# Table des matières

Systèmes de fichier

Étude didactique – FAT / NTFS / FUSE

Système d’exploitation

Enseignant :  
Cortinovis Claudio

Projet réalisé par :  
Lovis Thomas, Ombang Ndo Charles, Vulliemin Kevin

INF3\_DLM\_B

[Table des matières 0](#_Toc513146297)

[Introduction 2](#_Toc513146298)

[FAT 4](#_Toc513146299)

[Composition 4](#_Toc513146300)

[Types de fichiers 4](#_Toc513146301)

[Types de clusters 4](#_Toc513146302)

[Entrées de dossier 4](#_Toc513146303)

[Table FAT 4](#_Toc513146304)

[Les deux premières entrées FAT 5](#_Toc513146305)

[Sources 5](#_Toc513146306)

[Example fragmentation 5](#_Toc513146307)

[Must have 5](#_Toc513146308)

[Autres 5](#_Toc513146309)

[NTFS 6](#_Toc513146310)

[Introduction 6](#_Toc513146311)

[Structure globale 6](#_Toc513146312)

[Démarrage 7](#_Toc513146313)

[MBR / GUID 7](#_Toc513146314)

[Processus 7](#_Toc513146315)

[Master File Table 8](#_Toc513146316)

[Enregistrement de fichier / dossier 8](#_Toc513146317)

[Types de fichier 9](#_Toc513146318)

[Système didactique 9](#_Toc513146319)

[Sources 9](#_Toc513146320)

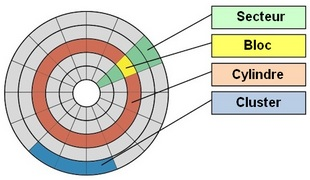
[FUSE 10](#_Toc513146321)

# Introduction

Les disques durs, aussi petits soient-ils, contiennent des millions de bits, il faut donc organiser les données afin de pouvoir localiser les informations, c'est le but du système de fichiers.

Un disque dur est, rappelons-le, constitué de plusieurs plateaux circulaires tournant autour d'un axe. Les pistes (zones concentriques écrites de part et d'autre d'un plateau) sont divisées en quartiers appelés secteurs (d'une taille de 512 octets).

Le formatage logique d'un disque permet de créer un système de fichiers sur le disque, qui va permettre à un système d'exploitation (Windows, UNIX, MacOS, ...) d'utiliser l'espace disque pour stocker et utiliser des fichiers. Le système de fichiers est basé sur la gestion des clusters (en français « unité d'allocation »), c'est-à-dire la plus petite unité de disque que le système d'exploitation est capable de gérer.



Un cluster est constitué d'un ou plusieurs secteurs, ainsi plus la taille d'un cluster est importante, moins le système d'exploitation aura d'entités à gérer.

En contrepartie, étant donné qu'un système d'exploitation ne sait gérer que des unités d'allocation entière, c'est-à-dire qu'un fichier occupe un nombre entier de cluster, le gaspillage est d'autant plus grand qu'il y a de secteurs par cluster. On comprend alors toute l'importance du choix du système de fichiers.

**Système de fichier et système d’exploitation**

En réalité le choix du système de fichiers se fait en premier lieu suivant le système d'exploitation que vous utilisez. D'une manière générale, plus le système d'exploitation est récent plus le nombre de systèmes de fichiers supportés sera important. Cette introduction n’aborde que l’évolution du système d’exploitation Windows.

Ainsi (si on suit une évolution essentiellement basée sur Windows) sous l’ancien DOS et sur les premières versions de Windows 95, la FAT16 était de rigueur compte tenu de la taille des disques à l’époque. Si jamais la taille de la partition était supérieure à 2Go, le système de fichier FAT16 était exclu.

On a donc dû le remplacer par un système pouvant exploiter des disques plus volumineux, le système FAT32. À partir de Windows NT5, la sécurité était de mise notamment grâce à l’évolution d’internet, de l’agrandissement des disques et de l’utilisation d’ordinateur par plusieurs personnes. C’est ainsi que naquit le NTFS avec la gestion des droits et une certaine garanti de récupération de données perdues.

Sur les système Windows de nos jours, le système NTFS (5.1 pour sa dernière version) est conseillé car il procure une sécurité plus grande ainsi que des performances accrues par rapport au système FAT32. Pour des raisons de comptabilité (Partition de swap) ou d’espace, vous pouvez toutefois opter pour une partition de type FAT32. Voici une liste non exhaustive des plus connus :

|  |  |
| --- | --- |
| Système d’exploitation (OS) | Système de fichier (FS) |
| **Windows XP, 7, 10+** | FAT16, FAT32, NTFS, ExFAT |
| **GNU / Linux, Android** | Ext, Ext2 / 3 / 4 |
| **MacOS, iOS** | APFS |

# FAT

## Composition

Il est nécessaire de clarifier les composants suivants :

Secteurs (hardware) : sont de minimum 512 octets. C’est la plus petite quantité de données que l'électronique du matériel sait manipuler (plus petite unité physique).

Cluster (software) : un ensemble de secteurs consécutifs. Comme vu précédemment, c’est la plus petite quantité de données en une fois que sait manipuler le système d'exploitation (multiple de taille secteurs)

Il faut savoir qu’un fichier contenant un seul caractère va occuper tout le cluster quel que soit sa taille. Cela peut engendrer une perte de taille de données considérable (plus petite unité logique).

## Types de fichiers

Étant donné que FAT est un format très léger, il autorise uniquement l’inscription de données suivantes :

* Nom de fichier : Case sensitive (Linux) et case insensitive (Windows, OSX)
* Dossier
* Meta data

## Types de clusters

Les clusters peuvent être de 2 types différents :

* Clusters de données contenant du contenu de fichier
* Clusters de dossier contenant des structures d’entrée de dossier

## Entrées de dossier

Fichier système metadata pour tous les fichiers

Contient les noms de fichiers, timestamps, et cluster de départ du fichier

Un cluster utilisé pour l’entrée ne sera pas utilisé pour les fichiers

## Table FAT

* Fournit le mappage de fichier à cluster
* Chaque cellule représente un cluster
* L’adressage dépend de la version de FAT

Les valeurs importantes :

* 0 : non alloué
* ff7, fff7 or fff, fff7: bad cluster
* ff8, fff8 or fff, fff8: end of file
* Toutes les autres valeurs indiquent que le cluster est alloué

Si le cluster vaut 0x000000, le cluster correspondant est libre.

Si le cluster vaut 0xFFFFFF7, le cluster correspondant est déclaré « bad cluster ».

Si le cluster vaut un nombre supérieur ou égal à 0x0FFFFFF8, le cluster est le dernier de la chaîne. On dit qu’il porte une marque EOC (End Of Cluster Chain).

Dans tous les autres cas, le cluster indique le numéro du cluster suivant dans la chaîne.

## Les deux premières entrées FAT

Le 1er cluster de la zone de données est le cluster #2 ceci laisse libre les deux premières entrées de FAT inutilisé.

Dans le 1er byte de la 1ère entrée, une copie du « media descriptor » est stocké. Les bits restants sont fixés à 1. Dans la seconde entrée le marquer « end-of-life » est stocké.

The high order two bits of the second entry are sometimes, in the case of FAT16 and FAT32,

used for dirty volume management: high order bit 1: last shutdown was clean; next highest bit 1: during the previous mount no disk I/O errors were detected.

## Sources

### Example fragmentation

http://assiste.com/Assiste/media/images/Fragmentation\_Defragmentation.png

### Must have

<http://pic18.free.fr/fat32.php>

### Autres

https://www.youtube.com/watch?v=iGO7N7eoqZU

https://www.benjv.fr/blog/systeme-fichier-fat32/

https://www.pctechguide.com/hard-disks/file-systems-fat-fat8-fat16-fat32-and-ntfs-explained

https://technet.microsoft.com/en-us/library/cc776720%28v=ws.10%29.aspx?f=255&MSPPError=-2147217396

https://technet.microsoft.com/en-us/library/cc938438.aspx?f=255&MSPPError=-2147217396

https://www.pjrc.com/tech/8051/ide/fat32.html

http://www.forensicswiki.org/wiki/FAT

https://www.cs.drexel.edu/~jjohnson/2012-13/fall/cs370/resources/File%20Allocation%20Table.pdf

http://poloastucien.free.fr/mbr\_fat\_secteurs\_boot\_h.html

http://www.c-jump.com/CIS24/Slides/FAT/lecture.html

http://www.dewassoc.com/kbase/hard\_drives/clusters.htm

https://www.easeus.com/data-recovery-ebook/file-deletion-in-FAT32.htm

https://sitelec.org/cours/caleca/pc/fat16.html

http://assiste.com/Defragmentation.html (super bien)

http://www.courstechinfo.be/OS/FileSys.html

http://proquest.tech.safaribooksonline.de/0321268172/ch10#X2ludGVybmFsX0h0bWxWaWV3P3htbGlkPTAzMjEyNjgxNzIlMkZjaDEwbGV2MXNlYzQmcXVlcnk9KChGaWxlJTIwU3lzdGVtJTIwRm9yZW5zaWMlMjBBbmFseXNpcykp

<https://books.google.ch/books?hl=fr&lr=&id=Zpm9CgAAQBAJ&oi=fnd&pg=PT13&dq=fat32+file+system&ots=6LCI8igNJC&sig=mdILcrp-0j9FqaMs578e5cmu71o&redir_esc=y#v=onepage&q=fat32%20file%20system&f=false>

# NTFS

## Introduction

Le NTFS de « New Technologie File System » ou de « Windows NT File System » fourni une combinaison de performance, fiabilité et compatibilité supplémentaire aux autres formats de fichier tel que FAT.

Il a été conçu pour répondre de manière rapide à des opération de traitement standard sur de gros disques (maximum environ 2 TB) : écrire, lire et rechercher. NTFS dispose également d’une fonction de récupération de fichier.

La sécurité est le principal atout du NTFS car il permet de définir des attributs différents pour chaque fichier. La version 5 apporte encore une optimisation de la performance ainsi que des quotas d’espace disque pour chaque utilisateur. La version 5 devrait également apporter

Cette partie du document explique globalement comment NTFS se comporte mais des explications plus approfondies sont disponibles aux liens cités dans les sources.

## Structure globale

Pour avoir un aperçu complet des différentes parties d’un disque au format NTFS, voyons comment celui-ci se présente lors d’un formatage.

Premièrement le formatage entraine la création de plusieurs espace disques :



*Fig. 1. Structure organisationnelle NTFS*

|  |  |
| --- | --- |
| **Component** | **Description** |
| **NTFS Boot Sector** | Contient le block de paramètre BIOS qui stocke les informations sur la disposition du volume et sa structure de fichier.  Contient également le code de démarrage qui va charger le système d’exploitation. |
| **Master File Table** | Contient les informations nécessaires pour retrouver des fichiers sur une partition NTFS comme les attributs de fichiers. |
| **File System Data** | Contient les données de fichiers non-enregistré dans la Master File Table. |
| **Master File Table Copy** | Contient des copies des enregistrements essentiellement pour la récupération de fichier si l’enregistrement original est corrompu ou est inatteignable. |

Formatter un volume NTFS entraîne également la création de quelques fichiers système (Metadata) comme le $MFT, $Bitmap, $LogFile et d’autres, comprenant des informations sur tous les fichiers et tous les dossiers contenus sur le volume.

La liste complète des Metadata est disponible à l’adresse suivante :

<http://ntfs.com/ntfs-system-files.htm>

## Démarrage

Ce chapitre sera uniquement survolé étant donné que le projet se concentre essentiellement sur la répartition des données de fichier sur le disque.

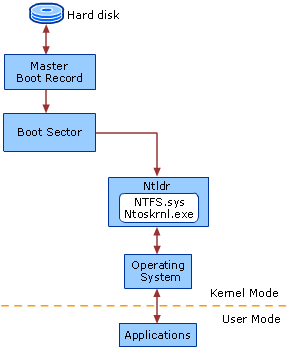
### MBR / GUID

Dans les disques MBR, le secteur de boot, qui est localisé au premier secteur logique de chaque partition, est une structure de disque critique pour démarrer votre ordinateur. Il contient le code exécutable et les données requises pour le fonctionnement de ce code, incluant les informations que le système de fichier utilise pour accéder au volume. À la fin du secteur de boot, il y a une structure 2-byte appelé « mot de signature » ou « marqueur de fin de secteur », qui possède toujours la valeur 0x55AA.

Les disques avec partitions de table GUID (GPT) sont similaires aux disques avec MBR, excepté qu’ils utilisent des structures de partitions primaires et de backup pour fournir une redondance. Ces structures sont placées au début et à la fin des disques. GPT identifie ces structures et les localise grâce au bloc d’adresse logique (LBA). Les éléments suivants sont présents dans le secteur de boot qu’il soit basique ou dynamique.

### Processus

Le secteur de boot de la partition active se charge dans la mémoire et démarre le programme Ntldr qui va s’occuper de charger le menu de boot utilisateur si plusieurs OS sont installés, ou charger le système directement si c’est le seul installé.



*Fig. 2. Structure de démarrage NTFS*

De plus amples informations sur le processus de démarrage est disponible à la rubrique « Master Boot Code startup process » et « Boot Sector startup process » de l’adresse suivante :

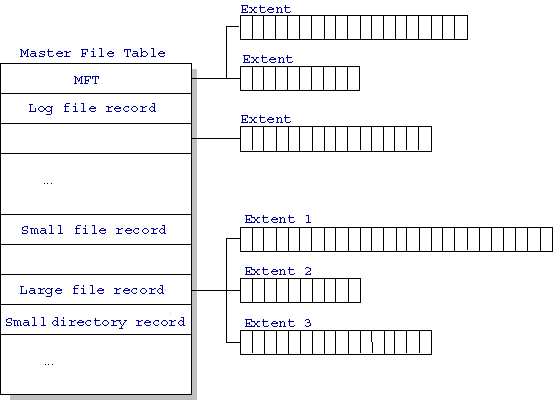
<https://technet.microsoft.com/pt-pt/library/cc781134(v=ws.10).aspx>

## Master File Table

Chaque fichier NTFS est représenté par un enregistrement dans un fichier spécial appelé le « Master File Table » (MFT). NTFS réserve les 16 premières entrées pour des informations spéciales.

La 1ère entrée de la table décrit la Master File Table elle-même suivi d’un fichier miroir. Si la 1ère entrée MFT est corrompue, NTFS va lire la seconde entrée pour trouver le fichier miroir possédant une 1ère entrée identique au MFT. Leurs localisations respectives sont définies dans le secteur de Boot.

La figure suivante résume la structure de MFT de manière simplifiée :



*Fig. 3. Structure MFT*

La MFT alloue une certaine quantité d’espace pour chaque entrée de fichier. Les attributs de fichier sont écrits dans cet espace alloué. Les petits fichiers (typiquement 512 bytes ou plut petit) peuvent être entièrement écrits à l’intérieur de la MFT. La MFT prend en général 12.5% du volume.

### Enregistrement de fichier / dossier



*Fig. 4. Petit enregistrement MFT*

Cette conception permet un accès très rapide aux fichiers. Prenons par exemple un système de fichier FAT. Il utilise une table d’allocation de fichier pour lister le nom et l’adresse de chaque fichier. Les enregistrements de dossier FAT contiennent un index de la table d’allocation de fichier.

Quand vous voulez retrouver un fichier, FAT va d’abord s’assurer de l’existence du fichier en lisant la table d’allocation de fichier. Ensuite FAT va retrouver le fichier en cherchant dans la chaîne d’unité d’allocation assigné au fichier. Avec NFTS, aussitôt qu’on retrouve le fichier, il est disponible à l’utilisation.

Les dossiers sont aussi hébergés dans la MFT mais au lieu de contenir des données, ils vont contenir des index. Contrairement aux petits dossiers, les gros dossiers sont organisés en B-Tree, ayant des entrées avec des pointeurs sur des clusters externes.

## Types de fichier

Tout les fichier (et les dossier) sont vues comme des ensembles attributs de fichier. Les éléments tels que le nom du fichier, les informations de sécurité et même les données sont des attributs de fichier. Chaque attribut est défini par un type, un code, et éventuellement un nom.

Quand un attribut de fichier peut être contenu dans la MFT, il est appelé « Attributs résidents ». Par exemple les informations tels que le nom du fichier et le timestamp sont toujours inclus dans l’entrée MFT relative.

Quand toutes les informations pour un fichier sont trop volumineuses pour être contenu dans une entrée MFT, certains attributs ne seront pas résident de la MFT. Les attributs non-résident sont alloué ailleurs sur le disque (Partition Data) dans un ou plusieurs clusters. Si tous les attributs ne peuvent être contenu dans une entrée MFT, NTFS crée une entrée additionnelle MFST et écrit la liste des attributs dans la 1ère entrée de fichier MFT pour décrire leurs emplacements.

La liste complète des attributs est disponible à l’adresse suivante :

<http://ntfs.com/ntfs-files-types.htm>

## Système didactique

Pour le système didactique, les actions vont être simplifiées de la façon suivante :

Les partitions « secteur de boot » et « copie de la MFT » sont omis.

La récupération de fichier et l’utilisation des dossiers n’est pas abordée.

Les petits fichier (< 1 KB) vont être enregistré dans la Master File Table.

Les données de gros fichiers (> 1 KB) seront externalisées sur la partie « Data » du disque. L’adresse de leur localisation remplace la partie « Données » dans l’entrée MFT.

## Sources

<http://www.ntfs.com/ntfs_basics.htm>

<https://technet.microsoft.com/pt-pt/library/cc781134(v=ws.10).aspx>

<https://blogs.technet.microsoft.com/askcore/2010/02/18/understanding-the-2-tb-limit-in-windows-storage/>

<http://lemouillour.eric.free.fr/systemes/nt/ntfs.htm>

# FUSE

FUSE pour **F**ilesystem in **Use**rspace, soit système de fichier en espace utilisateur, est un logiciel libre permettant à un utilisateur sans privilège particulier d’accéder à un système de fichiers sans qu’il soit nécessaire de modifier les sources du noyau (wikipedia).

FUSE nous permet d’implémenter nos propres systèmes de fichiers pour linux dans l’espace utilisateur. En effet, une implémentation côté kernel est bien plus complexe et nécessite des connaissances profondes.

