Guide pratique SquashFS

Version française du *SquashFS HOWTO*

Artemiy I. Pavlov, ArtemioLabs [http://artemiolabs.com/] `<ap CHEZ artemio POINT net>`

Adaptation française: Matthieu Thomassin

Relecture de la version française: Yvon Benoist

Préparation de la publication de la v.f.: Jean-Philippe Guérard

1.7.fr.1.0

2008-03-06

Historique des versions

Version 1.7.fr.1.0	2008-03-06	MT, YB, JPG
Première adaptation française.		
Version 1.7	2005-03-25	AP
Changements selon la version 2.1 de SquashFS.		
Version 1.6	2004-11-10	AP
Changements selon la version 2.0 de SquashFS. Corrections dans le texte.		
Version 1.5	2004-06-07	AP
Changements selon la version 2.0 alpha de SquashFS. Beaucoup de descriptions améliorées et clarifiées. Découpage des instructions pour le noyau Linux 2.6.x (nouveau) et les versions 2.4.x.		
Version 1.1	2004-05-22	AP
Changements selon la version 1.3r3 de SquashFS.		
Version 1.0	2004-02-19	AP
Version initial, revu par LDP.		
Version 0.2	2003-12-08	AP
Corrections du texte, licence ajoutée.		
Version 0.1	2003-11-24	AP
Version initial. Instructions pour la version 1.3r2 de SquashFS.		

Résumé

Ce guide pratique décrit l'utilisation de SquashFS — un système de fichier en lecture seule hautement compressé pour Linux, qui est prévu pour être utilisé dans des systèmes embarqués et de petite taille, et partout où on peut avoir besoin d'utiliser un système de fichier compressé. Avec ce document, vous allez apprendre comment préparer un noyau Linux à l'utilisation de SquashFS, créer un système de fichier compressé et l'utiliser en toute sérénité.

Table des matières

1. Qu'est-ce que SquashFS ... 2
    1.1. Introduction ... 2
    1.2. Aperçu de SquashFS ... 2
    1.3. Mettons cela au clair ... 3
2. On se prépare pour SquashFS ... 3
    2.1. Acquisition de SquashFS ... 3
    2.2. Préparation d'un noyau compatible SquashFS ... 3
    2.3. Compilation de l'outil **mksquashfs** ... 5
3. L'outil mksquashfs expliqué ... 6
    3.1. Utilisation de mksquashfs ... 6

3.2. Options en lignes de commande 7
4. Création et utilisation des systèmes de fichiers compressés 8
4.1. Étapes principales 8
4.2. Compression des systèmes de fichiers 9
4.3. Création de micro-systèmes incorporés/minuscules 10
5. Remerciements 12
6. Licence 12

1. Qu'est-ce que SquashFS

1.1. Introduction

Lors de la création de systèmes Linux de petite taille et embarqués, chaque octet du périphérique de stockage (disquette, disque à mémoire flash, etc.) est très important, ainsi la compression est utilisée partout où elle est possible. En outre, des systèmes de fichiers compressés sont fréquemment nécessaires pour l'archivage. Pour les très grosses archives publiques, ainsi que pour les archives sur support personnel, elle est essentielle.

Avec SquashFS, on passe à un niveau supérieur. C'est un système de fichier en lecture seule qui permet de compresser l'intégralité d'un système de fichiers ou un répertoire unique, les écrire sur d'autres périphériques/partitions ou vers des fichiers ordinaires, pour ensuite les monter directement (si c'est un périphérique) ou en utilisant un dispositif en local (s'il s'agit d'un fichier). Par sa conception compacte et modulaire, le système SquashFS n'est que pur bonheur. Pour l'archivage, SquashFS offre une souplesse et des performances beaucoup plus grandes qu'une archive .tar.gz.

SquashFS est distribuée comme correctif applicable sur le noyau source de Linux (qui active dans le support de SquashFS dans le noyau), et l'outil **mksquashfs**, qui crée un système de fichiers compressé (dans un fichier ou un périphérique d’entrée-sortie par blocs).

La dernière version de SquashFS est la 2.x, la première étant la 1.x. Ce document décrit les deux versions avec des notes données propres à l'une ou l'autre. Par exemple, s’il y a des caractéristiques ou paramètres différents entre ces versions, ce sera écrit de cette façon : *nouvelle valeur (2.x) ou ancienne valeur (1.x)*

1.2. Aperçu de SquashFS

- Données, nœuds d'index et répertoires compressés
- SquashFS enregistre tous les uid/gids (32 bits), et l'heure de création de fichier
- Fichiers jusqu’à 2^32 octets supportés; les systèmes de fichiers peuvent aller jusqu’à 2^32 octets
- Nœuds d'index et répertoires de données hautement compressés, et groupés par multiplets; chaque nœud d'index compressé fait en moyenne 8 octets de long (la longueur exacte varie selon le type de fichier, c'est-à-dire que les nœuds d’index de fichier courant, de répertoire, de lien symbolique ainsi que les nœuds des unités d’entrée-sortie de caractères ou par blocs ont des tailles différentes)
- SquashFS peut utiliser des blocs dont la taille va jusqu'à 32 Kb (1.x) et 64 Kb (2.x), ce qui garantit un meilleur rapport de compression qu'avec un bloc normal de 4 Kb
- SquashFS 2.x introduit le concept de *blocs de fragments* : la capacité à joindre/regrouper de multiples fichiers plus petits que la taille d'un bloc dans un seul bloc, ce qui améliore les taux de compression d'archivage
- Les fichiers dupliqués sont détectés et supprimés

- Les architectures big endian et little endian [gros-boutiens et petits-boutiens] sont supportées toutes les deux; SquashFS peut monter des systèmes de fichiers crées sur des machines ayant un ordre d'octets différents

1.3. Mettons cela au clair

Assurons-nous maintenant que vous comprendrez plus facilement tout ce qui sera abordé ici. La procédure de base pour faire marcher SquashFS est la suivante :

1. Application du correctif et recompilation du noyau Linux cible pour activer le support de SquashFS
2. Compilation de l'outil **mksquashfs**
3. Création d'un système de fichier compressé avec **mksquashfs**
4. Essai : montage d'un système de fichiers compressé à un emplacement temporaire
5. Modification du fichier `/etc/fstab` ou des scripts de démarrage de votre système Linux cible pour monter le nouveau système de fichiers compressé quand ce sera nécessaire

2. On se prépare pour SquashFS

2.1. Acquisition de SquashFS

Le site principal de SquashFS est basé à l'adresse http://squashfs.sourceforge.net/ ; il contient les informations pour les dernières versions et la liste des changements apportés, ainsi que les informations générales sur SquashFS. On peut obtenir la dernière version de SquashFS en allant sur la page [http://sourceforge.net/project/showfiles.php?group_id=63835] du projet SquashFS de SourceForge.

2.2. Préparation d'un noyau compatible SquashFS

Pour pouvoir l'utiliser, SquashFS doit être supporté dans votre noyau, exactement comme s'il s'agissait d'un système de fichiers `reiserfs` ou `ext3` . Il faut s'assurer qu'il existe un correctif approprié pour la version de votre noyau ; il devrait être dans le sous-répertoire `linux-2.x.y` de l'arborescence de SquashFS. En outre, rappelez vous que dans la plupart des cas vous aurez besoin d'un noyau source Linux *propre* (original) récupérer sur kernel.org [http://kernel.org/]. Si votre noyau source vient d'un revendeur de distribution, il aura peut-être été pré-corrigé par le revendeur, et l'application du correctif de SquashFS ne donnera très certainement aucun résultat, car les correctifs SquashFS sont faits par rapport aux noyaux Linux *d'origine*

2.2.1. Application du correctif sur la source du noyau

Avec la source du noyau et le correctif SquashFS qui convient, il n'y a qu'à faire ceci (on suppose que la source de votre noyau est dans `/usr/src/linux` et que la source de SquashFS est dans `/usr/src/squashfs`) :

Passer dans le répertoire source de SquashFS et copiez le correctif du noyau (on supposera qu'il se nomme `squashfs-patch`) vers `/usr/src/linux`.

```
bash# cd /usr/src/squashfs
bash# cp linux-2.x.y/squashfs-patch /usr/src/linux
```

Allez dans le répertoire source du noyau Linux `/usr/src/linux` :

```
bash# cd /usr/src/linux
```

Note

Il faut se rappeler qu'on ne quittera pas ce répertoire à aucun moment au cours des procédures prochaines liées au noyau, et que tous les chemins seront donnés par rapport au chemin `/usr/src/linux`.

Maintenant appliquez le correctif de SquashFS sur la source :

```
bash# patch -p1 < squashfs-patch
```

2.2.2. Compilation d’un noyau 2.6.x

Nettoyez et préparez la source du noyau :

```
bash# make distclean
bash# make mrproper
```

Configurez le noyau en utilisant votre méthode préférée (config/menuconfig/xconfig/gconfig) :

```
bash# make menuconfig
```

1. Dans la partie « *File systems* », sous-partie « *Miscellaneous file systems* », activez l'option « *Squashed filesystem* », aussi bien comme module ou empaqueté dans le noyau. Il n'est nécessaire de compiler SquashFS à l'intérieur du noyau que si vous prévoyez d'utiliser un disque RAM initial compressé (**initrd**).

2. Si vous voulez utiliser un disque RAM initial compressé, activez le « *Initial RAM disk support* » dans la partie « *Device drivers* », sous-partie « *Block devices* ».

3. Si vous voulez avoir, à l'avenir, la possibilité de monter le système de fichier compressé via un *périphérique local*, vous devrez activer le « *Loopback device support* » dans la partie « *Device drivers* », dans la sous-partie« *Block devices* ».

Maintenant on peut compiler le noyau et les modules :

```
bash# make
```

2.2.3. Compilation d'un noyau 2.4.x

Configuration du noyau :

```
bash# make menuconfig
```

1. Dans la partie « *File systems* », activez l'option « *Squashed filesystems* », aussi bien comme module ou empaqueté dans le noyau. Il n'est pas nécessaire de compiler SquashFS à l'intérieur du noyau que si vous prévoyez d'utiliser un disque RAM initial compressé (**initrd**).

2. Si vous voulez utiliser un disque RAM initial compressé, activez le « *Initial RAM disk support* » dans la partie « *Block devices* ».

3. Si vous voulez avoir, à l'avenir, la possibilité de monter le système de fichiers compressé via un *périphérique local*, vous devrez activer le « *Loopback device support* » dans la partie « *Block devices* ».

Maintenant on peut compiler le noyau et les modules :

```
bash# make dep
bash# make bzImage
bash# make modules
```

2.2.4. Installation et test du noyau

C’est maintenant que l’on va installer le nouveau noyau qui est prêt pour SquashFS. Les instructions ci-dessous expliquent l'installation et le démarrage du noyau sur la machine hôte. Si on veut, on peut l'installer et le tester sur le système cible.

On suppose que le noyau était compilé pour une architecture x86, et que l'image du noyau compressé est situé dans le sous-répertoire `arch/i386/boot/` de l'arborescence du noyau. Maintenant copiez le noyau dans le répertoire `/boot` (et appelez-le `bzImage-sqsh` par commodité si vous voulez) :

```
bash# cp arch/i386/boot/bzImage /boot/bzImage-sqsh
```

N'oubliez pas d'installer les modules du noyau si vous en avez :

```
bash# make modules_install
```

Modifiez la configuration de votre chargeur de démarrage pour inclure votre nouveau noyau et installez (mettez à jour) le chargeur de démarrage. Maintenant vous pouvez redémarrer avec votre nouveau noyau. Quand la machine redémarre, vérifiez que tout va bien :

```
bash# cat /proc/filesystems
```

Ou, si vous avez construit votre support de SquashFS un module du noyau :

```
bash# insmod squashfs
bash# cat /proc/filesystems
```

Si vous voyez la ligne `squashfs` à l'intérieur d'autres systèmes de fichiers, cela signifie que vous avez correctement activé SquashFS dans votre noyau.

2.3. Compilation de l'outil mksquashfs

Maintenant il faut compiler **mksquashfs**, l'outil pour créer des systèmes de fichiers compressés.

```
bash# cd /usr/src/squashfs/squashfs-tools
```

Compilez et installez **mksquashfs** :

```
bash# make
bash# cp mksquashfs /usr/sbin
```

Si tout s’est bien passé, en tapant **mksquashfs** dans un interpréteur de commandes vous devriez avoir le message sur« l'utilisation ».

3. L'outil mksquashfs expliqué

3.1. Utilisation de mksquashfs

mksquashfs est un outil qui permet de créer de nouveaux systèmes de fichiers compressés, et d'ajouter de nouvelles données dans des systèmes de fichiers compressés existants. Le format général de la ligne de commande pour **mksquashfs** est :

```
bash# mksquashfs source1 source2 ... destination [options]
```

- `source1`, `source2`, etc. : les fichiers et répertoires à ajouter au système de fichiers obtenu, donnés avec les chemins relatifs ou absolus, ou les deux.
- `destination` : fichier ordinaire (fichier image d'un système de fichiers), ou un périphérique de type bloc (tel que `/dev/fd0` ou `/dev/hda3`) où vous voulez mettre votre système de fichiers compressés.

Remarques concernant le comportement par défaut de **mksquashfs** :

- Quand on ajoute les nouveaux fichiers au nouveau système de fichiers ou qu'on les rajoute à un système de fichiers existant, **mksquashfs** va automatiquement renommer les fichiers avec des noms dupliqués de la façon suivante : si deux fichiers ou plus nommés `text` apparaissent dans le même répertoire, le second fichier sera renommé `text_1`, le troisième `text_2` et ainsi de suite.
- Les fichiers dupliqués seront supprimés ; il n'y aura donc qu'une seule instance physique d'un fichier (avec SquashFS 2.x, on peut désactiver la détection/suppression des fichiers dupliqués avec l'option **-no-duplicates**).
- Si la *destination* possède déjà un système de fichiers compressés, par défaut, les nouveaux éléments *sources* seront ajoutés au répertoire racine existant. Voir les tables d'options en ci-dessous pour forcer **mksquashfs** à écraser par réécriture sur l'ensemble de la destination et/ou changer la façon dont les nouveaux éléments source sont ajoutés. A noter qu'il n'est pas possible d'ajouter un système de fichiers créé avec **mksquashfs** version 1.x en utilisant **mksquashfs** version 2.x. Il faudra monter le système de fichiers au format SquashFS 1.x et copier les fichiers au même endroit, puis les assembler avec les autres fichiers nécessaires pour créer un système de fichiers au format SquashFS 2.x.
- Si un seul fichier source ou répertoire est donné, il devient la racine du système de fichiers nouvellement créé. Si a plus d'un fichier source, ils deviendront tous des sous-éléments de la racine du nouveau système de fichiers ; il en sera de même pour les répertoires.
- Les espaces inoccupés du système de fichiers se compléteront par blocs multiples de 4 Ko : ceci est indispensable pour pouvoir utiliser les systèmes de fichiers sur des périphériques de bloc. Si vous êtes absolument sûr que vous n'en avez pas besoin, utilisez l'option **-nopad** pour désactiver cette fonctionnalité.

Voir la section suivante pour plus d'information sur l'ensemble des options disponibles.

3.2. Options en lignes de commande

Tableau des options pour **mksquashfs** :

Tableau 1. Options en ligne de commande de l'outil mksquashfs

Option	Description
-2.0	Force **mksquashfs** version 2.1 à créer un système de fichiers en version 2.0
-all-root ou **-root-owned**	Avec cette option, tous les fichiers du système de fichiers cible appartiendront au compte root (UID=0, GID=0)
-always-use-fragments	Divise en fragments tous les fichiers dépassant la taille d'un bloc (version 2.x uniquement, les taux de compression en seront plus élevés)
-b [block size]	Sert à définir la taille d'un bloc [block size] du système de fichiers (32 Ko par défaut) ; cela peut être aussi 512, 1024, 2048, 4096, 8192, 16384 ou bien 32768
-be ou **-le**	Force un système de fichiers, respectivement, en big endian ou en little endian
-check-data	Active des vérifications supplémentaires du système de fichiers
-e [file1] ([file2] ...)	Spécifie les fichiers et/ou répertoires sont à ne pas mettre dans le nouveau système de fichiers qui va être crée
-ef [file]	Spécifie un `fichier` qui contient la liste des fichiers/répertoires à exclure
-force-gid [GID]	Met tous les identifiants de groupe du système de fichier cible à la valeur [GID] (peut être spécifié avec un nom ou un nombre)
-force-uid [UID]	Met tous les identifiants d'utilisateur du système de fichier cible à la valeur [UID] (peut être spécifié avec un nom ou un nombre)
-info	Affiche les fichiers, leurs tailles originales et le taux de compression, tels qu'ils ont été ajoutés dans le système de fichiers
-keep-as-directory	Si la source est un répertoire unique, force ce répertoire à devenir un sous-répertoire de la racine du système de fichiers créé
-noappend	Avec cette option, si le fichier/périphérique de destination contient déjà un système de fichiers compressés, il sera écrasé, et les nouvelles données ne seront donc pas rajoutées dans le système de fichiers existant
-no-duplicates	Ne détecte/ni ne supprime les noms de fichiers dupliqués
-noD ou **-noDataCompression**	Ne compresse pas les données

Option	Description
-noF ou **-noFragmentCompression**	Ne compresse pas les fragments de fichier (versions 2.x uniquement)
-no-fragments	Ne génère pas de blocs en fragments (versions 2.x uniquement, on aura alors pratiquement le même système de fichiers qu'avec les versions 1.x)
-noI ou **-noInodeCompression**	Ne compresse pas la table d'inodes
-nopad	Pour supprimer le remplissage du système de fichiers par blocs en multiples de 4 Ko
-root-becomes [nom]	Peut être utilisé pendant l'ajout à un système de fichiers déjà existant : cela créera un nouveau compte root, et le répertoire [nom] contiendra tous les fichiers/répertoires déjà existants
-version	Affiche un message avec la version, les droits et la licence

Dans la plupart des cas, vous devrez laisser toutes les options de compression/bloc par défaut, car elles permettent à **mksquashfs** de réaliser les meilleurs taux de compression possibles.

4. Création et utilisation des systèmes de fichiers compressés

4.1. Étapes principales

Pour créer un système de fichiers compressés à partir d'un répertoire unique (disons, `/répertoire/quelconque`), et de le sortir dans un fichier ordinaire (produisant ainsi une image du système de fichiers), vous n'avez qu'à dire la phrase magique :

```
bash# mksquashfs /répertoire/quelconque dir.sqsh
```

mksquashfs exécutera la compression et affichera le nombre d'inodes et la taille des données écrits qui en résultent, ainsi que le taux de compression moyen. Maintenant vous avez l'image de votre répertoire `/répertoire/quelconque` dans le fichier `dir.sqsh`. Vous pouvez maintenant utiliser la commande **mount** pour la monter en utilisant un dispositif local :

```
bash# mkdir /mnt/dir
bash# mount dir.sqsh /mnt/dir -t squashfs -o loop
```

Pour vérifier si on obtient bien le résultat escompté :

```
bash# ls /mnt/dir
```

Si vous voulez faire sortir directement le système de fichiers dans un périphérique (disons, votre lecteur de disquette `/dev/fd0`) :

```
bash# mksquashfs /répertoire/quelconque /dev/fd0
```

Ensuite il n'y a qu'à monter le périphérique avec la commande **mount** :

```
bash# mount /dev/fd0 /mnt/floppy -t squashfs
```

Puis vérifiez si c'est bon :

```
bash# ls /mnt/floppy
```

4.2. Compression des systèmes de fichiers

Les opérations décrites ici correspondent à la plupart des cas où le système de fichiers en lecture seule peut être utilisé, que vous le vouliez sur un périphérique de bloc ou dans un fichier. Il pourra s'agir tout aussi bien des immenses archives FTP/HTTP qui ne changent pas souvent, que l'obtention d'une partition compressée `/usr` et toute chose du même genre.

4.2.1. Exemple 1

Supposons que vous ayez un répertoire `/var/arch` avec beaucoup de fichiers et que vous voulez le transformer en un système de fichiers compressés et le garder sur votre partition racine en tant que fichier (il sera une image du système de fichiers que vous monterez en local). Voici les opérations à exécuter

Compressez le répertoire, puis montez le en local pour le tester :

```
bash# mksquashfs /var/arch /var/arch.sqsh
bash# mkdir /mnt/tmp
bash# mount /var/arch.sqsh /mnt/tmp -t squashfs -o loop
bash# ls /mnt/tmp
```

Si tout marche comme prévu, faites monter ce système de fichiers automatiquement au moment du démarrage en ajoutant cette ligne à votre `/etc/fstab` :

```
/var/arch.sqsh /var/arch squashfs ro,defaults 0 0
```

Démontez le système de fichiers du point de montage temporaire, et montez en utilisant cette entrée fstab :

```
bash# umount /mnt/tmp
bash# mount /var/arch
```

Maintenant il n'y a plus qu'à vérifier que tout fonctionne bien :

```
bash# ls /var/arch
```

4.2.2. Exemple 2

Disons que vous avez deux partitions sur le disque dur, `/dev/hda6` (qui est vide) et `/dev/hda7` (elle est plus grosse que `/dev/hda6`, est montée sur `/var/arch`, contient des données et est pleine). Maintenant, disons que vous voulez compresser le système de fichiers `/dev/hda7` et le déplacer dans `/dev/hda6`, puis utiliser `/dev/hda7` pour d'autres choses. Nous supposerons que vous avez la ligne suivante dans `/etc/fstab` (**reiserfs** est juste un exemple de système de fichiers utilisé sur `/dev/hda7`) :

```
/dev/hda7 /var/arch reiserfs defaults 0 0
```

De la même manière que dans l'exemple précédent :

```
bash# mksquashfs /var/arch /var/arch.sqsh
bash# mkdir /mnt/tmp
bash# mount /var/arch.sqsh /mnt/tmp -t squashfs -o loop
bash# ls /mnt/tmp
```

Si tout s'est bien passé, démontez `/dev/hda7` (si besoin est) et utilisez **dd** pour copier `/var/arch.sqsh` dans `/dev/hda6` :

```
bash# umount /dev/hda7
bash# dd if=/var/arch.sqsh of=/dev/hda6
```

Maintenant remplacez la ligne `/dev/hda7` dans `/etc/fstab` par :

```
/dev/hda6 /var/arch squashfs ro,defaults 0 0
```

Montez le nouveau système de fichiers et vérifiez si tout s'est bien passé :

```
bash# mount /var/arch
bash# ls /var/arch
```

N'oubliez pas d'effacer l'image du système de fichiers dont on n'a plus besoin :

```
bash# rm /var/arch.sqsh
```

4.3. Création de micro-systèmes incorporés/minuscules

En disant "incorporés/minuscules", je veux parler de systèmes Linux construits pour démarrer à partir de disquettes, de disque IDE/USB à mémoire flash, de CD-ROM iso-9660, de petits disques durs et autres. Que vous vouliez avoir votre système de fichiers racine en entier sur un support unique (une seule partition, une seule disquette), ou avoir un système modulaire (plusieurs disquettes ou partitions de disques), la procédure est presque identique. La création des systèmes Linux en question n'entre pas dans le cadre de ce Guide Pratique ; il existe des guides pratiques et autres qui y sont consacrés (comme le *Bootdisk HOWTO* et *Linux From Scratch* ; visitez http://www.tldp.org pour récupérer ces documents).

4.3.1. Systèmes de fichiers compressés sur disquette/mémoire flash/disques durs

Pour utiliser SquashFS pour la création de systèmes Linux sur des petits disques, vous n'avez juste qu'à suivre les étapes habituelles de création d'un système minimal, en exécutant les opérations suivantes au moment où c'est nécessaire :

1. Quand vous déployez un noyau sur votre système, vérifiez que vous avez bien activer le support de SquashFS afin qu'il puisse monter les systèmes de fichiers compressés.

2. Utilisez **mksquashfs** pour la création des disques RAM initiaux en lecture seule et/ou du répertoire racine et/ou des autres systèmes de fichiers.

3. N'oubliez pas de paramétrer les types de système de fichiers à `squashfs` dans `/etc/fstab` ou les scripts de démarrage de votre système pour monter les systèmes de fichiers compressés.

Exemple avec la disquette. Disons que vous avez l'arborescence de votre système de disquettes dans `/home/user/floppylinux` et que vous voulez placer le système de fichiers racine sur l'une d'entre elles et `/usr` sur une autre. Voici ce que vous devrez faire :

```
bash# cd /home/user
bash# mksquashfs floppylinux root.sqsh -e usr
bash# mksquashfs floppylinux/usr usr.sqsh
```

Note

Vous voyez ici comment nous utilisons l'option **-e** pour exclure le répertoire `/usr` pour l'image des systèmes de fichiers racine.

Note

N'oubliez pas de spécifiez `squashfs` dans le `/etc/fstab` de votre disque racine ou dans les scripts de démarrage lorsque vous montez le système de fichiers `/usr`.

Insérez un disque racine dans votre lecteur de disquettes 3,5" (je suppose que vous avez lilo ou grub dessus et que, donc, un système de fichiers existe sur cette disquette, et le système de fichier racine sera sous le répertoire `/boot` de ce système de fichiers) :

```
bash# mount /mnt/floppy
bash# cp root.sqsh /mnt/floppy/boot
```

Quand c'est fait, démontez la disquette racine, remplacez la disquette par un disque `/usr` et utilisez **dd** pour transférer le système de fichiers `usr` :

```
bash# dd if=usr.sqsh of=/dev/fd0
```

4.3.2. Systèmes de fichiers compressés sur CD-ROM

Avec SquashFS, on peut compresser de gros systèmes de fichiers qu'on pourra mettre sur des CD "Live" (par exemple).

1. Activez SquashFS dans le noyau Linux du système cible.

2. Créez un système de fichiers compressés racine.

3. Modifier le `/etc/fstab` ou les scripts de démarrage du système cible pour monter le système de fichiers compressés quand vous en aurez besoin.

Si vous créez un système de fichiers racine à partir d'un système de fichiers Linux qui tourne, utilisez l'option **-e** pour que **mksquashfs** exclue tous les pseudos systèmes de fichiers tels que `/proc`, `/sys` (sur les noyaux Linux après 2.5.x) et `/dev` (quand vous utilisez DevFS). En outre, n'oubliez pas d'ajouter l'image du système de fichiers elle-même qui est créée avec **mksquashfs** (Je pense que vous comprenez les raisons de ces exclusions).

5. Remerciements

Je souhaiterais exprimer mes sincères remerciements et mon immense respect à :

- Phillip Lougher — pour son brillant travail sous SquashFS, pour la création d'un correctif exclusif pour Linux-2.4.18, pour l'aide qu'il a apporté à l'amélioration de ce Guide Pratique et ses réponses à mes courriels.
- Tabatha Marshall de TLDP pour m'avoir aidé à amener ce Guide Pratique à la version finale 1.0.
- Tous les participants à The Linux Documentation Project http://www.tldp.org pour leur superbe travail sur tous les guides pratiques et autres guides qui m'ont beaucoup aidé pour explorer et bidouiller Linux.
- Tous les abonnés des listes de distribution de TLDP qui m'ont aidé à démarrer.
- Je remercie infiniment et adresse mon respect à toutes les personnes qui développent les logiciels libres.

6. Licence