

SOMMAIRE

Rappel fonctionnement PC	p1
Composition du dispositif de stockage	
A) Composition physique	
B) Structuration du support – localisation des données	p2
C) Evolutions – améliorations	p3
D) Structure logique	p4
1-1) Structure logique d'un disque HDD (au point de vue « machine »)	p5
1-2) Structure logique d'un disque HDD (au point de vue « utilisateur »)	p6
1-3) Structure logique d'un disque SSD (au point de vue « machine »)	p7
E) Stockage externe	p9

Stockage des données dans un PC

Avant propos

Un PC pour fonctionner utilise des données numériques. Il est nécessaire d'écrire ces données dans un dispositif de stockage si on souhaite les conserver.

Dans ce document nous donnerons les bases pour comprendre la nature et le fonctionnement de ce dispositif de stockage de données.

Rappel du fonctionnement d'un PC

Dans un PC, lorsque vous créez ou modifiez des données, elles ne sont tout d'abord chargées que dans la mémoire RAM (volatile), ceci pour accélérer le fonctionnement de l'ordinateur et fluidifier votre travail ; ces données peuvent être du texte (comme celui que vous lisez en ce moment), un tableau numérique, une photographie que vous retouchez, etc ... La mémoire RAM est volatile, cela signifie que lorsque vous éteindrez le PC, le contenu de cette mémoire temporaire se videra et qu'aucune donnée qui y est inscrite ne sera conservée.

L'ordinateur possède aussi un 2ème dispositif de stockage des données (non volatil) ; les données qui y sont inscrites seront conservées après l'extinction du PC ; par contre, ce dispositif est plus lent que la RAM pour les échanges (écriture/lecture) avec le processeur de l'ordinateur.

Toutes les données du PC sont écrites dans ce dispositif de conservation ; cela inclut le système d'exploitation (GNU/Linux, Windows, etc ...) ainsi que les logiciels et toutes vos données personnelles locales ; l'écriture de ces données s'effectue quand vous installez un logiciel ou quand vous enregistrez des modifications apportées à un texte ou une photographie par exemple.

Composition du dispositif de stockage de données

A) Composition physique

Les dispositifs de stockage local sont, physiquement, des boîtiers électroniques de diverses tailles et connectiques.

Les plus anciens sont appelés disques durs (Hard Disk en anglais) car ils contiennent 1 ou plusieurs disques métalliques tournants recouverts d'un alliage fer/cobalt (il en a existé aussi en verre).

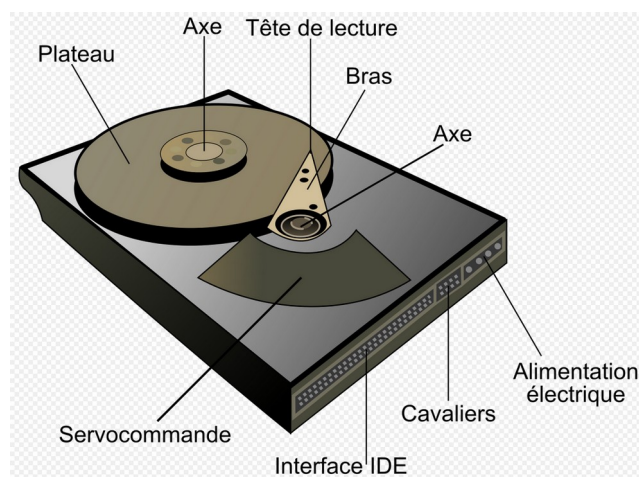
Le contrôle de la vitesse de rotation doit être très précis ; les disques les plus performants tournent à 7200 tr/mn. Leurs dimensions sont de 2,5 pouces de diamètre dans les pc portables, et 3,5 pouces dans les unités centrales (1 pouce = 25,4mm). Ils peuvent contenir de 1 à 4 plateaux pour les dispositifs grand public. L'air contenu dans ces boîtiers doit être très propre et la moindre poussière qui y entrerait détériorerait la couche de surface des plateaux, là où sont écrites les données.

Les disques durs à plateaux sont relativement lents en lecture/écriture mais peuvent être de très grande capacité.

Intérieur d'un disque dur de 3,5 pouces à 4 plateaux



Schéma des principales parties d'un disque dur



L'inscription des données se fait en orientant (haut-bas ou bas-haut) les nano-particules composant la couche de surface du disque grâce à un champ magnétique (un aimant pour faire simple) ; ces différences d'orientation seront interprétées lors de la lecture comme « 0 » ou « 1 ».

B) Structuration du support – comment l'ordinateur localise les données

Pour lire et écrire les données binaires (0 ou 1) sur les disques durs à plateaux, les concepteurs ont opté pour un système de rangement avec un repérage géométrique du disque (c'est l'adressage des données) :

- la première variable de l'adresse (appelée C) est celle du n° de Cylindre (les cylindres correspondent aux différentes positions que peut prendre la tête de lecture, chaque position correspondant à une piste ou couronne circulaire ; le n°0 => c'est celle la plus à l'extérieur) ;
- la seconde variable (H pour head) est le nombre de têtes de lecture utiles (en général une tête par face de plateau ; si le DDur a 4 plateaux, on aura 8 têtes de lecture numérotées de 0 à 7) ;
- la troisième et dernière variable de l'adresse (S) est celle du n° du secteur angulaire d'une piste (un plateau est divisé en secteurs angulaires – numérotés de 1 à 63 par exemple - à l'instar des portions d'un fromage).

Chaque paquet de données sera donc repéré par une adresse (dans l'ordre C/H/S) ; exemple : 0/0/1 sera l'adresse du paquet de données écrites sur le cylindre n° 0, lu par la tête n°0 dans le secteur n°1.

Le nombre de secteurs accessibles par ce système d'adressage sera donné par le produit :
nombre de Cylindres x nombre de têtes (H) x nombre de secteurs/cylindre (S)

exemple : 1024 cylindres, 8 têtes et 63 secteurs/ cylindre = 516096 secteurs ; à raison de 512 octets de données par secteur, on arrive à un DDur théorique de 264 241 152 octets.

Nota : il existe un procédé plus récent de repérage des secteurs du disque avec simplement un numéro d'ordre pour chaque secteur : c'est le « LBA » (Logical Block Accessing) (voir ici : <http://gecif.net/articles/linux/mbr.html>).

C) Améliorations - évolutions

Les améliorations technologiques des disques tournants ont concerné la nature de la couche de surface, la miniaturisation des têtes, la précision du contrôle de la vitesse de rotation des disques et la fréquence de lecture des données.

Les dispositifs de stockage de masse plus récents sont appelés SSD (solid-state drive = disque à semi-conducteurs), ils n'ont plus de disque tournant ; le plus grand a une taille de 2,5 pouces ; ils sont composés de microcircuits de stockage électroniques (mémoire flash à état solide comme pour les clés USB). Certains SSD d'ordinateurs portables peuvent être soudés sur la carte électronique principale du PC et ne sont donc plus remplaçables.

Stockage de masse SSD de 2,5 pouces dans un adaptateur 3,5 pouces permettant l'installation dans une baie 3,5 pouces connectique SATA)



SSD au format M2
(capacité 2Tbit)



Endroit/envers d'un SSD au format NVMe
(capacité 1 Tbit)

(connectique PCI-E)



On ne peut plus calculer aujourd'hui la capacité des dispositifs de stockage SSD en CxHxS comme on le faisait avec les disques tournants. En fait la mémoire « flash » est disposée sur le support en plusieurs modules ; la répartition et le rangement des données dans les modules sont effectués par un petit processeur intégré au support et doté d'un programme permettant la manipulation de divers paramètres. La connectique de ce nouveau support de stockage initialement de type SATA est progressivement remplacée par du PCI-E (Pci-express) plus performante.

Avec ces nouveaux dispositifs de stockage de masse, la taille physique a diminué et la capacité de stockage a fortement augmenté.

D) Structure logique

Préambule : Quelques définitions avant d'aller plus loin

« **partition** » : ou **région** ou **volume** ou parfois **disque** ; c'est une partie d'un support de stockage (disque dur, SSD, carte-mémoire...) ; le partitionnement est l'opération qui consiste à diviser le support physique en différentes zones (les partitions) dans lesquelles le système d'exploitation pourra gérer séparément les informations, généralement en y créant un ensemble de fichiers.

- Chez Windows les partitions sont désignées par des lettres (C: , D: , etc ...) ;
- Chez Mac OS elles sont désignées par *disk1s1*, *disk1s2* ... ;
- Chez Gnu/Linux elles sont désignées par *sda1*, *sda2*, *sdb1*, *sdb2*

« **Bios** » : (Basic Input Output System) c'est un microprogramme contenu dans la mémoire morte de la carte mère (carte principale) d'un ordinateur et lui permettant d'effectuer des opérations de

base lors de sa mise sous tension (identification des périphériques d'entrée/sortie connectés, lecture d'un secteur sur un disque, un CD ou une partie d'une clé USB, réglage de certains paramètres, etc ...). Sur les cartes récentes il est remplacé par sa version moderne l'**UEFI** (ou plus complètement le **Bios UEFI**).

« **Boot loader** » : ou « chargeur d'amorçage » ; c'est un petit programme (par exemple Lilo ou Grub pour Linux) dont la fonction principale est de renvoyer vers une partition où se trouve le système d'exploitation à lancer.

« **GRUB** (acronyme de « **GR**and **U**nified **B**ootloader ») est un des programmes d'amorçage Linux pour ordinateur ; il s'exécute après les séquences de contrôle interne du Bios et avant le lancement du système d'exploitation proprement dit, puisque son rôle est justement d'en organiser le chargement ; lorsque l'ordinateur héberge plusieurs systèmes (on parle alors de multi-amorçage ou de multi-boot), le GRUB permet à l'utilisateur de choisir quel système démarrer ; (pas mal d'infos sur GRUB [ici](#)).

1-1) Structure logique d'un disque HDD (envisagé du point de vue « machine »¹) :

La zone « clé » d'un disque HDD **de moins de 2 To** est le **Master Boot Record (MBR)** – (littéralement : enregistrement d'amorçage maître) ; c'est le nom du premier secteur du DD (repéré par une adresse C/H/S = 0/0/1) sur lequel se trouve écrit ce qui est nécessaire au lancement du système d'exploitation ; quand l'ordinateur démarre, c'est là qu'il commence à lire ; le MBR a une taille de 512 octets, et est organisé comme suit :

- 446 octets réservés au « boot Loader » ou plus précisément à l'une de ses parties.

- 4 zones de 16 octets (c'est la table des partitions) contenant chacune la description d'une des partitions possibles du disque ; chaque description indique :

- l'état de la partition (inerte ou bootable) - (1 octet)
- le « lieu » du début de la partition - (1 octet)
- le cylindre et le secteur du début de la partition - (2 octets)
- le format de la partition utilisé (système de fichier, ex: fat 32, ext2 etc...) - (1 octet)
- le « lieu » de la fin de la partition (1 octet)
- le cylindre et le secteur de la fin de la partition - (2 octets)
- le nombre de secteurs entre le MBR et le premier secteur de la partition - (4 octets)
- le nombre de secteur de la partition - (4 octets)

- 2 octets pour indiquer si le disque dur possède ou non un « Bootloader » ; s'il y en a un, le nombre inscrit ici est égal à « 55aa » en valeur hexadécimale.

Voici ce qui se passera au démarrage :

A la mise sous tension, le microprogramme « Bios » de la carte mère commence à exécuter du code routinier (c'est le POST) comme la vérification de la mémoire, la présence de différents équipements, etc ..., puis il va lire le secteur MBR du disque dur et vérifier s'il y a un chargeur d'amorçage ou « boot loader » en anglais;

1 Le « point de vue machine » correspond à peut près à ce qui est « vu » par le programme BIOS, c'est à dire pour ce qui nous intéresse présentement, le ou les disques durs, la ou les clés USB, le lecteur DVD.

- si dans les 2 derniers octets du secteur MBR le nombre inscrit est égal à 55aa, alors c'est qu'il y a plusieurs distributions, et le programme « Bios » charge et exécute les 446 premiers octets du MBR (où se trouve la première partie du « Boot loader »), qui va elle-même dans ce cas, rediriger la lecture vers le secteur du DD où se trouve le restant du programme « Boot loader » ; ce dernier (GRUB ou Lilo pour Linux) propose alors à l'utilisateur un écran de choix du système d'exploitation à lancer ; après validation, ou bien de façon automatique avec une temporisation, le système d'exploitation est à son tour chargé et exécuté (plus d'infos [ici](#) et [là](#)).

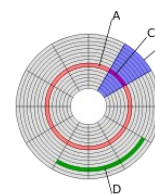
- si le nombre des 2 derniers octets est différent de 55aa, alors c'est qu'il n'y a qu'un seul système d'exploitation, et la partie du MBR écrite dans les 446 octets précités est une routine d'amorçage qui charge directement le noyau du système d'exploitation (plus d'infos [ici](#)).

Pour la machine, les secteurs restant du DD servent à ranger les fichiers des programmes exécutables installés (qui constitueront la partition « système ») et ceux des données que vous créerez (qui constitueront la partition « données ») ; c'est le programme pilote du disque dur (inclus dans la distribution) qui gèrera le rangement de ces fichiers sur le disque, d'une façon invisible pour l'utilisateur.

1-2) Structure logique d'un disque HDD (envisagé du point de vue « utilisateur »²) :

Un disque de stockage peut être divisé en une ou plusieurs parties logiques (ou partitions) ; chaque partition sera considérée (et affichée sur le bureau) comme un volume indépendant, même si physiquement, il s'agit des parties d'un même disque.

La taille et le nombre des blocs (ou cluster) d'une partition sont choisis lors du formatage de cette partition (plus d'infos sur ce sujet en fonction de la taille du disque et de l'usage qu'on veut en faire [ici](#)).



Structure d'un disque :

- (A) piste écrite (rouge)
- (B) secteur géométrique (violet)
- (D) bloc (ou cluster) de 3 secteurs

Les données (autres que le MBR) stockées sur une partition seront organisées sous forme de dossiers et de fichiers, selon une structure en arborescence inversée (une racine unique au départ et les différents niveaux de ramification des branches, constituées par les dossiers et/ou les fichiers, vers le bas).

Sous Linux, la racine unique est repérée « / » (elle s'affichera « / \$ » dans un terminal) ; on ne peut pas « remonter » plus haut ; si on regarde le premier « niveau » de dossiers et de fichiers situés juste « en dessous » de la racine, on verra dans un terminal (exemple pour une distribution « Xubuntu » avec une seule partition) :

[ici : la racine (/) et là la commande (ls -h) pour voir les dossiers et fichiers] de niveau « 1 »

```
linuxmaine@linuxmaine-20-12-04:/$ ls -h
bin    dev    initrd.img.old  media  root  srv    usr
boot  etc    lib             mnt    run   swapfile  var
cdrom  home  lib64          opt    sbin  sys     vmlinuz
core  initrd.img  lost+found    proc  snap  tmp     vmlinuz.old
linuxmaine@linuxmaine-20-12-04:/$
```

2 Le « point de vue utilisateur » correspond à ce qui est « vu » par le système d'exploitation de la distribution linux (et donc indirectement par l'utilisateur), c'est à dire pour le sujet des supports de stockage qui nous intéresse ici, **la ou les partitions** du ou des disques durs ou des clés USB, et non plus seulement les supports physiques de stockage comme le faisait le « point de vue machine » ; il y a là un point délicat et important à comprendre quand on débute.

Sur la première ligne à gauche on voit le dossier « bin » qui est le répertoire contenant les commandes ; c'est un ensemble de fichiers exécutables correspondant aux commandes que l'on peut faire exécuter par le système (comme la commande « ls -h » utilisée ici) ; on y trouve aussi des « utilitaires » permettant de travailler sur, ou avec ces commandes « système » ; pour découvrir les autres répertoires (« dev », « boot », « etc », ... et approfondir ce sujet, reportez-vous à [ce site](#).

Une autre commande (tree -d -L 1) permet de visualiser l'arborescence des 22 dossiers de niveau 1 (sans les fichiers qui peuvent aussi exister au niveau 1) :

```
Linuxmaine@linuxmaine-20-12-04:/$ tree -d -L 1
├── bin
├── boot
├── cdrom
├── dev
├── etc
├── home
├── lib
├── lib64
├── lost+found
├── media
├── mnt
├── opt
├── proc
├── root
├── run
├── sbin
├── snap
├── srv
├── sys
├── tmp
├── usr
└── var
22 directories
```

la racine

les dossiers de niveau 1

1-3) Structure logique d'un disque SSD (envisagé du point de vue « machine ») :

La mémoire « flash » des SSD est composée d'unités de stockage électroniques de 4 kB chacune groupées en « bloc » de 512 (soit $512 \times 4 \text{ kB} = 2048 \text{ kB}$) ; la machine peut lire une unité ou écrire dans une unité vierge, mais pour modifier une unité déjà écrite, elle est obligée d'effacer tout le bloc (de $512 \times 4 \text{ kB}$) et de ré-écrire chaque page en plus de celle qui doit être modifiée. Compte-tenu du nombre limité de cycles d'écriture/effacement des mémoires flash (c'est leur faiblesse par rapport à un HDD), le programme pilote du SSD répartit uniformément le nombre de ces cycles d'écriture/effacement sur la totalité de la place mémoire, ce qui fait que les données vont être déplacées régulièrement sur l'espace de stockage, et qu'on n'a plus une adresse « fixe » pour un paquet de données X ; toutefois ceci reste « opaque » pour le processeur de l'ordinateur qui « considère » lui qu'elles sont toujours à la même adresse ; en fait c'est le programme de gestion du SSD qui sert d'interface entre le processeur et l'espace mémoire et qui assure les correspondances « adresse initiale/adresse actuelle » ; en procédant ainsi, on augmente considérablement la durée de vie du SSD.

Les supports de stockage de type SSD étant généralement de grande taille, le dispositif de structuration du disque (c'est à dire l'équivalent du MBR pour un HDD) a dû évoluer en conséquence ; depuis 2007, pour les supports de stockage > à 2 To, le MBR qui peut fonctionner aussi pour un SSD mais qui ne gère pas plus de 2,2 To est remplacé par le GPT (plus d'infos [ici](#)) ; d'ailleurs pour des machines de type BIOS - unix, équipées de disque de grande taille > à 2,2 To, on utilise le GPT.

E) stockage externe

On peut connecter par le port USB des unités de stockage externes (clés USB ou disques HDD/SSD) ; les disques HDD restent plus fragiles face aux transports et aux chocs que les SSD mais ils sont assez fiables pour la conservation des données ; aucun système de stockage n'étant éternel, les sauvegardes externes, régulières, et sur plusieurs supports, rangés dans des lieux différents, sont donc fortement conseillées.