

<https://clx.asso.fr/spip/?Planter-Linux-sans-tout-planter>



FSCK SUCK(s)

Planter Linux sans tout planter

- Documentations - Technique -



Date de mise en ligne : lundi 26 août 2002

Copyright © Club LinuX Nord-Pas de Calais - Tous droits réservés

Installer du matériel, une carte, un scanner... Ce n'est pas toujours super facile. Et ça le devient encore moins, quand le matériel est vieux, et... hum... instable.

Comment éviter d'attendre 15 minutes après un plantage hard ? Comment éviter de perdre des données parce qu'il y a eu une panne de courant ? Les solutions existent !

Le matériel, ça a tendance à planter. C'est d'ailleurs là que se situe le point faible de Linux : il ne répare pas le matériel tout seul. Ben oui, c'est pour ça qu'on fait des systèmes embarqués (sous [EmbLinux](#) ;-)) mais ça c'est une autre histoire.

En cas de plantage, si on ne fait rien, les partitions montées en lecture/écriture ne sont pas démontées correctement. Deux problèmes se posent, qui sont liés au cache en écriture de linux :

– **Le cache n'est pas vidé**, il y a corruption de données. C'est à dire que des fichiers qui étaient en train d'être modifiés passent dans un état instable. Les données qu'ils contiennent ne sont pas complètes [1]. Il faut donc corriger le

contenu de ces fichiers, ou les supprimer.

– **Aucune corruption, mais le système ne le sait pas**. Un test de toute la partition sera effectué, ce qui risque de prendre un certain temps sur les partitions non journalisées comme celles formatées en ext2 [2].

Nota : Le test en question (fsck) n'est pas exécuté uniquement si le système détecte des erreurs. Chaque partition possède une valeur de *nombre maximal de montages* (ou *dump*) et un compteur de *montages* [3]. Au bout d'un nombre défini de montages, un test fsck est effectué sur cette partition. Le programme `tune2fs` permet de régler la valeur de ce compteur. Voir la suite de l'article.

Les magic SysReq

Avant que la situation ne soit vraiment catastrophique et, si vous en avez la possibilité, il vaut mieux essayer de copier le cache interne vers le disque dur, soit par la commande `sync`, soit par les *magic SysReq Keys* (traduction libre : les touches de requête système magiques).

Il s'agit d'une combinaison de touches permettant d'envoyer directement des ordres au kernel linux. Un [pointeur chez linuxHQ](#) ou une petite lecture du fichier `/usr/src/linuxXXXXX/Documentation/sysrq.txt` [4] vous en dira plus long là-dessus.

Pour utiliser ces combinaisons de touches, il faut avoir compilé le noyau du système avec le support des *magic sysreq keys* pour les noyaux 2.1 et supérieur, dans le menu "Kernel Hacking" [5]. Voir à ce propos : [La page kernel de Lea-linux.org](#).

N'oubliez pas de taper `lilo` après avoir copié le noyau fraîchement recompilé (!). Afin que le système prenne en compte les modifications que vous avez apportées au fichier `/etc/lilo.conf`.

Pour activer les touches magiques, il faut taper une commande à la main, ou l'ajouter dans un des scripts de

démarrage [6] :

```
echo "1" > /proc/sys/kernel/sysrq
```

Si votre machine se plante sans espoir de retour, vous pouvez donc taper dans l'ordre :

ALT - SysRq [Z] - s SYNChronise le cache et le disque

ALT - SysRq - u dé-moUnte les disques et les remonte en lecture seule

ALT - SysRq - b redémarre (reBoote) le système, sans écrire les données sur le disque

Notez bien que sur certaines architectures, la touche SysRq (ou Syst) ne sera pas fonctionnelle, voire carrément inexistante (comme sur une Sparc ou une Sun Ultra, par exemple, où il faut taper **Stop** au lieu de **SysRq**). Quoiqu'il en soit, toutes les infos sur les touches magiques sont disponibles (en Anglais) aux [quartiers généraux de Linux](#)

Notez également que ces trois commandes affichent comme résultat **OK** sur la console texte une fois le traitement achevé. Tant que vous n'avez pas vu le message OK, si vous n'êtes pas sous X-Window, bien sûr, ce n'est pas la peine de passer à la combinaison de touches suivante. Enfin, si le noyau (kernel) est vraiment bien planté, les combinaisons de touches SysReq ne serviront à rien du tout.

Bien structurer ses données

Les touches magiques ne sont pas toujours suffisantes pour éviter une perte de données. En cas de redémarrage complet (redémarrage en appuyant sur le bouton reset de l'ordinateur, ou initiative autonome de la part de votre ordinateur) — si vous n'avez pas le temps d'anticiper —, en cas de panne de courant ou d'appui impromptu sur le gros bouton rouge de la multiprise de courant, aucune intervention de secours n'est possible. Tout se passe en amont, pour éviter d'avoir trop de casse au *fsck* [8].

Tout d'abord, si vous disposez de plusieurs disques, il convient de séparer les données du système. Ca permet de limiter la casse en cas de pépin sur un disque, mais ça permettra également de monter des partitions en lecture seule. Et ça c'est vraiment intéressant.

Si vous n'avez qu'un disque, vous pouvez tout de même faire deux partitions, l'une pour le système et l'autre pour les données. Mieux, trois partitions. /, /home et /var

Je passe sur la création des partitions, chacun a sa propre technique, mais disons que PQMagic [9] (ou plutôt Partition Magic, de PowerQuest) tout en étant un logiciel commercial et propriétaire, permet, entre autres, de changer la taille des partitions ext2 tout en conservant les données intactes (enfin... normalement).

Notons qu'il existe d'autres outils plus récents, n'ayant donc pas la renommée de PQMagic comme [DiskDrake](#), inclus dans la distribution Mandrake, ou [GNU Parted](#) qui remplissent pourtant les mêmes fonctions. Là, ce sont des outils libres, mais personnellement, je ne les ai jamais utilisés.

Bien. Maintenant, on imagine que vous avez deux partitions, vous pouvez éventuellement déplacer des répertoires entiers de l'une vers l'autre, par exemple pour mettre /usr/src dans /home/src :

```
# mkdir /home/src # cp -rp /usr/src /home/src # mv /usr/src/ /usr/oldsrc # ln -s /home/src /usr/src
```

[10]Et voilà, un répertoire déplacé.

Pour information, sur ma machine, j'ai un très gros répertoire /home sur un disque à part, un repertoire racine (/) de 150 Mo, un répertoire /usr de 1 Go, et un répertoire /var à part. Mon /usr/src est sur /home. De temps en temps

(quand les besoins de place se font sentir), je déplace un répertoire de l'un vers l'autre (tout SAUF /lib qui contient des bibliothèques vitales pour les commandes mv, ls, rm, etc.)

La bonne idée étant de mettre les informations statiques à part, et de regrouper les informations dynamiques (/var/mysql, /var/spool, /var/log, /usr/src, /tmp, ...) sur une ou plusieurs partitions.

Monter une partition ro [\[11\]](#) au lieu de rw [\[12\]](#)

Maintenant que vous avez regroupé toutes les informations statiques, vous allez pouvoir passer les partitions en lecture seule.

Attention, un fichier est très important dans ces opérations de montage/démontage, il s'agit de /etc/mstab, dans lequel le système écrit les partitions montées avec leurs options, et leur point de montage. Il doit donc rester accessible en écriture. Cela dit, certains "fondus" sont capables de mettre /etc en lecture seule, mais personnellement je ne le recommande pas. A moins que ce ne soit absolument nécessaire...

Lorsque vous utilisez la commande *mount*, servant à monter les partitions, LA bonne idée consiste à configurer la partie "options", que l'on indique par un **-o**. Dans ces options, on trouve le type de montage (ro, rw), si c'est un montage "from scratch", ou un "remount", etc.

Il est ainsi tout à fait possible de changer les options d'une partition "à la volée" avec un **-o remount**. Attention, il faut toutefois garder à l'esprit que certains process "occupent" les partitions, et que ces opérations se font avec le compte root.

Changer les options de /, par exemple, ne peut se faire que si vous êtes en mode single-user, sinon les démons démarrés empêcheront le système de démonter/remonter [\[13\]](#).

Les répertoires qui ne risquent rien à être passés sur une partition en ro, sont /usr, bien sûr, /lib, /boot, /opt, /sbin, /bin, /var/www, /var/samba, et sans doute d'autres.

Les répertoires qu'il NE FAUT PAS passer en ro sont /var/spool, /var/log, /var/named, /tmp...

Enfin, les répertoires litigieux, sont /etc [\[14\]](#), /home, et /var/lib. Pour ceux là, tout dépend de votre humeur, et de l'utilisation de la machine. Un /home peut très bien être monté en ro sur un serveur par exemple, si ce n'est pas un serveur NFS ;-)

Pesez bien le pour et le contre de toutes vos manip, et surtout, faites des tests. Par exemple, déplacer /lib et /boot d'un système *en vie* n'est pas possible. Vous aller tout planter, dans la mesure où vous déplacez des fichiers utilisés par le noyau et par les commandes de base (ls, mkdir, mv....). Il faut donc le faire à partir d'un disque rescue (disque servant à se connecter sur le système afin de récupérer des données ou effectuer des opérations de maintenance disque), par exemple.

Pour changer l'état d'une partition, c'est à dire ro ou rw (read-write, lecture-écriture), il suffit de taper :

```
# mount /usr -o ro -o remount Pour la remonter rw (par exemple pour installer un package) : # mount /usr -o rw -o remount
```

Simple, non ?

Pour rendre ces changements définitifs, consultez la liste des partitions montées avec *mount*, et éditez le fichier

/etc/fstab comme cela, par exemple :

```
# /etc/fstab: static file system information. # # /dev/hda2 / ext2 defaults,errors=remount-ro 0 1
/dev/hda3 none swap sw 0 0 proc /proc proc defaults
0 0 /dev/hda1 /boot ext2 rw 0 2 /dev/hda5 /usr ext2 ro
0 2 /dev/hda6 /var ext2 rw,grpquota,usrquota 0 2 /dev/hda7 /tmp ext2 rw
0 2 /dev/hda8 /home ext2 grpquota,usrquota 0 2 /dev/fd0 /floppy
auto defaults,user,noauto 0 0 /dev/cdrom /cdrom iso9660 defaults,ro,user,noauto,exec 0 0
```

Voilà. Au prochain redémarrage violent, vous n'aurez pas de fsck à faire sur /usr. Cool, non ;-) ?

Passer en ext3

La solution ultime pour éviter les pertes de données à cause d'un plantage système, c'est bien sûr un système de fichiers journalisé. Oh bien sûr, vous pouvez choisir les *reiserfs*, *jfs* ou autres, mais le système de fichiers **ext3** fonctionne très bien. Il est disponible à partir des noyaux 2.4 et, je pense, pour ceux de la série 2.2.

Le système de fichiers ext3, possède des caractéristiques proches de celles d'ext2, avec en plus un journal qui stocke les modifications apportées sur les fichiers.

Grosso-modo, un démon supplémentaire est démarré, dont le rôle sera d'écrire la liste des changements effectués sur une partition.

En cas de crash, une lecture du journal à l'envers est effectuée, et les fichiers sont remis dans un état stable.

Compiler son noyau

Tout d'abord, si votre noyau ne l'a pas encore, il faut y inclure le support de l'ext3 [15]. Soit vous le compilez sous forme de module, si votre / et votre /lib/modules restent sur une partition ext2, par exemple en lecture seule [16], soit vous recompilez complètement votre noyau [17].

basculer la partition

Une fois votre nouveau noyau tout beau tout neuf recompile, vous allez utiliser **tune2fs**, qui sert à changer les paramètres des partitions ext2, pour activer le support du journal.

En fait, cette commande ne sert pas qu'à ça, mais aussi à changer le nombre de montages maxi entre deux fsck, le nombre de blocs réservés à root, etc.

```
man tune2fs TUNE2FS(8) NAME tune2fs - adjust tunable filesystem parameters on second extended filesystems
```

Une fois le fichier journal créé, un simple remontage de la partition en ext3 permettra d'activer la journalisation. Il n'est pas impossible qu'un message d'erreur (warning) apparaisse au remontage, sur une histoire de "flag".

Effectuez un fsck de la partition - en mode single user ou en désactivant l'écriture sur la partition (read-only) bien sûr - et tout rentrera dans l'ordre.

Enfin, modifiez le fichier /etc/fstab pour tenir compte du changement. Simple, non ?

Il suffit de remplacer ext2 par ext3 dans la ligne correspondant à la partition.

Et en cas de problème ?

Ah. Comment faire pour réparer une partition ext3 avec un vieux noyau qui ne prend pas en charge ce format... Il y a

une astuce.

Comme ext2 et ext3 sont compatibles (j'entends par compatible le fait qu'une partition ext3 puisse être montée comme une partition ext2 en perdant le bénéfice de la journalisation bien sûr), il est tout à fait possible de démarrer avec un CD-ROM de secours, type "rescue" de Debian, RedHat, Mandrake, les mini distributions Tomsrbt ou les CD Live comme DemoLinux, et de travailler sur les partitions après les avoir montées comme des partitions ext2 [18].

Autre solution, si vous avez l'habitude, comme moi, de travailler avec plusieurs noyaux (un 2.2, un 2.4 avec différentes options), vous pouvez configurer le système pour qu'il reconnaisse automatiquement le type de la partition, en changeant le type `ext3` par `auto` dans le `/etc/fstab`. Ce n'est pas vraiment conseillé, mais chez moi, ça a l'air de fonctionner (sauf pour la racine /).

Au boulot !

Post-scriptum :

Un très bon article (en anglais) sur les systèmes de fichiers journalisés est dispo sur la [Linux Gazette n°35](#).

Une [page de pointeurs](#) sur les systèmes d'urgence, ie. les rescue CD.

Une [page](#) en français sur ext3, trouvée sur linux-france.org.

Un [article](#) en français expliquant comment passer une partition en reiserfs.

[1] puisque certaines informations qui auraient dû être écrites sont restées dans le cache mémoire, et sont donc perdues au redémarrage

[2] Le type de partition le plus courant sous Linux

[3] ça vient de "to mount", et de la commande `mount` ;-)

[4] Sur une Redhat Mandrake ou SuSE, il s'agit de `/usr/src/linux`. Sur une Debian, c'est `/usr/src/kernel-2.XX`. Pour les autres distributions, je ne sais pas ;-)

[5] Les noyaux compilés avec la plupart des distributions Linux possèdent déjà le support des SysReq Keys, vous évitant une recompilation inutile

[6] Ça dépend des distributions, mais `/etc/rc.d/rc.local`, ou `code>/etc/rcS.d/S01sysreq` devraient faire l'affaire sur RedHat/Mandrake ou sur Debian... Sur une SuSE, il suffit d'activer un paramètre `ENABLE_SYSRQ` dans le fichier de configuration `/etc/rc.config` ou `/etc/sysconfig/sysctl` (SuSE 8.0).

[7] ou **Syst** ou [Impr Ecr] sur un clavier Français, c'est à dire la touche de gauche sur le paquet de trois en haut, à droite de **F12**

[8] Ce n'est pas une contrepétrie. C'est vraiment une [commande qui existe](#)

[9] Celui qui sort vos partitions de la "merde"

[10] Si vous vous sentez en veine,

```
tar cvpsf - /usr/src | tar -C /home/src -xf -
```

n'est pas mal non plus. C'est à vous de voir ;-)

[11] "read-only" c'est à dire en lecture seule

[12] "read/write", c'est à dire en lecture, *ET* en écriture

[13] Tiens, au passage, lorsque vous voyez le satané message *system busy...* il s'agit d'un ou plusieurs programmes démarrés sur le filesystem que vous essayez de modifier (monter, démonter, remonter). La commande `fuser` vous permet de savoir quel est le processus qui utilise un fichier donné :

```
gaetan@nutella:~$ fuser -v /bin/bash
USER          PID ACCESS COMMAND /bin/bash
mikael       7376 ....m bash
gaetan       8216 ...e. bash
nico
8515 ...e. bash
```

Vous pouvez même demander à fuser de tuer les processus en question avec l'option `-k`.

[14] A cause du fichier `mtab`

[15] Sur le CD n° 5 de la Debian Woody, ou sur le CD d'installation de la Mandrake, il y a un noyau qui comprend l'ext3. Il n'est pas impossible que le noyau présent sur votre machine en dispose déjà, mais là ça dépend de votre installation.

[16] Si vous compilez le support de l'ext3 sous forme de module, et que votre noyau (et vos modules) sont sur une partition ext3, que va-t-il se passer, à votre avis ?

[17] On me signale l'utilisation possible de `initrd` afin d'ajouter le support de l'IDE ou du SCSI (et des systèmes de fichiers compilés en module) avant le démarrage du système. Pour tout cela, référez vous à un article sur la compilation du noyau, par exemple : [La page kernel de Lea-linux.org](#)

[18] La dernière Mandrake possède tous les outils nécessaires pour redimensionner une partition, y compris ext3, et le CD n°5 de la Debian Woody dispose du support de l'ext3 et de `reiserfs`. Ce sont donc de bonnes bases de travail.