Partitionnement et systeme de fichiers Documentation

Version 0.0.1

Geoffroy Berry, Mohamed Missaoui

11 November 2016

CHAPITRE 1

LES SYSTEMES DE FICHIERS :

Quesqu'un systeme de fichier ?

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,. . .

Système de fichier journalisé

Le système de fichiers journalisé est un système de fichiers tolérant/résistant aux pannes qui permet d'assurer l'intégrité des données en cas de problème matériel, de panne de courant (ou débranchement à chaud) ou d'arrêt brutal du système. Cette fonctionnalité est assurée par la tenue d'un journal référençant les opérations d'écriture sur le support physique avant que ce dernier ne soit réellement mis à jour. Le système de fichiers doit permettre une reprise d'activité à la suite d'une coupure brutale, telle un arrêt électrique. Les métadonnées doivent alors rester cohérentes et à jour. La

journalisation permet d'optimiser le contrôle d'intégrité du système de fichiers, réduisant ainsi le temps de redémarrage du système, critère important dans les environnements qui ont besoin d'une haute disponibilité.

La journalisation du système de fichier assure la cohérence des données en utilisant un journal. Ce journal est un fichier spécial qui enregistre les changements destinés au système de fichier, dans une mémoire circulaire. À intervalles réguliers, le journal est appliqué sur le système de fichier. Si une interruption électrique intervient, le journal peut être utilisé comme point de départ afin de récupérer les informations non sauvegardées, et ainsi assurer l'intégrité des données du système de fichier.

Il y a 2 méthode de journalisation, celle avec des journaux physique et celle avec des journaux logique.

Journaux physique : Le journal physique enregistre les modifications de données sur le support avant que celles-ci soient opérées. Les entrées contenues dans ce journal comprennent les données à écrire, ainsi qu'une somme de contrôle, afin d'assurer l'intégrité de celles-ci. Ceci permet d'éviter d'écrire des données pour lesquelles l'entrée du journal est incomplète. Cette méthode pénalise les performances, car chaque écriture nécessite une double écriture sur le support physique, une pour le journal, et une pour les données effectives. Elle est néanmoins acceptée en raison de la garantie de la cohérence des données qu'elle permet.

Journaux logique : Le journal logique ne stocke que les métadonnées, sacrifiant la tolérance aux pannes pour de meilleures performances. Il permet lui aussi le rejeu des opérations, mais peut lier des métadonnées journalisées à des données non journalisées, causant ainsi une corruption des données.

Exemple de système de fichier journalisé :

Ext3, Ext4 ou encore BTRFS sont des système de fichier journalisé.

Ext3 : est une extension de Ext2, auquel on a ajouté une fonction de journalisation. Il est donc

possible de convertir une partition ext2 en une partition ext3 de manière simple, la seule modification à ajouter étant le journal. Il n'est utilisable que sur les systèmes Linux. Avec ce système de fichier l'utilisateur à la possibilité de choisir parmi trois mode de journalisation qui sont le mode Journal(correspond à la journalisation physique), mode Writeback, mode Ordered(mode par défaut).

Ext4 est son successeur, il permet la sauvegarde de fichier plus volumineux et est plus fiable grace à l'utilisation du checksum à chaque ecriture dans le journal et est plus performant sur les fichier volumineux.

BTRFS est un système de fichiers des années 2010 fondé sur le Copy-On-Write (copie sur écriture) sous licence GNU GPL, développé conjointement par Oracle, Red Hat, Fujitsu, Intel, SUSE, STRATO AG (en) et autres. En 2012, alors qu'il n'était pas encore considéré comme tout-à-fait stable7, un effort intense de développement et de test est fourni par la communauté afin de faire de Btrfs le successeur de ext4 et ext3, systèmes de fichiers habituels des distributions Linux. OpenSuse 13.2 propose dès son lancement Btrfs par défaut pour la partition racine afin d'assurer la sécurité et laisse le choix entre ext4 et XFS.

Btrfs offre les fonctionnalités suivantes absentes d'autres systèmes de fichiers :

- Instantané (snapshots),
- somme de contrôle (CheckSum).

Ces caractéristiques sont importantes pour les systèmes Linux, serveurs comme postes clients, car les tailles de stockage comme les configurations tendent à augmenter et à se complexifier. La technique de l'instantané, en particulier, garantit de pouvoir faire une sauvegarde cohérente des fichiers du système tels qu'ils étaient au moment précis de l'instantané, même si la sauvegarde dure plusieurs heures et que de nombreux fichiers sont modifiés entre temps.

Système de fichier non journalisé

Vous l'aurez compris un système de fichier non journalisé est un systeme de fichier moin tolérent au panne. Suite à un incident il existe 2 procédé :

- Procédé des mises à jour prudentes qui, à un instant précis, rend le système de fichiers cohérent.
- Procédé des programmes "éboueurs".

Le premier procédé consiste, à un instant précis, à ne pas laisser le système de fichiers incohérent. Ce procédé est très problématique car ne pas passer par une étape laissant le système dans une position incohérente est difficile. Par exemple, lorsqu'on crée un fichier, il faut ajouter ce fichier dans le répertoire et créer l'inode de ce répertoire. Ces opérations ne sont pas possibles simultanément et donc il se passe toujours un instant où l'inode est affecté au fichier mais pas au répertoire, ou l'inverse.

Le deuxième procédé est celui employé par le système de fichier ext2. Il utilise l'utilitaire fsck pour vérifier l'intégrité et réparer les données. L'utilitaire est lancé au démarrage si le système détecte un problème sur le système de fichiers. À la suite de cela, l'utilitaire vérifie l'intégralité du disque. L'inconvénient majeur de cette méthode est le temps d'exécution qui est proportionnel à la taille du support. Pour procéder à la vérification, le déroulement de l'utilitaire fsck se décompose en 6 phases :

- Passe 1 : vérification des i-nœuds, des blocs et des tailles
- Passe 2 : vérification de la structure des répertoires
- Passe 3 : vérification de la connectivité des répertoires
- Passe 3A : optimisation des répertoires
- Passe 4 : vérification des compteurs de référence
- Passe 5 : vérification de l'information du sommaire de groupe

On remarque que le temps d'exécution dépend de la taille du disque. De ce fait, plus la taille du support est importante, plus l'exécution de fsck est longue.

Il était donc intéressant de pouvoir mélanger les deux procédés pour avoir une méthode de récupération beaucoup plus rapide.

Exemple de système de fichier non journalisé :

lExt2 : Les fonctionnalités standard de permettent d'accéder à des partitions de 4 téraoctets (1 téraoctets = 1 024 gigaoctets), alors que la version ext1 ne permettait que des partitions de 2 gigaoctets (1 gigaoctets = 1 024 mégaoctets).

La taille maximale des fichiers avec un système ext2 standard est de 2 gigaoctets. De plus, lors de la création du système de fichiers, le système réserve une certaine quantité d'espace pour le super-utilisateur (root), en général 5 %. Ceci permet au super-utilisateur de pouvoir se connecter sur le système et de faire des tâches administratives quand le système de fichiers est plein pour les utilisateurs. Ext2Fs gère aussi les noms de fichiers longs (255 caractères) et prend en compte tous les caractères excepté "NUL" et "/".

Les lacunes : Bien qu'ayant été conçu de façon intelligente, ce système de fichiers possède des limites inacceptables pour des machines nécessitant une disponibilité importante. Le temps de vérification du système de fichiers est proportionnel à sa taille, puisque le système va lire tous les inodes et vérifier leurs cohérences. Par conséquent, plus le système de fichiers est important, et plus cette vérification est longue. De plus, un crash peut détruire, ou rendre inaccessibles des données pourtant déjà présentes sur le disque.

Une autre limitation importante vient de l'absence d'ACL (Access Control Lists soit listes de contrôle d'accès) utilisées au niveau du système de fichiers, de manière native par Linux. La présence d'ACL permet en effet de gérer de manière très fine les droits d'accès aux fichiers.

On peut donc considérer que ces aspects sont les caractéristiques qui vont permettre la création de nouveaux systèmes de fichiers. Afin de maintenir la plus haute disponibilité possible, ceux-ci devront permettre de redémarrer rapidement après un arrêt brutal, dans un temps qui n'est pas en relation directe avec leur taille. De plus, les coûts de maintenance de ces systèmes devront être les plus faibles possibles, autant au niveau du temps qu'au niveau des ressources matérielles. Le principal problème rencontré aujourd'hui reste la sauvegarde des données du système de fichiers. En effet, si on veut effectuer une sauvegarde rigoureuse du système de fichiers, il faut fixer une date précise à laquelle le système de fichiers sera sauvegardé. Cependant, cette opération n'étant pas instantanée, la modification de ce dernier durant la sauvegarde peut conduire, dans certains cas, à une incohérence de fichiers entre eux. La meilleure solution serait d'arrêter le système de fichiers pour effectuer la sauvegarde, ce qui n'est pas acceptable dans la plupart des cas. Il faudrait donc pouvoir figer l'aspect du système de fichiers, sans empêcher les utilisateurs de travailler dessus. Cette solution peut également apporter la fonctionnalité de pouvoir "défaire" une opération malencontreuse, comme un effacement accidentel d'un fichier, ou l'écrasement d'un fichier par un autre. Ensuite, il sera sans doute très utile de pouvoir redimensionner la taille du système de fichiers d'une manière souple, donc non-destructive.

Il reste également l'aspect de la confidentialité des données des utilisateurs sur le système de fichiers. Il faudrait qu'une personne ayant un accès physique sur le média ne soit pas capable de pouvoir reconstituer l'ensemble des données présentes sur ce dernier.

Système de fichier distribué et en réseau :

Le Network File System est une technologie (un mécanisme ou un protocole pour faire simple) permettant d'accéder aux fichiers présents sur des machines distantes exactement comme s'ils sont locaux.

Sous les systèmes UNIX (y compris linux), il est considéré comme un système de fichiers à part entière.

Proposé par Sun Microsystem, le protocole NFS s'est imposé comme un standard dans le monde UNIX mais dépasse le simple cadre du monde UNIX.

En effet il s'appuie sur une représentation standard des objets proposée par le protocole XDR et le mécanisme d'appels de procédures distantes implémenté par le protocole RPC (Remote Procedure Call). Tous ces deux protocoles sont aussi l'oeuvre de Sun Microsystem.

En fait NFS est composé de quatre protocoles distincts qui reposent tous sur les RPC et donc sur le programme rpc.portmap. Un des rôles de ce programme est de convertir les numéros de programmes RPC en numéros de ports. Quand un serveur RPC démarre, il va préciser à portmap quel port il utilisera et les numéros de programmes RPC qu'il gère. Quand un client souhaite envoyer une requête RPC vers un numéro de programme donné, il contacte d'abord le serveur portmap pour obtenir le numéro de port sur lequel tourne le programme souhaité. Ensuite, il adresse les paquets RPC au port concerné.

les commandes :

NFS côté Serveur :

```
apt-get install nfs-kernel-server
```

Editer le fichier

```
nano /etc/exports
```

Disont que l'on veut partager le repertoire /home/test à la machine debian1.domaine.org

```
/home/test debian1.domaine.org(rw,root_squash)
```

rw = read,write root_squash = le root de la machine debian1 n'aura pas les droit root sur /home/test.

```
/etc/init.d/nfs-kernel-server reload
```

NFS côté Client :

Pour monter le répertoire /home/test2/ partagé par la machine dont le nom DNS est debian2.domaine.org dans le répertoire /mnt/test déjà créé, utilisez la commande mount :

```
mount -t nfs debian2.domaine.org:/home/test2 mnt/test
```

umount de /mnt/test une fois que vous n'avez plus besoin du partage.

Disque virtuel :

Un disque dur virtuel fournit un espace de stockage pour une machine virtuelle. Au sein de la machine virtuelle, le disque dur virtuel est représenté comme un disque physique. La machine virtuelle l'utilise également comme s'il s'agissait d'un disque physique. Techniquement, le disque dur virtuel est un fichier résidant sur un disque physique

auquel le système d'exploitation hôte peut accéder. Sur le disque physique, le fichier du disque dur virtuel est stocké au format .vhd. En règle générale, vous pouvez stocker un fichier .vhd sur n'importe quel type de dispositif de stockage accessible par le système d'exploitation hôte.

Les types de disques et les exigences de stockage sont les suivants :

- Disque dur virtuel de taille dynamique : Ce type requiert un minimum de 8 Mo d'espace libre sur le support de stockage physique. La taille du disque (et du fichier .vhd) croît en proportion de l'utilisation du disque, jusqu'à la limite spécifiée lors de la création du disque.
- Disque dur virtuel fixe : Ce type de disque requiert un espace de stockage physique équivalent à la taille spécifiée lors de sa création. La taille du fichier .vhd est identique à celle du disque dur virtuel et reste inchangée.
- Disque dur virtuel de différenciation : Ce type requiert un espace de stockage physique réduit lors de la création du disque, puis un espace de plus en plus important à mesure que la taille du disque augmente. La taille maximale d'un disque de différenciation est régie par la taille maximale de son disque dur parent.

Snapshot :

En informatique, un instantané est l'état d'un système à un instant donné. On utilise particulièrement cette notion dans le cadre des systèmes de fichiers, des bases de données ou des machines virtuelles.

Une sauvegarde complète d'une grande quantité de données peut être longue à exécuter. Par ailleurs, celle-ci nécessite généralement le verrouillage des données à sauvegarder pour empêcher que des données en cours de sauvegarde ne soient modifiées et ne produisent ainsi une sauvegarde qui ne soit pas représentative du système à un instant donné. Cela ne pose pas de problème particulier pour effectuer des sauvegardes sur des ordinateurs personnels ou des serveurs internes de petites entreprises, puisqu'il est parfaitement acceptable qu'un tel système puisse être indisponible durant un certain laps de temps. Cependant, sur des serveurs utilisés à grande échelle, exigeant de grands uptimes, on ne peut recourir à de telles méthodes.

Pour éviter cela, on peut créer un instantané. On réserve d'abord sur le périphérique cible un espace de taille suffisante pour contenir les données à sauvegarder. Les données à copier et l'instantané ainsi initialisé sont divisés en blocs de données, de taille fixe ou variable ; dans l'instantané, aucune donnée n'est encore présente, mais chaque bloc contient un pointeur vers le bloc correspondant sur le système d'origine. Lorsqu'un bloc des données à sauvegarder est modifié, les données d'origine de ce bloc sont copiées dans le bloc correspondant de l'instantané avant que les modifications ne soient écrites.

Cette méthode ne permet cependant pas de protéger les données contre une panne de disque dur par exemple, puisque seules les données qui auront été modifiées seront copiées dans l'instantané. Pour pallier cela, on commence parfois par copier la totalité du système à sauvegarder par la méthode traditionnelle, après quoi on peut créer des instantanés successifs dont les parties non modifiées pointeront non pas vers les données originales mais vers la première copie complète.

Les commande :

Pour faire une sauvegarde complete d'une machine on peut utiliser la commande rsync qui peut nous copier le contenu de la machine dans un repertoire.

```
rsync -aAXv
--exclude={"/dev/*","/proc/*","/sys/*","/tmp/*","/run/*","/mnt/*","/media/*","/lost+found"
/* /mnt/backup/
```

Pour crée des snapshot régulière on peut utliser le logiciel rsnapshot.

```
apt-get install rsnapshot
nano /etc/rsnapshot.conf
```

Dans le fichier de conf on peut modifier l'interval de sauvegarde, le repertoire etc..

On peut automatiser la création de snapshot.

Pour l'activer manuellement on utilise la commande :

```
systemctl start rsnapshot@hourly
```

cette commande permet de prendre des snapshot toute les heurs.

SWAP

Le swap est appelé en français espace ou partition d'échange. | Linux divise la RAM (mémoire vive) physique en blocs de mémoire appelés pages. Le swapping est le procédé par lequel une page de mémoire est copiée vers un espace prédéfini sur le disque dur, appelé swap (espace d'échange), permettant de libérer cette page mémoire. Les tailles combinées de la mémoire physique et de l'espace d'échange correspondent à la quantité de mémoire virtuelle disponible.

L'espace d'échange, aussi connu sous le nom de mémoire virtuelle, peut être présent sous forme de partition ou de fichier, et peut être créé ou supprimé à n'importe quel moment. L'espace d'échange est recommandé pour les machines ayant moins de 1 Go de mémoire vive (RAM), mais dépend surtout des préférences personnelles du propriétaire de l'ordinateur. La seule contrainte vient lors de l'activation de l'hibernation, nécessitant un fichier d'échange de taille supérieur ou égale à la taille de mémoire vive physiquement installée.

FRAGMENTATION

Il n'y a pas vraiment de fragmentation sous linux étant donnée que les systeme de fichier sont fait de tel sorte à ne pas se fragmenter.

Les systèmes Ext2, Ext3, et Ext4 (systèmes de fichiers utilisés par la plupart des distributions Linux actuelles) organisent et répartissent les fichiers sur le disque dur de manière intelligente. Au lieu de placer les fichiers près les uns des autres sur le disque dur, ces systèmes de fichiers dispersent les fichiers sur tout le disque, laissant une grande quantité d'espace libre entre eux. Quand un fichier est modifié et que sa taille augmente, il y a généralement beaucoup d'espace libre pour enregistrer le fichier sur des secteurs contiguë. S'il y a fragmentation, le système de

fichiers tentera de déplacer les fichiers afin de réduire la fragmentation en utilisation normale sans avoir besoin d'un utilitaire de défragmentation.

En raison de la façon dont ces systèmes fonctionnent, il y aura fragmentation importante lorsque votre disque sur se remplit à une capacité de 95% (voire 80%). Cependant, le système de fichiers est conçu pour éviter la fragmentation en utilisation normale.

Si vous avez des problèmes avec la fragmentation sous Linux, vous avez probablement besoin d'un plus gros disque dur. Si vous avez réellement besoin de défragmenter un système de fichiers Linux, la façon la plus simple et sans doute la plus fiable de faire est de copier tous les fichiers de la partition sur un autre média, effacer les fichiers de la partition, puis copier les fichiers à nouveau sur la partition. Le système de fichiers saura répartir les fichiers intelligemment sur le disque.

Vous pouvez vérifier la fragmentation de votre disque à l'aide de la commande `fsck`.

FSTAB

Le fichier fstab (file systems table) est la table des différents systèmes de fichiers sur un ordinateur sous Unix/Linux : il contient une liste des disques utilisés au démarrage et des partitions de ces disques. Pour chaque partition, il indique comment elle sera utilisée et intégrée à l'arborescence du système de fichiers global (c'est-à-dire le point de montage). Il se trouve généralement à /etc/fstab.

Voici un exemple de fichier fstab :

<file system> <dir> <type> <options> <dump> <pass>
/dev/sda1 / ext4 defaults 1 1
/dev/hdxx /usr ext4 defaults 1 1
/dev/sda5 swap swap defaults 0 0

<file systems> - définit l'équipement de stockage

<dir> - indique à la commande de montage où monter le <file system>.

<type> - définit le type de système de fichiers pour monter le support ou la partition. Un grand nombre de systèmes de fichiers sont supportés, par exemple : ext2, ext3, ext4, reiserfs, xfs, jfs, smbfs, iso9660, vfat, ntfs, swap, and auto. Le type 'auto' laisse la commande de montage deviner quel type de système de fichiers utilisés.

<options> - définit des options particulières pour les systèmes de fichier. Certaines options sont pour le système de fichiers lui même.(auto,exec,ro,rw...)

<dump> - est utilisé par l'utilitaire dump pour décider quand faire des sauvegardes. Quand il est installé, dump vérifie le chiffre inscrit et décide si le système de fichiers doit être sauvegardé. Les valeurs possibles sont 0 et 1. Si 0, dump va ignorer le système de fichier, si 1, dump fera une sauvegarde.

<pass> fsck lit le chiffre <pass> et détermine dans quel ordre les systèmes de fichiers vont être vérifiés. Le champ peut prendre les valeurs 0,1 et 2. Le système de fichiers root devra avoir la priorité la plus haute : 1, tout les autres systèmes que vous voulez vérifier devront avoir un 2. Les systèmes de fichiers avec un <pass> à 0 ne seront pas vérifier par l'utilitaire fsck.

LES COMMANDES

mount :

La commande mount permet de relier une partition ou un périphérique à un répertoire, répertoire par lequel les données présentes sur la partition ou le périphérique sont accessibles.
Pour monter un périphérique ou une partition avec la commande mount, il faut indiquer :

- le type du système de fichiers par l'option -t
- le fichier spécial représentant le périphérique ou la partition (généralement /dev/*) ;
- le répertoire de montage.

Certaines indications peuvent être omises lorsqu'elles sont spécifiées dans le fichier de configuration listant les points de montage par défaut (/etc/fstab sous Linux). On peut omettre le type de système de fichiers si la version de mount utilisée est assez « intelligente ». Par contre, même en l'indiquant, on ne pourra jamais monter un système de fichiers que le noyau Unix ne sait pas gérer (parce qu'il n'a pas été configuré pour l'utiliser par exemple).

Lorsque le montage a réussi, une mise à jour est effectuée dans un fichier système recensant les montages en cours (fichier /etc/mtab sous Linux). L'option -n de mount permet d'éviter cette mise à jour dans des cas bien particuliers où le montage échouerait pour cette raison (si l'on travaille sur un système de fichiers chrooté en lecture seule par exemple).

On peut également sous les Unix modernes monter des fichiers qui constituent un système de fichiers à eux-seuls (loopback), grâce à l'option -loop. Ceci est particulièrement utile dans le cas d'images représentant des disquettes, CD-ROM, DVD. Les commandes dd et mkisofs peuvent aider à fabriquer de tels fichiers.

Il est possible, sous certaines configurations, de monter (recouvrement total ou partiel) par dessus d'autres systèmes déjà montés.

Exemple de commande avec mount :

Disont que l'on vient de modifier le fichier /etc/fstab pour qu'il monte automatiquement une certaine partition au démmarage. Pas besoin de redemarrer l'ordinateur on peut utiliser la commande :

```
mount –a
```

Si on veut monter le contenu de /dev/sda1 en ext2 dans le repertoire /mnt/test :

```
mount –t ext2 /dev/sda1 /mnt/test
```

Par exemple, la commande ci-dessous permet de lire un CD-ROM en montant le périphérique /dev/cdrom sur /media/cdrom en indiquant que le système de fichiers est ISO 9660.

```
# mount -t iso9660 -o loop/dev/cdrom /mnt/cdrom
```

L'option bind permet de lié un repertoire à un autre.

```
mount --bind /dev/sda1 /home/test
```

umount :

Pour démonter une partition ou un périphérique, il faut utiliser la commande umount. Par exemple :

```
umount /mnt/cdrom
```

Le démontage ne marche que si la partition n'est pas utilisée, à savoir :

- aucun fichier n'est en train d'être lu ou écrit sur la partition ;
- aucun processus n'a son répertoire de travail sur la partition.

Si le démontage est refusé, on peut utiliser la commande fuser/lsof pour savoir quels processus l'utilisent. Par exemple (si le démontage de /media/cdrom est refusé) :

```
fuser /mnt/cdrom
```

Lorsque le démontage a eu lieu, le fichier /etc/mtab (Linux) est mis à jour.

fdisk :

L'utilitaire fdisk de Linux permet de créer des partitions sur un disque dur.

Le partitionnement avec fdisk peut entraîner la perte de toutes les données présentes sur le disque sur lequel vous effectuez les opérations.

Il prend comme argument le chemin du fichier spécial associé au disque. À défaut, il utilisera le premier disque trouvé.

Cet utilitaire était un programme indépendant GNU fdisk (jusqu'à sa version 2.0 de décembre 2011) mais a été depuis intégré dans la suite GNU util-linux.

Le nom des périphériques de stockage varie selon qu'il s'agit de disques SCSI ou IDE.

— Pour des disques IDE, le premier sera nommé hda, le second hdb...
— Pour des disques SCSI, le premier s'appellera sda, le second sdb...

Exemple de commande avec fdisk :

Pour lister les partition on peut utiliser la comande :

```
fdisk -l
```

la commande fdisk peut être utiliser avec la commande blkid pour obtenir des information sur le complémentaire sur le disk tel que son UUiD ou ses attributs.

```
blkid /dev/sda1
```

Pour utiliser l'outil de partitionnement on utilise la la commande fdisk et on choisis le disk que l'on veut partitionner.

```
fdisk /dev/sda
```

Les commandes de fdisk sont appelées par des touches, voici la liste des plus importantes :

— d : destruction d'une partition
— l : liste des types de partitions
— m : impression du menu en cours
— n : création d'une nouvelle partition
— p : affichage des partitions
— q : sortie de fdisk sans sauvegarde des paramètres
— t : modification du type de partition
— v : vérification de la table des partitions
— w : sauvegarde des modifications et sortie de fdisk

Si ont veut juste crée une partiton de type ext nous avons juste besoin de crée une partiton via fdisk. puis d'utilisé mkfs pour lui donner le type de la partiton.

```
mkfs.ext4 /dev/sdb1
```

on pourra vérifier la crée de la partiton avec la commande :

```
blkid
```

On aura plus qu'a monter la partition à l'aide de la commande mount et de lui affecter un repertoire.
Pour ne pas avoir besoin de remonter le disque à chaque démarrage il faudra changer le fstab.

```
mount -t ext /dev/sdb1 /mnt/test
```

Ajouter dans le fichier fstab :

```
/dev/sdb1 /mnt/test ext4 defaults 0 0
```

source :

http://lea-linux.org/documentations/Fstab
http://www.linux-france.org/article/cel/alcove/etude-fichiers.html/ch1.html
http://www.tecmint.com/fdisk-commands-to-manage-linux-disk-partitions/
https://www.ibisc.univ-evry.fr/~petit/Enseignement/AdminSystem/IUP-ASR/2004-2005/snapshot.pdf
http://www.misfu.com/installation-configuration-nfs-linux.html
https://fr.wikipedia.org/wiki/Network_File_System
https://doc.ubuntu-fr.org/btrfs
https://fr.wikipedia.org/wiki/Network_File_System
https://fr.wikipedia.org/wiki/Syst%C3%A8me_de_fichiers_distribu%C3%A9
https://fr.wikipedia.org/wiki/Point_de_montage
https://fr.wikipedia.org/wiki/Ext4
https://fr.wikipedia.org/wiki/Ext2
https://fr.wikipedia.org/wiki/Disque_virtuel
https://fr.wikipedia.org/wiki/Instantan%C3%A9_(informatique)
https://fr.wikipedia.org/wiki/Fstab
https://fr.wikipedia.org/wiki/Fdisk
https://fr.wikipedia.org/wiki/D%C3%A9fragmentation_(informatique)
https://fr.wikipedia.org/wiki/XFS