



par Guido Socher ([homepage](#))

*L'auteur:*

Guido apprécie Linux pour sa communauté. C'est un énorme projet et, d'une certaine façon, toutes les personnes à travers le monde, si différentes soient-elles, joignent leurs efforts dans un même but : rendre ce système toujours meilleur, un peu plus chaque jour.

Mais Linux a "malheureusement" atteint une telle notoriété que l'on commence à voir des gens peu recommandables s'y intéresser, dont le but est de faire du profit sans la moindre contrepartie : l'on voit apparaître de plus en plus de petites sociétés prétendument "pro-Linux" qui s'avèrent en fait purement et simplement des opportunistes désireux de profiter d'un effet de mode.

*Traduit en Français par:*

Guillaume Baudot

<guillaume.baudot(at)caramail.com>

## Le mystère des points de montage



*Résumé:*

Cet article explique le concept de point de montage, avec l'espoir non dissimulé d'offrir une information propre à satisfaire la curiosité des novices comme des utilisateurs plus expérimentés.

---

# Introduction

Je vais commencer par expliciter le principe du montage d'un système de fichiers pour ensuite vous présenter des caractéristiques un peu plus poussées. Si vous pensez être un expert Linux, libre à vous de sauter quelques paragraphes...

## Comment ça marche

Aujourd'hui encore, ajouter un lecteur dans Windows se révèle souvent une véritable gageure. Par exemple, votre ancien lecteur D: est soudainement transformé en E:, ce qui provoque une confusion évidente pour les utilisateurs comme pour le système (N.d.T. et plus particulièrement la fameuse "base de registre" !).

Avec Unix, ce problème n'en est pas un, car il n'y a pas de notion telle que les lettres de lecteur : les partitions sont intégrées dans une seule et même arborescence de répertoires, comme vous pouvez le constater avec les commandes "mount" ou "df" :

```
# df
Filesystem 1K-blocks    Used Available Use% Mounted on
/dev/hda1   9070728 4701180   3908768   55% /
/dev/hda3   24950428 683412   22999584    3% /home
none        257236      0    257236    0% /dev/shm
```

```
# mount
/dev/hda1 on / type ext3 (rw)
none on /proc type proc (rw)
none on /dev/shm type tmpfs (rw)
/dev/hda3 on /home type ext3 (rw)
```

Nous voyons dans cet exemple deux partitions : l'une contient la presque totalité des fichiers du système, à l'exception de "/home" qui se trouve sur une partition séparée. Nous avons donc la racine du système de fichier ("/) qui est connectée à la première partition (1) du premier disque dur IDE (hda), et "/home" ("hda3") qui est monté dans la troisième : de fait, lorsque vous ouvrez le répertoire /home, vous accédez à des fichiers se trouvant sur la partition hda3, sans avoir à vous soucier de savoir si cette partition correspond au lecteur D:, E:, ou que sais-je encore ? C'est simplement un répertoire dans votre arborescence et sauf changement explicite de votre part, il n'y a aucune raison pour qu'il se retrouve ailleurs ou disparaisse...

## Le problème du "système de fichiers occupé"

Linux bufferise (cache en mémoire) les accès aux systèmes de fichiers, que ce soit en lecture ou en écriture. Vous pouvez le constater notamment en lisant un gros fichier sur clé USB (version 1.1) : la première ouverture du fichier sera particulièrement lente, mais si vous ouvrez une seconde fois ce même fichier, ce sera au contraire très rapide (et vous ne verrez même pas s'éclairer la diode du lecteur, puisque le fichier en question est déjà copié en mémoire !).

Mais à cause de cette particularité, vous ne pouvez pas retirer un disque en cours d'utilisation d'un simple claquement de doigts, il faut le "dé-monter" (umount) proprement, pour assurer la cohésion du système de fichiers (répercuter les éventuelles opérations d'écriture depuis le cache mémoire vers le disque, si besoin est). S'il est peu probable d'être confronté à ce problème pour des disques durs internes, c'est au contraire très fréquent pour les disques amovibles (CDrom, clés USB...).

Par exemple, si vous montez dans un terminal votre clé USB, disons dans /mnt/usb, tapez alors "cd /mnt/usb" : le shell utilise ce système de fichiers. Pour vous en convaincre, essayez donc la commande "umount /mnt/usb" : elle échouera, avec ce message d'erreur "file system busy"—"système de fichiers occupé". Il suffit de sortir de ce répertoire pour pouvoir ensuite le démonter... Si cet exemple est simple, il est parfois délicat de retrouver l'application qui cause la défaillance de la commande "umount" : c'est pourquoi je ne saurais trop vous conseiller le cas échéant de demander à l'ordinateur !

```
# fuser -m -u /mnt/usb
/mnt/usb: 1347c(root) 1348c(guido) 1349c(guido)
```

La commande ci-dessus renvoie la liste des processus qui utilisent le système de fichiers monté sur /mnt/usb. Libre à vous ensuite d'utiliser "ps auxw" ou plus brutalement "kill" pour faire en sorte que plus aucun processus n'utilise /mnt/usb, et qu'ainsi "umount" fonctionne.

## Mais Windows peut aussi bien le faire !?!

Pourquoi ne pas simplement débrancher la clé USB ? Après tout cela fonctionnait très bien ainsi avec les vieux DOS et autres Windows-3.1... Nul besoin non plus d'utiliser la commande "shutdown", il suffisait d'appuyer sur l'interrupteur pour éteindre l'ordinateur ! Mais depuis Windows-95, nous voici tous logés à la même enseigne : on utilise "shutdown" avant la mise hors-tension de la machine, et l'on dé-monte (ou croise les doigts pour que l'écriture soit bel et bien effectuée !) avant de retirer un disque (ou tout autre type de lecteur).

## Linux en est capable aussi !

Il existe des solutions sous Linux pour permettre de retirer ainsi un disque, sans avoir à le démonter auparavant : par exemple Mandrake propose le programme "supermount" qui vous offre les mêmes possibilités (ou devrais-je plutôt dire les mêmes problèmes !) que vous avez sous Windows. Mais je n'oserais vous le recommander, car il peut causer moult ennuis : instabilité, pertes de fichiers...

Il est bien plus raisonnable de faire appel aux "mtools" (<http://mtools.linux.lu/>). Il s'agit d'un ensemble de commandes pour accéder aux lecteurs disquettes, clés USB, etc. (en fait tout périphérique doté d'un système de fichiers FAT) sans avoir à les monter.

Voici comment cela fonctionne :

1. Éditez le fichier /etc/mtools.conf et insérez une ligne de ce genre :

```
drive u: file="/dev/sda1" # or sda4; some usb-stick have partition
# 4 created as factory default.
```

2. Et maintenant, ne montez surtout pas /dev/sda1. Mais tapez la commande :

mdir u:

...pour voir ce que contient votre clef.

mcopy \* u:

...pour copier le contenu du répertoire courant dans la clef.

```
mcopu u:\* .
```

...pour copier tous les fichiers de la clef dans le répertoire courant (n'oubliez pas l'antislash "\" devant le caractère spécial "\*" !).

Vous en conviendrez, il est difficile de faire plus simple !

## Permettre aux utilisateurs de monter un système de fichiers

Normalement, la commande `mount` est réservée au super-utilisateur (`root`). Il peut être pratique toutefois d'autoriser les utilisateurs courants à monter eux-mêmes certains périphériques (CDrom, clé USB...). Tout ce qu'il y'a à faire, c'est d'ajouter (voire modifier) une ligne dans le fichier `/etc/fstab`, comme suit :

```
# dev      mount point    fs type  flags
/dev/sda1  /mnt/usb       auto     noauto,user 0 0
```

Cette ligne indique que `sda1` (la première partition du premier disque SCSI – les lecteurs USB sont vus par Linux comme des disques SCSI !) ne doit pas être montée automatiquement au démarrage (`noauto`) et peut être montée/démontée par tous les utilisateurs (`user`). Le paramètre `auto` dans le champ FS-type permet la détection automatique du type du système de fichiers au moment du montage, ainsi la commande "`mount /mnt/usb`" fonctionnera toujours pour les utilisateurs, que la clef soit formatée avec un système `vfat`, `ext2` ou autre.

## Digression sur les portables et NFS

Le système de fichiers pour réseaux NFS (Network File System) est particulièrement adapté si vous desirez centraliser les répertoires personnels des utilisateurs d'un réseau. Il vous suffit d'avoir ces répertoires sur un serveur, et de monter les répertoires personnels quand un utilisateur se connecte (depuis n'importe quelle machine de ce même réseau) : où qu'il soit, l'utilisateur retrouve sa configuration, ses préférences et ses données personnelles : le top du top !

Mais comment tenir compte du fait que les utilisateurs de portables ne sont pas toujours connectés au réseau ? L'on pourrait certes copier systématiquement les données (avec `rsync` par exemple), mais le risque est grand d'arriver à des incohérences, faute de discipline... et de toute façon, l'on a pas toujours besoin de la totalité de ses données personnelles. personnellement, je préconise l'utilisation d'un second répertoire personnel avec comme contenu mes préférences et les quelques données dont j'ai besoin en déplacement.

En effet, l'une des particularités des systèmes Unix est que, lorsque vous montez un système de fichiers dans un répertoire, alors son contenu vous est caché et vous voyez à la place le contenu du système de fichiers nouvellement monté.

La voici, la solution ! Avant toute autre chose, créez un lien vers le répertoire personnel à utiliser lorsque l'utilisateur n'est pas connecté au réseau, par exemple :

```
ln -s /home_nonet/guido /home/guido
```

Et quand l'utilisateur est sur le réseau, faites en sorte que son script de connexion monte son répertoire personnel du serveur NFS dans `/home/guido` (cachant alors le lien vers `/home_nonet/guido`...).

Connecté, l'utilisateur verra le contenu de son répertoire personnel habituel (celui présent sur le serveur NFS) et pourra toujours effectuer des copies de/vers /home\_nonet/guido au besoin. Et en déplacement, le contenu affiché de /home/guido sera celui de /home\_nonet/guido.

Une solution aussi simple qu'élégante !

## Conclusion

La mise en place d'une couche d'abstraction pour le matériel a toujours été l'un des principaux buts affichés d'Unix. Et la notion de montage/démontage de systèmes de fichiers (ou de disques) est l'une des pièces maîtresses pour réussir un tel prodige ! Même si Unix est largement antérieur à la plupart des systèmes d'exploitation modernes pour PC, ce concept n'en reste pas moins aussi évolué et moderne qu'efficace.

---

<p><u>Site Web maintenu par l'équipe d'édition</u> <u>LinuxFocus</u> <u>© Guido Socher</u> "some rights reserved" see <a href="http://linuxfocus.org/license/">linuxfocus.org/license/</a> <a href="http://www.LinuxFocus.org">http://www.LinuxFocus.org</a></p>	<p>Translation information: en --&gt; -- : Guido Socher (<a href="#">homepage</a>) en --&gt; fr: Guillaume Baudot &lt;guillaume.baudot(at)caramail.com&gt;</p>
--	--

2005-01-22, generated by lfparsr\_pdf version 2.51