

Technicien Supérieur Gestionnaire  
de Ressources Informatiques  
et Réseaux

Partie 1

# Systemes de Fichiers

Ce(tte) œuvre est mise à disposition selon les termes de la [Licence Creative Commons Attribution - Pas d'Utilisation Commerciale 4.0 International](https://creativecommons.org/licenses/by-nc-sa/4.0/).



## Table des matières

Structure du système de fichiers.....	1
Exemples.....	2
Ext2.....	2
Ext3.....	3
Ext4.....	3
Attribut étendus.....	4
Linux.....	4
TOUT EST FICHER.....	5
Ls -l.....	6
Ls -a.....	7
ls -i.....	7
Arborescence.....	8
Man pages ls.....	8
Hierarchy Files System.....	9

# Files systèmes

Quel monde fascinant que le système de fichiers, je ne cesse d'en être émerveillé.

Déjà sa découverte lorsque j'ai tenté le dual boot avec Vista et Hardy Heron.

Là j'ai su que je ne savais pas. On peut faire plein de « trucs » c'est sure.

Penser qu'on est le boss.... Oui... , bien sure.

Je parle ici des types de système de gestion de volume.

## Structure du système de fichiers

En informatique, un système de fichiers (**FS ou FileSystem en anglais**) est une méthode d'organisation des données persistantes sur un média durable (disque dur, DVD, clé USB, etc.).

Le système de fichiers offre à l'utilisateur une vue abstraite sur ses données.

L'unité de stockage est le fichier, qui est une séquence d'octets ; les fichiers sont groupés dans des dossiers ; les dossiers sont organisés en arborescence (il y a un dossier racine et des sous-dossiers).

Chaque objet (dossier, fichier, etc.) est identifié par un nom et possède des propriétés qui dépendent du système de fichier (notamment la protection d'accès en lecture, en écriture ou des listes de dates).

Ce nom est une chaîne de caractères de taille parfois limitée dans laquelle certains caractères peuvent être interdits (par exemple : ":" ainsi que "/" ou "\" selon les systèmes de fichiers).

Via un système de fichiers, l'organisation apparente ne change pas, mais dans le moteur il peut y avoir divers fonctionnements (indexation, résistance aux pannes, sécurité, etc...).

Les systèmes de fichiers peuvent inclure la compression ou le chiffrement automatique des données, une gestion plus ou moins fine des droits d'accès aux fichiers, et une journalisation des écritures (pour la robustesse, en cas de défaillance du système).

De plus, certains systèmes de fichiers peuvent s'étendre sur un réseau entier, comme par exemple NFS ou SMB / CIFS.

Il existe d'autres façons d'organiser les données, par exemple les systèmes de gestion de base de données (notamment base de données relationnelle) et les fichiers indexés.

## Exemples

Voici des exemples de systèmes de fichiers :

### Ext2

Ext2 ou Ext2fs (en anglais, second extended file system) est le système de fichiers historique de GNU/Linux. Il a été créé à l'origine par **Rémy Card** et est basé sur l'extended file system.

Il a également été très largement influencé par le FFS.

La grande particularité de Ext2 est qu'il ne fragmente que très peu !  
Contrairement aux systèmes de fichiers FAT ou NTFS (tous deux utilisés par Windows), Ext2 range ses données de manière "intelligente".

Les performances n'en sont qu'améliorées, et il n'est par conséquent pas toujours nécessaire d'utiliser un outil de défragmentation, ou plus rarement.

Plus tard, une extension a été ajoutée qui a permis de journaliser le

système de fichiers. Celle-ci a fort logiquement été nommée ext3.

Notons qu'un système qui ne connaît que l'ext2 est parfaitement capable de lire et d'écrire de l'ext3, mais il n'y aura pas alors de journalisation.

La différence entre les deux systèmes réside dans l'adjonction d'une zone journal et la suppression des données, rendant la récupération de celle-ci impossible sur le FS ext3.

Il suffit de cocher une option dans son noyau et de le recompiler pour bénéficier du support de ext3. Pour pouvoir passer sa partition ext2 en ext3, cela se fait à l'aide de la commande `tune2fs`.

### Ext3

Il a l'avantage majeur de pouvoir être utilisé à partir d'une partition ext2, sans avoir à sauvegarder et à restaurer des données (un système de fichiers ext3 peut être monté et utilisé comme un système de fichier ext2). Tous les utilitaires de maintenance pour les systèmes de fichiers ext2, comme `fsck`, peuvent également être utilisés avec ext3.

Son avantage sur ReiserFS qui est un autre système de fichier journalisé est qu'il permet l'utilisation du logiciel `dump`, abondamment utilisé en entreprise pour les sauvegardes.

### Ext4

Il est basé sur la notion d'extent.

C'est une zone contiguë (qui peut atteindre plusieurs centaines de Mo) qui est réservée chaque fois qu'un fichier est enregistré sur le disque dur.

Cela permet, en cas d'écriture ultérieure sur le fichier, d'ajouter les nouvelles données dans l'extent au lieu de devoir écrire dans une autre zone du disque dur (ce qui augmente la fragmentation).

Les gros fichiers sont ainsi stockés de façon bien plus efficace et rationnelle.

Pour les petits fichiers Ext4 utilise une astuce puisqu'il stocke les

données directement dans le système d'extent lui-même sans avoir à allouer un bloc mémoire séparé.

L'ext4 permet de gérer les volumes d'une taille allant jusqu'à 1024 pébioctets.

Donc l'UEFI je m'en fou, puisqu'il oblige un système Microsoft à redescendre du NTFS au Fat32, ce qui n'a pas de sens, sur le plan technique.

Le système de fichiers ext4 a une compatibilité descendante avec ext3.

C'est à dire, qu'une partition ext3 peut toujours être montée comme ext4 (en utilisant le type de système de fichiers "ext4" lors du montage).

L'inverse peut être possible, à la condition que la partition ext4 n'ait jamais utilisé l'allocation par extents pour sauvegarder des fichiers, mais l'allocation binaire comprise par ext3.

Cette capacité de ext4 d'utiliser les formats de ext3 existe pour faciliter la conversion des partitions ext3 vers ext4.

Il existe une limitation présente dans la plupart des systèmes de fichiers : la limite des 255 caractères par fichier, voire par chemin d'accès !

## Attribut étendus

En informatique, les attributs étendus (EA - Extended Attributes) sont une fonctionnalité du système de fichiers.

Une règle importante dans les systèmes Gnu/Linux

Au contraire des attributs réguliers, les attributs étendus ne sont pas interprétés par le système de fichiers, mais sont stockés et gérés séparément.

Tout attribut étendu est identifié par une paire (nom:valeur) qui permet la manipulation des informations stockées. Les attributs étendus sont généralement limités en taille, mais ces limitations sont dépendantes du système de fichier utilisé et de sa configuration.

## Linux

Sous Linux, les systèmes de fichiers ext2, ext3, ext4, JFS, ReiserFS, XFS, Btrfs et OCFS2 1.6 incluent le support pour les attributs étendus.

Abrégée xattr, cette fonctionnalité doit toutefois être activée dans la configuration du noyau (libattr).

Tout fichier régulier peut disposer d'attributs étendus, chacun d'entre eux étant constitué d'un nom et de données associées.

Le nom doit être une chaîne de caractères se terminant par un octet nul, et faire partie d'un espace de nommage parmi utilisateur (user), fiabilité (trusted), sécurité (security) and et système (system).

- L'espace de nommage utilisateur n'a aucune restriction quant aux noms ou aux contenus.
- L'espace de nommage système est principalement utilisé par le noyau pour les listes des contrôles d'accès (ACL).
- L'espace de nommage sécurité est utilisé par exemple par SELinux.

Les attributs étendus ne sont pas couramment utilisés par les programmes en espace utilisateur, en dépit du fait qu'ils soient supportés depuis le noyau 2.6.1. Néanmoins, les logiciels Beagle et Dropbox utilisent déjà les attributs étendus, d'autre part freedesktop.org a recommandé officiellement leur utilisation.

## TOUT EST FICHER

Vos documents, absolument tous vos documents peu importe leur nature, photo, vidéo, musique, votre disque dur, SSD, chaque volume, clé USB, imprimante...tout; oui tout, ne cherchez pas plus loin , (allez pour être franc, ce n'est pas tout à fait vrai, nous y viendrons) .

"Oui mais il y a des dossiers quand même, et une clé USB c'est bien du matériel"

"Oui mais!!": TOUT EST FICHER

Les dossiers ou "répertoires", ne sont que des représentations graphique simplifié.

Votre clé est bien un matériel, nous verrons comment "ça monte", nous le verrons dans l'arborescence . Il existe donc des:

- fichiers standards du genre fichiers de données (.odt, .txt), fichiers images (.png, .jpg, .jpeg, .svg), fichiers audio (.mp3, .ogg), fichiers de code (.py, .php, .C), etc etc
- fichiers spéciaux pour les périphériques (voilà la clé et encore je prends un raccourci).

C'est un peu plus claire? En admettant que tout est fichier, c'est plus simple.

Oui mais les dossiers

ce sont des répertoires qui ne sont qu'une représentation graphique, en réalité vos fichiers et vos "répertoires" sont des **i-nodes (index de noeuds)** contenant la description et caractéristique du/des dit/s fichier/s.

un environnement graphique, nous faisons en générale beaucoup d'action à la souris, et trouver certaines information devient (franchement) fastidieux. Nous allons commencer à voir quelques commandes.

Dans le gestionnaire de fichiers, nous pourrions dire que nous exécutons la commande ls , ls correspond à lister un contenu.

Nous avons relativement peu d'info, bien sur ls peut s'accompagner d'options, ici ls -l (méta-données).

Ls -l

```
\O/ [xtbushido@zalem]-[~]-[21:26]-->
>~ ls -l
total 498M
-rw-r--r--  1 xtbushido xtbushido    0 mars   8 18:42 4444
drwxr-xr-x  3 xtbushido xtbushido 4,0K janv. 30 12:37 backup/
drwxr-xr-x  2 xtbushido xtbushido 4,0K mars  20 08:55 bin/
-rw-r--r--  1 xtbushido xtbushido  100 mars  16 20:48 cendio
-rw-----  1 xtbushido xtbushido 241M mars  18 15:33 core
-rw-r--r--  1 root      root      660 janv. 30 12:51 defaultdhcpserver.bak
drwxr-xr-x  2 xtbushido xtbushido 4,0K mars  20 17:03 Desktop/
drwxr-xr-x  2 xtbushido xtbushido 4,0K janv. 23 13:33 documents/
drwxr-xr-x  4 xtbushido xtbushido 4,0K mars  19 16:28 downloads/
-rw-r--r--  1 root      root      744 janv. 29 14:51 eth0
-rw-r--r--  1 root      root      744 janv. 29 14:51 eth1
drwxr-xr-x  6 xtbushido xtbushido 4,0K mars   7 20:38 Ghost/
drwxr-xr-x  3 xtbushido xtbushido 4,0K mars   7 20:37 ghost_ordinatous_github/
drwx----- 6 xtbushido xtbushido 4,0K mars   7 20:08 GitKraken-master/
```

L'i-node contient les informations suivantes que l'on obtient ls -l (les méta données si vous préférez) :

- types de fichier: (-) simple fichier, (d ) répertoire, (l) lien, il en existe d'autres que vous ne rencontrerez que dans quelques cas particulier.
- droit d'accès (r) read= lecture, (w) write= lecture , et (x) execute=exécution
- nombre de liens
- UID identifiant de l'utilisateur propriétaire
- GID identifiant du groupe propriétaire
- taille
- Un horodatage (différent suivant l'argument utilisé.
- atime: dernier accès (commande stat nom du fichier ; ou ls -lu)
- mtime: dernière modification du fichier (commande ls -l)
- ctime: dernière modification de l'i-node (commande stat ;nom du fichier; ou ls -lc) c'est à dire modification des droits d'accès d'un fichier
- adresse
- 

Vos « répertoires » sont en fait des fichiers contenant eux mêmes une liste **d'i-nodes** auquel correspond les noms des **fichiers**, sinon nous aurions en permanence des listes énormes d'i-nodes et de nom de fichiers affichés, au fur-et-à-mesure que l'on travaille sur la machine.

En utilisant simplement les commandes suivantes:

## Ls -a

```
\O/ [xtbushido@zalem]-[~]-[21:23]-->
>~ ls -a
./          jekyll_ordinatous_github/
../         .local/
4444        lynis/
backup/     .mozilla/
.bash_history music/
.bash_logout NetworkManager-1.0.10/
.bashrc     nginx/
.bashrc.bak NoMachine/
.bashrc.swp nomachine_5.1.9_6_amd64.deb
bin/        .nx/
.cache/     Oracle_VM_VirtualBox_Extension_Pack-4.3.36-
105129.vbox-extpack
cendio     Oracle_VM_VirtualBox_Extension_Pack-4.3.36.vbox-
extpack
.config/    Oracle_VM_VirtualBox_Extension_Pack-5.0.16-
105871.vbox-extpack
.conkyrc    .pbuilderrc
```

## ls -li

- ls -li dans un répertoire, vous accédez à la liste des i-nodes et les noms qui y correspondent.

```
\O/ [xtbushido@zalem]-[~]-[21:24]-->
>~ ls -l
25962975 4444
25952366 backup/
25952367 bin/
25963419 cendio
25962246 core
25959299 defaultdhcpserver.bak
25952421 Desktop/
25952368 documents/
25952369 downloads/
25952623 eth0
25952624 eth1
26872516 Ghost/
26871413 ghost_ordinatous_github/
```

A noter que c'est le principe d'i-node qui nous permet d'ouvrir le même fichier, avec plusieurs applications en même temps.

### *Arborescence*

Prenez un arbre, imaginez-le à l'envers, les racines en haut. Sous Gnu/Linux c'est pareil il y a "la racine" écrit comme ceci / (le slash) ou encore . (le point).

Restons dans l'imagé où une branche représenterait un "répertoire" contenant lui même des feuilles, des "fichiers". Voici les principaux répertoires d'un systèmes, on dit qu'ils sont à la racine. Les dossiers utilisateurs sont dans le home lui même à la racine.

La plupart des systèmes Gnu/Linux et Unix-like respectent cette organisation, à quelques différence près mais reste très similaire. Lors du partitionnement nous choisissons 2 partitions pour séparer le système de l'espace utilisateur, nous voyons que ceci est parfaitement transparent pour l'utilisateur, nous pourrions même choisir des systèmes différents pour le système et l'espace utilisateur ce serait tout aussi transparent.

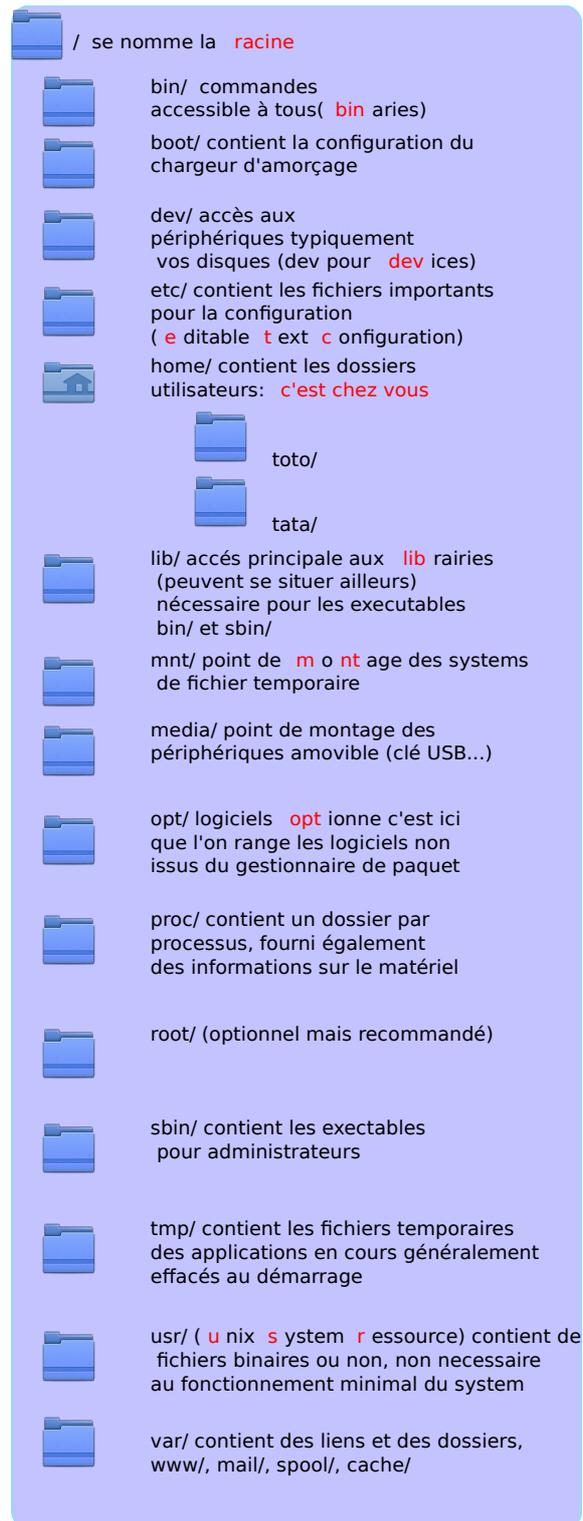
Voilà donc pour les fichiers, répertoires et l'organisation de votre système de fichiers, nous allons voir comment nous déplacer dans cette arborescence.

### *Man pages ls*

RTFM



## Hierarchy Files System



LA hiérarchisation vient du fait que pour démarrer les fichiers se chargent selon un certain ordre, c'est devenu réellement évident lorsqu'ils ne pouvaient plus tenir sur une disquette.

Il a donc fallut découper et monter tour à tour les « système de fichiers ».