Le problème des gros disques durs IDE

Ou l'histoire d'un véritable capharnaüm technologique

Écrit par Djalil CHAFAÏ dchafai@mail.dotcom.fr le 1998-01-18.

Révisions les 1998-01-19, 1999-01-15, 1999-01-18, 1999-01-20 et 1999-01-23.

Avertissements

J'espère que ce texte, malgré les erreurs qu'il peut contenir permettra à quelques uns de mieux comprendre les problèmes évoqués. J'ai personnellement mis du temps à comprendre certains aspects des choses et je ne suis pas sûr d'avoir encore tout compris.

Je serais très reconnaissant aux lecteurs de me faire part de leurs éventuelles critiques et remarques constructives afin que nous en profitions tous. Cela dit, bien que je comprends bien la détresse de certains, je n'ai ni le temps ni l'envie de faire du support technique, ne me demandez donc pas de résoudre vos problèmes de disques s.v.p.

Ce document ne traite pas du problème des disques durs «Ultra-DMA».

Je dégage toute responsabilité quant aux éventuelles conséquences résultant de l'utilisation de ce document de quelque façon que ce soit.

Les marques et noms déposés cités dans ce document sont la propriété de leur détenteurs respectifs.

1 Terminologie

IDE voudrait dire «Integrated Device Electronics» car ces disques possèdent leur propre électronique de contrôle intégrée. On devrait dire ATA (AT Attachement). ATA est le standard d'interface pour les disques durs, quant à IDE, il semble que ça soit un nom déposé par WESTERN DIGITAL. De même, on devrait dire ATA-2 au lieu de EIDE (Enhenced IDE ou Extented IDE selon les sources).

Quant à ATAPI, cela signifie «ATA Packet Interface», cette norme est apparue en même temps que l'ATA-2. Elle ressemble à SCSI et permet le contrôle de nombreux périphériques tels que les lecteurs de CD-ROM, de ZIP et de bandes connectés sur les mêmes supports que les disques ATA-2.

Dans la suite, j'utiliserai indifféremment IDE/ATA et EIDE/ATA-2.

2 La géométrie CHS et l'interruption 13 du BIOS

En théorie, un disque dur est constitué de plateaux magnétiques superposés découpés en secteurs et pistes. À chaque face de plateau correspond une tête de lecture. Sur chaque face de plateau, l'intersection des pistes et des secteurs forme des blocs de 512 octets référencés par trois numéros :

– le numéro de piste (ou de cylindre)

– le numéro de plateau (ou de tête)

– le numéro de secteur (sur le plateau sous la tête)

Ces blocs de 512 octets sont les plus petites unités de données adressables sur les disques durs. On les appelle aussi secteurs, ce qui peut porter à confusion. Un secteur fait 512 octet et est repéré par un numéro de cylindre, un numéro de tête et un numéro de secteur. On parle alors de géométrie CHS = Cylinders Heads Sectors. C'est ce que l'on appelle une géométrie cylindrique.

Un disque IDE est accessible via un port d'entrées/sorties standard. Ce port permet de communiquer avec la puce chargée de contrôler[1] les disques. L'interruption 13[2] du BIOS[3] se charge de communiquer à travers ce port. Sous MS-DOS, les programmes font appel au système d'exploitation pour accéder aux disques, lui même faisant appel à l'interruption 13 (héxa) du BIOS.

La transmission des paramètres CHS à l'interruption 13 du BIOS se fait sur 24 bits (via trois registres 8 bits) :

- 10 bits pour le numéro de cylindre (soit 1024 au maximum)
- 8 bits pour le numéro de tête (soit 256 au maximum)
- 6 bits pour le numéro de secteur (soit 64 au maximum)

On a donc au plus 2^{24} blocs de 512 octets, la taille maximale théorique du disque dur est donc de $2^{24+9} = 8$ Go[4].

Remarque

La géométrie reportée par les disques durs modernes ne correspond pas forcément à leur géométrie physique. Elle est en fait dictée par les limitations de l'interface IDE, l'électronique du disque se chargeant de la conversion automatique. Il se peut très bien par exemple que votre disque ne comporte que deux plateaux bifaces avec quatre têtes et que sa géométrie reportée par l'interface IDE soit de seize têtes. De tout manière, cela ne change pas grand chose car ces considérations ne concernent que l'«intérieur» du disque dur.

La géométrie CHS «externe» d'un disque est souvent imprimée sur son boitier métallique. Vous trouverez souvent des informations techniques (géométrie, cavaliers. . .) sur le site ouèbe[5] du constructeur.

3 Le secteur d'amorce : MBR et table des partitions

Sur PC, le premier secteur d'un disque dur (CHS=0,0,0) est appelé «secteur d'amorce». Les 446 premiers octets du secteur d'amorce constituent ce que l'on appelle le «Master Boot Record» (MBR). Il s'agit de code machine qui permet d'amorcer le démarrage du système d'exploitation.

Lors du boot, le BIOS, après avoir terminé sa tâche d'initialisation, copie le secteur d'amorce du premier disque dur en mémoire centrale puis donne la main au MBR en ordonnant au processeur d'exécuter le code machine y figurant. Ce code est responsable du chargement du système d'exploitation.

Bien entendu, les choses se passent différemment si l'on demande au BIOS de bouter sur disquette, CD-ROM ou autre périphérique.

`fdisk` de MS-DOS confectionne un MBR. Idem pour LILO. En faisant un `fdisk /MBR` sous MS-DOS, vous demandez à `fdisk` de recréer un MBR MS-DOS tout neuf.

Sous Linux, LILO écrit le MBR dans les 446 premiers octets du disque lors de sa phase d'installation. En fait, il peut aussi le mettre dans le premier secteur d'une partition étendue, c'est ce que permet de préciser le paramètre `boot=` du fichier `lilo.conf`.

Les octets restants du secteur d'amorce sont censés contenir une table appelée **table des partitions**. Cette table spécifie le découpage du disque en zones (partitions) référencées par une géométrie CHS. Cette géométrie à exactement

1. Chaque disque contient sa propre électronique de contrôle mais il semble que seule l'une d'entre elle doit être activée pour chaque nappe. Elle contrôlelerait alors les deux disques de la nappe ce qui expliquerait pourquoi on parle de maître et d'esclave.

2. Sur architecture Intel x86, une interruption est une instruction (INT) qui ordonne au processeur d'interrompre momentanément son travail en sautant à une adresse repérée par un numéro d'ordre dans la table des vecteurs d'interruptions (TVI) qui figure dans les 1024 premiers octets de la mémoire centrale (256 vecteurs d'interruptions possibles). Les interruptions permettent au BIOS et à certains systèmes d'exploitation comme MS-DOS de fournir leurs services aux programmes.

En détournant une interruption, un pilote peut effectuer une gestion supplémentaire d'un périphérique exotique non reconnu par le BIOS. Sur PC-AT, l'interruption 13 par exemple est réservée aux disques et est implémentée par le BIOS au départ. Les BIOS SCSI par exemple détournent l'interruption 13 du BIOS principal pour implémenter la gestion des périphériques SCSI. Celà explique pourquoi on ne peut pas bouter sur un périphérique SCSI si le contrôleur auquel il est connecté ne possède pas de BIOS.

3. BIOS signifie «Basic Input Output System». Sur PC, le BIOS est un programme écrit sur une puce intégrée à la carte mère. Il est adressé à la fin du premier Mo de la mémoire centrale. Son point d'entrée standard est en 0xFFFF0. C'est le premier programme exécuté par le processeur au démarrage de l'ordinateur. Il est responsable de la détection, l'initialisation et le pilotage des périphériques. Il fournit ses services au système(s) d'exploitation pauvres via des interruptions. Certaines cartes d'extention contiennent une extention du BIOS qui permet la gestion de périphériques exotiques ou trop complexes. C'est le cas par exemple des cartes vidéo ainsi que de de la plupart des contrôleurs SCSI.

4. Ici, 1 Ko = 2^{10} octets, 1 Mo = 2^{20} octets et 1 Go = 2^{30} octets. C'est très pratique, les multiplications de puissances de 2 se résumant à des additions. Beaucoup de constructeurs utilisent une convention commercialement plus avantageuse : 1 Go = 10^9 octets. Ces conventions différentes expliquent pourquoi la limite fatidique des 512 Mo est quelque fois appelée limite des 540 Mo.

5. `http://www.`*constructeur*`.com/` vous y mènera la plupart du temps.

les mêmes limitations que l'interruption 13 du BIOS, c'est à dire 1024 cylindres (10 bits), 256 têtes (8 bits) et 64 (6 bits) secteurs.

La consultation de la table des partitions peut permettre de reconstituer la géométrie du disque utilisée.

Vous trouverez à la fin de ce document une annexe (page 7) sur la structure de la table des partitions.

4 Le problème

Le problème vient du fait que l'interface IDE (i.e. l'électronique du disque) n'a pas les mêmes conventions que l'interruption 13 du BIOS, les voici :

- 16 bits pour le numéro de cylindre (soit 65536 au maximum)
- 4 bits pour le numéro de tête (soit 16 au maximum)
- 8 bits pour le numéro de secteur (soit 256 au maximum)

Soit une taille de disque maximale théorique de 128 Go. Ces limitations contradictoires entre interruption 13 du BIOS et interface IDE n'étaient pas gênantes tant que les disques ne dépassaient pas $1024 \times 16 \times 64 \times 512 = 512$ Mo. Voici un tableau récapitulatif

PARAMÈTRE	BIOS INT 13	INTERFACE IDE	DÉNOMINATEUR COMMUN
CYLINDERS	1024 (10 bits)	65536 (16 bits)	1024 (10 bits)
HEADS	256 (8 bits)	16 (4 bits)	16 (4 bits)
SECTORS	64 (6 bits)	256 (8 bits)	64 (6 bits)
«LARGEUR»	24 bits	28 bits	20 bits
TAILLE MAXI	8 Go	128 Go	512 Mo

Les disques étant de plus en plus gros, les fabricants de disques durs ont projeté de commercialiser des disques IDE avec une géométrie à 64 secteurs, 16 têtes et plus de 1024 cylindres. De tels disques posent des problèmes à l'interruption 13 et à la table des partitions car les cylindres dont le numéro est supérieur ou égal à 1024 sont inaccessibles. Par ailleurs, seuls les quatre premiers bits des huit prévus par l'interruption 13 du BIOS pour coder la tête sont utilisés car précisément l'interface IDE code la tête sur quatre bits.

5 Solutions

5.1 Le LBA de la nouvelle norme ATA-2

La nouvelle norme ATA-2 (dite EIDE) qui remplace l'ancienne ATA (dite IDE) introduit une nouvelle technique d'adressage des blocs de 512 octets des disques durs appelée «Logical Block Addressing[6]» (**LBA**).

Il s'agit d'un adressage linéaire sur 28 bits (et non plus tridimensionnel comme le CHS), recouvrant pleinement les capacités de l'interface IDE. La taille maximale d'un disque est alors de 128 Go.

On se débarrasse ainsi des limitations CHS de l'interface IDE, notamment de la limitation à 4 bits pour les têtes. Un BIOS reconnaissant le LBA se charge de maintenir un accès CHS au disque dur via l'interruption 13 en convertissant systématiquement les adresses 24 bits CHS en adresses linéaires LBA sur 28 bits.

Pour être plus précis, la norme EIDE (ATA-2) ajoute trois ports au port IDE standard mais la plupart des cartes mères n'en intègrent que deux au total. Chaque port permet de connecter une nappe. Chaque nappe peut porter jusqu'à deux périphériques (disques, lecteurs de CD-ROM, lecteurs ZIP et autres périphériques ATAPI). En voici la liste ainsi que les adresses d'entrées/sorties et IRQ associées :

```
ide0=0x1f0,0x3f6,14
ide1=0x170,0x376,15
ide2=0x1e8,0x3ee,11
ide3=0x168,0x36e,10
```

6. Certains disent que LBA signifie «Linear Block Adressing», cela semblerait plus logique mais j'ai vu les deux versions dans les documentations.

5.2 Translation CHS

Il s'agit d'une technique pour passer outre la limitation à 1024 cylindres (10 bits) de l'interruption 13.

L'idée est de simuler dans l'interruption 13 une géométrie «normale». Le BIOS divise par une puissance de 2 adéquate le numéro réel du cylindre et multiplie par la même puissance de 2 le numéro réel de la tête, de sorte que la géométrie CHS ainsi obtenue puisse passer par l'interruption 13. Le système d'exploitation qui utilise l'interruption 13 du BIOS ne voit rien : de son point de vue, le disque a une géométrie CHS tout à fait habituelle.

Un disque de géométrie CHS réelle de $2000 \times 16 \times 20$ par exemple sera vu par un système d'exploitation utilisant l'interruption 13 comme un disque de géométrie CHS de $1000 \times 32 \times 20$. On appelle cette technique la **translation CHS**. Elle permet d'utiliser un disque avec plus de 1024 cylindres via l'interruption 13, ce qui était impossible auparavant.

Notons au passage que les constructeurs de disques et les fabricants de BIOS n'ont fait qu'appliquer à l'IDE ce qu'ils utilisaient déjà pour les interfaces et BIOS SCSI.

6 Remarques

6.1 Désignations utilisées par les BIOS

Les désignations utilisées par les BIOS sont un peu curieuses, ainsi, la translation CHS est souvent appelée Large Mode tandis que le mode CHS traditionnel non translaté est appelé Normal Mode. Pour compliquer les choses, il faut préciser que sur beaucoup de BIOS, le LBA Mode entraîne une translation CHS (i.e. le Large Mode) alors qu'il s'agit de deux choses très différentes, ce qui est souvent source de confusions !

6.2 Vieux BIOS

Pour les vieux BIOS qui ne savent pas utiliser le LBA ni faire de translation CHS, il existe des pilotes qui détournent l'interruption 13 et fournissent ainsi un accès à ces disques aux systèmes d'exploitation faisant appel au BIOS comme MS-DOS. Une solution plus propre consisterait à mettre à jour le BIOS en changeant la ROM par exemple s'il n'est pas sur mémoire FLASH, ça n'est malheureusement pas toujours possible.

6.3 Disques de plus de 8 Go

Les constructeurs de disques durs IDE commencent à commercialiser des disques de plus de 8 Go, dépassant ainsi les limites de l'interruption 13 du BIOS et donc de la table des partitions, avec ou sans translation CHS.

Pour passer outre ces limitations, les concepteurs de BIOS ont étendu l'interruption 13. Mais ces extentions ne sont utilisées que par des logiciels y faisant appel comme `EZ-DRIVE` par exemple. Les `fdisk` traditionnels ne savent pas faire.

Cela va sans sans doute poser de nouveaux problèmes mais je n'en sais pas assez pour l'instant.

À partir de ce point, les disques dont je parle sont tous supposés être des disques dits EIDE de plus de 512 Mo et de moins de 8 Go. Je suppose aussi que le mode LBA et la translation CHS sont activés dans le menu de configuration du BIOS.

7 Et Linux ?

Ici, je parle des noyaux Linux récents, disons que je ne pense pas dire de bêtises pour un noyau 2.0.30.

En théorie, Linux n'a que faire de l'interruption 13 du BIOS puisqu'il ne l'utilise pas. Le pilote IDE de Linux communique directement avec les ports IDE en LBA. La translation CHS lui est donc inutile en théorie.

En fait les choses sont plus subtiles. La géométrie utilisée par la table des partitions souffre des mêmes limitations que l'interruption 13 du BIOS. De ce fait, la table des partitions doit être dressée en utilisant une géométrie translatée sous peine de débordements possibles. Cela permet aussi à Linux et MS-DOS d'être d'accord malgré le fait qu'MS-DOS utilise l'interruption 13 du BIOS et pas Linux.

De même, LILO fait appel à l'interruption 13 du BIOS pour accéder aux disques pour la simple et bonne raison qu'aucun système d'exploitation n'est chargé quand LILO s'execute au moment du boot.

Il est donc très important que la table des partitions soit rédigée en géométrie translatée. Le pilote IDE des noyaux Linux récents effectue en automatiquement la translation [7] lorsqu'on lui demande la géométrie d'un disque, ce qui permet

7. En tout cas pour les disques connectés sur le premier port IDE.

sous Linux à `fdisk` mais aussi à LILO lors de sa phase d'installation (confection du MBR) d'utiliser une géométrie translatée qui contente tout le monde.

La géométrie que reporte Linux est affichée lors du boot. Voici par exemple ce que donne chez moi un `dmesg | grep 'CHS='`

```
hda: QUANTUM FIREBALL ST4.3A, 4110MB w/81kB Cache, CHS=524/255/63, UDMA
```

Vous pouvez également obtenir ces informations en utilisant `hdparm -g /dev/hd?`.

7.1 LILO et l'option «linear»

Comme nous l'avons vu, LILO fait appel à l'interruption 13 du BIOS pour accéder aux disques pour la simple et bonne raison qu'aucun système d'exploitation n'est chargé quand LILO s'execute au moment du boot.

D'un autre côté, LILO, lors de sa phase de confection et d'installation du MBR sous Linux, utilise la géométrie que lui donne le noyau Linux. Il est donc important que cette dernière soit translatée pour qu'elle soit en accord avec le BIOS au moment du boot (exécution du MBR installé par LILO).

Si malgré tout votre Linux reporte une géométrie CHS non translatée, vous risquez d'avoir des problèmes avec LILO. La parade consiste à utiliser l'option `linear` dans `lilo.conf` lors de la configuration et installation de LILO. LILO mémorise alors une adresse linéaire obtenue à partir de la géométrie CHS. Lors du boot, cette adresse linéaire est reconvertie en adresse CHS mais cette fois ci avec les conventions du BIOS ! Du coup, LILO parvient à accéder au disque correctement et le tour est joué. Il semble que cette technique soit apparue à l'origine pour les disques SCSI, qui franchirent la barre fatidique des 512 Mo bien avant les disques IDE.

7.2 Les ports IDE 2, 3...

Je ne parlerai que du second port IDE car la plupart des cartes mères n'intégrent que deux ports. Mais je pense que ce que je dis à propos du second port IDE est encore vrai pour les ports IDE suivants.

D'après une source sûre, le pilote IDE de Linux ne sais pas comment obtenir la géométrie utilisée par le BIOS pour les disques du second port IDE. Si le disque possède déjà une table des partitions, le pilote utilise la géométrie de la table des partitions, sinon il utilise la géométrie non translatée que lui fourni l'interface IDE.

En effet, le BIOS utilise deux petites zones de la mémoire centrale pointées par les vecteurs d'interruption 0x41 et 0x46 pour stocker lors du boot la géométrie CHS des deux premiers disques durs (à l'époque, ça suffisait). Une copie de ces zones de données est sauvegardée [8] par le noyau Linux [9] lors du boot pour que les pilotes puissent y accéder ultérieurement. Le pilote IDE par exemple s'en sert pour déterminer la géométrie utilisée par le BIOS pour les deux premiers disques IDE.

Le manque d'informations de géométrie CHS BIOS pour les disques du second port IDE peut poser de graves problèmes comme nous allons le voir plus loin. L'idéal serait que Linux interroge le BIOS via l'interruption 13 lors du boot avant le chargement du noyau proprement dit [10]. Cela garantirait qu'il utilise la même convention que lui. Voici ce que dit Mark LORD dans le pilote IDE du noyau Linux (2.0.36) :

```
 *  To do, in likely order of completion:
 *      - modify kernel to obtain BIOS geometry for drives on 2nd/3rd/4th i/f
```

Malheureusement, personne ne l'a fait à l'heure actuelle. Le noyau Linux 2.2 souffrira donc du même problème jusqu'à nouvel ordre (patch pardon :-).

Fort heureusement, la plupart des disques qui sont vendus sont déjà partitionnés par le vendeur mais il peut arriver que ça ne soit pas le cas ou encore que la table a été effacée par une manip quelconque d'un utilisateur croyant bien faire en partant de zéro.

Une méthode très simple permet d'éviter tout problème : partitionner les disques IDE en les branchant sur le premier port IDE ou alors en spécifiant la géométrie translatée BIOS en paramètre au noyau lors du boot. Vous pourrez ensuite les rebrancher sur le second port sans problèmes.

---

8. Ainsi que d'autres données BIOS comme les paramètres vidéo etc.
9. Confer le fichier `linux/arch/i386/boot/setup.S` des sources du noyau Linux.
10. Par exemple dans `linux/arch/i386/boot/setup.S` et/ou `linux/arch/i386/kernel/setup.c`.

7.2.1 Exemple de problèmes

Supposons par exemple que vous disposiez d'un disque dur *sans table des partitions*. Si vous le branchez sur le *second port* IDE (en `hdc` ou `hdd`) et que vous le partitionnez *sous Linux* après avoir bouté sur un disque du premier port par exemple alors vous risquez d'avoir des problèmes.

En effet, comme nous l'avons vu, la table des partitions souffre des mêmes limitations géométriques que l'interruption 13 du BIOS. `fdisk` risque de se plaindre de débordements, et même si vous parvenez à créer une table, elle sera dressée en géométrie non translatée, ce qui rendra tout accès via MS-DOS impossible. De plus, si vous souhaitez utiliser LILO pour bouter Linux dessus, il vous faudra absolument utiliser l'option `linear` car LILO utilisera au boot la géométrie translatée du BIOS alors qu'il a utilisé lors de sa phase de confection d'installation du MBR sous Linux la géométrie non translatée que lui fourni Linux.

Bien évidement, vous risquez ne ne vous apercevoir de rien si vous utilisez sans le savoir l'option `linear` de LILO et que vous ne tentez pas d'y accéder sous MS-DOS et consort.

Plus grave encore, si vous déplacez ce disque en le branchant sur le premier port IDE (en `hda` ou `hdb`), vous ne pourrez plus y accéder via MS-DOS, ni LILO ni Linux sans le repartitionner car la géométrie qu'utilise la table des partition est une géométrie non translatée.

En revanche, si vous avez partitionné votre disque dur lorsqu'il était branché en `hda` ou `hdb` et que vous le connectez ensuite sur le second port en `hdc` ou en `hdd`, vous pourrez y accéder normalement aussi bien avec MS-DOS qu'avec LILO ou Linux car Linux utilisera la géométrie (translatée) de la table des partitions existante.

Remarque

Il peut arriver (je l'ai vu !) qu'un disque IDE reconnu habituellement en LBA par le BIOS ne le soit plus après qu'on l'ai utilisé avec Linux sur le second port IDE. Dans ce cas, une solution consiste à forcer le mode LBA dans le BIOS, ça n'est malheureusement pas possible sur tous les BIOS.

Références

Ce que je dis est surtout basé sur mon expérience personnelle ou celle des personnes qui ont bien voulu me parler de la leur, mais je n'aurais jamais rien compris sans avoir lu les documents suivants. Un grand merci à leur auteurs respectifs.

- La programmation avancée sous MS-DOS par Ph. MERCIER chez Marabout;
- La maîtrise du MS-DOS et du BIOS par Ph. MERCIER chez Marabout;
- `FAQ ATA/ATA-1/ATA-2/IDE/EIDE/etc` par Hale LANDIS;
- `Interrupt List, Release 60` par Ralf BROWN disponible sur ftp://ftp.cdrom.com/;
- Source du pilote IDE du noyau Linux `drivers/block/ide.c`;
- Source du noyau Linux `arch/i386/boot/setup.S`, `arch/i386/kernel/setup.c` et `include/linux/genhd.h`;
- `Documentation/ide.txt` des noyaux 2.0.8 et +;
- Documentation de LILO;
- `Large-Disk MINI-HOWTO` de Andries BROUWER.

Il y a aussi beaucoup d'autres documents dont j'ai oublié le nom et que j'ai lu il y a bien longtemps maintenant...

Épilogue

J'espère avoir été le plus clair possible et ne pas avoir dit de bêtises... :-)

Je remercie mon frère Karim ainsi que Damien CHRISMENT, Emmanuel BLINDAU, Lionel LOBJOIS, Marc LORD, François PÉTILLON, O_R (orou@my-dejanews.com) et Pablo SARATXAGA pour leur aide et remarques.

Un grand merci également aux abonnés à la liste diffusion Linux-31 qui ont bien voulu m'envoyer la géométrie de leur disque IDE sous Linux lorsque je leur ai demandé.

Annexe : Structure de la table des partitions

Voici la structure de la table des partitions :

Décalage	Taille en octets	Description
0x01BE	16	Quatrième partition
0x01CE	16	Troisième partition
0x01DE	16	Deuxième partition
0x01EE	16	Première partition
0x01EF	2	Signature (doit être 0xAA55)

Notez que 0x01BE = 446, la table des partitions vient donc juste après le MBR sur le secteur d'amorce du disque. Pour chaque partition, les seize octets contiennent les données suivantes :

Décalage	Taille en octets	Description
0x00	1	0x80 si boutable, 0x00 sinon
0x01	1	Tête du premier secteur de la partition
0x02	2	Premier secteur et premier cylindre de la partition
0x04	1	Type de la partition
0x05	1	Tête du dernier secteur de la partition
0x06	2	Dernier secteur et dernier cylindre de la partition
0x08	4	Numéro absolu du premier secteur de la partition par rapport au début du disque
0x0C	4	Nombre total de secteurs de la partition

Le numéro de premier secteur de la partition est codé sur 6 bits (bits 8-13 du mot au décalage 0x02) et celui du premier cylindre sur 10 bits (bits 14,15,0-7 du mot au décalage 0x02). Idem pour les dernier secteur et cylindre de la partition au décalage 0x06.

Le nombre de secteurs étant codé sur 4 octets, la taille maximale théorique d'une partition est de 2048 Go, ce qui peut sembler absurde car la largeur des champs de géométrie CHS de début et de fin ne le permet pas. En fait, pour repérer une partition, il suffit d'avoir le numéro du premier secteur et le nombre de secteurs de la partition tout deux codés sur 4 octets, sans tenir compte des champs CHS de début et de fin qui sont beaucoup trop exigus. Cette numérotation par secteurs absolus et relatifs est à la fois plus simple et plus souple, c'est l'idée même du LBA.

Voici une liste de types de partitions :

Code	Type	Code	Type	Code	Type	Code	Type
0x00	Vide	0x01	DOS FAT 12-bit	0x02	XENIX root	0x03	XENIX usr
0x04	DOS FAT 16-bit <32M	0x05	Étendue	0x06	DOS FAT 16-bit >=32	0x07	OS/2 HPFS
0x08	AIX	0x09	AIX boutable	0x0A	OS/2 Boot Manag	0x40	Venix 80286
0x51	Novell?	0x52	Microport	0x63	GNU HURD	0x64	Novell Netware
0x65	Novell Netware	0x75	PC/IX	0x80	Old MINIX	0x81	Linux/MINIX
0x82	Linux swap	0x83	Linux native	0x85	Linux étendue	0x93	Amoeba
0x94	Amoeba BBT	0xA5	BSD/386	0xA6	OpenBSD	0xA7	NEXTSTEP
0xB7	BSDI fs	0xB8	BSDI swap	0xC7	Syrinx	0xDB	CP/M
0xE1	DOS access	0xE3	DOS R/O	0xF2	DOS secondaire	0xFF	BBT

Une partition étendue peut contenir dans son premier secteur une nouvelle table des partitions. Mais ce mécanisme de chaînage s'arrête là.

Ce document a été rédigé en $\LaTeX 2_\varepsilon$. Vous trouverez la version HTML en ligne ainsi qu'une version imprimable PostScript sur ma page ouèbe ou encore sur le site du Club des Utilisateurs de Linux Toulousains et des environs à l'URL http://savage.iut-blagnac.fr/projets/doc/.
