

Rappel de l'architecture d'un disque dur et de la terminologie employée

Pour mieux comprendre les notions de cylindre, piste, secteur, cluster, coordonnées CHS... :

Un disque dur est constitué de plusieurs plateaux rigides solidaires d'un axe central auquel un moteur électrique imprime une vitesse de rotation constante (le plus souvent 5400 ou 7200 tours / mn). Chaque plateau est constitué d'un disque (en métal, verre ou céramique) sur la surface magnétique duquel sont stockées les données qui sont écrites en code binaire [0 ou 1] grâce à une tête de lecture/écriture. Il y a une tête par face, et deux faces par plateau. Ces têtes de lecture/écriture sont toutes reliées à un seul et unique bras rendant les têtes solidaires les unes des autres. Le bras est motorisé et les têtes vont donc se déplacer radialement à la surface des plateaux pendant qu'ils tournent, permettant ainsi d'accéder à la totalité des données. Ces têtes sont des électro-aimants qui se baissent et se soulèvent pour pouvoir lire l'information ou l'écrire. Leur géométrie leur permet de survoler la surface du plateau sans jamais le toucher : elles reposent sur un coussin d'air d'environ 10 nanomètres créé par la rotation des plateaux.



Les têtes commencent à inscrire des données de la périphérie vers le centre du plateau. Les données sont organisées en cercles concentriques appelés "**pistes**", créées par le formatage de bas niveau. Chaque piste est divisée en petits morceaux que l'on appelle "**secteurs**" et contenant les données (512 octets par secteur).

Le secteur est la plus petite unité fonctionnelle d'un disque dur. La juxtaposition de plusieurs secteurs constitue un "**cluster**" (unité d'allocation en français) qui représente la plus petite zone du disque que peut occuper un fichier. Le nombre de secteurs par cluster varie en fonction du système de fichier utilisé (FAT ou NTFS) et de la taille de la partition : il y a le plus souvent 2 à 64 secteurs par cluster. Ainsi un fichier, même minuscule, occupera au moins un cluster (d'où la perte d'espace +++ en fonction du système de fichier utilisé...). Les têtes étant liées entre elles, elles sont toujours à la verticale les unes des autres si bien que toutes les données lues ou écrites à un instant T vont se situer également à la verticale les unes des autres : on parle de **cylindre** de données pour désigner l'ensemble des données stockées sur une même verticale sur la totalité des disques. L'intérêt du cylindre est évident : toutes les têtes étant simultanément positionnées sur le même cylindre il est beaucoup plus rapide d'écrire sur les mêmes pistes des plateaux superposés (c'est-à-dire sur un cylindre donné) que de déplacer à chaque fois l'ensemble des bras l'adressage d'un fichier ou d'un secteur se fait donc de manière physique en définissant la position des données grâce aux coordonnées cylindre / tête / secteur (en anglais **CHS** pour Cylinder / Head / Sector).

Les cylindres sont codés sur 10 bits dans le MBR (donc 1024 au maximum), la numérotation commençant à 0

Les têtes sont codées sur 8 bits (256 maxi), la numérotation commençant également à 0

Les secteurs sont codés sur 6 bits (63 maxi), la numérotation commençant toujours à 1 (contrairement aux numérotations en secteurs absolus ou logiques qui, elles, commencent à 0 : voir ci-dessous)

1 secteur = 512 octet

La taille maximale de disque que peut donc gérer en théorie le MBR est de 8 Go environ (1024 x 256 x 63 x 512). Mais en pratique, au-delà de 8 Go, seules les valeurs exprimées en secteurs absolus sont valides (offsets 1C6h et 1CAh dans le MBR) alors que les valeurs de têtes et de cylindres sont fausses (offsets 1C0h et 1C4h du MBR) mais vont être extrapolées par le BIOS grâce aux extensions de l'interruption logicielle 13h qui vont permettre de gérer ces données en mode LBA (Logical Block Addressing).

Il faut bien distinguer secteur absolu et secteur logique : le terme de secteur absolu fait référence au DD entier alors que le terme de secteur logique fait référence à la partition.

* Lorsqu'on parle en **secteur absolu** (à l'échelle du DD), le 1er secteur (qui porte le numéro 0) correspond au MBR, le secteur de boot de la 1ère partition est donc le 63ème secteur et les données commencent au secteur absolu 79 (dans le cadre d'une partition NTFS).

* Lorsqu'on parle en **secteur logique** on se place toujours au sein d'une partition lambda et on ne prend pas en compte les 63 premiers secteurs du DD (renfermant MBR et secteurs réservés) ni les secteurs occupés par les partitions qui précèdent : le 1er secteur logique correspond donc au secteur de boot de la partition (et porte là encore le numéro 0) et les données commencent au 16ème secteur logique de la partition (si NTFS) ou au 32ème secteur (si FAT32).

Chaque emplacement du disque dur peut donc être défini en secteurs absolus (par rapport au DD), en secteurs logiques au sein d'une partition donnée, ou en coordonnées CHS : par exemple le secteur absolu 126 correspond au secteur logique 63 de la 1ère partition ou encore aux coordonnées CHS 0 2 1

Lorsqu'on installe un disque dur, il faut aller dans la gestion des disques pour "**l'initialiser**" (création du MBR), puis le "**partitionner**" (création de la table de partition dans le MBR), et enfin le "**formater**" (création du secteur de boot et de la Table d'Allocation des Fichiers) : c'est ce qu'on appelle le **formatage de haut niveau** qui, vous l'aurez compris, n'est pas un véritable formatage puisqu'il ne touche pas aux données éventuellement présentes sur la partition. A contrario, durant le **formatage de bas niveau** (qui est fait en usine avant qu'on achète le DD mais qu'on peut refaire à tout moment avec certains logiciels spécialisés), des 00 sont écrits sur chaque octet du DD et toutes les données présentes sur le DD sont intégralement effacées : c'est ce que l'on devrait toujours faire avant de donner un ordinateur si l'on a des données sensibles...

signification et correspondances entre bit, byte, octet, mots (word),

décimal et hexadécimal

Contrairement au **système décimal (base 10)** que l'on utilise tous les jours, la numérotation **hexadécimale fonctionne en base 16** (hexa = 6 + décimal = 10).

Pour ce faire, en plus des 10 chiffres utilisés dans le système décimal (de 0 à 9), on a introduit l'utilisation supplémentaire de 6 lettres (de A à F) pour les numérotations hexadécimales. Ainsi les 16 valeurs utilisées en hexadécimal sont 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E et F, et ce sont ces 16 valeurs que l'on retrouve par exemple dans les éditeurs hexadécimaux (voir EditHexa dans ma signature) qui permettent de présenter sous forme de tableau (à 16 colonnes et 32 lignes) les 512 octets d'un secteur.

Certains nombres hexadécimaux pouvant ne contenir que des chiffres, et pour éviter toute confusion avec une numérotation décimale, on rajoute par convention un "h" après le nombre pour indiquer qu'il s'agit d'un nombre hexadécimal (par exemple **245h**) alors qu'on laissera le nombre tel quel lorsqu'il s'agit d'une numérotation décimale (245).

A noter qu'au cours des processus de lecture/écriture sur les secteurs, les processeurs Intels et compatibles inversent les octets lors de la lecture d'un mot. Par exemple, la lecture des octets consécutifs 5Eh, A3h conduisent au

mot A35Eh. Ceci est important à connaître pour les conversions en secteurs des valeurs hexadécimales qui apparaissent dans le MBR et les secteurs de boot des partitions.

en code binaire (base 2), un bit est égal à 0 ou à 1. Ainsi :

* **une structure codée sur 6 bits** donne une valeur maximale de 11 1111 dont l'équivalent en numérotation décimale (base 10) est 63. En effet $2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0 = 32 + 16 + 8 + 4 + 2 + 1 = 63$.

Un exemple de structure codée sur 6 bits dans le MBR est le nombre de **secteurs** (63 au maximum).

* **une structure codée sur 8 bits** aura une valeur maximale de 1111 1111 :

$1111\ 1111 = 2^7 + 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0 = 128 + 64 + 32 + 16 + 8 + 4 + 2 + 1 = 255$.

Un exemple de structure codée sur 8 bits dans le MBR est le nombre de **têtes** (255 au maximum).

* **une structure codée sur 10 bits** aura une valeur maximale de 11 1111 1111 :

$11\ 1111\ 1111 = 2^9 + 2^8 + 2^7 + 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0 = 512 + 256 + 128 + 64 + 32 + 16 + 8 + 4 + 2 + 1 = 1023$.

Un exemple de structure codée sur 10 bits dans le MBR est le nombre de **cylindres** (1023 au maximum).

la conversion d'un nombre hexadécimal (base 16) en décimal (base 10) est à la fois simple (car logique) et compliquée (car on n'a pas l'habitude de raisonner en base 16) :

le nombre 3258 en décimal peut s'écrire ainsi : $3 \times 10^3 + 2 \times 10^2 + 5 \times 10^1 + 8 \times 10^0 = 3 \times 1000 + 2 \times 100 + 5 \times 10 + 8 \times 1$.

Le même nombre **3258 en hexadécimal** (qu'on écrit **3258h**) peut se convertir ainsi en décimal :

$3 \times 16^3 + 2 \times 16^2 + 5 \times 16^1 + 8 \times 16^0 = 3 \times 4096 + 2 \times 256 + 5 \times 16 + 8 \times 1 = 12288 + 512 + 80 + 8 = 12888$.

S'il y a des lettres dans le nombre en hexadécimal, il suffit d'**attribuer à la lettre le chiffre correspondant** (A=10, B=11, C=12, D=13, E=14, F=15) : ainsi un nombre total de secteur de **0B73E04F** (en hexadécimal) dans un descripteur de partition signifie qu'il y a :

$0 \times 16^7 + B \times 16^6 + 7 \times 16^5 + 3 \times 16^4 + E \times 16^3 + 0 \times 16^2 + 4 \times 16^1 + F \times 16^0 =$

$0 \times 16^7 + 11 \times 16^6 + 7 \times 16^5 + 3 \times 16^4 + 14 \times 16^3 + 0 \times 16^2 + 4 \times 16^1 + 15 \times 16^0 =$

$0 + 11 \times 16777216 + 7 \times 1048576 + 3 \times 65536 + 14 \times 4096 + 0 + 4 \times 16 + 15 \times 1 =$

$0 + 184549376 + 7340032 + 196608 + 57344 + 0 + 64 + 15 =$

192 143 439 secteurs = 192 143 439 x 512 octets = **96 Go**

Pour mémoire voici les principales correspondances :

Bits	Bytes (octets)	Words	Nombre hexadécimal maximal	Nombre décimal maximal	Exemples d'utilisation
6			3F	63	Nb de <u>secteurs</u> en CHS
8	1		FF	255	Nb de <u>têtes</u> en CHS
10			3FF	1023	Nb de <u>cylindres</u> en CHS
16	2	1	FFFF	65 535	"Mot" hexadécimal (Word)
24	3		FF FFFF	16 777 215	
28			FFF FFFF	268 435 456	Limite des BIOS classique (*)
32	4	2	FFFF FFFF	4 294 967 295	"double word" en hexa
48	6	3	FFFF FFFF FFFF	281 474 976 710 656	Limite des nouveaux BIOS (**)
64	8	4	FFFF FFFF FFFF FFFF	...	

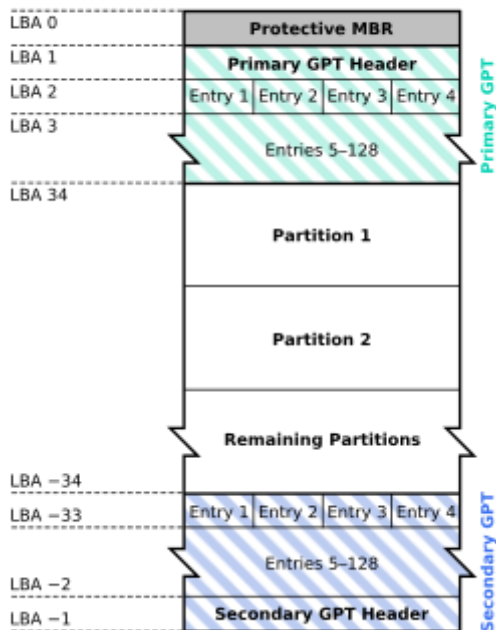
(*) : sur la plupart des ordinateurs actuels, le comptage des secteurs des DD dans le BIOS est codé sur 28-bits, ce qui explique que les BIOS ne reconnaissent pas les DD > 137 Go (268 435 456 secteurs x 512 octets = 137 Go), et au-delà il faut aller bidouiller la BdR pour activer le mode LBA-48 bits...

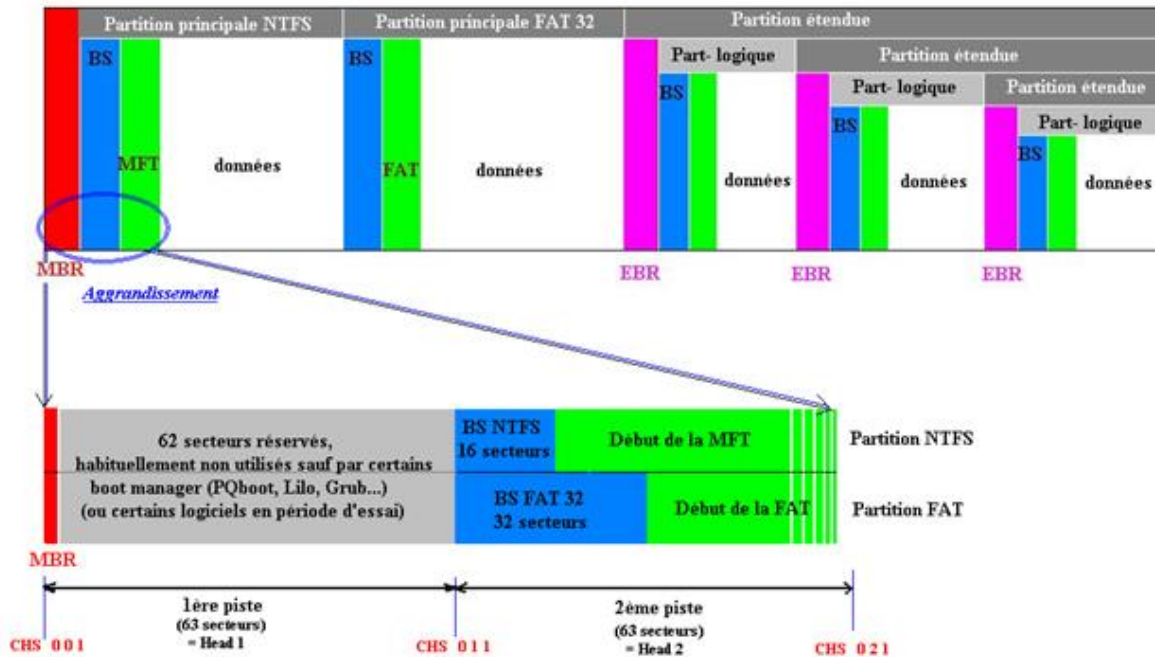
(**) : sur les BIOS les plus récents, le comptage des secteurs se fait sur 48-bits, ce qui permet la reconnaissance de DD de 281 474 976 710 656 secteurs = 144 115 188 Go = 144 PetaOctets (y'a de quoi faire !...)

La table de partitionnement

Une **table de partitionnement** est stockée sur un disque dur et contient les informations nécessaires pour **diviser un disque dur en partitions**. Il existe plusieurs modèles de table de partitionnement comme celle contenue dans le **MBR** (Master Boot Record) ou le **GPT** (GUID Partition Table).

GUID Partition Table Scheme





Un secteur de boot est situé au début de chaque partition. Il est créé lors du formatage de la partition et contient les caractéristiques de celle-ci ainsi que les coordonnées de la table d'allocation des fichiers (MFT ou FAT). Il comporte un nombre de secteur variable en fonction du système de fichier utilisé : 16 secteurs en cas de partition NTFS, 32 secteurs en cas de partition FAT 32. Une copie de sauvegarde du secteur de boot existe au sein de chaque partition. A noter qu'il n'y a pas de secteur de boot dans le MBR ni dans les EBR. D'autre part les secteurs de boot des partitions commencent toujours sur le 1er secteur de la 2ème piste d'un cylindre quelconque et leurs coordonnées CHS se présentent donc toujours sous la forme Z 1 1 (où Z est un numéro de cylindre quelconque dépendant de la taille des partitions qui précèdent)

une table d'allocation des fichiers enfin existe pour chaque partition, appelée **FAT (File Allocation Table)** pour les partitions FAT32, et **MFT (Master File Table)** pour les partitions NTFS. Ce n'est ni plus ni moins qu'un répertoire permettant de retrouver tous les fichiers stockés dans la partition. Si ce répertoire est effacé windows ne peut plus retrouver aucune des données stockées dans la partition : elles sont bien présentes mais inaccessibles (sauf pour certains logiciels de récupération de données)

DISQUE MBR

Depuis que le **BIOS** existe, nos disques durs utilisent la **table de partitions MBR** (Master Boot Record).

Le **MBR** (Master Boot Record) est le **premier secteur d'un disque dur**.

Si la liste de démarrage de votre PC commence par un disque dur, le **BIOS** va donc exécuter le **MBR**(Master Boot Record) de ce disque dur. Le **MBR** contient la **table de partitions** et une **routine d'amorçage** permettant de charger le système d'exploitation ou le chargeur d'amorçage (bootloader).

Le MBR (Master Boot Record) ou secteur principal de démarrage, est situé sur le **1er secteur** de chaque disque dur (1er secteur de la 1ère piste du 1er cylindre = CHS 0 0 1). Il y en a un seul par disque dur et il est créé lorsque le disque est initialisé dans la gestion des disques. Il contient les coordonnées CHS des secteurs de boot de chaque partition principale et sait reconnaître parmi celles ci laquelle est active et contient l'OS pour pouvoir lancer le système (OS = Operating System, c'est-à-dire windows XP par exemple...). La table des partitions qu'il renferme est écrite lorsque le disque dur est partitionné, et est modifiée chaque fois qu'on réajuste les partitions principales (suppression, fusion,...)

Le MBR comprend 3 grandes parties

Le code exécutable qui est lancé par le BIOS, la zone des messages d'erreur et la table de partition.

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
00000000	33	C0	8E	D0	BC	00	7C	FE	50	07	50	1F	FC	EE	1B	7C	À Z D W ú p P ú W
00000010	BF	1B	06	50	57	B9	E5	01	F3	A4	CE	ED	BE	07	B1	04	¿ . . P W ^ Á ó H H W ±
00000020	38	6E	00	7C	09	75	13	83	C5	10	E2	F4	CD	18	8B	F5	8 n u f Á á ó í < ó
00000030	83	C6	10	49	74	19	38	2C	74	F6	A0	B5	07	E4	07	8B	f E I t 8 , t ó u ' <
00000040	F0	AC	3C	00	74	FC	EB	07	00	E4	0E	CD	10	EB	F2	88	ä - < t ú » ' í e ó ^
00000050	4E	10	E8	46	00	73	2A	FE	46	10	80	7E	04	0B	74	0B	N è F s * b F e ~ t
00000060	80	7E	04	0C	74	05	A0	B6	07	75	D2	80	46	02	06	83	e ~ t t Q u ó E F f
00000070	46	08	06	83	56	0A	00	E8	21	00	73	05	A0	B6	07	EB	F f v . è ! s Q è
00000080	BC	81	3E	FE	7D	55	AA	74	0B	80	7E	10	00	74	C8	A0	W D > b } U ^ t e ~ t È
00000090	B7	07	EB	A9	8B	FC	1E	57	8B	F5	CB	BF	05	00	8A	56	' . e @ < ú W < ó È ; Š V
000000A0	00	E4	08	CD	13	72	23	8A	C1	24	3F	98	8A	DE	8A	FC	' í r # Š Á ? ~ Š Š Ů
000000B0	43	F7	E3	8E	D1	86	D6	E1	06	D2	EE	42	F7	E2	39	56	C + ä < N + ó ± 0 i B + á 9 V
000000C0	0A	77	23	72	05	39	46	08	73	1C	B8	01	02	BB	00	7C	w x r 9 F s . . .
000000D0	8E	4E	02	8B	56	00	CD	13	73	51	4F	74	4E	32	E4	8A	< N < V . í s Q ó t N 2 á Š
000000E0	56	00	CD	13	EB	E4	8A	56	00	60	BB	AA	55	E4	41	CD	V í . e ä Š V ' » U ' Á í
000000F0	13	72	36	81	FE	55	AA	75	30	F6	C1	01	74	2B	61	60	r ó ú U ^ u ó é Á t + á
00000100	6A	00	6A	00	FF	76	0A	FF	76	08	6A	00	68	00	7C	6A	j j j ŷ v ŷ v j h l j
00000110	01	6A	10	E4	42	8B	F4	CD	13	61	61	73	0E	4F	74	0E	j ' B < ó í a a s ó t
00000120	32	E4	8A	56	00	CD	13	EB	D6	61	F9	C3	54	61	62	6C	2 á Š V í . e ó u á Á Tabl
00000130	65	20	64	65	20	70	61	72	74	69	74	69	6F	6E	20	6E	e d e p a r t i t i o n n
00000140	6F	6E	20	76	61	6C	69	64	65	00	45	72	72	65	75	72	o n v a l i d e E r r e u r
00000150	20	6C	6F	72	73	20	64	75	20	63	68	61	72	67	65	6D	l o r s d u c h a r g e m
00000160	65	6E	74	20	64	75	20	73	79	73	74	8A	6D	65	20	64	e n t d u s y s t é m e d
00000170	27	65	78	70	6C	6F	69	74	61	74	69	00	53	79	73	74	' e x p l o i t a t i S y s t
00000180	8A	6D	65	20	64	27	65	78	70	6C	6F	69	74	61	74	69	Š m e d ' e x p l o i t a t i
00000190	6F	6E	20	61	62	73	65	6E	74	00	00	00	00	00	00	00	o n a b s e n t
000001A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00 J 8 < 9 < . e
000001B0	00	00	00	00	00	2C	4A	7C	38	3C	39	3C	00	00	80	01 b ŷ ŷ ? ' ^ 8 ' .
000001C0	01	00	07	FE	FF	FF	3F	00	00	00	5E	38	27	03	00	00 Á ŷ b ŷ ŷ □ 8 ' ŷ ŷ Ń . .
000001D0	C1	FF	0F	FE	FF	FF	9D	38	27	03	24	FF	D1	0A	00	00 U ^
000001E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
000001F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	

- « BS » BOOT SECTOR Les 300 premiers octets du MBR en bleu (de 00h à 12Bh) correspondent au code exécutable qui démarre le lancement du disque après que le BIOS ait passé en revue les différents éléments matériels de l'ordinateur (ce qu'on appelle le POST = Power-On Self Test = auto-test de démarrage). (Les 2 derniers octets du MBR (en bleu) correspondent au nombre magique 55AA appelé aussi "marque de secteur exécutable". C'est une signature qu'il est impératif de trouver à la fin du MBR puisque c'est cette marque que va rechercher le BIOS et qui va lui permettre de reconnaître le MBR pour pouvoir lancer son code exécutable).
- Les 80 à 110 octets suivants en vert (de 12Ch à 198h dans mon exemple, nombre variable selon la langue utilisée) correspondent aux messages d'erreur que lance le MBR en cas de problème de configuration
- Les 4 octets en jaune (de 1B8h à 1BBh) correspondent à la signature disque que Windows marque dans le MBR à l'installation du disque mais que l'on retrouve également dans différentes clés de la base de registre. Ces clés de registre utilisant la signature disque dans leur valeur contiennent de nombreuses autres informations concernant notamment la table de partition et il y a là possiblement une des causes du passage en RAW de certaines partitions (par corruption de la BdR, ou en tout cas inadéquation entre les valeurs de ces clés et les données du MBR)

```

+ 1&30a96598&0&Signature3C393C38Offset17507C9A00Length23D9BA600
+ 1&30a96598&0&Signature3C393C38Offset198E18BE00Length26456C400
+ 1&30a96598&0&Signature3C393C38Offset64E71B800Length2F9612400
+ 1&30a96598&0&Signature3C393C38Offset7E00Length64E70BC00
+ 1&30a96598&0&Signature3C393C38Offset947D35A00LengthE08A8C200

```

<i>Valeur de Parent</i>	<i>Prefix</i>	<i>Signature</i>	<i>Nombre d'octet</i>	<i>Taille exacte</i>
<i>que l'on trouve dans :</i>				
<i>HKEY_LOCAL_MACHINE\</i>		<i>disque</i>	<i>en hexadécimal</i>	<i>en octet de la</i>
<i>SYSTEM\CurrentControlSet</i>			<i>avant le début</i>	<i>partition</i>
<i>\Enum\Root\fdisk\0000</i>			<i>de la partition</i>	

Exemples de clés montrant la signature disque dans :
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet
\Enum\STORAGE\Volume

- Les **64 octets** en rose (de 1BEh à 1FDh) correspondent à la table de partition, centre névralgique du MBR. Cette table comporte seulement 4 descripteurs de partition de 16 octets (4 x 16 = 64) et il ne peut donc pas y avoir plus de 4 partitions principales sur un disque dur. Pour contourner le problème on a inventé les partitions étendues dont le principe consiste, à partir d'une partition étendue mère dont le descripteur est présent dans le MBR, à créer des entrées disséminées sur le DD (les EBR) pour des partitions étendues filles, chaque partition étendue contenant une partition logique et une autre partition étendue...

Le MBR est compatible avec tous les systèmes d'exploitation 32bits et 64 bits. Le MBR possède néanmoins des limitations, devenues de sérieux inconvénients à l'heure d'aujourd'hui :

- 4 partitions maximum
- Taille d'une partition limitée à 2,2 To (2 200Go)
- Impossible de démarrer un disque dur MBR avec un système UEFI

Les limitations actuelles du **MBR** ont poussé les fabricants à se tourner vers une **nouvelle table de partitionnement** plus performante, introduite par Intel : le **GPT**. Le GPT est d'ailleurs le **standard de l'UEFI**, le remplaçant du BIOS.

Quelles versions de Windows peuvent lire, écrire et booter sur un disque dur MBR ?

Toutes les versions de Windows !

Peut-on démarrer sur un disque dur MBR avec un système UEFI ?

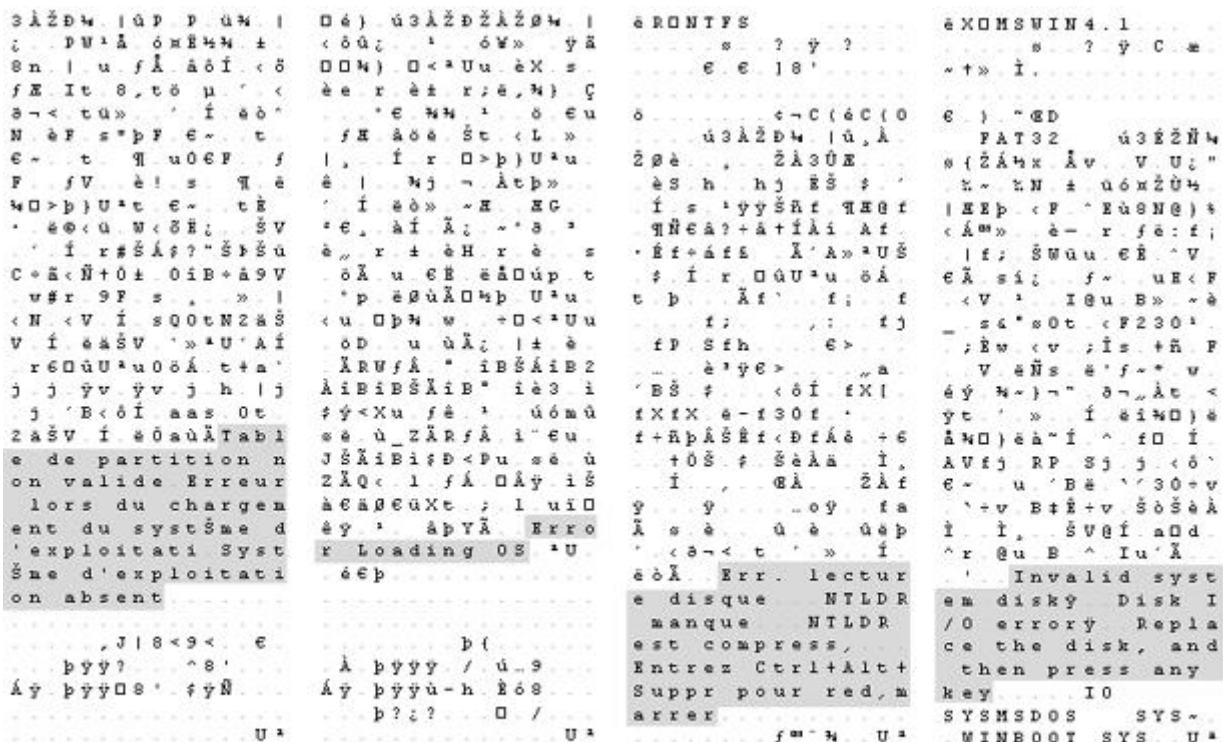
Non, un système UEFI ne peut démarrer que sur un disque dur utilisant la table de partitions GPT.

Un système d'exploitation UEFI peut-il lire et écrire sur un disque dur MBR ?

Oui, une version UEFI de Windows reconnaît parfaitement un disque dur MBR, vous pouvez modifier ou supprimer vos fichiers comme d'habitude.

Compléments

A chaque étape cependant un plantage peut empêcher le lancement de l'OS et le message d'erreur qui va s'afficher, inscrit dans le MBR ou le secteur de boot, dépend entièrement du moment précis où ça plante. **Connaissant cela on comprend mieux les messages d'erreur susceptibles de s'afficher en fonction de la panne ainsi que les solutions envisageables** (voir utilisation d'EditHexa dans ma signature) :



MBR de disque système (disque physique N°1)

MBR de disque de données (disque physique N°2)

Secteur de boot de partition NTFS

Secteur de boot de partition FAT 32

* **si le code exécutable du MBR est endommagé** on va se retrouver face à un écran noir sans aucun message d'erreur : en effet c'est le code exécutable qui entraîne l'affichage des messages d'erreur qui suivent (et donc sans code exécutable, pas de message d'erreur). **La solution consiste dans ce cas à faire un fixmbr sous la console de récupération pour restaurer ce secteur exécutable.**

* **si la marque de secteur exécutable est corrompue (le 55AA à la fin du MBR)**, on va voir s'afficher un message du genre "operating system not found" signifiant que le BIOS ne trouve pas de secteur exécutable. **Là encore un fixmbr devrait solutionner le problème.**

* **si la partition système n'est pas activée (corruption de la table de partition avec un 00 au lieu de 80 à l'offset 01BEh)**, on va voir le même message "operating system not found" : cela signifie que le chargeur ne trouve pas la partition système à laquelle il doit donner la main. Dans ce cas le fixmbr n'aura aucune efficacité car il n'agit pas sur la table de partition, et seul PTEDIT (ou un éditeur hexadécimal) peut permettre de diagnostiquer et résoudre le problème (d'autres utilitaires genre fdisk pouvant également permettre d'activer telle ou telle partition).

* **si à contrario une partition non système (de données par exemple) est activée par erreur (code 80 au lieu de 00, là encore par corruption de la table de partition)**, l'exécutable ayant trouvé le 80 va essayer de booter sur celle-ci, va échouer (puisqu'il n'y a pas de système d'exploitation sur cette partition), va rechercher un autre périphérique pour essayer de booter (floppy ou lecteur CD), et s'il ne trouve rien sur ces autres emplacements il va afficher le message d'erreur suivant : **"Non-system disk or disk error, Replace and press any key when ready"**. là aussi seuls PTEDIT, fdisk ou un éditeur hexadécimal peuvent résoudre le problème.

* **si le descripteur de la partition système est corrompu (ligne 1C0 remplie de 00 par exemple) à l'exception du 80** qui est sur la ligne au dessus, l'exécutable ayant trouvé le 80 va quand même essayer de booter sur cette partition, mais bien sûr il ne la trouvera pas et va afficher le message d'erreur suivant : **"missing operating system"**.

* **petite digression : comment agit un virus touchant le MBR et pourquoi un fixmbr est-il dangereux dans ce cas ?**

En fait le virus va s'installer à la place du MBR sur le 1er secteur du DD et transférer le secteur original du MBR à un autre endroit du disque dur (souvent le dernier secteur). Au démarrage, lorsque le BIOS va vouloir activer le MBR, il va en réalité activer le virus et charger celui-ci en mémoire, mais le virus va charger à son tour le MBR original si bien que le processus de démarrage va sembler normal. Si l'on fait un fixmbr, le virus va être (en partie) détruit, mais le véritable MBR ayant été déplacé il ne pourra plus être chargé en mémoire et sa table de partition ne sera plus du tout accessible, d'où plantage irrémédiable... certains virus de secteurs de boot se comportent exactement de la même manière, et dans ce cas c'est la commande fixboot (et non plus fixmbr) qui va tout planter... Il faut savoir également que certains virus infectant le MBR peuvent effacer la marque de secteur exécutable 55 AA : or dans ce cas le fixmbr fait encore plus de dégâts car s'il ne trouve pas de 55 AA il va effacer complètement la table de partition en mettant des 00 partout... donc prudence quand même avec cette commande.

DISQUE GPT (GUID Partition Table)

L'**UEFI**, le remplaçant du BIOS, change la donne et exige l'utilisation d'une nouvelle **table de partitionnement** : le **GPT** (GUID Partition Table).

Le **GPT** (GUID Partition Table) est un nouveau standard pour décrire la **table de partitionnement d'un disque dur**, il est amené à remplacer le **MBR** à cause des limitations de ce dernier qui limite la taille d'une partition à 2,2 To.

Le **GPT** fait partie du standard **UEFI**, c'est-à-dire qu'un **système d'exploitation UEFI** ne peut démarrer que sur un disque dur utilisant une **table de partitions GPT**. De manière globale, un BIOS traditionnel ne peut démarrer sur un disque dur GPT. Mais le GPT est aussi utilisé par certains BIOS à cause des limitations de la **table de partitions MBR**.

Disque dur 16hexa octets max

Quelle est le nombre limite de partitions sur un disque dur GPT ?

128 partitions maximum sur Windows.

Quelle est la taille limite d'une partition sur un disque dur GPT ?

256 To par partition sur Windows.

Quelles versions de Windows peuvent lire le contenu d'un disque dur GPT ?

Toutes les versions 32 bits et 64 bits de Windows Vista, Windows 7, Windows 8 et la version 64 bits de Windows XP.

Quelles sont les versions de Windows qui peuvent démarrer sur un disque dur GPT ?

Pour booter sur un disque dur GPT, vous devez activer l'UEFI dans les paramètres de votre carte mère et avoir une version 64 bits de Windows Vista SP1, Windows 7 ou Windows 8. Aucune version 32 bits de Windows ne peut démarrer sur un disque dur GPT. Windows XP 64 bits ne peut pas non plus booter sur un disque dur GPT.

Que voit une version de Windows qui ne supporte pas les disques durs GPT ?

Un Windows ne supportant pas le GPT comme Windows XP 32 bits verra une unique partition *Protective MBR*. Cette partition ne sera pas montée par Windows et vous ne pourrez pas accéder à son contenu.

Peut-on avoir à la fois un disque dur MBR et GPT sur un même système ?

Oui, vous devez pour cela posséder un Windows supportant le format GPT et démarrez votre ordinateur sur un disque dur GPT.

Peut-on convertir un disque dur MBR en GPT et inversement ?

Oui, mais il faut que votre disque dur soit vide : il faut supprimer toutes les données et toutes les partitions de votre disque dur. Seulement à partir de là, vous pourrez convertir votre disque dur GPT en MBR et inversement.

PARTITION NTFS

Cluster pour disque dur de plus de 4Go = 4 Ko (au lieu de 32 ko pour fat 32)- si 1 fichier pas cluster limite $2^{32}-1$ fichiers

voir la taille hdd (16 To) supérieure à la taille maximale de disque (2 To).

Cela est dû aux limitations supplémentaires sur la taille d'un disque (bits réservés dans l'adressage d'un secteur,...).

La taille **théorique** maximale d'un disque **NTFS** est non pas de 2 To mais de 2^{64} soit **16777216 To** !

en ntfs pour un fichier il y a une option dans la table des fichiers qui est « la table acl » qui définit les droits d'accès et il n'y a pas sur fat32

Sur une partition NTFS le secteur de boot occupe en théorie **16 secteurs**, et c'est effectivement ce qu'XP réserve au secteur de boot lors de la création de la partition et ce sont bien 16 secteurs qui sont chargés en mémoire par le MBR lorsque cette partition est appelée.

Mais en réalité il faut distinguer 3 parties au sein de ces 16 secteurs :

- **tout d'abord le 1er secteur** qui correspond au secteur de boot proprement dit, le seul qui soit vital pour la partition et que je détaille ci-dessous. A noter qu'il existe une sauvegarde de ce secteur de boot située sur le dernier secteur de la partition, ce qui est primordial en cas de gros plantage ... " .
- **les 6 secteurs suivants** correspondent à un code exécutable de boot (bootstrap code) qui est le même dans toutes les partitions NTFS mais qui n'a d'utilité qu'en cas de partition système : en effet ce "bootstrap code" (N.T.L.D.R secteurs) permet, dans une partition système, le lancement du système d'exploitation (et sa corruption est une cause de plantage), mais à contrario il ne joue aucun rôle dans le cas d'une partition de données : on peut effacer ces 6 secteurs et les remplir de 00, les données seront toujours accessibles de la même manière, aussi bien en lecture qu'en écriture.
- **Les 9 derniers secteurs** enfin sont habituellement remplis de 00 et n'ont pas d'utilité clairement définie (peut être réservés à d'éventuelles évolutions futures).

1er secteur de la partition, le secteur de boot proprement dit. On y retrouve, comme dans le MBR, différentes parties :

- d'abord le **code exécutable d'assemblage** de 303 octets (de 054h à 182h) qui va explorer le BPB afin d'analyser les caractéristiques physiques de la partition et ensuite charger en mémoire, dans le cas d'une partition système, le "bootstrap code" (NTLDR) situé sur les 6 secteurs suivants et qui est essentiel au lancement du système d'exploitation. A noter qu'il existe 3 octets importants au début du secteur de boot (EB 52 90) qui contiennent une instruction de saut (jump) vers le code exécutable d'assemblage : or ce "jump" est le premier élément chargé en

mémoire par le MBR et sa corruption empêcherait le chargement du code exécutable d'assemblage (donc pas d'analyse du secteur de boot et pas de chargement du bootstrap code...)

- les 110 à 120 octets (de 183h à 1Eah, variables selon la langue utilisée) correspondent aux **messages d'erreur** diffusés par le code exécutable d'assemblage lorsque le "bootstrap code" occupant les 6 secteurs suivants ne peut être lancé. à noter qu'on retrouve là le classique "**NTLDR manquant**" ou encore "**entrez Alt + Ctrl + Suppr pour redémarrer**", et qu'une des solutions consiste donc à restaurer ces 6 secteurs.

- 4 octets, comme dans le MBR, aux **Offsets des messages d'erreur** et qui permettent au code exécutable de trouver ces messages.

- on retrouve également, en fin de secteur, la marque de secteur exécutable 55 AA (indispensable)

- les 84 premiers octets (de 00h à 53h) que l'on appelle le **BPB (bios Parameter Block)** qui détaille précisément les caractéristiques de la partition (codées sur 1 à 8 octets chacune) que l'on retrouve sous forme de tableau dans PTEDIT (en cliquant sur "boot record") ou dans les éditeurs hexadécimaux :

Structure du secteur de boot			
Adresse	Intitulé	Valeur	
0x00	Jump (hex)	EB 52 90	<i>instruction de saut vers le code exécutable à l'Offset 54h</i>
0x03	Nom du fabricant	NTFS	<i>nom du système de fichier codé sur 3 octets</i>
0x0B	Bytes par secteur	512	<i>512 octets par secteurs</i>
0x0D	Nombre de secteurs par cluster	8	<i>chaque cluster est composé de 8 secteurs (habituel)</i>
0x0E	Nombre de secteurs réservés	0	<i>toujours à 0 pour NTFS</i>
0x10	Nombre de FAT	0	<i>toujours à 0 pour NTFS</i>
0x11	Nombre total d'entrées dans le répertoire racine	0	<i>toujours à 0 pour NTFS</i>
0x13	Pas utilisé sous NTFS	0	<i>toujours à 0 pour NTFS</i>
0x15	Type de media (hex)	F8	<i>toujours F8 (non utilisé par Win2K ou XP)</i>
0x16	Nombre de secteurs par FAT	0	<i>toujours à 0 pour NTFS</i>
0x18	Nombre de secteurs par piste	63	<i>nombre de secteurs par piste</i>
0x1A	Nombre de têtes	255	<i>nombre de pistes par cylindre</i>
0x1C	Secteurs cachés	63	<i>toujours 63 si partition logique ou 1ère part- principale</i>
0x20	Pas utilisé sous NTFS	0	<i>toujours à 0 pour NTFS</i>
0x24	Pas utilisé sous NTFS	8388736	<i>toujours cette valeur (équivalent décimal de 30003000)</i>
0x28	Nombre total de secteurs NTFS	117724256	<i>longueur de la partition en secteurs +++</i>
0x30	Cluster de départ de la Master File Table(MFT)	786432	<i>départ de la MFT (1er cluster) dans cette partition</i>
0x38	Miroir de la MFT	16	<i>départ (1er cluster) de la sauvegarde de la MFT...</i>
0x40	Clusters par 'File record Segment'	246	<i>nb de cluster par segment d'enregistrement (tjs 246)</i>
0x44	Clusters par 'Index Block'	1	<i>toujours à 1 (nombre de cluster par répertoire)</i>
0x48	Numéro de série du volume (hex)	B2B860E2B860A697	<i>numéro attribué par XP à la partition</i>
0x50	Checksum	0	<i>toujours à 0 pour NTFS</i>
0x1FE	Signature du secteur de boot (hex)	AA55	<i>marque de secteur exécutable (en fin de secteur)</i>

le BPB donne également des **informations importantes sur la MFT** (Master File Table) que NTLDR utilise pour localiser cette MFT pendant le processus de démarrage. Or ceci est primordial, car contrairement aux partitions en FAT16 ou FAT32, la MFT n'est pas localisée à un endroit prédéfini de la partition NTFS : l'avantage est qu'en cas de secteur défectueux Windows va déplacer la MFT à un autre endroit de la partition (en corrigeant le BPB) ce qui accroît la souplesse et la sécurité du système, mais l'inconvénient est qu'en cas de corruption du BPB, la MFT pouvant se trouver n'importe où sur la partition elle ne pourra plus être localisée par NTLDR et Windows renverra un message d'erreur disant que la partition n'est pas formatée (probablement la cause principale des passages en RAW...). **La corruption de ce secteur de boot explique donc de nombreuses situations de partitions disparues...**

PARTITION FAT32

Sur une partition FAT32 le "secteur" de boot s'étend sur **32 secteurs** et la FAT commence invariablement au 33ème secteur. Ces 32 secteurs sont répartis ainsi :

- **3 secteurs réservés au secteur de boot** proprement dit (secteurs logiques 0 à 2)
- **3 secteurs non utilisés remplis de 00** (secteurs logiques 3 à 5)
- **3 secteurs occupés par une copie du secteur de boot** = backup boot sector (secteurs logiques 6 à 8)
- **23 secteurs réservés par l'OS mais qui ne sont jamais utilisés et en fait remplis de 00...**

CAPACITE MAX : 2 To (CLUSTER 32 Ko) FICHIER MAX 4Go (2²²-1 nombre de blocs appelés aussi secteurs)

4 partitions primaire et autant de lecteurs logique dans la partition étendue (nombre étendue selon nombre primaire

Les 3 premiers secteurs correspondant au secteur de boot proprement dit et d'abord

le 1er secteur (qui est assez similaire au secteur de boot d'une partition NTFS) :

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F												
00000000	EB	58	90	4D	53	44	4F	53	35	2E	30	00	02	20	20	00	èX	MSDOS5.0	èX	MSWIN4.1	...	#						
00000010	02	00	00	00	F8	00	00	3F	00	FF	00	3F	00	00	00	00	è	*	?	ÿ	?	?						
00000020	77	05	42	02	1C	24	00	00	00	00	00	00	02	00	00	00	w	B	?	?	?	?						
00000030	01	00	06	00	00	00	00	00	00	00	00	00	00	00	00	00												
00000040	80	00	29	83	AB	73	EC	4E	4F	20	4E	41	4D	45	20	20	e)	f	«	s	i	N	O	N	A	M	E
00000050	20	20	46	41	54	33	32	20	20	20	33	C9	8E	D1	BC	F4	F	A	T	3	2	?	?	?	?	?	?	
00000060	7B	8E	C1	8E	D9	ED	00	7C	88	4E	02	8A	56	40	B4	08	{	?	?	?	?	?	?	?	?	?	?	
00000070	CD	13	73	05	B9	FF	FF	8A	F1	66	0F	B6	C6	40	66	0F	í	.	?	?	?	?	?	?	?	?	?	
00000080	B6	D1	80	E2	3F	F7	E2	86	CD	C0	ED	06	41	66	0F	B7	T	?	?	?	?	?	?	?	?	?	?	
00000090	C9	66	F7	E1	66	89	46	F8	83	7E	16	00	75	38	83	7E	É	f	+	á	f	?	?	?	?	?	?	
000000A0	2A	00	77	32	66	8B	46	1C	66	83	C0	0C	BB	00	80	B9	*	w	?	?	?	?	?	?	?	?	?	
000000B0	01	00	E8	2B	00	E9	48	03	A0	FA	7D	B4	7D	8B	F0	AC	à	+	é	?	?	?	?	?	?	?	?	
000000C0	84	C0	74	17	3C	FF	74	09	B4	0E	EB	07	00	CD	10	EB	À	t	<	?	?	?	?	?	?	?	?	
000000D0	EE	A0	FB	7D	EB	E5	A0	F9	7D	EB	E0	98	CD	16	CD	19	í	?	?	?	?	?	?	?	?	?	?	
000000E0	66	60	66	3E	46	F8	0F	82	4A	00	66	6A	00	66	50	06	f	?	?	?	?	?	?	?	?	?	?	
000000F0	53	66	68	10	00	01	00	80	7E	02	00	0F	85	20	00	B4	S	f	?	?	?	?	?	?	?	?	?	
00000100	41	EB	AA	55	8A	56	40	CD	13	0F	82	1C	00	81	FB	55	A	*	U	S	V	?	?	?	?	?	?	
00000110	AA	0F	85	14	00	F6	C1	01	0F	84	0D	00	FE	46	02	B4	*	?	?	?	?	?	?	?	?	?	?	
00000120	42	8A	56	40	8B	F4	CD	13	B0	F9	66	58	66	58	66	58	B	S	V	?	?	?	?	?	?	?	?	
00000130	66	58	EB	2A	66	33	D2	66	0F	B7	4E	18	66	F7	F1	FE	r	X	*	f	3	0	f	.	N	f	+	
00000140	C2	8A	CA	66	8B	D0	66	C1	EA	10	F7	76	1A	86	D6	8A	À	S	?	?	?	?	?	?	?	?	?	
00000150	56	40	8A	EB	C0	E4	06	0A	CC	B8	01	02	CD	13	66	61	V	?	?	?	?	?	?	?	?	?	?	
00000160	0F	82	54	FF	81	C3	00	02	66	40	49	0F	85	71	FF	C3	T	?	?	?	?	?	?	?	?	?	?	
00000170	4E	54	4C	44	52	20	20	20	20	20	20	00	00	00	00	00	N	T	L	D	R							
00000180	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00												
00000190	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00												
000001A0	00	00	00	00	00	00	00	00	00	00	00	00	0D	0A	52	65												
000001B0	74	69	72	65	7A	20	6C	65	73	20	64	69	73	71	75	65	t	?	?	?	?	?	?	?	?	?	?	
000001C0	73	FF	0D	0A	45	72	72	2E	20	64	69	73	71	75	65	FF	s	?	?	?	?	?	?	?	?	?	?	
000001D0	0D	0A	50	72	65	73	73	65	7A	20	75	6E	65	20	74	6F												
000001E0	75	63	68	65	20	70	6F	75	72	20	72	65	64	82	6D	61	u	?	?	?	?	?	?	?	?	?	?	
000001F0	72	72	65	72	0D	0A	00	00	00	00	AC	C2	DO	00	00	55	A	A										

* d'abord une instruction de saut sur les 3 premiers octets (EB 58 90) appelant en mémoire le code exécutable d'assemblage de 278 octets (en vert, offsets 05Ah à 16Fh), qui va lui-même charger en mémoire l'exécutable qui se trouve sur le 3ème secteur (voir plus bas) qui contient la plupart des routines permettant au secteur de boot de lire les entrées de la table d'allocation des fichiers (FAT), ce qui va permettre à windows de charger en mémoire les différents fichiers système nécessaires au lancement de l'OS et dont on trouve des références dans ce 1er secteur : NTLDR en gris sur cette copie d'écran (partition formatée sous XP en utilisant MSDOS5.0), ou IO.SYS si le système de formatage utilise MSWIN4.1 comme indiqué ci-dessus à droite (je n'ai représenté ici que la partie texte de l'éditeur)

* on voit ensuite les messages d'erreurs sur 60 octets (en bleu, de 1ACh à 1E7h) avec 3 octets donnant les offsets des messages d'erreur (en jaune, de 1F9h à 1FBh). Là encore les messages d'erreur ne sont pas les mêmes selon le système DOS qui a formaté la partition

* marque de secteur exécutable 55 AA (présente à la fin de chacun des 3 secteurs constituant le secteur de boot de la partition FAT32)

* et enfin le BPB qui occupe 90 octets en début de secteur (en rose, de 00h à 59h) et que les éditeurs hexadécimaux (ainsi que PTEDIT) retranscrivent, comme pour les partitions NTFS, sous la forme d'un tableau :

Adresse	Intitulé	Valeur	
0x00	Jump (hex)	EB 58 90	Instruction de saut vers le code exécutable à l'offset 5Ah
0x03	Nom du fabricant	MSWIN4.1	Système de fichiers : MSWIN4.1 ou MSDOS5.0 selon logiciel de formatage
0x0B	Bytes par secteur	512	512 octets par secteur (exceptionnellement 1024, 2048 ou 4096)
0x0D	Nombre de secteurs par cluster	16	16 secteurs par cluster : varie en fonction de la taille de la partition (2 à 128)
0x0E	Nombre de secteurs réservés	36	Signifie que la FAT commence au 36ème secteurs (les 32 du secteur de boot + 4 ici)
0x10	Nombre de FAT	2	Nombre de Table d'Allocation des Fichiers : toujours 2 (la FAT et sa copie)
0x11	Nombre total d'entrées dans le répertoire racine	0	Toujours à 0 pour les partitions FAT32
0x13	Nombre total de secteurs	0	Toujours à 0 pour les partitions FAT32
0x15	Type de media (hex)	F8	Toujours F8 (non utilisé par Win2K ou XP)
0x16	Nombre de secteurs par FAT	0	Toujours à 0 pour les partitions FAT32
0x18	Nombre de secteurs par piste	63	nombre de secteurs par piste
0x1A	Nombre de têtes	255	nombre de pistes par cylindre
0x1C	Secteurs cachés	63	toujours 63 si partition logique ou 1ère partition principale
0x20	Nombre de secteurs dans le volume	28113672	nombre total de secteurs dans la partition
0x24	Nombre de secteurs par FAT32	13714	nombre de secteurs composant la FAT (la copie de la FAT commence ici à 13715)
0x28	Flag (mirroring)	0	FAT active
0x2A	Version système de fichier	0	habituellement à 0
0x2C	Premier cluster du répertoire racine	2	habituellement à 2
0x30	Numéro de secteur du fichier d'information système	1	Numéro du secteur de FSInfo (2ème secteur du secteur de boot)
0x32	Secteur de backup du boot	6	emplacement de la copie du secteur de boot (6ème secteur du secteur de boot)
0x34	Réservés	000000000000	Non utilisé
0x40	Numéro du disque physique (hex)	80	Identité du disque pour Windows
0x41	Réservé pour NT (hex)	00	non utilisé
0x42	Signature NT (hex)	29	Extension de signature : 29 signifie que les 3 champs suivants doivent être renseignés
0x43	Numéro de série du disque (hex)	1E6F16DC	Windows génère ce numéro lors du formatage en fonction de la date et de l'heure
0x47	Nom du volume	NO NAME	nom attribué à la partition lors du formatage
0x52	Type de partition	FAT32	Système de fichiers utilisé
0x1FE	Signature du secteur de boot (hex)	AA55	marque de secteur exécutable (présent en fin de secteur)

le 2ème secteur donne des informations à Windows quant à l'utilisation qui est faite de la partition en indiquant le nombre de clusters non utilisés dans la partition ainsi que le 1er cluster libre pour inscrire de nouvelles données. C'est ce qu'on appelle le **FSInfo (File System Information)** :

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
00000000	52	52	61	41	00	00	00	00	00	00	00	00	00	00	00	00	R R a A
00000010	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000020	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000030	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000040	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000050	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000060	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000070	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000080	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000090	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
000000A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
000000B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
000000C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
000000D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
000000E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
000000F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000100	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000110	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000120	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000130	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000140	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000150	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000160	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000170	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000180	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000190	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
000001A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
000001B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
000001C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
000001D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
000001E0	00	00	00	00	72	72	41	61	50	6F	04	00	2B	F9	00	00	r r A a P o + ù
000001F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	55	AA	U *

* Les 4 premiers octets en rouge (52 52 61 41 = RRaA) indiquent à Windows qu'il s'agit bien du secteur FSInfo

* les 4 octets en bleu situés entre 01E4h et 01E7h (72 72 41 61 = rrAa) indiquent à Windows que les informations dont il a besoin se trouvent juste après : - 4 octets en jaune qui indiquent le nombre d'octets libres disponibles sur la partition (50 6F 04 00 dans mon exemple, correspondant en décimal à clusters de libres)

- 4 octets en vert qui indiquent à Windows quel est le 1er cluster libre disponible pour écrire de nouvelles

données, ce qui permet au pilote de la FAT de ne pas perdre de temps à explorer tous les clusters à la recherche du 1er cluster disponible (2B F9 00 00 dans mon exemple, ce qui signifie que le 1er cluster utilisable est le numéro) Ces données sont mises à jour à chaque création ou suppression de fichiers

*** et enfin toujours la marque de secteur exécutable 55 AA à la fin**

Le 3ème secteur est occupé par le **code exécutable** chargé en mémoire par le 1er secteur et qui va permettre le lancement de Windows lorsqu'il s'agit d'une **partition système** (si par contre il s'agit d'une simple partition de données ce 3ème secteur ne joue probablement aucun rôle, comme c'est le cas pour le « bootstrap code » des partitions de données NTFS.)

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
00000000	FA	66	0F	B6	46	10	66	8B	4E	24	66	F7	E1	66	03	46	ú f q F f < N \$ f + á f F
00000010	1C	66	0F	B7	56	0E	66	03	C2	33	C9	66	89	46	FC	66	f . V f À 3 é f k F ú f
00000020	C7	46	F8	FF	FF	FF	FF	FA	66	8B	46	2C	66	83	F8	02	Ç F ø ý ý ý ý ú f < F, f f ø
00000030	0F	82	CF	FC	66	3D	F8	FF	FF	0F	0F	83	C5	FC	66	0F	, I ú f = ø ý ý . . f Å ú f
00000040	A4	C2	10	FB	52	50	FA	66	C1	E0	10	66	0F	AC	D0	10	 Å ú R P ú f Á à f - Ð
00000050	66	83	E8	02	66	0F	B6	5E	0D	8B	F3	66	F7	E3	66	03	f f è f q ^ < ó f + á f
00000060	46	FC	66	0F	A4	C2	10	FB	BB	00	07	8B	FB	B9	01	00	F ú f  Å ú » < ú ' .
00000070	E8	BE	FC	0F	82	AA	FC	38	2D	74	1E	B1	0B	56	BE	D8	è  ü ,  ü 8 - t ± V  0
00000080	7D	F3	A6	5E	74	19	03	F9	83	C7	15	3B	FB	72	E8	4E) ó ! ^ t . ú f Ç ; ú r è N
00000090	75	D6	58	5A	E8	66	00	72	AB	83	C4	04	E9	64	FC	83	u Ö X Z è f r « f Å é d ú f
000000A0	C4	04	8B	75	09	8B	7D	0F	8B	C6	FA	66	C1	E0	10	8B	Ä < u < } < E ú f Á à <
000000B0	C7	66	83	F8	02	72	3B	66	3D	F8	FF	FF	0F	73	33	66	Ç f f ø r ; f = ø ý ý . s 3 f
000000C0	48	66	48	66	0F	B6	4E	0D	66	F7	E1	66	03	46	FC	66	H f H f q N f + á f F ú f
000000D0	0F	A4	C2	10	FB	BB	00	07	53	B9	04	00	E8	52	FC	5B	 Å ú » S ' è R ú [
000000E0	0F	82	3D	FC	81	3F	4D	5A	75	08	81	BF	00	02	42	4A	, = ú □ ? M Z u □ ; B J
000000F0	74	06	BE	80	7D	E9	0E	FC	EA	00	02	70	00	03	C0	13	t  € } é ü è . p . À .
00000100	D2	03	C0	13	D2	E8	18	00	FA	26	66	8B	01	66	25	FF	O . À . O è . ú s f < f % ý
00000110	FF	FF	0F	66	0F	A4	C2	10	66	3D	F8	FF	FF	0F	FB	C3	ý ý f  Å f = ø ý ý ú Å
00000120	BF	00	7E	FA	66	C1	E0	10	66	0F	AC	D0	10	66	0F	B7	¿ ~ ú f Á à f - Ð f .
00000130	4E	0B	66	33	D2	66	F7	F1	66	3B	46	F8	74	44	66	89	N f 3 Ó f + ñ f ; F ø t D f k
00000140	46	F8	66	03	46	1C	66	0F	B7	4E	0E	66	03	C1	66	0F	F ø f F f . N f Á f
00000150	B7	5E	28	83	E3	0F	74	16	3A	5E	10	0F	83	A4	FB	52	· ^ (f ä t : ^ . f  ú R
00000160	66	8B	C8	66	8B	46	24	66	F7	E3	66	03	C1	5A	52	66	f < È f < F \$ f + á f . Á Z R f
00000170	0F	A4	C2	10	FB	8B	DF	B9	01	00	E8	B4	FB	5A	0F	82	 Å ú < B ' è ' ú Z ,
00000180	9F	FB	FB	8B	DA	C3	00	00	00	00	00	00	00	00	00	00	Ý ú ú < Ú Å
00000190	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000001A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000001B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000001C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000001D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000001E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000001F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	55 AA	U 

Les tables d'allocation de fichiers (FAT et MFT)

Trois fonctions principales :

1) le suivi de l'espace alloué et libre, 2) la gestion des répertoires et noms de fichiers et 3) le suivi de l'emplacement dans lequel les différentes parties de chaque fichier sont physiquement stockées sur le disque.

La FAT (File Allocation Table) est une structure contenant la liste des clusters utilisés et non utilisés. Chaque partition comporte, en plus de la FAT originale, une copie de sauvegarde située juste après la FAT originale en début de partition (juste après le secteur de boot), afin que le système puisse récupérer celle-ci si la FAT originale s'avérait corrompue. Chaque FAT occupe plusieurs milliers de clusters sur la partition.

Lorsque l'on crée un fichier ou un sous-répertoire, les informations correspondantes sont stockées dans la FAT sous la forme de données hexadécimales détaillant le nom et la taille du fichier, la date et l'heure de sa dernière modification, le numéro de cluster de départ et l'attribut (Archive, Caché, Système...). C'est une liste de valeurs numériques permettant de décrire l'allocation des clusters d'une partition, c'est-à-dire l'état de chaque cluster de la partition dont elle fait partie. La table d'allocation est en fait un tableau dont chaque cellule contient un chiffre qui permet de savoir si le cluster qu'elle représente est utilisé par un fichier, et, le cas échéant, indique l'emplacement du prochain cluster que le fichier occupe. On obtient donc une chaîne FAT, c'est-à-dire une liste chaînée de références pointant vers les différents clusters successifs jusqu'au cluster de fin de fichier. Chaque entrée de la FAT a une longueur de 16 ou 32 bits (selon qu'il s'agit d'une FAT16 ou d'une FAT32). Les deux premières entrées permettent de stocker des informations sur la table elle-même, tandis que les entrées suivantes permettent de référencer les clusters. Certaines entrées peuvent contenir des valeurs indiquant un état du cluster spécifique. Ainsi la valeur 0000 indique que le cluster n'est pas utilisé, FFF7 permet de marquer le cluster comme défectueux pour éviter de l'utiliser, et les valeurs comprises entre FFF8 et FFFF spécifient que le cluster contient la fin d'un fichier. Voici comment se présente le **1er secteur de la FAT** (33ème secteur de la partition): les 8 premiers octets sont toujours ceux que j'ai soulignés en rouge (plus rarement on peut avoir FF à la place de 0F sur le 4ème octet, et/ou 0F à la place de FF sur le 8ème octet, voire F7 ou 07 en 8ème position si l'on boot sous Win 9x) :

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
00000000	FB	FF	FF	0F	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	ø ŷ ŷ ŷ ŷ ŷ ŷ ŷ ŷ ŷ ŷ ŷ ŷ ŷ ŷ ŷ
00000010	FF	FF	FF	0F	FF	FF	FF	FF	FF	FF	FF	FF	02	00	0C	00	ŷ ŷ ŷ ŷ ŷ ŷ ŷ ŷ ŷ ŷ ŷ ŷ ŷ ŷ ŷ ŷ
00000020	09	00	00	00	0A	00	00	00	00	0B	00	00	00	00	0C	00	ŷ ŷ ŷ ŷ ŷ ŷ ŷ ŷ ŷ ŷ ŷ ŷ ŷ ŷ ŷ ŷ
00000030	0D	00	00	00	0E	00	00	00	00	0F	00	00	00	10	00	00	ŷ ŷ ŷ ŷ ŷ ŷ ŷ ŷ ŷ ŷ ŷ ŷ ŷ ŷ ŷ ŷ
00000040	11	00	00	00	12	00	00	00	00	13	00	00	00	14	00	00	ŷ ŷ ŷ ŷ ŷ ŷ ŷ ŷ ŷ ŷ ŷ ŷ ŷ ŷ ŷ ŷ
00000050	15	00	00	00	16	00	00	00	00	17	00	00	00	18	00	00	ŷ ŷ ŷ ŷ ŷ ŷ ŷ ŷ ŷ ŷ ŷ ŷ ŷ ŷ ŷ ŷ
00000060	19	00	00	00	1A	00	00	00	00	1B	00	00	00	1C	00	00	ŷ ŷ ŷ ŷ ŷ ŷ ŷ ŷ ŷ ŷ ŷ ŷ ŷ ŷ ŷ ŷ
00000070	1D	00	00	00	1E	00	00	00	00	1F	00	00	00	20	00	00	ŷ ŷ ŷ ŷ ŷ ŷ ŷ ŷ ŷ ŷ ŷ ŷ ŷ ŷ ŷ ŷ
00000080	21	00	00	00	22	00	00	00	00	23	00	00	00	24	00	00	! " # \$
00000090	25	00	00	00	26	00	00	00	00	27	00	00	00	28	00	00	% & ' (
000000A0	29	00	00	00	2A	00	00	00	00	2B	00	00	00	2C	00	00) * + ,
000000B0	2D	00	00	00	2E	00	00	00	00	2F	00	00	00	30	00	00	- . / 0
000000C0	31	00	00	00	32	00	00	00	00	33	00	00	00	34	00	00	1 2 3 4
000000D0	35	00	00	00	36	00	00	00	00	37	00	00	00	38	00	00	5 6 7 8
000000E0	39	00	00	00	3A	00	00	00	00	3B	00	00	00	3C	00	00	9 : ; <
000000F0	3D	00	00	00	FF	FF	FF	0F	3F	00	00	00	00	40	00	00	= ŷ ŷ ŷ ? @
00000100	41	00	00	00	42	00	00	00	00	43	00	00	00	44	00	00	A B C D
00000110	45	00	00	00	46	00	00	00	00	47	00	00	00	48	00	00	E F G H
00000120	49	00	00	00	4A	00	00	00	00	4B	00	00	00	4C	00	00	I J K L
00000130	4D	00	00	00	4E	00	00	00	00	4F	00	00	00	50	00	00	M N O P
00000140	51	00	00	00	52	00	00	00	00	53	00	00	00	54	00	00	Q R S T
00000150	55	00	00	00	56	00	00	00	00	57	00	00	00	58	00	00	U V W X
00000160	59	00	00	00	5A	00	00	00	00	5B	00	00	00	5C	00	00	Y Z [\
00000170	5D	00	00	00	5E	00	00	00	00	5F	00	00	00	60	00	00] ^ _ `
00000180	61	00	00	00	62	00	00	00	00	63	00	00	00	64	00	00	a b c d
00000190	65	00	00	00	66	00	00	00	00	67	00	00	00	68	00	00	e f g h
000001A0	69	00	00	00	6A	00	00	00	00	6B	00	00	00	6C	00	00	i j k l
000001B0	6D	00	00	00	6E	00	00	00	00	6F	00	00	00	70	00	00	m n o p
000001C0	71	00	00	00	72	00	00	00	00	73	00	00	00	74	00	00	q r s t
000001D0	75	00	00	00	FF	FF	FF	0F	77	00	00	00	00	78	00	00	u ŷ ŷ ŷ w x
000001E0	79	00	00	00	7A	00	00	00	00	7B	00	00	00	7C	00	00	y z {
000001F0	7D	00	00	00	7E	00	00	00	00	7F	00	00	00	80	00	00	} ~ □ €

La MFT (Master File Table) constitue la structure centrale du système de fichiers NTFS, et comme pour la FAT il existe une copie de sauvegarde sur la partition (**MFT Mirror**) permettant de se protéger d'une perte éventuelle des données. Là encore la MFT occupe plusieurs milliers de clusters et par défaut NTFS lui réserve **12,5 % de l'espace disponible** sur la partition (les données ne pouvant pas être écrites sur cette zone réservée), ce qui évite la fragmentation de la MFT. On peut définir dans la base de registre la taille de la MFT en fonction du type de fichiers que l'on a l'habitude d'enregistrer : si l'on enregistre beaucoup de petits fichiers (de nombreux documents textes par exemple) il vaut mieux avoir une grande MFT (car il y aura beaucoup d'entrées), mais si on enregistre des fichiers peu nombreux mais volumineux (des films par exemple), il vaut mieux avoir une petite MFT pour laisser le plus

possible de place sur la partition pour ces gros fichiers :

en d'autres termes, on peut augmenter la MFT (et ça peut améliorer les performances) tant qu'il reste suffisamment d'espace libre sur la partition. Par contre s'il reste peu d'espace libre sur la partition, augmenter la MFT risque de favoriser sa fragmentation (car des fichiers ne trouvant pas de place sur la partition viendront s'y loger) et on aura l'effet inverse de celui attendu... seuls certains logiciels permettent de défragmenter la MFT en faisant une défragmentation au démarrage de l'ordinateur.

Pour le **réglage de la taille de la MFT dans la base de registre**, il faut sauvegarder au préalable son registre avec Erunt puis chercher la clé suivante : HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\FileSystem > clic droit sur la valeur NtfsMftZoneReservation et lui attribuer une valeur entre 1 et 4 selon l'espace que l'on veut réserver pour la MFT (12.5 %, 25 %, 37.5 %, 50 % de l'espace partitionné). Puis quitter le registre et redémarrer le PC...

A noter qu'une **défragmentation préalable** en mode sans échec s'impose, mais dans l'idéal il est préférable de fixer la taille de la MFT sur une **partition fraîchement formatée** avant d'y mettre les données lorsqu'on sait par avance quel type de données on va y mettre : de l'intérêt d'avoir autant de partition qu'on a de type de données (films, musiques et photos, documents), ceci permettant au mieux d'adapter la taille de la MFT en fonction de ce qu'on va mettre dans la partition... Voici pour exemple le **1er secteur de la MFT** (quelque part sur la partition...) :

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
00000000	46	49	4C	45	2A	00	03	00	A6	7E	C7	17	00	00	00	00	FILE +
00000010	01	00	01	00	30	00	01	00	90	01	00	00	00	04	00	00	0
00000020	00	00	00	00	00	00	00	00	06	00	F6	00	00	00	00	00	ö
00000030	10	00	00	00	60	00	00	00	00	00	18	00	00	00	00	00	,
00000040	48	00	00	00	18	00	00	00	60	07	4B	92	12	A0	C0	01	H
00000050	60	07	4B	92	12	A0	C0	01	60	07	4B	92	12	A0	C0	01	' K' À ' K' À
00000060	60	07	4B	92	12	A0	C0	01	06	00	00	00	00	00	00	00	' K' À
00000070	00	00	00	00	00	00	00	00	00	00	00	00	00	01	00	00	
00000080	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000090	30	00	00	00	68	00	00	00	00	00	18	00	00	00	01	00	0 h
000000A0	4A	00	00	00	18	00	01	00	05	00	00	00	00	00	05	00	J
000000B0	60	07	4B	92	12	A0	C0	01	60	07	4B	92	12	A0	C0	01	' K' À ' K' À
000000C0	60	07	4B	92	12	A0	C0	01	60	07	4B	92	12	A0	C0	01	' K' À ' K' À
000000D0	00	C0	E7	01	00	00	00	00	C0	E7	01	00	00	00	00	00	À Ç
000000E0	06	00	00	00	00	00	00	00	04	03	24	00	4D	00	46	00	\$ M F
000000F0	54	00	00	00	00	00	00	00	80	00	00	00	48	00	00	00	T € H
00000100	01	00	00	00	00	00	03	00	00	00	00	00	00	00	00	00	
00000110	83	2A	00	00	00	00	00	00	40	00	00	00	00	00	00	00	f + @
00000120	00	40	A8	02	00	00	00	00	40	A8	02	00	00	00	00	00	@ " @ "
00000130	00	40	A8	02	00	00	00	00	32	84	2A	00	00	0C	00	00	@ " 2 " +
00000140	B0	00	00	00	48	00	00	00	01	00	40	00	00	00	04	00	° H @
00000150	00	00	00	00	00	00	00	00	07	00	00	00	00	00	00	00	
00000160	40	00	00	00	00	00	00	00	80	00	00	00	00	00	00	00	@ €
00000170	48	15	00	00	00	00	00	00	48	15	00	00	00	00	00	00	H H
00000180	31	08	92	43	31	00	00	00	FF	FF	FF	FF	00	00	00	00	1 ' C 1 ŷ ŷ ŷ ŷ
00000190	00	00	66	01	00	00	00	00	00	00	66	01	00	00	00	00	f f
000001A0	12	60	16	0A	00	00	01	00	B0	00	00	00	48	00	00	00	' ° H
000001B0	01	00	40	00	00	00	04	00	00	00	00	00	00	00	00	00	@
000001C0	07	00	00	00	00	00	00	00	40	00	00	00	00	00	00	00	@
000001D0	00	80	00	00	00	00	00	00	30	0B	00	00	00	00	00	00	€ 0
000001E0	30	0B	00	00	00	00	00	00	11	08	02	00	00	00	00	00	0
000001F0	FF	FF	FF	FF	00	00	00	00	00	00	00	00	00	00	F6	00	ŷ ŷ ŷ ŷ ö

Successeur du **BIOS**, l'**UEFI** (Unified Extensible Firmware Interface) ajoute des fonctionnalités intéressantes mais certaines empêchent l'installation d'un autre système d'exploitation : **Secure Boot**, **CSM**, **GPT**... **Enlever ces restrictions permettra d'installer un autre système d'exploitation** comme **Windows 7, 8 ou 10 version 32 bits**, **Windows XP** ou une **distribution Linux**.

D'une manière générale, toutes les **versions 32 bits de Windows sont incompatibles avec l'UEFI**

- Windows Vista, 7 ou 8 version 32 bits
- Windows XP
- Une distribution Linux (Debian, Arch Linux, Ubuntu...) qui ne supporte pas l'UEFI

Windows 10, 8, ou 7 en version 64 bits : compatibles UEFI

Pour pouvoir **installer un Windows 32 bits ou une distribution Linux** sur une carte mère avec **firmware UEFI**, il faut que vous suiviez ces 3 étapes : la **désactivation du Secure Boot**, l'**activation du CSM**(Compatibility Support Module) et la **conversion de votre disque dur en MBR**.

Désactiver le Secure Boot

Rendez-vous dans la **configuration du BIOS UEFI** en appuyant sur la touche appropriée lors du démarrage de l'ordinateur (souvent associée au terme *BIOS Setup*). Désactivez le **Secure Boot**, fonctionnalité dont le but est de bloquer tout élément étranger au système (comme un autre système d'exploitation).

Activer le CSM (Compatibility Support Module)

Le **CSM** (Compatibility Support Module) est un composant qui permet d'**émuler un BIOS traditionnel**, permettant ainsi d'installer des systèmes d'exploitation non compatibles UEFI. Il faut obligatoirement activer cette option pour **installer un Windows 32 bits** par exemple. En activant cette option, l'**UEFI sera désactivé**. Après, c'est comme si vous utilisiez un **BIOS traditionnel**.

Dans le **BIOS UEFI**, vous devez modifier le champ *Boot Mode* et mettre la valeur *CSM Boot* ou *Legacy Boot*. Sur certaines cartes mères, l'option consiste à mettre le *Boot UEFI* sur *Disabled*.

Convertir le disque dur GPT en MBR

Sur une carte mère équipée d'un **firmware UEFI**, il y a de très fortes chances que votre disque dur utilise la **table de partitionnement GPT**. Or, il est impossible de démarrer un système d'exploitation sur un **disque dur GPT** avec un BIOS traditionnel (**UEFI DESACTIVE**). Le BIOS traditionnel ne peut démarrer que sur un disque dur qui utilise la **table de partitionnement MBR**.

Attention, **vous allez perdre toutes les données présentes sur votre disque dur** ainsi que votre partition *Recovery* ! Donc soyez certain d'avoir **sauvegarder tous vos documents** et prenez le temps de créer une copie de votre système.

Comment **convertir le disque dur GPT en MBR** pour que le système d'exploitation puisse booter au démarrage du PC.

Avec le programme d'installation de 7, 8 ou 10

- 1- Lancez le **programme d'installation de Windows**
- 2- entrez la langue et les préférences de votre choix, puis cliquez sur *Installer maintenant*.
- 3- Acceptez les termes du contrat de licence de Windows puis sélectionnez le type d'installation **Personnalisée (option avancée)**
- 4- Vous devez normalement être sur la fenêtre *Où souhaitez-vous installer Windows ?*.
- 5- Ouvrez l'**invite de commandes Windows** en appuyant simultanément sur les touches *Shift + F10*. Tapez ensuite la commande *diskpart* pour lancer le programme Microsoft Diskpart. C'est lui qui va nous permettre de **convertir notre disque dur en MBR**.
 - a. Listez les disques durs de votre système en tapant : *list disk*.
 - b. Sélectionnez le disque dur que vous souhaitez **convertir en MBR** en tapant : *select disk #* (remplacer # par le numéro faisant référence à votre disque dur). Exemple *select disk 0*.
 - c. Saisissez la commande *clean* pour **effacer toutes les partitions et toutes les données de votre disque dur**. Votre disque dur sera totalement nettoyé ne laissant qu'un *Espace non alloué sur le disque*.
 - d. Tapez ensuite *convert mbr* pour **convertir votre disque dur en MBR**.
 - e. C'est fini, votre **disque dur a bien été converti en MBR** ! Quittez diskpart en tapant *exit*, puis fermez l'invite de commandes Windows.
- 6- **Actualisez** l'affichage des partitions pour que les modifications que vous venez de faire soient prises en compte.

Avec le logiciel GParted

>> **Convertir un disque dur en MBR avec GParted** est ultra-simple ! :D

Tout d'abord, vous devez télécharger l'ISO de GParted puis le graver sur un CD-R. Vous pouvez aussi installer GParted sur une clé USB.

Démarrez votre ordinateur à partir du CD ou de la clé USB contenant GParted. Sélectionnez *GParted Live (Default Settings)* à l'écran de démarrage de GParted.

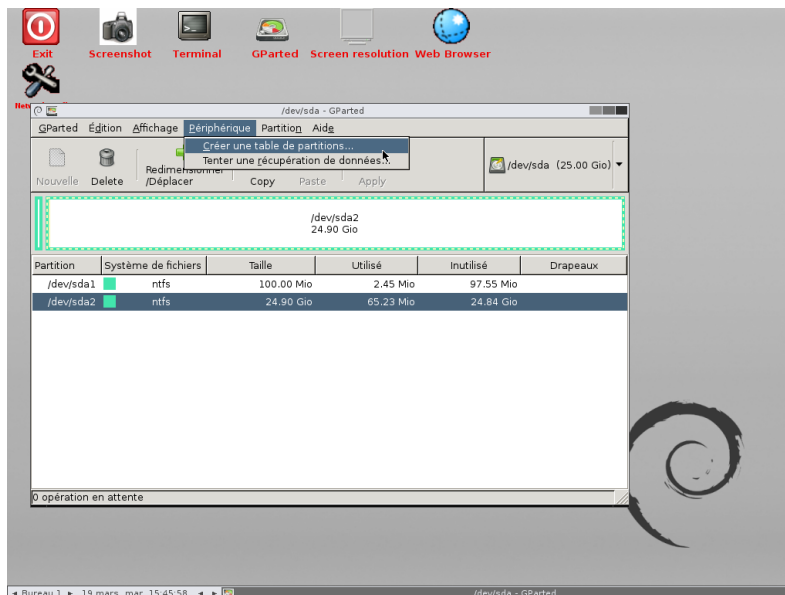
Démarrez votre ordinateur à partir du CD ou de la clé USB contenant GParted. Sélectionnez *GParted Live (Default Settings)* à l'écran de démarrage de GParted.

On vous demande ensuite de choisir une keymap, c'est-à-dire la disposition de votre clavier. Choisissez dans l'ordre *Select keymap from arch list > azerty > French > With Euro (latin 9)*.

Sélectionnez la langue française en saisissant les chiffres *08* puis en tapant sur *Entrée*.

Tapez enfin sur *Entrée* pour laisser la configuration du serveur X par défaut.

La fenêtre principale de GParted s'ouvre. Nous allons créer une nouvelle table de partition, ce qui aura pour effet de supprimer toutes les partitions et toutes les données de notre disque dur. Pour cela, cliquez sur *Périphérique > Créer une table de partition...*



Une fenêtre s'ouvre. Cliquez sur *Avancé* et vérifiez que le type de la table partitions est bien défini sur *gpt*. Validez en cliquant sur le bouton *Apply*.

Votre disque dur est maintenant vide et a bien été **converti en GPT** !

Démarrer le DVD d'installation en mode UEFI

L'UEFI utilise la **table de partitionnement GPT** et non plus le **MBR** pour démarrer un système d'exploitation sur un disque dur. Vous devez donc utiliser un système d'exploitation qui prend en charge le **démarrage sur les disques durs GPT**. Windows 7, 8 ou 10 le permettent mais **il faut lancer l'installation correctement** !

Si vous démarrez **l'installation de Windows en mode BIOS**, cela lancera la version BIOS de Windows. Comme cette version ne prend pas en charge le démarrage sur un disque GPT, vous aurez droit à un message d'erreur lors de l'installation.

Pour débiter l'installation de **Windows en version UEFI**, il faut démarrer votre PC sur le **DVD d'installation en mode UEFI** ! Regardez attentivement la capture ci-dessous. Il faut absolument que vous démarriez sur le lecteur CD/DVD (ici un lecteur Mastshita) en mode UEFI. On distingue bien les deux possibilités de démarrage du lecteur : UEFI (GPT) et P2 (MBR).

Pour ce faire, allumez votre ordinateur puis lors du démarrage du BIOS, sélectionnez l'option « Boot Order » ou « Ordre de démarrage » en appuyant sur la touche appropriée pour sélectionner votre lecteur CD/DVD. NB : la touche est propre au modèle de votre carte mère, vous devez donc regarder en bas de votre écran pour voir la touche en question (dans la majorité des cas il s'agit des touches Fn de type F2, F8, F11...).



Une fois que vous avez sélectionné votre lecteur CD/DVD en mode UEFI, le programme d'installation de Windows débute.

Formattage : défini table des fichiers et la taille de clusters et découpe le hdd en cluster (numérote) et enregistre les num clusters dans la table (unité la plus petite = la taille du cluster) et cré la racine (le root « \ »)

installation os active la partition et inscrit dans le mbr le flag de quelle partition est active

Petit disque utilise fat32 est mieux (512 oc pour 1 secteur est le mini) et en ntfs seulement hdd ~1 Go

<http://www.bellamyjc.org/fr/theoriemultiboot1.html>

<http://www.bellamyjc.org/fr/theoriemultiboot2.html>