

# Ubuntu-fr

[Communauté francophone d'utilisateurs d'Ubuntu](#)

Recherche : <input type="text" value="Recherche rapide...."/>	Chercher dans : <input type="text" value=""/>
Identifiant : <input type="text" value="Identifiant"/>	Mot de passe : <input type="password" value="....."/> <input type="text" value=""/> / <a href="#">inscription</a>

## Qu'est-ce qu'un système de fichiers ?

### Les qualifications d'un système de fichiers

La taille maximale d'un fichier

La taille maximale d'une partition

La gestion des droits d'accès aux fichiers et répertoires

La journalisation

Particularité des systèmes de fichiers \*nix : inode

### Comparaison de systèmes de fichiers

\* Systèmes de fichiers FAT32 et NTFS

### Quelques commandes utiles

### Plus d'informations

[installation](#), [système](#), [partitions](#)

## Les systèmes de fichiers sous Linux2(by somebody)

Le système de fichiers est un élément-clé de votre ordinateur. C'est lui qui gère l'organisation de vos données. Il existe de nombreux systèmes de fichiers, aux caractéristiques diverses qui offrent des avantages intéressants pour certains types d'utilisation... et des désavantages pour d'autres. Connaître les bases des systèmes de fichiers est essentiel dans le monde informatique.

### Qu'est-ce qu'un système de fichiers ?

Dans le cœur d'un ordinateur, tout est constitué de 1 et de 0, mais l'organisation de ces données n'est pas aussi simple. Un bit est un 1 ou un 0 ; un octet (*byte* en anglais) est composé de huit bits ; un kilo-octet binaire(*kibibyte*) est un groupe de 1024 octets ; un méga-octet binaire(*mebibyte*) se constitue de 1024 kilo-octets binaires ; et ainsi se poursuit la chaîne.

Un disque dur (*hard drive* ou *hard disk*) stocke toutes vos données : à chaque fois que vous enregistrez un fichier, vous écrivez des milliers de 1 et de 0 sur un disque métallique, le piquant littéralement et créant une indentation qui pourra plus tard être relue en tant que 1 ou 0 par votre ordinateur. En réalité, vous magnétisez des milliers de petits bâtons (que nous appellerons *clusters*) qui seront polarisés dans un sens ou dans un autre (N-S ou S-N) ; cette polarisation différente sera ré-interprétée plus tard par l'ordinateur en tant que 1 ou 0.

Il y a tellement de données sur un disque dur qu'il doit obligatoirement y avoir un moyen de les

organiser. C'est un peu comme les anciens classeurs de cartes d'identification de livres dans une bibliothèque municipale, dans lesquels tous les livres sont recensés : sans ces index, il serait impossible de retrouver facilement les livres que nous recherchons. Les bibliothèques utilisent pour la plupart le système décimal Dewey pour organiser les livres en sujets ; il existe aussi d'autres systèmes de classification arrivant à un résultat similaire, bien qu'aucun d'entre eux n'ait atteint la même popularité que le système de Dewey.

Les systèmes de fichiers jouent exactement le même rôle que ces index : organiser les fichiers de votre ordinateur sur votre disque dur de façon à pouvoir les retrouver lorsque vous en aurez besoin. Les systèmes de fichiers les plus utilisés à l'heure actuelle sont sûrement le FAT32 et le NTFS, qui sont les deux seuls systèmes de fichiers que Microsoft® Windows® peut nativement lire. Mais, tout comme il existe d'autres systèmes pour classer des livres dans une bibliothèque, il existe de nombreux autres systèmes de fichiers : ext2, ext3, ReiserFS, JFS, XFS,...

## Les qualifications d'un système de fichiers

De nombreux attributs différents sont nécessaires afin de définir un système de fichiers. Ils incluent entre autres la taille maximale que peut avoir un fichier dans ce système de fichiers, la taille maximale d'une partition et la journalisation ou non du système de fichiers.

Ce critère définit la taille maximale qu'un fichier quelconque enregistré sur un système de fichiers peut avoir. Ainsi, si vous possédez un fichier de 100 mégaoctets et que vous tentez de l'enregistrer sur un système de fichiers n'acceptant pas les fichiers plus grands que 90 mégaoctets, l'opération d'enregistrement ne pourra pas être complétée et vous disposerez d'un fichier corrompu, incomplet.

Ce critère définit la taille maximale que peut avoir une partition sur un disque dur. De nombreuses caractéristiques causent des limitations plus ou moins grandes quant à la taille d'une partition formatée dans un système de fichiers donné. Vous retrouverez, plus bas dans cette page, un tableau résumé des principaux systèmes de fichiers rencontrés dans le monde linuxien; nous y avons indiqué la taille maximale que peut avoir une partition selon chacun de ces systèmes de fichiers.

Ce critère définit s'il est possible ou non d'attribuer la possession d'un fichier ou d'un répertoire à un utilisateur et à un groupe d'utilisateurs, de même que de définir quelles actions les utilisateurs ont le droit d'effectuer sur les fichiers et répertoires, selon qu'ils sont propriétaires du fichier, membre du groupe propriétaire du fichier ou ni l'un ni l'autre. La possession et la gestion des droits d'accès associés s'effectue individuellement avec chaque fichier et répertoire.

Les **droits d'accès** que l'on trouve habituellement sont la lecture du fichier ou répertoire, l'écriture dans celui-ci et son exécution. Par exemple, l'utilisateur *toto* dispose des droits de lecture et d'exécution sur le répertoire `foo`, mais pas d'écriture; *toto* peut donc lancer les programmes présents dans ce répertoire et ouvrir les fichiers qu'il contient, mais ne peut pas les modifier ni en créer de nouveaux.

Parlons maintenant de la journalisation. Un système de fichiers journalisé est plus fiable lorsqu'on entre dans le domaine du stockage des données. Il a été expliqué plus haut ce qui se produit réellement lorsqu'un fichier est enregistré sur un disque dur (une suite de 1 et de 0 est inscrite sur le disque) ; mais que se produit-il si l'écriture de la chaîne est interrompue avant son terme (ce qui se produit, par exemple, lors d'une coupure de courant) ? Votre fichier devient « corrompu », incomplet.

Un système de fichiers journalisé travaille de façon à prévenir une telle corruption : lors de la sauvegarde d'un fichier, au lieu d'écrire immédiatement sur le disque dur les données à l'endroit exact où elles devraient être enregistrées, le système de fichiers écrit les données dans une autre partie du

disque dur et note les changements nécessaires dans un journal, et ensuite, en arrière-plan, il repasse chacune des entrées du journal et termine le travail commencé ; lorsque la tâche est accomplie, il raye la tâche de la liste.

Mais comment cela prévient-il la perte de données ? Prenons un exemple : disons que vous cliquez sur le bouton *Enregistrer* de votre logiciel de traitement de texte pour sauvegarder le fichier `foo.txt`. L'ordinateur écrit d'abord un « brouillon » de `foo.txt` dans une partie différente du disque dur et écrit le changement dans le journal du système de fichiers. Une fois cela effectué, l'ordinateur commence à retranscrire le fichier (*la suite de 1 et de 0*) à son endroit définitif sur le disque dur. Soudain, il survient une panne de courant ; alors la transcription du fichier est interrompue. Lorsque le courant revient, même si le « propre », la version finale de votre fichier est incomplète, vous possédez toujours votre brouillon dans le journal du système de fichiers ; l'ordinateur recommence donc la retranscription du fichier, écrasant les données corrompues.

Et si, par hasard, le courant était coupé lorsque l'ordinateur écrivait dans le journal, vous disposeriez toujours d'un brouillon précédemment écrit dans le journal pour récupérer votre travail.

« Mais, direz-vous, il est beaucoup plus avantageux d'utiliser un système de fichiers journalisé ! C'est bien plus sécurisé ! Pourquoi utiliserais-je un système de fichiers non journalisé ? » L'utilisation d'un journal requiert des capacités de stockage importantes sur vos périphériques ; ces systèmes de fichiers ne sont donc pas adaptés aux médias de faible capacité, telles les cartes mémoires (*memory sticks*) et les disquettes.

Les utilisateurs habitués de Microsoft Windows savent qu'il n'est normalement pas possible de modifier les propriétés d'un fichier (renommer, déplacer, supprimer...) quand celui-ci est ouvert par un programme ; cette restriction n'existe pas sur les systèmes de fichiers de type Unix (`ext2`, `ext3`, `ReiserFS`...). La raison est que sur les systèmes de fichiers *\*nix*, les fichiers sont indexés selon un numéro, appelé *inode* ou *i-node*, et que chaque inode possède de nombreux attributs associés à lui, tels les droits d'accès, le nom du fichier, etc. Lorsque vous supprimez un fichier, ce qui se produit réellement est que l'inode est « délié » du système de fichiers (c'est-à-dire qu'il n'est plus indexé), mais si des programmes ont un lien avec l'inode (par exemple, si un document texte est ouvert dans un éditeur de texte, il a un lien avec cet éditeur de texte), le fichier auquel l'inode est associé continue d'exister dans le système d'exploitation et continue d'être mis à jour. Un fichier n'est réellement effacé que lorsque tous les liens avec son inode ont été coupés (*et encore, les données résident encore sur le disque dur, mais comme elles ne sont plus indexées, il est difficile de les récupérer*). Ce que tout ceci signifie est que vous pouvez supprimer des programmes alors qu'ils sont en fonction sans faire planter votre système d'exploitation, renommer ou déplacer des fichiers avant qu'ils aient fini d'être téléchargés sans les corrompre et supprimer un fichier alors qu'il est encore ouvert dans un logiciel.

## Comparaison de systèmes de fichiers

Ci-dessous se trouve une très brève comparaison des systèmes de fichiers les plus communs et les plus utilisés dans le monde linuxien.

Nom du système de fichiers	Taille maximale d'un fichier	Taille maximale d'une partition	Journalisée ou non ?	Gestion des droits d'accès?	Notes
					Extended File System est le système de fichiers natif de Linux. En ses versions 1 et

<b>ext2fs</b> (Extended File System)	2 TiB	4 TiB	Non	Oui	2, on peut le considérer comme désuet, car il ne dispose pas de la journalisation. Ext2 peut tout de même s'avérer utile sur des disquettes 3½ et sur les autres périphériques dont l'espace de stockage est restreint, car aucun espace ne doit être réservé à un journal.
<b>ext3fs</b>	2 TiB	4 TiB	Oui	Oui	ext3 est essentiellement ext2 avec la gestion de la journalisation. Il est possible de <b>passer une partition formatée en ext2 vers le système de fichiers ext3</b> (et <i>vice versa</i> ) sans formatage.
<b>ext4fs</b>	16 TiB	1 EiB	Oui	Oui	ext4 est le successeur du système de fichiers ext3. Il est cependant considéré par ses propres concepteurs comme une solution intermédiaire en attendant le vrai système de nouvelle génération que sera Btrfs
<b>ReiserFS</b>	8 TiB	16 TiB	Oui	Oui	Développé par Hans Reiser et la société Namesys, ReiserFS est reconnu particulièrement pour bien gérer les fichiers de moins de 4 ko. Un avantage du ReiserFS, par rapport à ext3, est qu'il ne nécessite pas une hiérarchisation aussi poussée: il s'avère intéressant pour le stockage de plusieurs fichiers temporaires provenant d'Internet. Par contre, ReiserFS n'est pas recommandé pour les ordinateurs portables, car le disque dur tourne en permanence, ce qui consomme beaucoup d'énergie.
<b>FAT</b> (File Allocation Table)	2 GiB	2 GiB	Non	Non*	Développé par Microsoft, ce système de fichiers se rencontre moins fréquemment aujourd'hui. Il reste néanmoins utilisé sur les disquettes 3½ formatées sous Windows et devrait être utilisé sous Linux si une disquette doit aussi être lue sous Windows. Il est aussi utilisé par plusieurs constructeurs comme système de fichiers pour cartes mémoires ( <i>memory sticks</i> ), car, bien documenté, ce système de fichiers reste le plus universellement utilisé et accessible.
					Ce système de fichiers, aussi créé par Microsoft, est une évolution de son

<b>FAT32</b>	4 GiB	8 TiB	Non	Non*	prédécesseur. Depuis ses versions 2000 SP4 et XP, Windows ne peut pas formater (ou bloque volontairement le formatage) une partition en FAT32 d'une taille supérieure à 32 Go. Cette limitation ne s'applique pas sous Linux, de même qu'avec des versions antérieures de Windows. Une partition FAT32 d'une taille supérieure à 32 Go déjà formatée pourra être lue par Windows, peu importe sa version.
<b>NTFS</b> (New Technology File System)	Limitée par la taille de la partition	2 TiB	Oui	Oui*	Ce système de fichiers a aussi été développé par Microsoft, et il reste très peu documenté. L'écriture depuis Linux sur ce système de fichiers est stable à l'aide du pilote <b>ntfs-3g</b> . Ce pilote est inclus de base dans <b>Ubuntu 7.10</b> , et disponible en paquets dans les <b>dépôts</b> pour les versions antérieures.

**EiB = Exbiocets (1024 pébiocets) :: PiB = Pébiocet (1024 tébiocet) :: TiB = Tébiocet (1024 gibiocets) :: GiB = Gibiocet (1024 mibiocets) [ [Plus d'informations sur la Wikipedia](#) ]**

Ci-dessus vous voyez aussi une comparaison de deux autres des principaux attributs d'un système de fichiers, à savoir la taille maximale d'un fichier sur une partition et la taille la plus grande que peut avoir une partition avec un système de fichiers particulier.

Parmi les systèmes de fichiers précédents, les seuls sur lesquels on ne peut pas installer Linux sont le NTFS, la FAT et la FAT32. En théorie, il pourrait être possible d'installer Linux sur l'un de ces systèmes de fichiers, mais comme ce système de fichiers ne gère pas les droits d'accès, il résulterait un système d'exploitation hautement non-sécurisé.

Les systèmes de fichiers créés par Microsoft (FAT, FAT32 et NTFS) ne gèrent pas les droits d'accès aux fichiers comme les systèmes de fichiers de type Unix (ext2, ext3, ReiserFS...). Toutefois, pour être accessible sous Linux, un système de fichiers doit disposer du système de droits d'accès; pour contourner cette limitation, les droits d'accès de type Unix sont émulsés sur ces systèmes de fichiers. Cette émulation est effectuée au montage d'une partition de disque dur formatée selon l'un de ces systèmes de fichiers, et ces droits sont immuables tout le temps que ce système de fichiers est en cours d'utilisation.

Il n'est pas possible de modifier les permissions ou les propriétés d'un fichier ou d'un dossier situé dans un système de fichiers NTFS ou FAT32. Le seul moyen de changer ce comportement est de démonter le système de fichiers puis le remonter avec des options de montage différentes.

Sous Microsoft® Windows®, le système de fichiers NTFS gère les droits d'accès basé sur une liste de contrôle (ACL) qui n'est pas prise en compte sous Linux. Linux peut aussi prendre en charge une forme d'ACL, mais celle-ci est différente de celle implantée dans NTFS. Pour cette raison, les ACL des systèmes de fichiers NTFS sont simplement ignorées.

**«Je désire créer une partition d'échange de fichiers entre Ubuntu et Microsoft® Windows®. Quel svstème de fichiers devrais-je préférer ?»**

Vous devriez faire une partition d'échange formatée en NTFS. Elle sera pleinement gérée par Windows NT/2000/XP/Vista et Ubuntu.

Le désavantage de cette option est que les droits d'accès aux fichiers ne sont pas disponibles, et donc l'ensemble des données seront accessibles sous Ubuntu à tous les utilisateurs. Ceci peut être déroutant sur des systèmes multi-utilisateurs qui doivent partager des fichiers sensibles.

Il n'existe aucun moyen de contourner ce problème, vous pouvez créer une partition d'échange formatée en ext3, mais le problème se retrouvera sous Windows ce coup ci, avec des problèmes de sécurité plus importants encore.

## Quelques commandes utiles

Afficher le système de fichier d'une partition

- partition montée

```
df -h
```

- partition quelconque (nécessite que le paquet **parted** soit installé et que le disque /dev/sda existe)

```
sudo parted /dev/sda print all
```

## Plus d'informations

- [Wikipedia : Comparison of File Systems](#) (en)
- [Système de fichiers](#) (fr)
- [Ext2](#), [Ext3](#), [Ext4](#), [ReiserFS](#) (fr)
- [Comparaison de performance](#) (en)
- [Vérifier son système de fichiers sous Linux](#)
- [OpenDedup \(SDFS\)](#) : système de fichier supportant la déduplication.

---

Contributeurs: [AlexandreP,fnx](#)

Basé sur [Linux Filesystems Explained](#)

systeme\_de\_fichiers.txt · Dernière modification: Le 13/02/2012, 16:12 par Canigou 66

Le contenu de ce wiki est sous licence : [CC BY-SA v3.0](#)

- [Contact](#)
- [Ressources](#)
- [Licences](#)
  
- [Propulsé par Dokuwiki](#)