

# LE DISQUE DUR

## Historique des technologies.

- **1957** Apparition du premier modèle, 50 disques d'un diamètre de 24 pouces (61 cm), capacité 5 Mo.
- **1983** IBM propose le modèle XT qui est équipé d'un disque dur de 10 Mo. La technologie est très onéreuse. Seagate Technologie propose au début des années 80 une nouvelle interface plus connue sous le nom d'**interface ST506**, elle utilisait l'encodage des informations en MFM puis RLL. Le formatage physique du disque dépendait de l'ordinateur sur lequel il avait été fait, on ne pouvait pas déplacer le disque d'une machine vers une autre sans le reformater.
- **1984** Western Digital et IBM mettent au point le standard connu sous le nom WD003 pour les disques durs. La partie du bios gérant le disque dur devenait une partie du bios de la carte mère. La gestion du disque n'était plus dépendante de la carte contrôleur du disque dur mais du bios de l'ordinateur.
- **1985** IBM propose l'**interface ESDI (Enhanced Small Devices Interface)** sur ses premiers PS/1, c'est une version améliorée de ST506.

Cependant ces premières technologies de disques ne comportaient pas beaucoup d'électronique, et devaient être contrôlés par l'unité centrale pour chaque opération. L'unité centrale devait tout d'abord envoyer une commande pour positionner le bras. Puis elle devait signaler au lecteur laquelle des têtes devait effectuer la lecture, et quel secteur devait être lu. Ensuite, l'unité centrale attendait que le secteur désiré se retrouve sous la tête de lecture, et commençait le transfert de données.

## 1986 : IDE (Integrated Device Electronique) : La nouvelle référence.

Western Digital et Compaq proposent une nouvelle mouture du WD1003 en intégrant le contrôleur sur le disque lui même mais en gardant la compatibilité WD1003, c'est le standard **IDE** également nommée ATA. Un petit processeur permet de libérer des ressources de l'unité centrale en implémentant un ensemble de commandes plus sophistiquées. Quand on déplace le disque on n'a plus besoin de le reformater mais seulement de le déclarer au BIOS. Avec les disques IDE, l'unité centrale a juste besoin de connaître le CHS/CTS (Cylindre Tête Secteur) du secteur qu'elle veut lire, et le processeur intégré au disque va se charger de positionner lui même les têtes, puis appeler l'unité centrale lorsque le disque est positionné pour commencer le transfert. Le mode CHS est à réservé au disques de moins de 528 Mo, il reste le standard des disques jusqu'en 1994.

### Les limites des premiers disques durs de type IDE.

- **Le contrôleur ne sait gérer que 2 disques.** Chaque disque a une capacité limitée à 528 millions d'octets car le BIOS ne sait détecter que **1024** cylindres (le numéro de cylindre est code sur 10 bits, soit une plage 0-1023) de **63** secteurs (le numéro de secteur est code sur **6 bits**, soit une plage 1-63 car le secteur 0 n'est pas adressé) de 512 octets soit pour un disque IDE limité à 16 têtes (WD1003 ne gère que **16** têtes)  **$1024 \times 63 \times 16 \times 512 = 528482304$  octets = 504 Mo plus exactement.** Pour dépasser cette limite sur les anciennes machines on trompe le BIOS avec un programme, Disk Manager ou EZ Drive .
- **Le système d'exploitation lui même est un frein.** Il faut attendre MSDOS 4.0 pour franchir la barre des 32 Mo. Un autre goulot d'étranglement est qu'au départ IDE n'autorise qu'un transfert à la vitesse de 3,3 Mbits/s. On parle de PIO mode 0, mais le mode de transfert des données va très vite s'améliorer..

## 1993 : L'E-IDE (Enhanced-IDE ).

Le standard **E-IDE** appelé aussi **Fast ATA-2**, (IDE étant considérée comme l'ATA 1) apporte de nombreuses améliorations techniques.

- **EIDE possède deux canaux, qui peuvent chacun recevoir deux périphériques de type IDE, soit 4 périphériques en tout**, comme le disque dur, des lecteurs CD-ROM ou DVD, des graveurs de CD, des lecteurs ZIP pourvu qu'ils soient de type IDE. Les deux canaux fonctionnent indépendamment l'un de l'autre, cependant, si deux périphériques sont installés sur un même canal, ils ne peuvent fonctionner que l'un après l'autre. C'est pourquoi, lorsque l'on ne possède qu'un disque dur et un CD-ROM, on préfère les installer chacun sur un canal différent
- **Le contrôleur peut gérer des disques ayant une capacité de 8 Giga octets.** Cette norme supporte des disques durs de **255 têtes** (au lieu de 16). Donc la capacité maximale devient :  $1024 \times 63 \times 255 \times 512 = 8422686720 \text{ octets} = 8032 \text{ Mo}$  (arrondi à **8 Go**). Pour franchir la limite des 528 Mo, les BIOS doivent être compatibles EIDE (c'est le cas de ceux postérieurs à Juillet 94). Par translation d'adresse un BIOS compatible EIDE va "transformer" l'adressage CHS/CTS (cylindre, tête secteur) "*physique*" en adressage LBA.
- **Le passage du mode CHS à LBA.** Au lieu d'utiliser le mode CHS pour repérer un secteur, on utilise le mode LBA (Logical Block Addressing). Dans le mode LBA, un programme n'a besoin de connaître que le nombre de secteur depuis le début du disque jusqu'à la position du secteur recherché (les secteurs sont numérotés en commençant à zéro). Quasiment tous les systèmes d'exploitation modernes utilisent l'adressage LBA, mais la notation CHS n'a pas encore disparu complètement. Tout d'abord, MS-DOS, dont les premières versions remontent à près de 20 ans, n'utilise que le mode CHS. D'autres programme, comme Partition Magic, ne fonctionneraient pas si les partitions ne commençaient pas au début d'un cylindre ou d'une face. Les valeurs maximales reconnues en mode CHS sont **0 à 1023 cylindres, 0 à 254 têtes et 1 à 63 secteurs**. En multipliant ces valeurs, on remarque que le disque dur le plus gros, que l'on puisse gérer avec le mode CHS, a une capacité de **8 Go**. En conséquence, si un disque fait 12 Go, beaucoup de programmes ne le verront que comme un disque de 8 Go, parce qu'ils utilisent le mode CHS.
- **Le taux de transfert des données passe à 16,6 Mb/s (PIO Mode 4).** Le mode PIO, **Programmed Input/Output** désigne le transferts de données réalisés par le processeur.

Voici les différents taux de transfert maximum théorique :

PIO mode 0	3.3 Mo/s	
PIO mode 1	5.2 Mo/s	
PIO mode 2	8.3 Mo/s	
PIO mode 3	11.1 Mo/s	
PIO mode 4	16.6 Mo/s	
Ultra DMA 33	33.3 Mo/s	

En **Mode PIO4**, le chipset qui contrôle l'interface IDE intègre un accélérateur de transfert de données appelé **PIIX (PCI ISA IDE XCELERATOR)**. Il permet, entre autre, via le canal DMA (Direct Memory Access) d'envoyer les données du disque vers la mémoire en sollicitant très peu le processeur, cette technologie est encore désignée sous le nom de Bus Mastering. L'utilisation du disque dur en mode DMA, se repère dans les propriétés système du poste de travail, onglet gestionnaire de périphériques, rubrique lecteurs de disques. Pour chaque disque, on peut voir par ses propriétés si dans l'onglet paramètres, la case DMA est validée.

## L'Ultra ATA / UDMA.

Dans ce que l'on désigne sous le terme **Ultra ATA** (norme ATA, ATAPI 4) c'est l'apparition d'un mode d'échange de données

entre le disque dur et la mémoire qui se libère du processeur. On parle aussi souvent d'**UDMA**.

- **1996 L'UDMA/33. L'Ultra DMA (ou Ultra ATA)** qui permet un transfert théorique à 33 Mo/s a été développé conjointement par Quantum et Intel. Cette fonction est incluse dans les chips Intel 430 TX pour Pentium et 440 LX pour Pentium II, qui intègrent tout deux le nouveau pont PIIX4. L'UDMA est adopté en 1996 par les principaux constructeurs de disque durs. (Seagate, WD, IBM). Cependant pour utiliser cette fonctionnalité, il faut installer le pilote Intel PIIX Bus Master IDE. Jusqu'en février 1996 les bios (et bien souvent aussi l'OS) ne savaient pas franchir la barrière des 2,1 Go. En janvier 98 on accède à la barrière des 8,4 Go.
- **1999 L'UDMA/66** (norme ATA, ATAPI 5) permet un débit théorique de 66 Mo/s, mais il faut une carte mère et un chipset qui le supporte comme l'i820. D'un point de vue technique l'Udma/66 améliore la protection des données grâce à l'utilisation d'une nappe de 80 fils au lieu des 40 actuels. Ces 40 fils supplémentaires correspondent à des lignes de masse qui agissent comme des blindages. Cependant si le nombre de fils augmente, le connecteur garde un brochage à 40 fils pour assurer une compatibilité avec l'existant. Correspond aussi au franchissement d'une nouvelle barrière de reconnaissance des capacités, de stockage, 32 Go en juin 1999.
- **2000/2002 L'UDMA/100** (norme ATA, ATAPI 6) **et 133** (norme ATA, ATAPI 7) où l'on retrouve les caractéristiques de l'UDMA/66, mais pour tirer pleinement partie il faut une nappe à 80 fils. Sur UDMA 100 la limite de capacité au delà de laquelle il faudra partitionner est de 137 Mo. Pour contourner cette limite plusieurs constructeurs travaillent sur une nouvelle norme, "Big Drive" ou ATA/ATAPI-6 définie sur 48 bits contre 28 actuellement ce qui devrait permettre de supporter des disques allant jusqu'à 144 pétaoctets (144 000 000 Go).

### Connexion d'un disque UDMA.

La fiche bleue de la nappe est destinée à être branchée sur la carte mère ensuite, le premier disque rencontré doit être l'esclave, et enfin le maître.

## 2002 : Le Serial ATA.

Pour remplacer l'Ultra ATA. On se retrouve avec une interface série et une disparition des claviers à déplacer sur les disques pour éviter les conflits d'adressage. On pourra connecter le disque à chaud. Le chipset KT400 de Via gère actuellement le serial ATA. Le débit théorique de 1,5 Gbits/s (150 Mo/s) n'est qu'un faible progrès par rapport à l'Udma 133 (133 Mo/s) mais le consortium Serial ATA prévoit une évolution vers les 300 puis 600 Mo/s.

<http://www.serialata.com>

## Les "gros disques durs".

Aujourd'hui les disques durs vont bien au delà des 8 Go de la norme EIDE. Dans ce cas pour un BIOS uniquement compatible EIDE, il faudra comme au premier temps de l'IDE (limité à 528 Mo) utiliser un programme additionnel, installé dans le **MBR**, pour tromper le BIOS ( par exemple EZ-Install de [Western Digital](#)). Cependant depuis 1998 de nouveaux BIOS permettent de dépasser la limite des 8 Go grâce à l'utilisation de nouvelles fonctions de **l'interruption 13h**. (Une interruption logicielle, qui relève du BIOS afin de régler les accès au disque). Un BIOS doté de ces fonctions supplémentaires (on parle de EBIOS pour les fonctions 41h à 49h) permet la reconnaissance des gros disques durs avec le mode **LBA (Logical Block Addressing)**.

**Récapitulatif :** On peut dire que selon le Bios, mais aussi l'OS on ne pouvait pas dépasser une capacité de disque de

- 528 Mo avant août 1994
- 2,1 Go avant février 1996
- 8,4 Go avant janvier 1998

- 32 Go avant juin 1999
- 137 Go actuellement

## Branchement d'un disque

Aujourd'hui les câbles reliant les disques durs à la carte mère intègrent des connecteurs de couleur qui se standardisent. Le connecteur bleu est à mettre sur la carte mère, le noir sur le premier disque et le gris sur le second disque.

### Le partitionnement.

#### Définition :

Le **partitionnement** est une opération qui permet de découper un disque dur en plusieurs parties désignées sous les noms de partitions **primaires** ou **principales** (4 au maximum) et partition **étendue** (une seule, mais qui se substitue à une des 4 partitions primaires) dans laquelle on peut ensuite créer des partitions **logiques**. Cette opération se fait à partir d'un programme de partitionnement comme **Fdisk** (le plus connu car faisant partie de MSDOS) sur un disque neuf car ce découpage entraîne la perte des données présente sur le disque. Cependant certains programmes commerciaux comme Partition Magique permettent ce travail sans perte de données. Partitions **primaire, étendue et logique** sont des notions indépendantes de tout système d'exploitation, on les doit uniquement au système PC.

#### Pourquoi partitionner ?

- Pour installer différents OS sur son disque dur.
- Pour mieux organiser son espace disque, séparer système et données.
- Pour gagner de la place et contourner certaines limitation des systèmes..

Tout disque dur se compose de **cylindres**, eux-mêmes divisés en **secteurs** physiques de **512 octets** dans lesquels sont stockées les données. Plusieurs secteurs forment des **clusters**. Le **cluster** peut être défini comme **l'unité d'espace minimal occupé par un fichier**. **L'adresse du cluster a été codée sur 12 bits, puis sur sur 2 octets (16 bits) c'est la Fat 16 et maintenant sur 4 octets (32 bits), c'est la Fat 32.**

Le disque parfait serait celui dans lequel chaque cluster correspondrait à un secteur, mais le système d'exploitation ne sait gérer qu'un nombre limité de clusters. Pour résoudre le problème, le système associe un nombre fixe de secteurs à chaque clusters (en général 4 ou 8 pour les disques de capacité moyenne). On neutralise donc rapidement de la place quand on sait que chaque cluster n'est affecté qu'à un seul et unique fichier. Ainsi sur un disque où les clusters sont de 2048 octets, un fichier de 2049 octets a besoin de 2 clusters de 2048 octets, le second cluster est quasiment vide mais perdu pour y mettre tout autre information. Plus les fichiers sont petits plus la place perdue est considérable.

**Windows 95 dans sa première version** ne peut gérer que des disques limités à 2 Go car en FAT 16 on ne dispose que de  $2^{16}$  soit 65 536 emplacements adressables pour stocker les fichiers, soit 65 536 clusters. Comme un **cluster** ne peut dépasser (sous DOS, Windows 95 et Windows 98) **32 768 octets**, la **taille maximale d'une partition FAT16** est donc égale à  $65536 \times 32768 = 2147483648$  octets (**2 Go**). En FAT 16 la taille minimum occupée par un fichier va varier en fonction de la taille du disque dur. Elle sera de **Capacité du disque divisé par 65536**. Un disque de 256 à 512 Mo possède des clusters de 8 Ko, au delà de 512 Mo, le cluster passe à 16 Ko, à partir de 1,2 Go il sera de 32 Ko. Un fichier de 250 octets comme par exemple un raccourci sous Windows va occuper en réalité 32 Ko sur un disque de 1,5 Go. Que de place perdue.

Pour restreindre le problème on peut créer des partitions de taille variable car c'est au moment du partitionnement que le système détermine le nombre de secteurs affectés aux clusters, ce chiffre étant déterminé par la taille de la partition.

**1997** : La nouvelle version de **Windows 95 OSR2** propose la FAT 32 qui permet de limiter enfin la taille de chaque cluster à 4 Ko (4096 octets). Comme la FAT 32 autorise  $2^{32}$  soit 4,2 milliards de clusters de 4 Ko, le système OSR2 peut gérer

théoriquement un disque de 2 To. (2048 Go) mais en 97 le Bios ne reconnaît que des disques de 8,4 Go. Oui je sais c'est le bordel mais c'est comme d'habitude :o).

## Installer plusieurs systèmes d'exploitation.

Les partitions peuvent être de différents types car **chaque OS gère le disque dur avec son propre système de fichiers**. Si on veut installer plusieurs OS sur un disque dur chaque OS doit avoir sa propre partition. Si le partitionnement est un procédé commun à tous les systèmes d'exploitation, chaque système possède son propre outil de partitionnement.

# Les systèmes de fichiers.

Le disque dur recevant les fichiers, il doit être préparé, c'est l'objet du formatage qui selon les OS (systèmes d'exploitations) organise ces fichiers sous différentes formes.

**FAT** (Files Allocation Table). Il doit son nom à une table d'allocation des fichiers de 16 bits qui adresse un maximum de 65 536 clusters. Tout fichier sur disque dur est répertorié sur 32 octets. (8 + 3 pour le nom, 1 pour l'attribut, 10 sont inutilisés, 2 + 2 pour heure et date, **2 pour l'adresse (soit 16 bits)**, 4 pour la taille.

Jusqu'au Dos 3.3, on ne pouvait pas formater une partition FAT à plus de 32 Mo, l'adresse de chaque cluster était codée sur 12 bits. Chaque cluster était limité à 512 octets.

A partir du Dos 4, l'adressage du cluster sur 16 bits donne accès à 65 536 clusters, mais la partition étant limitée à 2 Go, la taille des clusters peut varier, de 1 Ko (1024 octets) sur des disques de 80 Mo à 32 Ko sur les disques de 2 Go. Sur un disque de 512 Mo le cluster occupe  $512/65\,536 = 8$  Ko.

**VFAT en 1995** (Virtual File Allocation Table) est une forme d'interface qui convertit la **FAT** réelle (12, 16, 32 bits) en un modèle unique en mémoire exclusivement 32 bits. **VFAT ne remplace pas** le système de partition **FAT16** ou **FAT32**, qui concerne seulement la structure physique des partitions, mais au contraire s'y **ajoute**, en constituant une couche supérieure dans le système de fichiers.. Etait déjà présent avec Windows 3.11 en tant que système optionnel activé par le panneau de configuration, icône 386 étendue, mémoire virtuelle.

**FAT 32** en **1997** de l'OSR2 demande pour être installé le reformatage du disque après utilisation de Fdisk. On adresse les clusters sur 32 bits soit (4,2 milliards de clusters ramenés à une taille de 4 Ko). Le système OSR2 peut gérer un disque de 2 To. (2048 Go) mais le Bios ne sait reconnaître que des disques limités à 8,4 Go.

**FAT 32X, en 1998** (C'est tout neuf, de Microsoft et j'ai pas encore bien compris). Sert à exploiter les disques **durs dont la capacité dépasse 8,4 Go**. La FAT est déportée à la fin du disque.

**HPFS** Mis au point en **1989 par MICROSOFT et IBM** alors associés dans le développement d'OS/2 (ce qui a bien changé). Son but était de pallier au défaut de la FAT. Il permet par une méthode relativement complexe de garder des clusters de 512 octets. La fragmentation devient quasiment absente car un fichier HPFS est obligatoirement constitué de blocs contigus quitte à déplacer les autres fichiers. Il s'agit en quelque sorte d'un Defrag permanent. HPFS utilise des noms longs (254 caractères maximum).

**NTFS (4)** Le système de fichier à la HPFS de Windows NT. Peut lire les fichiers de la FAT 16 mais pas l'inverse. Ses points forts :

- Permet de limiter l'accès des fichiers aux utilisateurs.
- Supporte les noms longs
- Les activités disques sont enregistrés dans un fichier journal.

**NTFS 5** Apparu avec Windows 2000 et XP, il n'est lisible que par ces deux versions. Ajoute essentiellement le cryptage et la

compression des fichiers par rapport à NTFS.

**WinFS** (Windows Futur Storage) qui devrait être dans Longhorn, le futur XP. Tous les fichiers sont indexés automatiquement, y compris leur contenu et toutes les modifications sont historisées en temps réel. WinFs est basé sur l'intégration de SQL Server au sein du système d'exploitation.

**CDFS** Le système de fichiers des CD ROM. Sous Dos ou Windows 3.x, il est activé par le programme MSCDEX. Sous W 95, un pilote virtuel CDFS se charge lorsqu'on fait appel au CD.

**OFS** Object File System Nouveau système de gestion des fichiers de Cairo futur NT en 64 bits.

EXT1 puis EXT2 et **EXT3** Le système de fichiers sous Linux.

Certains systèmes d'exploitation peuvent accéder directement ou à l'aide d'outils spécifiques à des systèmes de fichiers prévus pour d'autres systèmes. Par exemple **Linux** sait lire les partitions de type **FAT**, mais pour qu'un poste Windows puisse lire les fichiers EXT2 il lui faut l'utilitaire [explore2fs](#). Si NT4 sait lire les partitions FAT 16, il faut un utilitaire supplémentaire pour lire les partitions FAT 32 ([FAT32.SYS](#)) et il faut [NTFS98](#) si on veut lire les fichiers **NTFS** sous **Windows 95/98**.

Hubert GREGOIRE

Dernière modification le 06/04/04

---

" [Retour en page d'accueil des Services](#) "

" [Retour en page d'accueil de CIEL](#) "

---