un « sudo update-grub » ajoutera une entrée Windows au démarrage,

comment sauvegarder son mbr ?

https://doc.ubuntu-fr.org/tutoriel/comment_sauvegarder_le_mbr

ce qu'il faut retenir :

si votre pc est conçu avec bios → créez une table de partition en mode « msdos »

si votre pc est conçu avec UEFI → créez une table de partition en mode « GPT »

pour convertir un disque dur de l'un vers l'autre vous devrez effacer intégralement toutes les partitions et leur contenu de ce disque dur. (il est néanmoins possible d'avoir deux disques durs l'un en msdos, et l'autre en gpt si votre ordinateur le permet)

à compléter...

diverses commande de gestion de disque dur

Attention toute manipulation sur des disques durs est dangereuse. La moindre erreur peut causer des dégâts sur les données présentes !!!!

attention entre les « i », « l », « I », « L », « 0 », « o », « O » entre autre !!

ces opérations sont souvent très longues (plusieurs heures parfois plusieurs jours...)

un outil sous Linux aussi puissant que dangereux :

<https://doc.ubuntu-fr.org/dd>

tester l'état d'usure d'un dd et éliminer les blocs défectueux avec badblocks

<https://www.forum-francophone-linuxmint.fr/viewtopic.php?t=2054>

<https://www.aplu.fr/v2/post/2016/01/07/forcer-un-disque-a-reallouer-des-secteurs-defectueux>

<https://linux-actif.fr/badblocks>

deux exemples avec le logiciel DD

effacer un dd (formatage de bas niveau)

```
dd if=/dev/zero of=/dev/sda conv=notrunc
```

note :

vous aurez peut-être remarqué la similitude avec la commande pour créer une clé usb live

Linux :

```
dd if=mon_image.iso of=/dev/sdX
```

« dd if=source of=destination » est un outil extrêmement puissant et plein de possibilités...

effacer un dd (avec une sécurité gouvernementale) :

Si vous êtes inquiet au sujet d'espions avec des détecteurs supra-conducteurs à interférences quantiques, vous pouvez toujours ajouter une boucle 'for' pour effacer le disque d'un niveau de sécurité gouvernemental : copier coller les deux lignes suivantes dans un éditeur de texte :

```
#!/bin/bash
```

```
for n in `seq 7`; do dd if=/dev/urandom of=/dev/sdX bs=8b conv=notrunc; done
```

Maintenant vous avez un script shell qui exécute sept passes d'inscriptions aléatoires de caractères sur tout le disque.

Faites : `chmod a+x <fichier>` pour le rendre exécutable.

liens

https://fr.wikipedia.org/wiki/GUID_Partition_Table

<https://ubuntu-fr.org/mbr>

<https://doc.ubuntu-fr.org/gpt>

<https://lecrabeinfo.net/les-tables-de-partitionnement-mbr-et-gpt.html>

<https://www.qwant.com>

<https://www.google.fr>



rédigé par ailes-52
contact@ailes-52.org