

ROYAUME DU MAROC

مكتب التكوين المهني وإنعاش الشغل

Office de la Formation Professionnelle et de la Promotion du Travail

Les systèmes de fichiers sous Gnu/Linux

www.ofppt.info



**DIRECTION RECHERCHE ET INGENIERIE DE FORMATION
SECTEUR NTIC**

Sommaire

1.	Introduction	2
2.	Création des systèmes de fichiers.....	2
3.	Montage des systèmes de fichiers	2
4.	Démontage des systèmes de fichiers	5
5.	Vérification des systèmes de fichiers.....	6
6.	Configuration du montage des systèmes de fichiers.....	9
7.	Montage des systèmes de fichiers à la demande	13

OFPPT @	Document	Millésime	Page
	Les systèmes de fichiers sous Gnu/Linux	janvier 15	1 - 16

1.Introduction

Ce chapitre vous présentera les opérations de base sur les systèmes de fichiers, telle que leur création, leur montage et démontage, et leur vérification. Vous y trouverez également la manière de monter automatiquement les systèmes de fichiers les plus utilisés au démarrage du système, ainsi que la manière de monter « à la demande » les systèmes de fichiers amovibles. Vous verrez enfin comment réaliser des agrégats de volumes et comment chiffrer vos systèmes de fichiers.

2.Création des systèmes de fichiers

Linux peut gérer de nombreux systèmes de fichiers, et leur création peut se faire de manière tout à fait semblable à la création d'un système de fichiers EXT2.

Pour chaque système de fichiers, une commande spécifique est fournie afin de le créer. Toutefois, une commande générique de création de systèmes de fichiers permet d'uniformiser la manière de créer les systèmes de fichiers, rendant ainsi inutile la connaissance de ces commandes spécifiques.

Cette commande générique est la commande **mkfs**. Elle prend en paramètre le type de système de fichiers à créer, que l'on peut spécifier à l'aide de l'option -t. En fait, **mkfs** appelle systématiquement la commande de création du système de fichiers spécifié en paramètre. Pour que cela fonctionne, il est nécessaire que cette commande ait un nom de la forme **mkfs.type**, où **type** est le nom du système de fichiers. Ainsi, pour créer un système de fichiers JFS (système de fichiers journalisé créé par IBM) sur la première partition du premier disque SCSI, il suffit d'exécuter la commande suivante :

```
mkfs -t jfs /dev/sda1
```

En général, les commandes **mkfs.type** ne sont rien d'autre que des liens vers les programmes de création spécifiques des systèmes de fichiers.

Les commandes de création des systèmes de fichiers peuvent prendre des options particulières, qu'il faut donc pouvoir leur fournir via **mkfs**. **mkfs** transfère donc toutes les options qu'il trouve après la spécification du type de système de fichiers telles quelles aux programmes de création spécifique des systèmes de fichiers. La liste des options effectivement disponibles peut être consultée dans les pages de manuel respectives de ces programmes.

3.Montage des systèmes de fichiers

Comme il l'a été vu dans le chapitre expliquant les généralités sur Unix, les systèmes de fichiers ne sont donc pas accessibles directement. Ils doivent en effet subir une opération que l'on nomme le *montage* avant de pouvoir être utilisés. Le montage est donc l'opération qui consiste à associer un répertoire au point d'entrée d'un système de fichiers. Une fois monté, les données d'un système de fichiers sont accessibles à partir de ce répertoire. L'opération de

OFPPT @	Document	Millésime	Page
	Les systèmes de fichiers sous Gnu/Linux	janvier 15	2 - 16

Les systèmes de fichiers sous Gnu/Linux

montage permet ainsi de réaliser une abstraction du support des systèmes de fichiers, qui peuvent se trouver aussi bien sur disque qu'en mémoire ou que sur un réseau.

L'opération permettant de monter un disque suit la syntaxe suivante :

`mount [-t type] fichier base`

où fichier est le fichier contenant le système de fichiers à monter (en général, il s'agit d'un fichier spécial de périphérique, mais ce peut également être une image disque), et base est le *point de montage*, c'est-à-dire le répertoire à partir duquel le système de fichiers doit être accédé. L'option `-t` permet d'indiquer le type du système de fichiers. Notez qu'en général il n'est pas nécessaire de le préciser, car le noyau sait reconnaître la plupart des systèmes de fichiers automatiquement.

Pour information, les types de systèmes de fichiers les plus utilisés sont les suivants :

- `ext2`, pour les systèmes de fichiers EXT2 ;
- `ext3`, pour les systèmes de fichiers EXT3. Il est nécessaire d'avoir créé le journal du système de fichiers au préalable, avec l'option `-j` de la commande **mke2fs** ou de la commande **tune2fs** ;
- `reiserfs`, pour les systèmes de fichiers ReiserFS ;
- `iso9660`, pour les CD-ROM (qu'ils soient avec extensions Joliet ou Rock Ridge ou en mode ISO 9660 pur) ;
- `ntfs`, pour les systèmes de fichiers NTFS ;
- `msdos`, pour les systèmes de fichiers FAT normaux ;
- `vfat`, pour les systèmes de fichiers FAT32.

Si le répertoire de montage n'est pas vide, les fichiers qui s'y trouvent sont masqués par le système de fichiers monté. Il est donc recommandé de ne monter les systèmes de fichiers que dans des répertoires vides.

Pour les supports de système de fichiers amovibles, il arrive parfois que Linux ne puisse pas déterminer la géométrie ou la table de partition du support de données. Par exemple, lorsque l'on insère une carte mémoire dans un lecteur de carte, Linux considère que le périphérique n'a pas changé (puisque le lecteur de carte est toujours branché) et ne relit donc pas les informations de la carte mémoire. Il peut donc être nécessaire de demander explicitement au système de relire la table de partition du périphérique. Cela peut être réalisé avec l'option `--rereadpt` de la commande **blockdev** :

`blockdev --rereadpt périphérique`

où périphérique est le fichier spécial de périphérique dont la table de partition doit être relue.

La commande **mount** peut prendre diverses options pour le montage des systèmes de fichiers. Par exemple, elle permet de monter des systèmes de fichiers en lecture seule, ou de monter des systèmes de fichiers placés dans des images disques. Ces options sont introduites par l'option de ligne de commande `-`

OFPPT @	Document	Millésime	Page
	Les systèmes de fichiers sous Gnu/Linux	janvier 15	3 - 16

o, et doivent être séparées les unes des autres par des virgules.

Par exemple, pour monter un système de fichiers ISO9660 en lecture seule, on utilisera la ligne de commande suivante :

```
mount -t iso9660 -o ro fichier base
```

Une autre option utile pour le montage des CD-ROMs est sans doute l'option `session`, qui permet d'indiquer le numéro de la session à monter dans le cas des CD-ROMs multisessions. Par exemple, pour monter la deuxième session d'un CD-ROM multisession, on utilisera une ligne de commande telle que celle-ci :

```
mount -t iso9660 -o ro,session=2 fichier base
```

Le système de fichiers EXT3 prend également des options supplémentaires par rapport au système de fichiers EXT2. Ces options permettent de contrôler la manière dont la journalisation des opérations sur disque est réalisée. Avec EXT3, le journal peut être utilisé pour stocker toutes les opérations concernant la structure de données même du système de fichiers (c'est-à-dire ce que l'on appelle les « méta-données » du système de fichiers) et les opérations concernant les données des fichiers elles-mêmes. La différence est importante et il faut bien la comprendre. Si l'on choisit de ne journaliser que les méta-données du système de fichiers, les informations concernant les répertoires, les fichiers et les droits d'accès seront toujours dans un état cohérent. En revanche, les données stockées dans les fichiers eux-mêmes peuvent être a priori fausses à la suite d'un redémarrage impromptu. Si, en revanche, on décide de stocker également les informations concernant les données des fichiers dans le journal, le contenu des fichiers sera également garanti, au détriment d'une perte de performances notable. Le mode de fonctionnement à utiliser est spécifié à l'aide de l'option `data` du système de fichiers, qui doit donc être fixée lors de l'opération de montage. Cette option peut prendre l'une des trois valeurs suivantes :

- `journal`, qui permet d'effectuer une journalisation complète des méta-données et des données des fichiers. Il est donc garanti que le contenu des fichiers est toujours cohérent, tout comme l'est le système de fichiers. Cette option procure le maximum de sécurité, mais c'est également celle qui pénalise le plus les performances du système (les données sont écrites deux fois, une fois dans le journal, et une fois sur disque) ;
- `ordered`, qui est l'option par défaut et qui permet de ne journaliser que les méta-données, mais qui garantit également que les tampons d'écriture sont vidés avant chaque journalisation d'une opération disque. Tout comme avec l'option `journal`, il est garanti que le système de fichiers est dans un état cohérent. Les données des fichiers seront également cohérentes avec les structures de données du système de fichiers. Cependant, rien ne garantit que le contenu des fichiers sera lui aussi cohérent en interne, car même si les données sont écrites avant toute modification du système de fichiers, aucun contrôle n'est effectué pour que les écritures soient réalisées dans l'ordre dans lequel les applications les ont effectuées. Cela dit, les performances sont meilleures qu'avec l'option `journal`, tout en garantissant une sécurité quasi totale des fichiers ;
- `writeback`, qui permet de ne journaliser que les méta-données du système

OFPPT @	Document	Millésime	Page
	Les systèmes de fichiers sous Gnu/Linux	janvier 15	4 - 16

de fichiers. Le contenu des fichiers peut donc être incorrect, voire même contenir des données aléatoires à la suite d'un arrêt brutal du système, mais le système de fichiers est toujours dans un état correct. Cette option permet donc simplement de rendre facultative la vérification et la réparation des systèmes de fichiers EXT2 au redémarrage. Les performances sont quasiment aussi bonnes que pour le système de fichiers EXT2.

Enfin, il est possible de monter un système de fichiers plusieurs fois, éventuellement avec des options différentes, dans différents points de montage. De même, il est possible de monter une partie d'un système de fichiers seulement dans un autre répertoire, par exemple pour réaliser un raccourci vers un sous ensemble du système de fichiers hôte. Vous pouvez consulter la page de manuel de la commande **mount** pour obtenir plus de détail à ce sujet.

Vous pourrez trouver la liste des autres options acceptées par **mount** et par les systèmes de fichiers dans la page de manuel mount.

4. Démontage des systèmes de fichiers

Les systèmes de fichiers sont gérés de manière très efficace par les systèmes d'exploitation. Des mécanismes de copie en mémoire des données sont utilisés afin d'accélérer leur lecture, et les écritures peuvent être différées pour être regroupées et ainsi optimiser les transferts. De ce fait, l'état du système de fichiers sur le support n'est quasiment jamais exactement en phase avec celui du système de fichiers en mémoire lorsqu'il est en cours d'utilisation. Il est donc nécessaire de signaler au système d'exploitation que l'on désire l'arrêter avant de couper le courant, ou que l'on désire retirer un lecteur amovible avant de le faire, afin qu'il puisse effectuer les synchronisations nécessaires. Ne pas le faire risquerait de provoquer des pertes de données irrémédiables. Cette opération s'appelle simplement le *démontage*.

Note : Bien entendu, les commandes d'arrêt du système se chargent (entre autres) de démonter tous les systèmes de fichiers avant d'éteindre l'ordinateur.

La commande permettant de démonter un système de fichiers est beaucoup plus simple que celle permettant de les monter, car aucune option n'est nécessaire. Il suffit en effet d'exécuter l'une des commandes suivantes :

umount fichier
ou :
umount base

Dans ces commandes, fichier représente le fichier contenant le système de fichiers à démonter, et base est le répertoire dans lequel ce système de fichiers est monté. On peut utiliser l'un ou l'autre de ces paramètres, la commande **umount** se débrouillera pour retrouver l'autre automatiquement. On notera qu'il est impossible de démonter un système de fichiers qui est en cours d'utilisation par quelqu'un. En particulier, il ne faut pas être dans le répertoire servant de point de montage pour pouvoir démonter un système de fichiers, car dans ce cas

OFPPT @	Document	Millésime	Page
	Les systèmes de fichiers sous Gnu/Linux	janvier 15	5 - 16

on est en train de l'utiliser. Faites bien attention à l'orthographe de la commande **umount**, elle a perdu son 'n' depuis bien longtemps déjà, et on ne l'a jamais retrouvé. Si vous savez où il se trouve, faites-le savoir.

Pour les supports de systèmes de fichiers amovibles, le démontage du système de fichiers peut ne pas être suffisant. En effet, il peut être nécessaire d'arrêter le périphérique correctement avant de l'ejecter. C'est en particulier le cas pour les systèmes de fichiers sur les clefs USB par exemple. Cette opération peut être réalisée à l'aide de la commande **eject** :

eject périphérique

où périphérique est le fichier spécial du périphérique à ejecter. Cette commande réalise également l'opération de démontage sur les systèmes de fichiers montés avant d'ejecter le disque.

5. Vérification des systèmes de fichiers

La vérification des systèmes de fichiers est une opération que l'on ne devrait jamais avoir à faire. Il y a plusieurs raisons à cela. Premièrement, si l'on arrête le système correctement avant d'éteindre la machine, et si l'on démonte bien les systèmes de fichiers avant de retirer les lecteurs amovibles, les systèmes de fichiers sont normalement toujours dans un état correct. Deuxièmement, les systèmes de fichiers Unix sont réputés pour être très fiables. Troisièmement, une vérification périodique est faite par le système au bout d'un certain nombre de démarrages. Quatrièmement, si un système de fichiers n'est pas démonté correctement avant l'arrêt du système, celui-ci sera vérifié automatiquement au démarrage suivant, ce qui fait qu'il n'y a pas lieu de le faire soi-même. Enfin, pour les systèmes de fichiers journalisés tels que EXT3, JFS ou ReiserFS, cette opération peut être réalisée très rapidement à l'aide des informations qui sont stockées dans le journal des transactions du système de fichiers.

Toutefois, même le meilleur système du monde ne saurait être à l'abri des secteurs défectueux du disque dur sur lequel il est installé. Il est donc parfois nécessaire d'effectuer une vérification manuelle des systèmes de fichiers, et il faut savoir le faire même quand plus rien ne fonctionne.

Un système de fichiers ne se manipule que lorsqu'il est démonté. Cela pose évidemment quelques problèmes pour le système de fichiers racine, puisqu'on ne peut pas accéder aux outils de vérification sans le monter. Pour ce système de fichiers, il n'y a donc que deux possibilités :

- soit on utilise une disquette de démarrage contenant les outils de vérification et de réparation des systèmes de fichiers ;
- soit on monte le système de fichiers racine en lecture seule.

La deuxième solution est la seule réalisable si l'on ne dispose pas de disquette de démarrage. Par conséquent, c'est cette méthode qui sera décrite ici.

La première étape consiste à passer en mode mono utilisateur, afin de s'assurer

OFPPT @	Document	Millésime	Page
	Les systèmes de fichiers sous Gnu/Linux	janvier 15	6 - 16

que personne ni aucun programme n'accède au système de fichiers racine en écriture. Pour cela, il suffit de taper la commande suivante :

`init 1`

qui fait passer le système dans le niveau d'exécution 1. On peut également passer le paramètre single au noyau lors de l'amorçage du système, comme il l'a été expliqué dans la section précédente. Ensuite, il faut s'assurer que le système de fichiers racine est en lecture seule, ce qui se fait avec la commande suivante :

`mount -n -o remount,ro /`

L'option `remount` permet de démonter et de remonter le système de fichiers racine, et l'option `ro` indique qu'il doit être remonté en lecture seule (« `ro` » signifie « Read Only »). Les options sont séparées par des virgules (attention, il ne faut pas insérer d'espace). De plus, l'option `-n` indique à **mount** qu'il ne doit pas écrire dans le fichier `/etc/mstab` lorsqu'il aura remonté le système de fichiers, parce que ce fichier sera alors également en lecture seule et qu'on ne pourra pas y écrire. Ce fichier est utilisé par **mount** pour mémoriser les systèmes de fichiers qui sont montés, afin de pouvoir en donner la liste (ce que la commande **mount** fait lorsqu'elle est appelée sans paramètres) et de permettre à la commande **umount** de vérifier que les systèmes de fichiers à démonter ne le sont pas déjà.

Note : Normalement, le noyau monte toujours le système de fichiers racine en lecture seule lors de l'amorçage. Ce sont les scripts de démarrage du système, lancés par **init**, qui le remontent en lecture / écriture s'il est dans un état correct. Il est donc fortement probable, si votre système ne démarre plus correctement, que le système de fichiers racine soit déjà en lecture seule après un démarrage en mode de maintenance. La commande précédente n'est donc décrite qu'à titre indicatif.

Une fois le système de fichiers racine monté en lecture seule, on peut utiliser le programme **fsck** afin de le vérifier et éventuellement le réparer. En réalité, ce programme ne fait rien d'autre que d'appeler un programme spécifique pour chaque système de fichiers, de la même manière que **mkfs** appelle des programmes spécifiques pour les créer. Par exemple, pour les systèmes de fichiers EXT2 et EXT3, **fsck** appelle le programme **fsck.ext** (qui est un lien vers **e2fsck**).

La ligne de commande à utiliser pour vérifier un système de fichiers avec **fsck** est la suivante :

`fsck -a fichier`

où `fichier` est le fichier spécial du périphérique ou le fichier image contenant le système de fichiers à vérifier. Il faut donc, généralement, spécifier la partition sur laquelle le système de fichiers racine se trouve. L'option `-a` demande à **fsck** d'effectuer les éventuelles corrections automatiquement en cas d'erreur sur le système de fichiers ainsi vérifié, sans confirmation de la part de l'utilisateur.

Il est également possible de demander la vérification de tous les systèmes de

OFPPT @	Document	Millésime	Page
	Les systèmes de fichiers sous Gnu/Linux	janvier 15	7 - 16

fichiers enregistrés dans le fichier de configuration `/etc/fstab`. Ce fichier contient la liste des systèmes de fichiers les plus utilisés et leurs options respectives. Il suffit donc d'ajouter l'option `-A` :

```
fsck -a -A
```

Il n'est évidemment plus nécessaire de spécifier le fichier spécial du périphérique contenant le système de fichiers à vérifier, puisque cette information est enregistrée dans le fichier `/etc/fstab`. La syntaxe de ce fichier sera décrite dans la section suivante.

Si le disque dur contient des secteurs défectueux, il peut être nécessaire de les marquer comme tels dans les structures du système de fichiers afin de ne pas les utiliser par la suite. De manière générale, la recherche de ces blocs peut être faite à l'aide du programme **badblocks**. Cette commande effectue un test de lecture de tous les blocs du disque sur lequel le système de fichiers se trouve, et génère une liste des blocs défectueux. Vous pouvez l'appeler directement et fournir cette liste au programme **e2fsck** à l'aide de son option `-l`, mais le plus simple est encore de demander à **e2fsck** d'appeler **badblocks** lui-même. Pour cela, il suffit de lui passer l'option `-c`, ce qui se fait en faisant précéder cette option d'un double-tiret dans la ligne de commande de **fsck** :

```
fsck -a -- -c périphérique
```

Note : L'option `-c` est spécifique à **e2fsck** et peut ne pas fonctionner avec d'autres systèmes de fichiers. En particulier, certains systèmes de fichiers ne sont pas capables de gérer correctement les blocs défectueux des disques durs. C'est le cas du système de fichiers ReiserFS.

Le programme **badblocks** peut également effectuer un test d'écriture sur le disque dur, si on lui communique l'option `-w`. Il va de soi que ce type de test est destructif, car toutes les données du disque sont alors écrasées par des motifs particuliers. Il ne faut donc *JAMAIS* utiliser cette option sur un système de fichiers contenant des données !

De manière générale, il vaut mieux prévenir que guérir, aussi est-il recommandé d'utiliser la commande **badblocks** au moins une fois avant d'utiliser un système de fichiers. Cette vérification peut être réalisée de manière automatique lors de la création du système de fichiers à l'aide de l'option `-c` de la commande **mke2fs**.

Une fois que vous aurez terminé la vérification du système de fichiers, vous pourrez le remonter en lecture et écriture avec la commande suivante :

```
mount -n -o remount,rw /
```

Cette commande est similaire à celle que l'on a vue pour monter le système de fichiers en lecture seule, à ceci près que l'option `rw` est utilisée à la place de l'option `ro`. Cette option permet de remonter le système de fichiers en lecture et en écriture (« `rw` » est l'abréviation de l'anglais « Read Write »).

OFPPT @	Document	Millésime	Page
	Les systèmes de fichiers sous Gnu/Linux	janvier 15	8 - 16

6. Configuration du montage des systèmes de fichiers

Le montage des systèmes de fichiers peut devenir très vite une opération assez fastidieuse. Heureusement, elle peut être automatisée au démarrage pour les systèmes de fichiers situés sur les disques fixes, et simplifiée pour les systèmes de fichiers amovibles. Pour cela, il faut enregistrer ces systèmes de fichiers et leurs options de montage dans le fichier de configuration `/etc/fstab`.

Ce fichier contient, entre autres, le répertoire de montage, le type du système de fichiers et le fichier de périphérique à utiliser pour chaque système de fichiers. De cette manière, il est possible d'utiliser la commande **mount** de manière simplifiée, en ne précisant que le répertoire servant de point de montage ou le fichier spécial de périphérique.

Le fichier `/etc/fstab` contient une ligne pour chaque système de fichiers enregistré. Chaque ligne contient plusieurs champs séparés par des espaces. Les informations suivantes sont enregistrées dans ces champs :

- le fichier spécial permettant d'accéder au système de fichiers ;
- le répertoire servant de point de montage par défaut ;
- le type du système de fichiers ;
- les options de montage pour ce système de fichiers ;
- un entier indiquant si le système de fichiers doit être sauvegardé ;
- un entier indiquant l'ordre que ce système de fichiers doit avoir dans la liste des systèmes de fichiers à vérifier.

Grâce à ces informations, l'emploi de la commande **mount** est plus simple :

`mount périphérique`

ou :

`mount répertoire`

où `périphérique` est le fichier spécial de périphérique contenant le système de fichiers à monter, et `répertoire` est le répertoire servant de point de montage indiqué dans le fichier `/etc/fstab`. Il est possible d'utiliser indifféremment le fichier spécial de périphérique ou le répertoire du point de montage.

Le type du système de fichiers est l'un des types disponibles acceptés par la commande **mount**. Consultez la page de manuel de cette commande pour plus de renseignements à ce sujet. Les principales options disponibles pour le montage sont les suivantes :

- l'option `defaults`, qui permet de choisir les options par défaut pour ce système de fichiers ;
- l'option `auto`, qui permet de faire en sorte que le système de fichiers soit monté automatiquement au démarrage du système ;
- l'option `user`, qui permet d'autoriser le montage de ce système de fichiers par les utilisateurs ;

OFPPT @	Document	Millésime	Page
	Les systèmes de fichiers sous Gnu/Linux	janvier 15	9 - 16

Les systèmes de fichiers sous Gnu/Linux

- l'option `ro`, qui permet de monter le système de fichiers en lecture seule ;
- l'option `rw`, qui permet de monter le système de fichiers en lecture et écriture ;
- l'option `exec`, qui permet d'autoriser l'exécution des fichiers exécutables sur ce système de fichiers si celui-ci ne supporte pas la notion de droit d'exécution ;
- l'option `acl`, qui permet d'autoriser l'utilisation des ACLs (« Access Control List ») pour fixer les droits des utilisateurs sur les fichiers ;
- l'option `uid=utilisateur`, qui permet de spécifier le numéro utilisateur de l'utilisateur propriétaire du répertoire racine de ce système de fichiers ;
- l'option `gid=groupe`, qui permet de spécifier le numéro groupe du groupe d'utilisateurs auquel le répertoire racine du système de fichiers appartient ;
- l'option `mode=valeur`, qui permet de fixer les droits sur le répertoire racine du système de fichiers à monter. La valeur valeur est donnée en octal ;
- l'option `umask=valeur`, qui permet de fixer les droits sur les fichiers qui ne sont pas gérés par le système de fichiers. La valeur valeur est donnée en octal ;
- les options `codepage=cp` et `iocharset=charset`, qui permettent de fixer les tables de conversion des caractères pour les systèmes de fichiers pour lesquels les noms de fichiers doivent être transcodés.

Nous allons détailler un peu quelques-unes de ces options.

Par défaut, les utilisateurs n'ont pas le droit de monter et de démonter les systèmes de fichiers. L'option `user` permet de désactiver cette protection. Elle peut être utile pour permettre le montage et le démontage des disquettes et des CD-ROM. De même, l'exécution des fichiers exécutables n'est par défaut pas autorisée sur les systèmes de fichiers. Cette restriction permet d'éviter l'exécution de programmes placés sur des systèmes de fichiers de systèmes d'exploitation différents. Elle peut être levée grâce à l'option `exec`.

Tous les systèmes de fichiers disposant de l'option `auto` seront montés automatiquement au démarrage du système par la commande **mount -a**. Les autres systèmes de fichiers sont montables manuellement, avec les autres options indiquées dans le fichier `/etc/fstab`.

Les options `ro` et `rw` permettent d'indiquer à **mount** si le système de fichiers doit être monté en lecture seule ou en lecture et écriture. Les systèmes de fichiers devant être réparés doivent être montés en lecture seule si l'on ne peut pas les démonter (c'est le cas notamment du système de fichiers racine). Il en va de même pour les CD-ROM, car on ne peut bien entendu pas écrire dessus.

Les options `uid` et `gid` permettent de spécifier le propriétaire et le groupe du répertoire racine du système de fichiers à monter. Par défaut, c'est l'utilisateur `root` qui devient propriétaire de ce système de fichiers.

L'option `mode` permet de spécifier les droits d'accès sur tous les fichiers du système de fichiers à monter. L'option `umask` permet quant à elle de fixer les droits qui ne sont pas gérés par le système de fichiers. Ce peut être utile pour les systèmes de fichiers FAT et FAT32. Il est ainsi possible de donner les droits de

OFPPT @	Document	Millésime	Page
	Les systèmes de fichiers sous Gnu/Linux	janvier 15	10 - 16

lecture et d'exécution pour les fichiers de ces systèmes avec une valeur de masque nulle. Cela permet de monter les systèmes de fichiers FAT et FAT32 de telle sorte que tous les fichiers appartiennent à l'utilisateur root par défaut, et de donner cependant tous les droits à tous les utilisateurs sur ces fichiers. On prendra garde à ces options, car elles permettent à quiconque d'écrire des fichiers sous le nom de root, et donc constituent un grave défaut dans la sécurité du système.

Les options `codepage` et `iocharset` permettent de spécifier respectivement la page de codes et le jeu de caractères utilisés pour les systèmes de fichiers qui ne stockent pas les noms de fichiers Unix nativement. En particulier, elles permettent de modifier la page de code et le jeu de caractères sélectionnés par défaut dans la configuration du noyau pour les systèmes de fichiers FAT. L'option `codepage` est utilisée pour donner la page de codes utilisée pour la conversion des noms de fichiers en noms de fichiers courts. En général, pour les systèmes français, la page de codes utilisée est la page de codes 850. Il faut donc donner à cette option la valeur `cp850`. L'option `iocharset` quant à elle est utilisée pour faire la conversion des noms de fichiers Unix en Unicode. Elle est utilisée pour les systèmes de fichiers VFAT, car ceux-ci stockent les noms de fichiers longs en Unicode. Pour la plupart des systèmes, le jeu de caractères le plus approprié est sans doute le jeu de caractères ISO8859-1. Aussi faut-il généralement donner à cette option la valeur `iso8859-1`. Cette option n'est pas nécessaire pour les systèmes de fichiers FAT purs, puisque ceux-ci ne savent pas gérer les noms de fichiers longs.

Les deux derniers champs de `/etc/fstab` spécifient des options pour des programmes annexes. L'avant-dernier contrôle le comportement du programme de sauvegarde **dump**, et le dernier celui du programme de vérification de système de fichiers **fsck**. Consultez les pages de manuel pour plus de détails à ce sujet.

Si vous disposez de systèmes de fichiers FAT ou FAT32, vous pourrez monter ces partitions automatiquement lors du démarrage du système. Comme les systèmes de fichiers basés sur la FAT ne peuvent pas gérer les droits des utilisateurs, vous allez devoir faire un choix pour fixer ces droits à une valeur raisonnable. Vous pouvez par exemple donner le droit de lecture à tous les utilisateurs, mais le droit d'écriture uniquement à l'administrateur système. La ligne à ajouter dans le fichier `/etc/fstab` sera alors la suivante :

```
/dev/partition répertoire vfat auto,exec,codepage=cp850,iocharset=iso8859-1  
0 0
```

où `partition` est la partition contenant le système de fichiers FAT, et `répertoire` est le répertoire servant de point de montage pour cette partition. Cette ligne permettra de monter automatiquement ce système de fichiers en tant que FAT32 au démarrage du système. Les fichiers binaires seront exécutables, bien qu'ils ne soient pas stockés sur un système de fichiers EXT2. Si vous voulez laisser les droits d'écriture aux utilisateurs, vous pouvez utiliser la ligne suivante à la place de celle indiquée ci-dessus :

```
/dev/partition répertoire vfat auto,codepage=cp850,iocharset=iso8859-  
1,umask=0 0 0
```

OFPPT @	Document	Millésime	Page
	Les systèmes de fichiers sous Gnu/Linux	janvier 15	11 - 16

Les systèmes de fichiers sous Gnu/Linux

Cette ligne permet de monter le système de fichiers FAT en laissant les droits d'exécution et d'écriture aux utilisateurs. Cependant, aucun fichier exécutable de ce système de fichiers ne pourra être lancé, car l'option `exec` n'a pas été précisée.

Par ailleurs, certaines distributions spécifient des options incorrectes pour le système de fichiers `/dev/pts/` dans le fichier `/etc/fstab`. Veuillez vous assurer que la ligne utilisée pour ce système de fichiers est bien identique à la ligne suivante :

```
none      /dev/pts    devpts auto,gid=5,mode=620 0 0
```

Si ce n'est pas le cas, certains émulateurs de terminaux développés récemment ne fonctionneront pas correctement. Le système de fichiers `/dev/pts/` est en effet un système de fichiers virtuel, géré directement par le noyau, dans lequel des fichiers spéciaux de périphériques utilisés par les émulateurs de terminaux sont placés. Si les droits ne sont pas correctement fixés sur le répertoire racine de ce système de fichiers, les émulateurs de terminaux utilisant cette fonctionnalité ne pourront se connecter au système que sous le compte `root`. Il faut donc impérativement corriger cette ligne, si vous voulez que les utilisateurs normaux puissent utiliser ces émulateurs. Notez également qu'il faut que tout le monde ait les droits d'écriture et de lecture sur le fichier spécial de périphérique `/dev/ptmx` pour que les utilisateurs non privilégiés puissent utiliser ce système de fichiers virtuel.

Vous devrez également ajouter le système de fichiers virtuel `/dev/shm/` dans votre fichier `/etc/fstab`. Ce système de fichiers permet aux applications qui le désirent d'utiliser des segments de mémoire partagée de manière compatible avec la norme POSIX, par exemple pour réaliser des communications inter-processus. Ce système de fichiers permet en effet de créer des fichiers qui sont directement stockés dans la mémoire virtuelle et qui peuvent être ouverts par plusieurs processus simultanément, ce qui permet d'utiliser les fonctionnalités classiques de partage de fichiers pour partager des zones de mémoire entre plusieurs processus. Ce type de communication peut être utilisé par des processus courants, aussi faut-il vous assurer que la ligne suivante se trouve bien dans la fichier `fstab` :

```
tmpfs     /dev/shm    tmpfs defaults          0 0
```

Bien entendu, vous devrez créer le point de montage `/dev/shm/` si celui-ci n'existe pas avant de monter ce système de fichiers virtuels.

Vous pouvez monter les systèmes de fichiers virtuels du noyau, qui vous permettront d'obtenir des informations sur votre système et d'en modifier le comportement, en ajoutant les deux lignes suivantes dans le fichier `/etc/fstab` :

```
proc      /proc       proc defaults          0 0
sysfs     /sys        sysfs defaults          0 0
```

La première ligne monte le système de fichiers `/proc/`, qui contient tous les paramètres du noyau et permet de paramétrer son comportement, et la deuxième contient le système de fichiers `/sys/`, qui contient une vue de tous les objets systèmes gérés par le noyau. Ce système de fichiers n'est

OFPPT @	Document	Millésime	Page
	Les systèmes de fichiers sous Gnu/Linux	janvier 15	12 - 16

disponible qu'avec les versions 2.6 et plus du noyau. Ces deux lignes ne sont généralement pas nécessaires, car ces systèmes de fichiers sont montés automatiquement dans les scripts d'initialisation des distributions.

Enfin, si vous utilisez des périphériques USB, il vous faudra également monter le système de fichiers virtuel `/proc/bus/usb/` en ajoutant la ligne suivante dans le fichier `fstab` :

```
none /proc/bus/usb usbfs defaults 0 0
```

Cette ligne doit être placée après celle qui effectue le montage du système de fichiers virtuel `/proc/`.

7. Montage des systèmes de fichiers à la demande

Les systèmes de fichiers sur les supports amovibles ne peuvent pas être montés automatiquement au démarrage du système, étant donné qu'ils ne sont pas forcément présents à ce moment. Les entrées qui les décrivent dans le fichier `fstab` doivent donc utiliser l'option `noauto`. De ce fait, l'utilisation des systèmes de fichiers stockés sur les supports amovibles nécessite toujours un montage et un démontage manuel, et même si ces opérations sont simplifiées, elles restent gênantes.

Il est toutefois possible de faire en sorte que ces systèmes de fichiers soient montés à la demande, c'est à dire dès qu'un accès est réalisé dans les répertoires de montage dédiés à ces systèmes par un quelconque programme. Ainsi, il n'est plus nécessaire de monter ces systèmes de fichiers manuellement. Le démontage quant à lui peut être réalisé automatiquement après un temps d'inactivité, ou manuellement, lorsqu'on veut retirer le lecteur amovible. Notez que le démontage reste vital si le système de fichiers est accédé en écriture, car des données pourraient être perdues si l'on supprime le lecteur avant qu'elles ne soient écrites.

Les opérations de montage et de démontage sont réalisées par un démon, que le noyau appelle automatiquement dès qu'une opération susceptible de concerner un système de fichiers amovible se produit. Ce démon se nomme **automount**, et il s'appuie donc sur une fonctionnalité dédiée du noyau Linux. Pour que le montage automatique des systèmes de fichiers fonctionne, il faut activer soit l'option « Kernel automounter support », soit l'option « Kernel automounter version 4 support (also supports v3) » du menu « File systems » du programme de configuration du noyau.

Note : Un démon est un processus qui tourne en arrière plan dans le système. Les démons fonctionnent souvent dans le compte root et offrent des services de base que les autres programmes utilisent. Le terme « démon » provient de la traduction littérale « daemon », ce qui signifie « ange » en réalité. Le démon se place donc en tant qu'intermédiaire entre Dieu (c'est-à-dire le noyau) et les hommes (c'est-à-dire les applications normales). Le terme « daemon » a été

OFPPT @	Document	Millésime	Page
	Les systèmes de fichiers sous Gnu/Linux	janvier 15	13 - 16

ensuite éclairci et défini comme l'acronyme de l'anglais « Disk And Execution MONitor ».

Le démon **automount** peut surveiller plusieurs points de montage, s'ils sont tous placés dans le même répertoire. Le noyau surveille simplement les accès sur les sous-répertoires de ce répertoire, et signale à **automount** toutes les opérations susceptibles de nécessiter un montage de système de fichiers.

Les informations spécifiques aux points de montage doivent être stockées dans un fichier de configuration, que l'on doit fournir à **automount** en ligne de commande. Cette ligne de commande est donc typiquement de la forme suivante :

`automount -t durée répertoire file fichier`

où durée est la durée d'inactivité avant démontage automatique du système de fichiers, répertoire est le répertoire dont les sous-répertoires seront utilisés comme points de montage des systèmes de fichiers (il s'agit généralement du répertoire /mnt/), et fichier est le fichier de configuration pour les points de montage placés dans ce répertoire. La durée d'inactivité doit être spécifiée en secondes, et la valeur par défaut est de cinq minutes. Pour désactiver le démontage automatique, il suffit de spécifier une valeur nulle.

Note : **automount** est capable d'utiliser d'autres méthodes de paramétrage des systèmes de fichiers amovibles que des fichiers de configuration, principalement utilisées dans le contexte du montage automatique des systèmes de fichiers en réseau. Ces méthodes de paramétrage ne seront toutefois pas décrites ici, vous pouvez consulter la page de manuel du démon **automount** pour plus de détails à ce sujet.

Les fichiers de configuration pour **automount** sont généralement placés dans le répertoire /etc/, et portent un nom de la forme « auto.nature », où nature est la nature des systèmes de fichiers décrits par ce fichier. Par exemple, le fichier de configuration pour les lecteurs amovibles locaux pourrait s'appeler /etc/auto.local. Ce fichier de configuration est constitué de plusieurs lignes, à raison d'une ligne par point de montage. Chaque ligne utilise la syntaxe suivante :

`répertoire -options périphérique`

où répertoire est le sous-répertoire du répertoire spécifié en ligne de commande à **automount** et devant servir de point de montage, options est la liste des options de montage du système de fichiers, séparées par des virgules, et périphérique est la localisation du système de fichiers. Pour les lecteurs amovibles, il s'agit du fichier spécial de périphérique permettant d'y accéder, précédé du caractère ':'.

Par exemple, pour réaliser le montage automatique du lecteur de CD-ROM et du lecteur de disquettes, vous devez ajouter deux lignes telles que celles-ci dans le fichier de configuration d'**automount** :

`cdrom -fstype=iso9660,ro :/dev/cdrom`

OFPPT @	Document	Millésime	Page
	Les systèmes de fichiers sous Gnu/Linux	janvier 15	14 - 16

Les systèmes de fichiers sous Gnu/Linux

```
floppy -fstype=auto,umask=0 :/dev/fd0
```

Vous noterez que les options sont précédées d'un tiret ('-'), et que les fichiers spéciaux de périphérique sont précédés de deux points (':'). Une fois ce fichier défini, vous n'aurez plus qu'à lancer **automount** avec la ligne de commande suivante :

```
automount -t 60 /mnt file /etc/auto.local
```

En général, les distributions lancent automatiquement le démon **automount** au démarrage, via le script d'initialisation rc.autofs. Ce script utilise le fichier de configuration /etc/auto.master pour déterminer les noms des fichiers de configuration à fournir en paramètre aux démons **automount**, ainsi que pour déterminer les options avec lesquelles ils doivent être lancés. Ce fichier est constitué de lignes, à raison d'une ligne par instance du démon **automount** qui doit être lancée. Le format de ces lignes est le suivant :

répertoire fichier options

où répertoire est le répertoire spécifié en ligne de commande à **automount**, fichier est le fichier de configuration, et option est la liste des options à lui fournir. Par exemple, pour lancer automatiquement **automount** comme dans l'exemple précédent, la ligne suivante devrait être ajoutée dans le fichier /etc/auto.master :

```
/mnt /etc/auto.local -t 60
```

Note : Vous noterez que lorsque les systèmes de fichiers ne sont pas montés, le répertoire contenant leurs points de montage est vide. De ce fait, les chemins sur ces répertoires devront être saisis explicitement pour forcer le montage des systèmes de fichiers. Cela peut être gênant lorsque l'on utilise les fonctionnalités de complétion automatique des shells, ou lorsqu'on cherche à localiser un fichier dans un de ces répertoires avec un gestionnaire de fichiers ou une boîte de dialogue d'un programme graphique. Une solution pour résoudre ce problème est de forcer une référence aux points de montage. Pour cela, il suffit de définir des liens symboliques sur les points de montage, et d'utiliser ces liens symboliques vers ces points de montage. Par exemple, on pourra définir les liens symboliques /cdrom et /floppy pour référencer les points de montage /mnt/cdrom et /mnt/floppy, et laisser **automount** monter les systèmes de fichiers relatifs et créer les répertoires des points de montage dès qu'on accèdera aux liens symboliques.

OFPPT @	Document	Millésime	Page
	Les systèmes de fichiers sous Gnu/Linux	janvier 15	15 - 16