
Guide pratique de sauvegarde et de récupération sous Linux

Version française du guide pratique *Linux Complete Backup and Recovery HOWTO*

Charles Curley <charlescurley CHEZ charlescurley
POINT com>

Adaptation française: Denis Berhaut

Relecture de la version française: Guillaume Lelarge

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

1.6.fr.1.0

10 octobre 2004

Historique des versions		
Version 1.6.fr.1.0	2004-10-10	DB, GL, JPG
Première adaptation française.		
Version 1.6	2004-04-29	C^2
Ajout de notes sur Knoppix, Syslinux, PPARTE, QtParted, quelques autres CDROM de secours, ainsi que quelques corrections. (<i>Added Knoppix notes, Syslinux, PPARTE, QtParted, some other rescue CDs, and made some fixes.</i>)		
Version 1.5	2003-12-19	C^2
Notes sur Fedora et GRUB. (<i>Fedora and GRUB notes.</i>)		
Version 1.4	2003-08-17	C^2
Quelques notes sur le gravage des CDROM et rajouts concernant les fichiers à exclure. (<i>Some notes on burning CD-ROMs, and more on files to exclude.</i>)		
Version 1.3	2003-04-24	C^2
Substitution d'une nouvelle adresse de courrier électronique et d'une URL aux anciennes. (<i>Substituted new email address and URL for old.</i>)		
Version 1.2	2003-02-12	C^2
Ajout de notes pour Red Hat 8.0, pour le support de FAT32, scission de la première étape des scripts de restauration, ainsi que quelques changements mineurs. Notes sur Amanda. (<i>Added Red Hat 8.0 notes, support for FAT32, split the first stage restore scripts, and other minor changes. Notes on Amanda.</i>)		
Version 1.1	2002-09-10	C^2
Nouveau code pour prendre en charge les partitions ext3 dans le fichier <code>make.fdisk</code> , ainsi qu'une note sur <code>initrd</code> . (<i>New code to handle ext3 partitions in make.fdisk, and a note on initrd.</i>)		
Version 1.0	2002-07-24	C^2
Désormais, nous utilisons la compression bz2 pour la première étape, disposons de l'option « run time » pour la vérification des blocs défectueux et avons créé un script qui exécute entièrement la première étape. (<i>We now use bz2 compression in the first stage, have the run time option to check for bad blocks, and have a script that runs the entire first stage.</i>)		

Résumé

Imaginez une seconde que votre disque dur vient de se transformer en un palet de hockey hors de prix. Imaginez qu'après un incendie, votre ordinateur ressemble à un sujet que Salvador Dali apprécierait. Et maintenant ?

La restauration complète, que l'on appelle parfois restauration intégrale de système, est un processus consistant à remonter un ordinateur après une panne catastrophique. Pour effectuer une restauration complète, vous devez disposer de sauvegardes complètes, non seulement de vos systèmes de fichiers, mais aussi des informations sur vos partitions, ainsi que sur d'autres données. Ce guide pratique est un tutoriel qui vous montrera pas à pas comment sauvegarder un ordinateur sous

Linux pour être en mesure d'effectuer une restauration intégrale de système, et comment effectuer cette restauration intégrale de système. Il comprend des scripts destinés à ces tâches.

Table des matières

1. Introduction	3
1.1. Droits d'utilisation (<i>Copyright Information</i>)	3
1.2. Limitation de Responsabilité (<i>Disclaimers</i>)	3
1.3. Nouvelles versions	4
1.4. Historique	4
1.5. Commentaires et réactions	5
1.6. Traductions	5
2. Tour d'horizon	5
2.1. Limitations	6
3. Préparation	7
3.1. Installation du lecteur ZIP	7
4. Création de la sauvegarde de l'étape 1	7
4.1. Variations sur le thème	9
5. Première étape de restauration	11
5.1. Démarrer tomsrtbt	11
5.2. Restauration	11
6. Seconde étape de la restauration	13
7. Notes spécifiques aux distributions	15
7.1. Fedora	15
7.2. Red Hat 9	15
7.3. Red Hat 8.0	16
7.4. Red Hat 7.1	16
7.5. Red Hat 7.0	16
7.6. Knoppix	16
8. Notes concernant certaines application	17
8.1. GRUB	17
8.2. Tripwire	17
8.3. Squid	17
8.4. Arkeia	17
8.5. Amanda	18
9. Quelques conseils pour une récupération après un désastre	19
10. Et maintenant ?	19
10.1. Liste de travail	19
11. Les scripts	20
11.1. Première étape	20
11.2. Deuxième étape	37
11.3. Scripts de sauvegarde du serveur	40
12. Ressources	43
A. License GNU Free Documentation	45
0. PREAMBULE	45
1. APPLICABILITY AND DEFINITIONS	45
2. VERBATIM COPYING	46
3. COPYING IN QUANTITY	46
4. MODIFICATIONS	47
5. COMBINING DOCUMENTS	48
6. COLLECTIONS OF DOCUMENTS	48
7. AGGREGATION WITH INDEPENDENT WORKS	48
8. TRANSLATION	49
9. TERMINATION	49
10. FUTURE REVISIONS OF THIS LICENSE	49
11. How to use this License for your documents	49

1. Introduction

Le processus de restauration intégrale de système consiste à : installer le système d'exploitation à partir des disques du produit. installer le logiciel de sauvegarde de façon à pouvoir restaurer vos données. Restaurer vos données. Puis, il vous faudra restaurer les fonctionnalités en vérifiant vos fichiers de configuration, les droits, etc.

Le processus et les scripts décrits dans ce guide pratique sauvegarderont la réinstallation du système d'exploitation. Le processus décrit ici restaurera uniquement les sauvegardes des fichiers de l'ordinateur de production. La restauration vous rétablira votre configuration intacte, vous épargnant des heures de vérification de votre configuration et des données.

1.1. Droits d'utilisation (*Copyright Information*)

Copyright © 2001, 2002, 2003 Charles Curley pour la version originale.

Copyright © 2004 Denis Berhaut, Guillaume Lelarge, Jean-Philippe Guérard pour la version française.

Ce document est distribué selon les termes de la licence GNU Free Documentation License (GFDL), décrite ci-dessous. Permission est accordée de copier, distribuer et de modifier ce document selon les termes de la GNU Free Documentation License, version 1.1 ou ultérieure publiée par la Free Software Foundation à condition qu'il ne contienne ni Section Inaltérable, ni texte de première ou de quatrième de couverture. Une copie de la licence est incluse dans la section « Licence GNU Free Documentation ».

Copyright © 2001, 2002, 2003 Charles Curley and distributed under the terms of the GNU Free Documentation License (GFDL) license, stated below. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with no Invariant Sections, with no Front-Cover Texts, and with no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have any questions, please contact <linux-howto at metalab.unc.edu>.

Pour toute question relative à la version originale de ce document, veuillez contacter en anglais <linux TIRET howto CHEZ metalab POINT unc POINT edu>.

N'hésitez pas à faire parvenir tout commentaire relatif à la version française de ce document à <commentaires CHEZ traduc POINT org> en précisant son titre, sa date et sa version.

1.2. Limitation de Responsabilité (*Disclaimers*)

Ni l'auteur de ce document, ni le Projet de documentation Linux [<http://www.tldp.org/>] ou qui que ce soit d'autre ne pourra être tenu responsable du contenu de ce document. L'utilisation des concepts, exemples et autres contenus du document s'effectue à vos risques et périls. Il peut comporter des erreurs ou des inexactitudes pouvant endommager votre système. Procédez prudemment, et bien qu'il n'y ait probablement pas d'erreur, l'auteur dégage toute responsabilité à leur sujet.

No liability for the contents of this documents can be accepted by the author, the Linux Documentation Project [<http://www.tldp.org/>] or anyone else. Use the concepts, examples and other content at your own risk. There may be errors and inaccuracies that may damage your system. Proceed with caution, and, although errors are unlikely, the author take no responsibility for them.

Sauf mention spécifique, les droits d'auteur sont la possession de leurs propriétaires respectifs. L'utilisation d'un terme dans ce document ne devrait pas être considérée comme ayant une influence sur la validité d'une quelconque marque déposée ou marque de services.

All copyrights are held by their by their respective owners, unless specifically noted otherwise. Use of a term in this document should not be regarded as affecting the validity of any trademark or service mark.

Le fait de nommer un produit ou une marque ne doit pas être considéré comme une approbation.

Naming of particular products or brands should not be seen as endorsements.

Nous vous recommandons fortement d'effectuer une sauvegarde de votre système avant une installation importante et d'effectuer des sauvegardes à intervalles réguliers. De plus, nous vous recommandons fortement d'utiliser un ordinateur expérimental dédié lorsque vous mettrez les mains dans le cambouis des matériaux de ce guide pratique, en particulier les scripts.

You are strongly recommended to take a backup of your system before major installation and backups at regular intervals. In addition, you are strongly recommended to use a sacrificial experimental computer when mucking with the material, especially the scripts, in this HOWTO.

1.3. Nouvelles versions

Vous pourrez trouver ce document à sa page d'accueil [<http://www.charlescurley.com/Linux-Complete-Backup-and-Recovery-HOWTO.html>] ou sur le site internet du projet de documentation Linux [<http://www.tldp.org/>] dans de nombreux formats. Envoyez vos commentaires à l'adresse <charlescurley CHEZ charlescurley POINT com>.

En fonction de votre navigateur, il vous faudra peut-être maintenir la touche majuscule appuyée pour les télécharger lorsque vous cliquerez dessus.

- compression bzip2 de divers fichiers (beaucoup de petites pages. Lecture plus rapide.) HTML [<http://www.charlescurley.com/Linux-Complete-Backup-and-Recovery-HOWTO/Linux-Complete-Backup-and-Recovery-HOWTO.chunky.html.tar.bz2>]
- compression bzip2 « douce » (une énorme page — pas de petits fichiers. Recherches plus faciles). HTML [<http://www.charlescurley.com/Linux-Complete-Backup-and-Recovery-HOWTO/Linux-Complete-Backup-and-Recovery-HOWTO.smooth.html.tar.bz2>]
- compression bzip2 postscript (format lettre US) [<http://www.charlescurley.com/Linux-Complete-Backup-and-Recovery-HOWTO/Linux-Complete-Backup-and-Recovery-HOWTO.ps.bz2>]
- compression bzip2 PDF (format lettre US) [<http://www.charlescurley.com/Linux-Complete-Backup-and-Recovery-HOWTO/Linux-Complete-Backup-and-Recovery-HOWTO.pdf.bz2>]
- compression bzip2 texte brut ASCII [<http://www.charlescurley.com/Linux-Complete-Backup-and-Recovery-HOWTO/Linux-Complete-Backup-and-Recovery-HOWTO.txt.bz2>]
- Utilisez les sources [<http://www.charlescurley.com/Linux-Complete-Backup-and-Recovery-HOWTO/Linux-Complete-Backup-and-Recovery-HOWTO.tar.bz2>], Luke.

1.4. Historique

Ce document provient de deux articles publiés à l'origine dans *le Linux Journal* [<http://www.linuxjournal.com/>]. Je tiens à remercier *le Linux Journal* pour avoir modifié les droits sur ces articles, rendant la rédaction de ce guide pratique possible.

Je remercie particulièrement Joy Y Goodreau pour son excellente mise en forme du guide pratique, et David Palomares pour avoir corrigé l'orthographe du prénom de Salvador Dali.

D'autres remerciements vont à Pasi Oja-Nisula [<mailto:pon@iki.fi>] pour avoir corrigé un bogue et fourni des informations sur Knoppix [<http://www.knoppix-fr.org/>].

1.5. Commentaires et réactions

Les réactions que vous pourrez faire à ce document sont bienvenues. Sans vos corrections, suggestions et autres apports, ce document n'existerait pas. Envoyez-moi, en anglais, vos ajouts, commentaires et critiques à cette adresse : <charlescurley CHEZ charlescurley POINT com>.

N'hésitez pas à faire parvenir tout commentaire relatif à la version française de ce document à <commentaires CHEZ traduc POINT org> en précisant son titre, sa date et sa version.

1.6. Traductions

Tout le monde ne parle pas anglais. Les volontaires sont bienvenus.

Traduction en français [<http://www.traduc.org/docs/howto/lecture/>] par Denis Berhaut <denis POINT berhaut CHEZ free POINT fr>

Relecture par Guillaume Lelarge <gleu CHEZ wanadoo POINT fr>

2. Tour d'horizon

Mettre en œuvre le processus présenté ci-dessous n'est pas facile et peut être dangereux pour vos données. Entraînez-vous avant d'en avoir besoin ! Faites comme moi et *utilisez un ordinateur sacrifié d'avance* !

Dans ce guide pratique, l'ordinateur cible est un Pentium. À l'origine, la version de Red Hat [<http://www.fr.redhat.com>] 7.1 Linux serveur ou poste de travail était installée sur un disque dur IDE. Depuis, il a été mis à jour vers la Red Hat 8.0 et Fedora Core 1 [<http://fedora.redhat.com/>]. L'ordinateur cible ne contient pas beaucoup de données dans la mesure où c'est une « machine sacrifiée » consacrée aux tests. En fait, je ne voulais pas tester ce processus avec un ordinateur de production et des données de production. Aussi, j'ai effectué une installation avant d'engager les tests pour être en mesure de réinstaller si j'avais besoin de retourner à une configuration connue.



N.B.

Les exemples de commandes montreront, en général, ce que j'ai dû effectuer pour récupérer le système cible. Vous utiliserez sans doute des commandes similaires, mais avec des paramètres différents. C'est à vous de vous assurer que vous dupliquez votre configuration, et non pas la configuration de l'ordinateur de test.

La procédure initiale a été mise au point dans le livre de W. Curtis Preston, *Unix Backup & Recovery* [<http://www.oreilly.com/catalog/unixbr/>], O'Reilly & Associates, 1999, que j'ai approuvé dans le *Linux Journal* [<http://www2.linuxjournal.com/lj-issues/issue78/3839.html>]. Cependant, le livre est un peu léger en ce qui concerne les questions spécifiques, concrètes. Par exemple, quels fichiers faut-il sauvegarder ? Quelles méta données sont à conserver, et comment ?

Avant de démarrer le processus publié dans ce guide pratique, il vous faudra sauvegarder votre système à l'aide d'un outil de sauvegarde classique comme Amanda, BRU™, tar, Arkeia® ou cpio. La question suivante sera de déterminer comment exécuter l'outil qui restaurera vos données à partir d'un matériel hors d'usage.

Les utilisateurs du Red Hat Package Manager (RPM) des distributions Linux devront aussi sauver les méta données RPM en tant que parties intégrantes de leurs sauvegardes normales. Une instruction du type :

```
bash# rpm -Va > /etc/rpmVa.txt
```

dans votre script de sauvegarde vous donnera une base de comparaison du résultat à obtenir après une restauration intégrale de système.

Pour arriver à ce point, il vous faut :

- Votre matériel remonté et de nouveau en état de marche, les composants ayant été remplacés. Le BIOS devrait être configuré correctement, y compris l'heure et la date, ainsi que les paramètres du disque dur. À ce stade, il n'y a pas de raison d'utiliser un disque dur différent.
- Un lecteur Iomega® [<http://www.iomega.com/>] ZIP® sur port parallèle [http://www.iomega.com/zip/products/par100_250.html] ou équivalent. Vous aurez besoin d'au moins 30 Mo d'espace libre.
- Votre média de sauvegarde.
- Un système Linux minimal pour vous permettre de lancer le logiciel de restauration.

Pour en arriver là, au moins deux étapes de sauvegarde sont nécessaires, peut-être trois. La nature exacte de ce que vous sauvegardez et à quelle étape vous sauvegardez est déterminée par votre processus de restauration. Par exemple, si vous restaurez un serveur de bandes, il se peut que vous n'ayez pas besoin du réseau pendant le processus de restauration. Donc, sauvegardez le réseau uniquement lors de vos sauvegardes habituelles.

De plus, vous restaurerez par étapes. À l'étape une, nous construisons les partitions, les systèmes de fichiers, etc. et restaurons un minimum de fichiers du disque ZIP. L'objectif de la première étape est de pouvoir initialiser un ordinateur disposant d'une connexion réseau, de lecteurs de bandes, d'un logiciel de restauration ou de tout ce qui est nécessaire pour l'étape deux.

La seconde étape si nécessaire, consiste à restaurer le logiciel de sauvegarde et les bases de données associées. Par exemple, supposons que vous utilisiez Arkeia et que vous prépariez un disque zip de restauration intégrale de système pour votre serveur de sauvegarde. Arkeia occupe énormément d'espace pour sa base de données sur les disques durs du serveur. Si vous le désirez, vous pouvez récupérer la base de données à partir des bandes. À la place, pourquoi ne pas archiver et compresser avec tar et gzip l'intégralité du répertoire arkeia (/usr/knox), et l'enregistrer sur un autre ordinateur à l'aide de nfs ou ssh ? La première étape, comme nous l'avons défini plus bas, ne comprend pas X. Vous aurez donc quelques tests à effectuer si vous désirez sauvegarder X avec votre programme de sauvegarde. Pour certains programmes de restauration, X est indispensable.

Bien sûr, si vous utilisez d'autres programmes de sauvegarde, vous aurez peut-être un travail de détective à effectuer. Vous devrez déterminer les répertoires et les fichiers nécessaires à son fonctionnement. Si vous utilisez tar, gzip, cpio, mt ou dd comme outils de sauvegarde et de restauration, ils devront être enregistrés sur votre disque ZIP et restaurés pendant la première étape décrite plus bas.

La dernière étape est une restauration intégrale à partir d'une bande ou d'un autre média. Après avoir terminé la dernière étape, vous devriez pouvoir démarrer un système entièrement restauré et opérationnel.

2.1. Limitations

Ce guide pratique se limite à la création d'une sauvegarde minimum de sorte que, ayant ensuite restauré cette sauvegarde vers un nouveau matériel (« restauration intégrale de système »), vous pourrez ensuite utiliser vos sauvegardes traditionnelles pour restaurer un système totalement opérationnel. Ce guide pratique ne traite pas de vos sauvegardes traditionnelles.

Même dans ce cadre étroit, ce guide pratique n'est pas exhaustif. Vous devrez encore faire des recherches, éditer des scripts et effectuer des tests.

Les scripts présentés restaurent les données des partitions telles qu'elles sont sur le disque dur d'origine. Ce serait formidable si vous pouviez restaurer sur un ordinateur identique ou au moins un disque dur identique, mais c'est rarement le cas. Pour l'instant, il y a deux remèdes (qui prendront plus de sens après que vous ayez lu le reste du guide pratique) :

- Éditez le fichier d'entrée de la table des partitions. Je l'ai fait quelquefois. Vous pouvez aussi re-

courir à ce moyen pour ajouter de nouvelles partitions ou en supprimer (mais éditez les scripts qui utilisent aussi le fichier d'entrées de la table des partitions).

- Créez à la main une nouvelle table des partitions et partez de ce point. c'est une des raisons qui fait que `restore.metadata` n'appelle pas le script de reconstruction du disque dur. Utilisez le script de reconstruction.

Les scripts présentés ici prennent en charge uniquement ext2fs, FAT12, FAT16 et FAT32. Vous aurez besoin d'autres outils pour sauvegarder et restaurer des systèmes de fichiers que nous n'avons pas couverts, à moins que des volontaires passionnés ne codent les scripts pour le faire. Partition Image [<http://www.partimage.org/>] se présente comme un candidat utile.

3. Préparation



AVERTISSEMENT

Effectuez vos sauvegardes traditionnelles de façon régulière. Ce guide pratique est sans intérêt si vous ne le faites pas.

Fabriquez vous-même un disque de secours. J'utilise tomsrtbt [<http://www.toms.net/rb>]. Il est bien documenté et tient sur une disquette qui comprend de nombreux outils utiles. Sa liste de discussion est active et les quelques questions que j'ai posées ont trouvé rapidement une réponse précise. J'apprécie cela dans un produit dont ma boutique peut dépendre un jour.

Ensuite, imaginez comment sauvegarder le système d'exploitation dont vous aurez besoin pour restaurer vos sauvegardes traditionnelles. J'ai suivi les conseils de Preston et j'ai utilisé un lecteur ZIP sur port parallèle. Les lecteurs contiennent à peu près 90 Mo utiles. J'ai besoin d'environ 85 Mo pour sauvegarder mon ordinateur, ce qui fait qu'un lecteur de 100 Mo peut être votre salut.

3.1. Installation du lecteur ZIP

L'installation du lecteur ZIP est décrite dans le guide pratique des disques ZIP [<http://www.tldp.org/HOWTO/mini/ZIP-Drive.html>], qui est disponible auprès du Linux Documentation Project [<http://www.tldp.org/>] et, pour sa traduction française, à l'adresse <http://www.traduc.org/docs/HOWTO/vf/ZIP-Drive.html> [<http://www.traduc.org/docs/howto/lecture/ZIP-Drive.html>].

4. Création de la sauvegarde de l'étape 1

Votre sauvegarde de production étant effectuée, il est indispensable de conserver les informations de partitions pour pouvoir les reconstruire.

Le script `make.fdisk` recherche les informations des partitions du disque dur et les enregistre dans trois fichiers. Le premier est un script exécutable, nommé `make.dev.x` (où « x » est le nom du fichier de périphérique, c'est-à-dire `hda`). Le second est `mount.dev.x`, qui crée les points de montage et qui y monte les partitions nouvellement créées. Le dernier, `dev.x`, est constitué des commandes nécessaires à **fdisk** pour construire les partitions. Vous précisez quel disque dur vous désirez, pour construire les scripts associés (et donc les noms de fichiers) en nommant le fichier de périphérique associé comme argument de `make.fdisk`. Par exemple, sur un système IDE classique,

```
bash# make.fdisk /dev/hda
```

produit les scripts `make.dev.hda`, `mount.dev.hda` ainsi que le fichier en entrée de **fdisk**, `dev.hda`.

De plus, si `make .fdisk` rencontre une partition FAT, il conserve le secteur d'amorçage de la partition dans un fichier nommé `dev.xy`, où `x` est le nom de périphérique du disque (c'est-à-dire `sd`, `hda`) et `y` est le numéro de la partition. Le secteur d'amorçage est le premier secteur (512 octets) de la partition. Ce secteur est restauré en même temps que les partitions sont reconstruites, dans le script `make .dev.hda`.

Heureusement, le prix des disques durs s'effondre presque aussi vite que la confiance du public envers les politiciens après une élection. Donc, c'est une bonne chose que les fichiers produits soient au format texte et puissent être édités manuellement. C'est la façon la plus difficile mais la plus flexible pour effectuer une reconstruction sur un plus grand disque de remplacement. (Voir la liste du carnet de route.)

Les autres méta données sont conservées dans le script `save.metadata`. Le script enregistre les informations de partition dans le fichier `fdisk.hda` à la racine du disque ZIP. C'est une bonne idée d'imprimer ce fichier ainsi que votre `/etc/fstab` pour avoir une copie durable au cas où vous devriez restaurer manuellement les données de la partition. Vous pouvez enregistrer une arborescence en alternant l'utilisation de deux consoles virtuelles, par le lancement de **fdisk** dans l'une et en affichant par la commande `cat /etc/fstab` ou `/fdisk.hda` au besoin. Cependant, pratiquer ainsi peut mener à l'erreur.

Vous voudrez aussi conserver les fichiers se rapportant à votre méthode de restauration. Par exemple, si vous utilisez `nfs` pour sauvegarder vos données, vous devrez conserver `hosts.allow`, `hosts.deny`, `exports`, etc. De même, si vous utilisez un processus de restauration adossé au réseau comme `Amanda` ou `Quick Restore`, il vous faudra conserver les fichiers de configuration réseau tels que `HOSTNAME`, `hosts`, etc., ainsi que l'arborescence logicielle qui s'y rapporte.

La façon la plus simple de régler ces questions ainsi que d'autres similaires est de conserver l'intégralité du répertoire etc.

Il est impossible à un lecteur ZIP de 100 Mo de contenir à lui tout seul l'installation d'une distribution moderne d'un serveur Linux. Nous ne pouvons pas conserver l'ensemble du zinzin. Il nous faut être bien plus sélectifs. De quels fichiers avons-nous besoin ?

- Le répertoire `/boot`.
- Le répertoire `/etc` et ses sous-répertoires.
- Les répertoires nécessaires au moment du démarrage.
- Les fichiers de périphériques dans `/dev`.

Pour déterminer les répertoires nécessaires au démarrage, il nous faut regarder dans le fichier d'initialisation du démarrage `/etc/rc.sysinit`. Il détermine son propre chemin ainsi :

```
PATH=/bin:/sbin:/usr/bin:/usr/sbin
export PATH
```

Les essais et les erreurs ont indiqué que nous avons besoin d'autres répertoires, tels que `/dev`. Sous Linux, vous ne pouvez pas faire grand-chose sans les fichiers de périphériques.

Lorsque vous lirez le script `save.metadata`, vous remarquerez que nous ne sauvegardons pas nécessairement les fichiers qui sont appelés par leur chemin absolu.

Il nous faudra peut-être procéder à plusieurs itérations de sauvegarde, tester la restauration intégrale du système, réinstaller d'un CDROM et essayer encore, avant d'obtenir un script de sauvegarde fonctionnel. Pour la préparation de ce guide pratique, j'ai fait cinq itérations avant de réussir une restauration. C'est la raison pour laquelle il est nécessaire d'utiliser des scripts dès que possible. Testez soigneusement !

L'une des choses à faire avec un système basé sur RPM est d'utiliser le programme **rpm** pour déter-

miner l'emplacement des fichiers. Par exemple, pour obtenir la liste complète des fichiers utilisés par le paquet openssh, saisissez :

```
bash# rpm -ql openssh
```

Certains éléments ne sont pas nécessaires, comme les pages man. Vous pouvez les inspecter un par un et décider de les sauvegarder ou non.



AVERTISSEMENT

La seconde étape de la restauration est menée sans que les fichiers précédemment restaurés soient écrasés. Cela signifie que les fichiers restaurés durant la première étape sont ceux qui seront utilisés après la fin de la restauration. Donc, mettez à jour vos sauvegardes intégrales de système à chaque fois que vous mettez à jour des fichiers de ces répertoires.



AVERTISSEMENT

La version de **tar** fournie avec tomsrtbt [<http://www.toms.net/rb>] ne préserve pas le propriétaire lors de la restauration. Ceci peut poser problème à des applications comme Amanda. Amanda, outil de sauvegarde et de restauration, a plusieurs répertoires dont le propriétaire est son utilisateur éponyme. La solution est :

- Notez quels sont les répertoires et les fichiers dont root n'est pas propriétaire.
- Notez l'identifiant de leurs propriétaires.
- Faites en sorte que le processus de restauration comprenne le rétablissement du propriétaire. Par exemple :

```
bash# chown -R amanda:disk /var/lib/amanda
```

vous pouvez aussi ajouter cette ligne à vos scripts de la deuxième étape de restauration, comme dans `restore.test`.

4.1. Variations sur le thème

4.1.1. Pas de lecteur ZIP

Ce processus de sauvegarde vous oblige à disposer du lecteur de disque ZIP à chaque sauvegarde. Vous pouvez créer le contenu du disque ZIP dans un répertoire, puis le sauvegarder via le réseau. Ensuite, il vous faudra simplement construire un disque ZIP sur le serveur de sauvegarde (avec la commande **cp -rp**) lorsque vous aurez besoin de restaurer.

Le processus de sauvegarde sera plus rapide, mais vérifiez tout de même que le répertoire créé tiendra sur votre disque ZIP (avec la commande **du -hs \$cible.zip**) ! Vous devrez éditer la définition de la variable `zip` dans le fichier `save.metadata`.

Mon portable ne supportant pas la présence simultanée d'une carte réseau et d'un lecteur ZIP, c'est le processus que j'utilise pour le sauvegarder. Je conserve une image de sauvegarde en même temps que la courante, ce qui fait que je dispose d'un système de secours au cas où l'ordinateur planterait pendant une sauvegarde.

Vous pouvez aussi sauvegarder sur plusieurs disques ZIP des données qui en valent la peine, et les transférer sur le système pendant la restauration.

4.1.2. Cédérom

Ceci est similaire à l'option « pas de lecteur ZIP » ci-dessus. Enregistrez vos sauvegardes dans un répertoire de votre disque dur, comme mentionné. Puis, utilisez la commande **mkisofs** pour créer une image ISO 9660 de ce répertoire et gravez-la.

De nos jours, beaucoup d'ordinateurs sont livrés avec un lecteur de CDROM mais pas de disquette. De plus, les lecteurs de disquette tombent en panne. C'est pourquoi, graver votre CDROM avec une image qui permette d'amorcer le système est une bonne idée. La mauvaise nouvelle, c'est que le format « El Torito » supporte les disquettes de 1,2 Mo, 1,44 Mo et 2,88 Mo, et que `tomsrtdt` [<http://www.toms.net/rb>] utilise une disquette de 1,7 Mo. La bonne nouvelle, c'est que vous pouvez obtenir une version en 2,88 Mo, `tomsrtdt-2.0.103.ElTorito.288.img`, des mêmes miroirs que là où vous vous procurez l'image pour disquette. Placez une *copie*¹ dans le répertoire racine des fichiers de sauvegarde. Puis, utilisez l'option `-b` de la commande **mkisofs** pour spécifier que `tomsrtdt-2.0.103.ElTorito.288.img` est le fichier de l'image d'amorçage.

Le seul point noir de ce processus est que beaucoup de vieux BIOS ne supportent pas les images de disquette de 2,88 Mo sur CDROM. Beaucoup de ceux-ci s'amorceront avec une disquette `tomsrtdt` [<http://www.toms.net/rb>].

On peut aussi utiliser Syslinux [<http://syslinux.zytor.com/>]. Il n'est pas dépendant de l'image d'une disquette et vous pouvez construire votre propre CD avec un certain nombre d'outils tels que `tomsrtdt` [<http://www.toms.net/rb>].

Il vous faudra peut-être ajuster les options du BIOS pour pouvoir amorcer sur le lecteur de CDROM.

Testez vos CDROM sur le lecteur que vous utiliserez pour la restauration. S'il se trouve que vous devez modifier les scripts, vous pouvez les copier dans `/tmp`, un disque en mémoire vive sous `tomsrtdt` [<http://www.toms.net/rb>], et les y éditer. Les scripts s'exécuteront à cet emplacement. Un disque en MEV étant volatil par nature, enregistrez vos changements avant de réamorcer !

4.1.3. Disques ZIP multiples

En séparant les deux premiers scripts de la première étape, `restore.metadata` et `save.metadata`, vous pouvez répartir les méta données de la première étape sur plusieurs disques ZIP.

4.1.4. Exclusions de l'enregistrement de la première étape

Lors de la première étape, il est parfois nécessaire de compresser quelques méga-octets de données, particulièrement lorsque vous atteignez les limites de votre disque ZIP. La fonction **crunch** du script `save.metadata` accepte de nombreux paramètres pour alimenter la commande **tar**. Elle accepte aussi le paramètre **--exclude**. Ainsi, par exemple, vous pouvez exclure les répertoires `samba` et `X11` situés dans `/etc` de cette façon :

```
crunch etc etc --exclude etc/samba --exclude etc/X11
```

Pourquoi ces deux-là ? Parce qu'ils consomment beaucoup d'espace disque et que nous n'en avons pas besoin pour l'amorçage.

Si vous possédez un nombre important de noyaux, vous pouvez éliminer les modules de tous les noyaux sur lesquels vous n'amorcerez pas. vérifiez votre `lilo.conf` ou `grub.conf` pour connaître le noyau que vous utiliserez, puis vérifiez `/lib/modules` pour connaître les répertoires des modules que vous pouvez exclure.

Où trouver d'autres bons candidats à l'exclusion ? Listez les répertoires cible avec la commande **ls -alSr** pour les fichiers individuels, et avec la commande **du | sort -n** pour les répertoires.

¹ Je fais ressortir copie parce que **mkisofs** détruira le fichier dans le répertoire où il fabrique l'image ISO.

Une autre façon (probablement plus nette), d'exclure les répertoires est d'écrire la liste complète des répertoires dans un fichier puis d'y faire référence via l'option `tar --exclude-from=FILENAME`.

4.1.5. Initrd

Si votre système utilise un disque MEV pour initialiser, ou `initrd` pour amorcer, assurez-vous que `restore.metadata` crée le répertoire `/initrd`. La façon la plus simple de le faire est de s'assurer qu'ils figurent dans la liste des répertoires utilisés dans la boucle de création des répertoires (vers la fin).

Votre système utilise probablement `initrd` s'il démarre sur un disque SCSI ou si sa racine est sur une partition `ext3fs`. Vérifiez dans `/etc/lilo.conf` s'il appelle un tel script.

5. Première étape de restauration

5.1. Démarrer tomsrtd

La première chose à faire avant de démarrer le processus de restauration est de vérifier que l'heure du système est correctement réglée. Pour ce faire, configurez le BIOS. La précision du réglage de l'heure dépendra de vos applications. Pour une restauration, une précision de quelques minutes devrait être suffisante. Cela devrait permettre aux événements dépendant de l'heure de redémarrer à leur point d'arrêt lorsque vous lancerez finalement la restauration du système.

Avant de démarrer `tomsrtd` [<http://www.toms.net/rb>], vérifiez que votre lecteur ZIP est installé sur un port parallèle, soit `/dev/lp0` soit `/dev/lp1`. Le logiciel d'initialisation chargera pour vous le pilote du lecteur zip sur le port parallèle.

L'étape suivante est la configuration du mode vidéo. D'habitude, j'aime afficher à l'écran tout ce qui est possible. À l'apparition de la sélection du mode vidéo, je sélectionne le mode 6, 80 colonnes par 60 lignes. Il se peut que votre matériel ne soit pas capable de supporter de si hautes résolutions, aussi effectuez des tests.

5.2. Restauration

Une fois `tomsrtd` [<http://www.toms.net/rb>] démarré et que vous voyez une console, montez le lecteur ZIP. C'est probablement une bonne idée de le monter en lecture seule :

```
# mount /dev/sda1 /mnt -o ro
```

Vérifiez sa présence :

```
# ls -l /mnt
```

À ce stade, vous pouvez lancer la restauration automatiquement ou manuellement. Utilisez la restauration automatique si vous n'avez pas besoin d'effectuer des changements pendant celle-ci.

Une remarque cependant si vous avez plusieurs disques durs. Si votre installation de Linux monte des partitions à partir de plusieurs disques durs, vous devez monter la partition racine en premier. Ceci pour être certain que les répertoires des points de montage sont créés sur la partition à laquelle ils appartiennent. Le script `first.stage` lancera les scripts de montage des lecteurs dans leur ordre de création. Si vous les avez créés (dans le script `save.metadata`) dans un ordre découlant de l'arbre racine, le processus de montage devrait se dérouler correctement.

Si vous avez plusieurs disques durs, et qu'ils se montent de façon croisée, faites-le à votre main. Vous pouvez combiner et éditer les scripts pour les monter dans le bon ordre, où les monter manuellement.

5.2.1. Automatisée

Le processus automatique lance tous les scripts manuels dans le bon ordre. Il ne permet pas les interventions manuelles, telles que la création de systèmes de fichiers que ce guide pratique ne prend pas en charge. Pour lancer la première étape de restauration automatiquement, saisissez :

```
# /mnt/root.bin/first.stage
```

Si vous désirez rechercher les blocs endommagés, ajoutez l'option « -c ».

5.2.2. Manuelle

Pour lancer le processus manuel, positionnez-vous dans le répertoire où se trouvent les scripts, dans le lecteur ZIP.

```
# cd /mnt/root.bin
```

Lancez maintenant le(s) script(s) qui restaurer(a/ont) les informations de partition et créer(a/ont) les systèmes de fichiers. Vous pouvez les lancer dans n'importe quel ordre, par exemple :

```
# ./make.dev.hda
```

Si vous désirez qu'une recherche de blocs endommagés s'effectue, ajoutez l'option « -c ».

Ce script va :

- Nettoyer les 1024 premiers octets du disque dur, détruire toutes les tables des partitions existantes ainsi que le bloc de démarrage (MBR).
- Recréer les partitions des informations collectées quand vous avez lancé `make.fdisk`.
- Créer correctement des partitions ext2 et ext3 ainsi que des partitions swap. Si vous ajoutez l'option « -c » au script, il vérifiera aussi les blocs défectueux.
- Créer différents types de partitions FAT.



N.B.

Si vous devez restaurer d'autres systèmes d'exploitation, c'est le moment de le faire. Lorsque c'est terminé, redémarrez avec `tomsrtbt` [<http://www.toms.net/rb>] et continuez la restauration de Linux.

À présent, lancez le(s) script(s) qui crée(nt) des points de montage et montez les partitions.

```
# ./mount.dev.hda
```

Une fois que vous avez créé tous vos répertoires et que vous y avez monté les partitions, vous pouvez lancer le script `restore.metadata`. Il restaurera le contenu du lecteur zip sur le disque dur.

Vous devriez voir un répertoire du répertoire racine du lecteur zip, puis une liste des fichiers d'archives tels qu'ils sont restaurés. L'exécution de la commande `tar` sous `tomsrtbt` [<http://www.toms.net/rb>] vous dira que la taille d'un bloc de `tar` est de 20, et c'est très bien ainsi. Vous pouvez l'ignorer. Assurez-vous que `lilo` affiche ses résultats :

```
Added linux *
```

Un affichage suivra, correspondant à la commande « **df -m** ».

5.2.3. Touche finale

Si d'habitude vous démarrez directement sous X, vous aurez peut-être quelques problèmes. Pour les éviter, changez temporairement votre niveau de démarrage. Dans le fichier `/etc/getty/ettab`, trouvez la ligne qui ressemble à :

```
id:5:initdefault:
```

et transformez-la en :

```
id:3:initdefault:
```

Vous pouvez maintenant démarrer en douceur. Retirez la disquette `tomsrtbt` [<http://www.toms.net/rb>] de votre lecteur si vous ne l'avez pas encore fait, et faites le salut à trois doigts à votre ordinateur (N.D.T. : CTRL+MAJ+SUP), ou son équivalent :

```
# reboot
```

L'ordinateur va s'arrêter et redémarrer.

6. Seconde étape de la restauration

Au redémarrage de l'ordinateur, vérifiez dans le BIOS que l'heure est à peu près correcte.

Une fois la vérification terminée, sortez du BIOS et redémarrez sur le disque dur. Laissez simplement la séquence de démarrage normal de l'ordinateur se dérouler. Vous verrez un nombre important de messages d'erreurs, essentiellement de type « Impossible de trouver bla-bla ! Ouahhh ! » Si vous avez bien travaillé jusqu'à maintenant, ces messages d'erreurs seront sans importance. Vous n'avez pas besoin de `linuxconf` ou d'`apache` pour cette opération.



N.B.

Vous pouvez aussi démarrer en mode utilisateur unique (single user) : à l'invite de `lilo`, saisissez : **linux single**, mais il vous faudra configurer manuellement votre réseau et lancer `sshd` ou tous les démons nécessaires au redémarrage de votre système. Chaque système a sa méthode spécifique.

Vous devriez vous identifier sur une console en tant que super utilisateur (root) (désolé, pas de session X, pas d'utilisateurs). À présent, vous devriez pouvoir utiliser le réseau, par exemple pour monter la sauvegarde de votre système via `nfs`.

Si vous avez effectué la sauvegarde en deux étapes que j'ai suggérées pour Arkeia, vous pouvez maintenant restaurer les exécutable et la base de données d'Arkeia. Vous devriez pouvoir lancer

```
/etc/rc.d/init.d/arkeia start
```

et démarrer le serveur. Si vous avez une interface graphique installée sur un autre ordinateur, vous devriez pouvoir vous connecter à Arkeia sur votre serveur de bandes et préparer la restauration.



N.B.

Pendant la restauration, lisez attentivement la documentation de restauration de vos programmes. Par exemple, d'habitude, tar ne restaure pas certaines caractéristiques de fichiers, telles que le bit suid. Les droits sur les fichiers sont fixées par l'utilisateur avec la commande umask. Pour restaurer vos fichiers exactement comme vous les avez sauvegardés, utilisez l'option tar avec l'option p. De la même manière, vérifiez que votre logiciel de restauration restaure les données de façon identique à la sauvegarde.

Pour restaurer l'ordinateur de test, saisissez :

```
bash# restore.all
```

Si vous avez sauvegardé et restauré avec tar, et avez utilisé l'option -k (conserver les anciens fichiers, ne pas écraser), vous constaterez un nombre important de :

```
tar: usr/sbin/rpcinfo: Could not create file: File exists
tar: usr/sbin/zdump: Could not create file: File exists
tar: usr/sbin/zic: Could not create file: File exists
tar: usr/sbin/ab: Could not create file: File exists
```

Ceci est normal, tar refusant d'écraser les fichiers que vous avez restaurés pendant la première étape.

Puis redémarrez. À la fermeture, vous verrez un nombre important de messages d'erreurs, tels que « no such pid ». Ils font partie du processus. Le programme de fermeture utilise les fichiers pid des démons qui tournaient au moment de la sauvegarde pour fermer les démons qui n'avaient pas été lancés au dernier démarrage. Et évidemment, il n'ont pas de PID.

Votre système devrait s'initialiser normalement avec beaucoup moins d'erreurs que précédemment et idéalement sans erreurs. Le test décisif du bon fonctionnement de votre restauration sur un système basé sur des RPM est de vérifier tous les paquets :

```
bash# rpm -Va
```

Si vous constatez des messages d'erreurs de dépendances, vous pouvez lancer la commande **/etc/cron.daily/prelink** pour les enlever.

Certains fichiers, tels que les fichiers de configuration et les journaux, auront naturellement changé ; vous devriez pouvoir mentalement les exclure du rapport. Vous pouvez rediriger la sortie dans un fichier, et le comparer grâce à diff avec celui effectué lors de la sauvegarde (/etc/rpmVa.txt), ce qui accélérera considérablement cette étape. Les utilisateurs d'Emacs devraient se renseigner sur ses aptitudes à comparer des fichiers.

À présent, votre système devrait être lancé et fonctionner. C'est le moment de tester vos applications, particulièrement celles lancées en tant que démons. Plus les applications seront sophistiquées, plus il vous faudra effectuer des tests. S'il y a des utilisateurs distants, interdisez leur l'accès du système, ou passez le en « lecture seule » le temps des tests. Ceci vaut particulièrement pour les bases de données, afin empêcher une corruption ou une perte de données pire que ce qu'il pourrait déjà y avoir.

Si d'habitude vous redémarrez directement sous X et que vous l'avez désactivé plus haut, testez X avant de le réactiver. Réactivez-le en rétablissant cette unique ligne dans le fichier /etc/inittab :

```
id:5:initdefault:
```

Vous pourrez alors aller faire la fête, prendre un peu d'aspirine et aller au dodo.

7. Notes spécifiques aux distributions

Vous trouverez ci-dessous des remarques nées de l'expérience, distribution par distribution. Si vous avez des remarques concernant d'autres distributions, envoyez-les-moi.

7.1. Fedora

Fedora [<http://fedora.redhat.com/>] est issue de Red Hat 9. Effectuez les modifications suivantes :

- Fedora utilise grub, et non lilo. Effectuez les modifications suivantes.
- Il n'est plus nécessaire de sauvegarder les bibliothèques kerberos séparément, comme dans la Red Hat 8.0. Enlevez ou mettez en commentaire cette ligne du fichier `save.metadata` :

```
# RH8. Fedora 1 puts them in /lib # crunch kerberos usr/kerberos/lib/
```

- Les noms des bibliothèques nécessaires à la sauvegarde de SSH ont changé. Aussi, dans le fichier `save.metadata` :

```
# save these so we can use ssh for restore. *crack* for RH 7.0 login
# authentication.
# RH 8.0
# crunch usr.lib usr/lib/*crack* usr/lib/libz* usr/lib/libssl*
usr/lib/libcrypto*
# Fedora 1
crunch usr.lib usr/lib/*crack* usr/lib/libz* usr/lib/libwrap* usr/lib/libk
usr/lib/libgss*
```

- Dans le fichier `restore.metadata`, il y a un plus grand nombre de répertoires à sauvegarder.

```
# If you boot via an initrd, make sure you build a directory here so
# the kernel can mount the initrd at boot. tmp/.font-unix is for the
# xfs font server.
for dir in mnt/save mnt/zip mnt/cdrom mnt/floppy mnt/imports mnt/dosc mnt/
proc initrd tmp/.font-unix var/empty/sshd var/lock/subsys var/log; do
```

Certains modes sont à configurer :

```
chmod a-w $target/proc # Restore /proc's read-only permissions
# Set modes
chmod 0111 $target/var/empty/sshd
chmod 0775 $target/var/lock
# For Fedora. First two for xfs.
chroot $target
chown xfs:xfs /tmp/.font-unix
chmod 1777 $target/tmp/.font-unix
# set the sticky bit.
chmod 1777 $target/tmp
```

7.2. Red Hat 9

Je n'ai jamais travaillé sous Red Hat 9. Les commentaires concernant Fedora devraient s'appliquer également à la Red Hat 9.

7.3. Red Hat 8.0

Il est nécessaire de modifier Red Hat 8.0 uniquement si vous vous connectez après la première étape de restauration via ssh. Si vous effectuez la deuxième étape de restauration avec ssh, effectuez les modifications suivantes :

- Dans le fichier `save.metadata`, ajoutez les bibliothèques kerberos à la première étape de sauvegarde :

```
crunch kerberos usr/kerberos/lib/
```

- Dans le fichier `restore.metadata`, il y a une boucle qui crée plusieurs répertoires. Tout d'abord, ajoutez le nom de ces deux répertoires à la liste : `/var/empty/sshd` et `/var/lock/subsys`. Red Hat 8.0 utilisant par défaut ext3fs, a besoin d'un disque en MEV au démarrage. Aussi assurez-vous que `initrd` est dans la liste. Puis, s'il n'y est pas encore, ajoutez l'argument `-p` à la commande `mkdir`.

Le groupe propriétaire du répertoire `/var/lock/subsys` est le groupe `lock`, aussi changez son propriétaire.

```
chroot $target /bin/chown root:lock /var/lock
```

Pour finir, `usr/lib/libcrypto*` a disparu, aussi vous pouvez le retirer de la ligne qui traite `usr/lib`.

7.4. Red Hat 7.1

À l'origine, j'ai utilisé cette distribution sur mon ordinateur de test. Je n'ai eu aucun problème avec.

7.5. Red Hat 7.0

Il semble que cette version ait besoin de `libcrack` (dans `/usr/lib`) et de ses fichiers pour authentifier les utilisateurs. Aussi, dans le fichier `save.metadata`, ajoutez ce qui suit à la ligne qui sauvegarde `/usr/lib` : `/usr/lib/*crack*` et activez cette ligne.

7.6. Knoppix

Je n'ai pas utilisé Knoppix [<http://www.knoppix.org/>], mais d'autres que moi l'ont fait. Pasi Ojanisula [<mailto:pon@iki.fi>] nous en parle :

Pour moi, le grand avantage de knoppix est que je n'ai pas besoin d'un médium de démarrage propre à chaque machine mais que je peux utiliser les mêmes outils tout le temps. Et le support matériel de Knoppix est vraiment excellent. Je n'ai pas tant d'expérience que ça avec les différentes plates-formes mais toutes les machines que j'ai essayé ont bien fonctionné, les pilotes scsi ont été trouvés, etc...

J'effectue le travail de restauration en copiant les sauvegardes sur une autre machine du réseau. Restaurer implique de démarrer à partir du CDROM de Knoppix, d'aller chercher le fichier `metadata.tar.gz` sur l'autre machine du réseau. Puis de d'appeler `make.dev`, `mount.dev`, d'aller chercher les autres fichiers `tar.gz`, `grub` et de redémarrer. Il y a bien quelques saisies mais, grâce à vos scripts, c'est très simple. À moins que vous ne passiez de ide à scsi ou quelque chose de ce genre, mais même ainsi, ce n'est pas si difficile dans la mesure où Linux peut facilement être restauré sur différents matériels.

Jetez aussi un coup d'œil à « Récupération de système avec Knoppix (System recovery with Knoppix, N.D.T. : pas de version française) »
[<http://www-106.ibm.com/developerworks/linux/library/l-knopp.html?ca=dgr-lnxw04Knoppix>] .

8. Notes concernant certaines application

Vous trouverez ci-dessous quelques remarques sur la sauvegarde de certaines applications.

8.1. GRUB

Le chargeur d'amorçage par défaut de Fedora est le Grand Chargeur D'amorçage Unifié (Grand Unified Bootloader (GRUB)) [<http://www.gnu.org/software/grub/>] . Il doit être lancé à la fin de la première étape, sans quoi vous ne pourrez pas démarrer ensuite. Pour l'inclure dans la première étape de restauration, effectuez les modifications suivantes :

- Éditez la pénultième stance de `restore.metadata` :

```
# Now install the boot sector.
# chroot $target /sbin/lilo -C /etc/lilo.conf
chroot $target /sbin/grub-install /dev/hda
```

- Ajoutez la stance suivante à `save.metadata` :

```
# Grub requires these at installation time.
crunch usr.share.grub usr/share/grub
```

8.2. Tripwire

Si vous utilisez Tripwire ou une autre application qui utilise une base de données de métadonnées en fichiers, reconstruisez cette base de données immédiatement après la restauration.

8.3. Squid

Squid est un serveur HTTP de cache et de proximité. Il stocke donc une grande quantité de données temporaires sur le disque dur. Il n'y a aucune raison de les sauvegarder. Insérez « `--exclude /var/spool/squid` » dans la commande `tar` appropriée du script de sauvegarde de la deuxième étape. Puis, laissez squid reconstruire sa structure de répertoires lui-même. Ajoutez une commande pour que squid s'initialise tout seul à la fin du script de restauration de la deuxième étape. Voilà comment j'ai fait avec `ssh` dans le fichier `restore.test` :

```
ssh $target "mkdir /var/spool/squid ; chown squid:squid /var/spool/squid;\
/usr/sbin/squid -z ; touch /var/spool/squid/.OPB_NOBACKUP"
```

La dernière commande crée un fichier de longueur 0 appelé `.OPB_NOBACKUP`. Il est à destination d'Arkeia, et lui dit de ne rien sauvegarder en dessous de ce répertoire.

8.4. Arkeia

Ces notes sont basées sur des tests effectués avec Arkeia 4.2.

Arkeia [<http://www.arkeia.com/>] est un programme de sauvegarde et de restauration qui tourne sur une grande variété de plate-formes. Vous pouvez utiliser Arkeia dans votre programme de restauration intégrale de système, mais je ferais deux observations.

La première est probablement celle qui pose le plus de problèmes, dans la mesure où il n'y a pas de solution plus élégante que de sélectionner à la main et dans le navigateur les répertoires à restaurer. Ceci est dû au fait qu'apparemment, Arkeia ne dispose pas de mécanisme pour ne pas restaurer des fichiers déjà présents sur le disque, rien qui soit analogue à **tar** et à son option **-p**. Si vous paramétrez simplement une restauration complète, la restauration plantera car Arkeia écrasera une bibliothèque utilisée au moment de la restauration, à savoir `lib/libc-2.1.1.so`. Une sélection à la main des répertoires à restaurer est moins risquée, aussi je la recommande.

La seconde observation est qu'il vous faudra sauvegarder le dictionnaire des données et/ou les programmes d'Arkeia. Pour le faire, modifiez le script `save.metadata` en ajoutant Arkeia à la liste des répertoires à sauvegarder :

```
# arkeia specific: tar cf - usr/knox | gzip -c > $zip/arkeia.tar.gz
```

C'est ainsi que vous *devez* sauvegarder le dictionnaire des données car Arkeia ne le fait pas. C'est un des reproches que je fais à Arkeia, mais je le résous sur mon propre PC en sauvegardant le dictionnaire des données sur bande avec The TOLIS Group's BRU [<http://www.estinc.com/>].

Le script `restore.metadata` restaurera le dictionnaire des données automatiquement.

8.5. Amanda

Amanda [<http://www.amanda.org/>], archiveur automatique avancé sur disque en réseau du Maryland (The Advanced Maryland Automatic Network Disk Archiver), marche plutôt bien avec cet ensemble de scripts. Utilisez le processus normal de sauvegarde d'Amanda et créez votre première étape comme d'habitude. Amanda stocke les données sur bande au format `tar` ou `cpio`, donc vous pouvez restaurer des fichiers individuels jusqu'à des images complètes de sauvegarde. Ce qu'il y a de bien dans une restauration d'image complète est que vous pouvez utiliser les variantes de ce guide pratique pour restaurer à partir de l'image ou à partir de la bande directement. J'ai pu réparer ma machine de tests avec les instructions de W. Curtis Preston *et de son Unix Backup & Recovery* [<http://www.oreilly.com/catalog/unixbr/>]. Pour plus d'informations, consultez le lien Ressources. Le chapitre du livre qui concerne Amanda est en ligne [<http://www.backupcentral.com/amanda.html>].

J'ai effectué deux changements au script `restore.test`. Premièrement, je l'ai modifié pour qu'il accepte un nom de fichier comme argument. Puis, comme la commande d'Amanda **amrestore** décompresse les données pendant qu'il les restaure, je l'ai réécrit pour qu'il envoie, via la commande `cat`, le fichier dans le canal (pipe) plutôt que de le décompresser.

La ligne en question ressemble à :

```
cat $fichier | ssh $cible "umask 000 ; cd / ; tar -xpkf - "
```

où **\$fichier** est l'argument du script, l'image récupérée de la bande par la commande **amrestore**.

Comme les arguments de la ligne de commande de **tar** interdisent l'écrasement de fichiers, restaurez les images dans l'ordre *inverse* de leur ordre de création. Restaurez les plus récents en premier.

Avec Amanda, il n'est pas nécessaire de déterminer manuellement les propriétaires si vous sauvegardez le répertoire des données d'Amanda avec le fichier `save.metadata`. Ce devrait être une instruction du type :

```
bash# chown -R amanda:disk /var/lib/amanda
```

Vous pouvez aussi ajouter cette ligne à vos scripts de la deuxième étape de restauration, comme dans `restore.test`.

9. Quelques conseils pour une récupération après un désastre

Vous devriez placer les disques zip de chaque ordinateur et les sorties papier que vous avez effectuées dans un endroit sûr de votre boutique. Vous devriez conserver des copies de ces sauvegardes dans un site de stockage extérieur. L'intérêt principal d'un site de stockage extérieur est de rendre possible une récupération après un désastre ; restaurer chaque hôte sur un matériel de remplacement fait d'ailleurs partie d'une récupération après un désastre.

Vous devriez conserver plusieurs disquettes de tomsrtbt [<http://www.toms.net/rb>] ainsi que si possible plusieurs disques zip à l'extérieur. De plus, installez des copies de la distribution de tomsrtbt [<http://www.toms.net/rb>] sur plusieurs de vos ordinateurs, de façon à ce qu'ils puissent se restaurer mutuellement.

Vous devriez probablement conserver des copies de ce guide pratique, agrémentées d'annotations spécifiques à votre site, avec vos sauvegardes et sur votre site extérieur de conservation des sauvegardes.

10. Et maintenant ?

Ce guide pratique est le résultat d'expérimentations sur un seul ordinateur. Vous y trouverez sans aucun doute des répertoires ou des fichiers nécessaires à la première étape de votre sauvegarde. Je n'ai pas essayé de sauvegarder ni de restaurer X à la première étape, pas plus que je ne me suis occupé d'autres processeurs qu'Intel.

J'apprécierais que vous me fassiez des retours sur vos tests et les améliorations que vous avez apportées à ces scripts pour vos ordinateurs. J'encourage aussi les éditeurs de logiciels de sauvegarde à rédiger une documentation permettant d'effectuer une sauvegarde minimum de leurs produits.

J'aimerais que grâce à ces efforts la communauté Linux dorme un tout petit peu mieux chaque nuit.

10.1. Liste de travail

Les volontaires sont bienvenus. Avant de commencer un thème, consultez-moi pour savoir si quelqu'un d'autre y travaille déjà.

- Un éditeur de partitions pour ajuster les limites des partitions dans le fichier `dev.hdx`. Cela permettra aux utilisateurs d'ajuster les partitions sur un disque dur différent, ou sur le même, mais avec une géométrie différente, ou encore d'ajuster la taille des partitions sur le même disque dur. Une interface graphique serait certainement une bonne chose. D'un autre côté, le programme de la FSF `parted` [<http://www.gnu.org/software/parted>] semble pouvoir répondre partiellement au besoin. Il redimensionne les partitions existantes, mais avec certaines restrictions.
- `make.fdisk` est le seul actuellement à reconnaître certaines partitions FAT, pas toutes. Ajouter du code à `make.fdisk` pour qu'il reconnaisse les autres et écrire les instructions appropriées pour qu'il les reconstruise dans les fichiers en sortie.
- Pour les partitions FAT12 et FAT16, nous ne formatons pas, n'écrivons pas de zéro dans la partition de telle sorte que MS-DOS 6.x les reconnaisse correctement. Reportez-vous aux notes sur **fdisk** pour une explication du problème.
- Faire un script pour mettre les systèmes de fichiers ext2/3 sur disque ZIP.
- Traduire ce guide dans d'autres langues.
- Déterminer dans quelle mesure loadlin ou des programmes de ce type affectent le processus.
- J'ai mentionné le gestionnaire de paquets de (Red Hat Red Hat Package Manager : rpm) de temps en temps. Quelles sont les commandes deb équivalentes ?

11. Les scripts

Reportez-vous aux notes situées au début de chaque script pour voir le résumé de ce qu'il fait.

11.1. Première étape

11.1.1. `make.fdisk`

Ce script, lancé pendant la sauvegarde, crée des scripts semblables à `make.dev.hda` et `mount.dev.x`, plus bas, pour que vous les lanciez à la restauration. Il produit aussi des fichiers de données semblables à `dev.hda`, plus bas. Le nom du script et du fichier de données qui est produit dépend du périphérique donné en paramètre à ce script. Ce script, lancé à la restauration, crée les partitions sur le disque dur. `make.fdisk` est appelé par `save.metadata`, plus bas.

```
#!/usr/bin/perl

# A perl script to create a script and input file for fdisk to
# re-create the partitions on the hard disk, and format the Linux and
# Linux swap partitions. The first parameter is the fully qualified
# path of the device of the hard disk, e.g. /dev/hda. The two
# resulting files are the script make.dev.x and the data file dev.x
# (where x is the hard drive described, e.g. hda, sdc). make.dev.x is
# run at restore time to rebuild hard drive x, prior to running
# restore.metadata. dev.x is the input file for fdisk.

# Time-stamp: <2004-04-10 13:51:37 root make.fdisk>

# Copyright 2001 through the last date of modification Charles Curley
# except for the subroutine cut2fmt.

# cut2fmt Copyright (c) 1998 Tom Christiansen, Nathan Torkington and
# O'Reilly & Associates, Inc. Permission is granted to use this code
# freely EXCEPT for book publication. You may use this code for book
# publication only with the explicit permission of O'Reilly &
# Associates, Inc.

# This program is free software; you can redistribute it and/or modify it
# under the terms of the GNU General Public License as published by the
# Free Software Foundation; either version 2 of the License, or (at your
# option) any later version.

# This program is distributed in the hope that it will be useful, but
# WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
# General Public License for more details.

# You should have received a copy of the GNU General Public License along
# with this program; if not, write to the Free Software Foundation, Inc.,
# 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

# In addition, as a special exception, Tom Christiansen, Nathan
# Torkington and O'Reilly & Associates, Inc. give permission to use
# the code of this program with the subroutine cut2fmt (or with
# modified versions of the subroutine cut2fmt that use the same
# license as the subroutine cut2fmt), and distribute linked
# combinations including the two. You must obey the GNU General
# Public License in all respects for all of the code used other than
# the subroutine cut2fmt. If you modify this file, you may extend
# this exception to your version of the file, but you are not
# obligated to do so. If you do not wish to do so, delete this
# exception statement and the subroutine cut2fmt from your version.

# You can also contact the Free Software Foundation at
# http://www.fsf.org/
```

```
# Changes:

# 2004 04 10: fdisk v > 2.11 has wider columns. Added code to select
# the appropriate cut string based on fdisk's version.

# 2004 04 09: Added support for Mandrake's idea of devfs. On Mandrake,
# everything is mounted with devfs. So the mount devices are buried
# deep in places like /dev/ide/host0/bus0/target0/lun0/part1 instead
# of places like /dev/hda1, where $DEITY intended they should be. We
# have to reverse from the long devfs device to the shorter old style
# that tomsrtbt uses. The alternative is to keep track in an array of
# which devfs device belongs to which short device.

# 2003 12 29: Changed the regex for detecting whether a file system is
# read-write in the code that builds the mount file(s). The old test
# does not work if mount returns multiple parameters in the 5th field,
# e.g. (rw,errors=remount-ro) on some debian systems. This regex
# assumes that the rw parameter is always listed first, which may not
# always be the case. If it fails, take out the '\('. Thanks to Pasi
# Oja-Nisula <pon at iki dot fi> for pointing this out.

# 2003 01 09: Added support for FAT32. We now create two scripts for
# each hard drive, make.dev.[as]dx and mount.dev.[as]dx. These create
# and make file systems on each partition, and make mount points and
# mount them.

# 2002 12 25: added support to handle W95 extended (LBA) (f) and W95
# FAT 32 partitions. I have tested this for primary but not logical
# partitions.

# 2002 09 08: Added minimal support for ext3fs. We now detect mounted
# ext3fs partitions & rebuild but with no options. The detection
# depends on the command line "dumpe2fs <device> 2>/dev/null | grep -i
# journal" producing no output for an ext2fs, and output (we don't
# care what) for an ext3fs.

# This could stand extension to support non-default ext3 options such
# as the type of journaling. Volunteers?

# 2002 07 25: Bad block checking is now a command line option (-c) at
# the time the product script is run.

# 2002 07 03: Corrected the mechanism for specifying the default
# drive.

# 2001 11 25: Changed the way mke2fs gets its bad block
# list. badblocks does not guess at the block size, so you have to get
# it (from dumpe2fs) and feed it to badblocks. It is simpler to just
# have mke2fs call badblocks, but you do lose the ability to have a
# writing test easily. -- C^2

# 2001 11 25: Changed the regex that extracts partition labels from
# the mount command. This change does not affect the results at all,
# it just makes it possible to use Emacs' perl mode to indent
# correctly. I just escaped the left bracket in the regex. -- C^2

# Discussion:

# fdisk will spit out a file of the form below if you run it as "fdisk
# -l".

# root@tester ~/bin $ fdisk -l /dev/hda

# Disk /dev/hda: 64 heads, 63 sectors, 1023 cylinders
# Units = cylinders of 4032 * 512 bytes

#   Device Boot      Start         End      Blocks   Id  System
# /dev/hda1          1           9       18112+   83   Linux
# /dev/hda2          10        1023      2044224    5   Extended
```

```
# /dev/hda5          10          368      723712+   83   Linux
# /dev/hda6          369          727      723712+   83   Linux
# /dev/hda7          728          858      264064+   83   Linux
# /dev/hda8          859          989      264064+   83   Linux
# /dev/hda9          990         1022      66496+    82   Linux swap

# What fdisk does not do is provide output suitable for later
# importing into fdisk, a la sfdisk. This script parses the output
# from fdisk and creates an input file for fdisk. Use the input file
# like so:

# fdisk /dev/hdx < dev.hdx

# For the bare metal restore package, this script also builds a script
# that will execute the above command so you can run it from your zip
# disk. Because the bare metal restore scripts all are in /root/bin,
# the data file and script created by this script are also placed
# there. The same script also creates appropriate Linux file systems,
# either ext2fs, or Linux swap. There is limited support for FAT12,
# FAT16 and FAT32. For anything else, you're on your own.

# Note for FAT32: According to the MS KB, there are more than one
# reserved sectors for FAT32, usually 32, but it can vary. Do a search
# in M$'s KB for "boot sector" or BPB for the gory details. For more
# info than you really need on how boot sectors are used, see
# http://support.microsoft.com/support/kb/articles/Q140/4/18.asp

# You can also edit dev.x to change the sizes of partitions. Don't
# forget, if you change the size of a FAT partition across the 32MB
# boundary, you need to change the type as well! Run "fdisk /dev/hda"
# or some such, then the l command to see the available partition
# types. Then go ahead and edit dev.x appropriately. Also, when moving
# partition boundaries with hand edits, make sure you move both logical
# and extended partition boundaries appropriately.

# Bad block checking right now is a quick read of the partition. A
# writing check is also possible but more difficult. You have to run
# badblocks as a separate command, and pass the bad block list to
# mke2fs in a file (in /tmp, which is a ram disk). You also have to
# know how large the blocks are, which you learn by running
# dumpe2fs. It gets messy and I haven't done it yet. You probably
# don't need it for a new hard drive, but if you have had a hard drive
# crash on you and you are reusing it (while you are waiting for its
# replacement to come in, I presume), then I highly recommend it. Let
# me know how you do it.

# For more information contact the author, Charles Curley, at
# http://www.charlescurley.com/.

# cut2fmt figures out the format string for the unpack function we use
# to slice and dice the output from fdisk. From Christiansen and
# Torkington, Perl Cookbook 5.

sub cut2fmt {
    my (@positions) = @_ ;
    my $template    = '' ;
    my $lastpos     = 1 ;

    foreach $place (@positions) {
        $template .= "A" . ($place - $lastpos) . " " ;
        $lastpos = $place ;
    }

    $template .= "A*" ;
    return $template ;
}
```

```
# Sub gpl, a subroutine to ship the GPL and other header information
# to the current output file.

sub gpl {

print OUTPUT <<FINIS;

# Copyright 2001 through the last date of modification Charles Curley.

# This program is free software; you can redistribute it and/or modify it
# under the terms of the GNU General Public License as published by the
# Free Software Foundation; either version 2 of the License, or (at your
# option) any later version.

# This program is distributed in the hope that it will be useful, but
# WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
# General Public License for more details.

# You should have received a copy of the GNU General Public License along
# with this program; if not, write to the Free Software Foundation, Inc.,
# 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

# You can also contact the Free Software Foundation at http://www.fsf.org/

# For more information contact the author, Charles Curley, at
# http://www.charlescurley.com/.

FINIS

}

# Begin main line code.

# Provide a default device.

# print "\$ARGV[0] is \$ARGV[0].\n";

$device = defined ($ARGV[0]) ? $ARGV[0] : "/dev/hda";

# Need to check to see if $device is a sym link. If it is, the mount
# point is the target of the link. (Mandrake) Otherwise we search for
# mount points on $device. Fedora, Red Hat.

if ( -l $device) {

    # It is a sym link. Get the target of the link, then make it into
    # an absolute path, preserving the numbering.

    $mountdev = '/dev/' . readlink ($device);
    $mountdev =~ s|ide/host(\d+)/bus(\d+)/target(\d+)/lun(\d+)/disc|ide/host

} else {
    # not a sym link; just assign it.
    $mountdev = $device;
}

# print "Device is $device; mount device is $mountdev.\n";

# Prepare format string. Here are two format strings I have found
# useful. Here, column numbers are 1 based, i.e. the leftmost column
# is column 1, not column 0 as in Emacs.

# We select a format string according to fdisk's version.

$fdpid = open (FDVER, "fdisk -v |") or die "Couldn't fork: $!\n";
while (<FDVER>) {
```

```

    @_ = unpack ("A7 A*", $_);
    $fdver=$_[1];
    $fdver =~ s/[^\.d.]//g; # strip non-numbers, non-periods, as in "2.12pre".
}

# print "fdisk version is $fdver\n";

if ($fdver < 2.12) {
# fdisk to 2.11?? Red Hat, Fedora Core 1
    $fmt = cut2fmt (11, 19, 24, 34, 45, 49);
} else {
# fdisk 2.12 & up?? Mandrake 10.0, Fedora Core 2
    $fmt = cut2fmt (12, 14, 26, 38, 50, 55);
}
# print "Format string is $fmt.\n";

# define fields in the array @_.
$dev = 0;
$bootable = 1;
$firstcyl = 2;
$lastcyl = 3;
$parttype = 5;
$partstring = 6;

$target = "\/target";

$outputfilename = $device;
$outputfilename =~ s/\\//.g;
$outputfilename = substr ($outputfilename, 1, 100);

$outputfilepath = "\/root\/bin\/";

# Make a hash of the labels.
$mpid = open (MOUNT, "mount -l |") or die "Couldn't fork: $!\n";
while (<MOUNT>) {
    if ($_ =~ /^$mountdev/i) { # is this a line with a partition in it?
#        print $_;                # print it just for grins
        split;
        if ($_[6] ne "") {      # only process if there actually is a label
            $_[6] =~ s/[\\\/]//g; # strike [ and ].
            $labels{$_[0]} = $_[6];
#            print "The label of file device $_[0] is $labels{$_[0]}.\n";
        }
    }

    # We only mount if it's ext2fs or ext3fs and read and write.

    if ($_[4] =~ /ext[23]/ and $_[5] =~ /\(rw/ ) {
        if ($_[0] =~ /ide/i) {

            # We have a devfs system, e.g. Mandrake. This code
            # converts back from the devfs designation to the old
            # /dev/hd* designation for tomsrtb. I have NOT checked
            # this out for drives other than /dev/hda. Also, this
            # code does not handle SCSI drives.

            if ( $_[0] =~ /target0/ && $_[0] =~ /bus0/ ) {
                $letter = 'a';
            } elsif ( $_[0] =~ /target1/ && $_[0] =~ /bus0/) {
                $letter = 'b';
            } elsif ( $_[0] =~ /target0/ && $_[0] =~ /bus1/) {
                $letter = 'c';
            } else {
                $letter = 'd';
            }
            $_[0] =~ s|/ide/host\d+/bus\d+/target\d+/lun\d+/part|/hd|g;
            $_[0] =~ s/hd/hd$letter/;
        }
    }
}

```



```

        $mountpoints{$_[2]} = $_[0];
#         print "$_[2] is the mountpoint for tomsrtbt device $mountpoint.
    }
}
close (MOUNT);

$fpid = open (FDISK, "fdisk -l $device |") or die "Couldn't fork: $!\n";
open (OUTPUT, "> $outputfilepath${outputfilename}")
    or die "Couldn't open output file $outputfilepath${outputfilename}.\n";

while (<FDISK) {
    if ($_ =~ /^$device/i) {      # is this a line with a partition in it?
#         print $_;                # print it just for grins
        chop;                    # kill trailing \r
        @_ = unpack ($fmt, $_);

        # now strip white spaces from cylinder numbers
        @_[ $firstcyl ] =~ s/[ \t]+//;
        @_[ $lastcyl ]  =~ s/[ \t]+//;
        @_[ $parttype ] =~ s/[ \t]+//;

        $partnumber = substr(@_[ $dev ], 8, 10); # get partition number for th
        # just for grins
#         print "    $partnumber, @_[ $firstcyl ], @_[ $lastcyl ], @_[ $parttype ], @_

        # Here we start creating the input to recreate the partition
        # this line represents.

        print OUTPUT "n\n";
        if ($partnumber < 5) {
            # primary Linux partition
            if (@_[ $parttype ] == 83) {
                print OUTPUT "p\n$partnumber\n@_[ $firstcyl ]\n";
                if (@_[ $firstcyl ] ne @_[ $lastcyl ]) { # in case it's all on o
                    print OUTPUT "@_[ $lastcyl ]\n";
                }

                # Now detect if this is an ext3 (journaling)
                # partition. We do this using dumpe2fs to dump the
                # partition and grepping on "journal". If the
                # partition is ext2, there will be no output. If it is
                # ext3, there will be output, and we use that fact to
                # set a command line switch. The command line switch
                # goes into an associative array (hash) so we don't
                # have to remember to reset it to the null string when
                # we're done.

                $dpid = open (DUMPE2FS, "dumpe2fs @_[ $dev ] 2>/dev/null | grep
                    or die "Couldn't fork: $!\n";
                while (<DUMPE2FS) {
#                 print "Dumpe2fs: $_";
                    $ext3{$_[ $dev ]} = "-j ";
                    last;
                }
                close (DUMPE2FS);

                if ($labels{@_[ $dev ]}) { # do we have a label?
                    $format .= "echo\necho formatting $checking@_[ $dev ]\n";
                    $format .= "mke2fs $ext3{$_[ $dev ]}\$blockcheck -L $label
                } else {
                    $format .= "echo\necho formatting $checking@_[ $dev ]\n";
                    $format .= "mke2fs $ext3{$_[ $dev ]}\$blockcheck @_[ $dev ]\
                }

                # extended partition
            } elsif (@_[ $parttype ] == 5) {

```

```

        # print ("Creating Extended Partition.\n");
        print OUTPUT "e\n$partnumber\n@_[$firstcyl]\n";
        if (@_[$firstcyl] ne @_[$lastcyl]) {
            print OUTPUT "@_[$lastcyl]\n";
        }

        # extended partition, Win95 Ext'd (LBA)
    } elsif (@_[$parttype] eq "f") {
        # print ("Creating Extended LBA Partition.\n");
        print OUTPUT "e\n$partnumber\n@_[$firstcyl]\n";
        if (@_[$firstcyl] ne @_[$lastcyl]) {
            print OUTPUT "@_[$lastcyl]\n";
        }
        print OUTPUT "t\n$partnumber\nf\n";

        # primary Linux swap partition
    } elsif (@_[$parttype] == 82) {
        print OUTPUT "p\n$partnumber\n@_[$firstcyl]\n";
        if (@_[$firstcyl] ne @_[$lastcyl]) {
            print OUTPUT "@_[$lastcyl]\n";
        }
        print OUTPUT "t\n$partnumber\n82\n";
        $format .= "echo Making @_[$dev] a swap partition.\n";
        $format .= "mkswap \ $blockcheck @_[$dev]\n\n";

        # Primary mess-dos partition. We don't handle hidden
        # partitions.
    } elsif ( @_[$parttype] == 1 || @_[$parttype] == 4 || @_[$parttype] ==
        || @_[$parttype] eq "b" || @_[$parttype] eq "c"
        || @_[$parttype] eq "e" ) {
        # print ("Making DOS primary partition.\n");
        print ("dd if=@_[$dev] of=$outputfilepath$outputfilename$partnumber
        print (" bs=512 count=1\n");
        system ("dd if=@_[$dev] of=$outputfilepath$outputfilename$partnumber
        print OUTPUT "p\n$partnumber\n@_[$firstcyl]\n";
        if (@_[$firstcyl] ne @_[$lastcyl]) { # in case it's all on one cyl
            print OUTPUT "@_[$lastcyl]\n";
        }
        print OUTPUT "t\n$partnumber\n@_[$parttype]\n";
        $format .= "echo\nnecho formatting $checking@_[$dev]\n";
        $format .= "mkdosfs \ $blockcheck";
        if ( @_[$parttype] == b || @_[$parttype] == c) {
            # We have a W9x FAT32 partition. Add a command line switch.
            $format .= " -F 32";
        }
        $format .= " @_[$dev]\n";
        $format .= "# restore FAT boot sector.\n";
        $format .= "dd if=$outputfilename$partnumber of=@_[$dev] bs=512 co

    } else {
        # anything else partition
        print OUTPUT "p\n@_[$firstcyl]\n";
        if (@_[$firstcyl] ne @_[$lastcyl]) {
            print OUTPUT "@_[$lastcyl]\n";
        }
        print OUTPUT "t\n$partnumber\n@_[$parttype]\n";
    }
} else {
    # logical Linux partition
    if (@_[$parttype] == 83) {
        print OUTPUT "l\n@_[$firstcyl]\n";
        if (@_[$firstcyl] ne @_[$lastcyl]) {
            print OUTPUT "@_[$lastcyl]\n";
        }

        # Now detect if this is an ext3 (journaling)
        # partition. We do this using dumpe2fs to dump the
        # partition and grepping on "journal". If the

```

```

# partition is ext2, there will be no output. If it is
# ext3, there will be output, and we use that fact to
# set a command line switch. The command line switch
# goes into an associative array (hash) so we don't
# have to remember to reset it to the null string when
# we're done.

$dpid = open (DUMPE2FS, "dumpe2fs @_[$dev] 2>/dev/null | grep
    or die "Couldn't fork: $!\n";
while (<DUMPE2FS>) {
#   print "Dumpe2fs: $_";
    $ext3{$_[$dev]} = "-j ";
    last;
}
close (DUMPE2FS);

if ($labels{$_[$dev]}) { # do we have a label?
    $format .= "echo\necho formatting $checking@_[$dev]\n";
    $format .= "mke2fs $ext3{$_[$dev]}\$blockcheck -L $label";
} else {
    $format .= "echo\necho formatting $checking@_[$dev]\n";
    $format .= "mke2fs $ext3{$_[$dev]}\$blockcheck @_[$dev]\n";
}

# logical Linux swap partition
} elsif (@_[$parttype] == 82 ) {
    print OUTPUT "l\n@_[$firstcyl]\n";
    if (@_[$firstcyl] ne @_[$lastcyl]) {
        print OUTPUT "@_[$lastcyl]\n";
    }
    print OUTPUT "t\n$partnumber\n82\n";
    $format .= "echo Making @_[$dev] a swap partition.\n";
    $format .= "mkswap \$blockcheck @_[$dev]\n\n";

    # Logical mess-dos partition. We don't handle hidden
    # partitions.

} elsif ( @_[$parttype] == 1 || @_[$parttype] == 4 || @_[$parttype] == 5 ||
    @_[$parttype] eq "b" || @_[$parttype] eq "c"
    || @_[$parttype] eq "e" ) {
#   print ("Making DOS logical partition.\n");
    print ("dd if=@_[$dev] of=$outputfilepath$outputfilename$partnumber
    bs=512 count=1\n");
    system ("dd if=@_[$dev] of=$outputfilepath$outputfilename$partnumber
    bs=512 count=1\n");
    print OUTPUT "l\n$partnumber\n@_[$firstcyl]\n";
    if (@_[$firstcyl] ne @_[$lastcyl]) { # in case it's all on one cylinder
        print OUTPUT "@_[$lastcyl]\n";
    }
    print OUTPUT "t\n$partnumber\n@_[$parttype]\n";
    $format .= "echo\necho formatting $checking@_[$dev]\n";
    $format .= "mkdosfs \$blockcheck";
    if ( @_[$parttype] == b || @_[$parttype] == c ) {
        # We have a W9x FAT32 partition. Add a command line switch
        $format .= " -F 32";
    }
    $format .= " @_[$dev]\n";
    $format .= "# restore FAT boot sector.\n";
    $format .= "dd if=$outputfilename$partnumber of=@_[$dev] bs=512
    count=1\n";
} else {
    # anything else partition
    print OUTPUT "l\n@_[$firstcyl]\n";
    if (@_[$firstcyl] ne @_[$lastcyl]) {
        print OUTPUT "@_[$lastcyl]\n";
    }
    print OUTPUT "t\n$partnumber\n@_[$parttype]\n";
}
}
}

```

```
        # handle bootable partitions
        if (@_[$bootable] =~ /\*/) {
            print OUTPUT "a\n$partnumber\n";
        }
    }
}

print OUTPUT "v\nw\n";

close (OUTPUT);
close (FDISK);

open (OUTPUT, "> ${outputfilepath}make.$outputfilename")
    or die "Couldn't open output file ${outputfilepath}make.$outputfilename.\n";

print OUTPUT <<FINIS;
#! /bin/sh

# A script to restore the partition data of a hard drive and format
# the partitions. Created at bare metal backup time by the Perl script
# make.fdisk.
FINIS

&gpl;

print OUTPUT <<FINIS;

export blockcheck=\$1;

if [ "\$blockcheck" != "-c" ] && [ -n "\$blockcheck" ]
then
    echo "\${0}: automated restore with no human interaction."
    echo "\${0}: -c: block check during file system making."
    exit 1;
fi

FINIS

# Clean the old partition table out.
print OUTPUT "dd if=/dev/zero of=\$device bs=512 count=2\n\nsync\n\n";

print OUTPUT "fdisk \$device \< \$outputfilename\n\nsync\n\n";
print OUTPUT \$format;

print OUTPUT "fdisk -l \"\$device\"\n\n";

close (OUTPUT);

# Now build the script that will build the mount points on the root
# and other partitions.

open (OUTPUT, "> ${outputfilepath}mount.$outputfilename")
    or die "Couldn't open output file ${outputfilepath}make.$outputfilename.\n";

print OUTPUT <<FINIS;
#! /bin/sh

# A script to create a minimal directory tree on the target hard drive
# and mount the partitions on it. Created at bare metal backup time by
# the Perl script make.fdisk.
FINIS

&gpl;

print OUTPUT <<FINIS;

# WARNING: If your Linux system mount partitions across hard drive
```

```
# boundaries, you will have multiple "mount.dev.*" scripts. You must
# ensure that they run in the proper order. The root partition should
# be mounted first, then the rest in the order they cascade. If they
# cross mount, you'll have to handle that manually.

FINIS

# We have a hash of mount points and devices in %mountpoints. However,
# we have to process them such that directories are built on the
# appropriate target partition. E.g. where /usr/local is on its own
# partition, we have to mount /usr before we build /usr/local. We can
# ensure this by sorting them. Shorter mount point paths will be built
# first. We can't sort a hash directly, so we use an array.

# We build commands to create the appropriate mount points and then
# mount the partitions to the mount points. This is in preparation for
# untarring the contents of the ZIP disk, done in restore.metadata.

foreach $point ( sort keys %mountpoints) {
    print OUTPUT "\n# $point is the mountpoint for tomsrtbt device $mountpoint\n";
    print OUTPUT "mkdir $target$point\n";
    print OUTPUT "mount $mountpoints{$point} $target$point\n";
}

print OUTPUT "\nmount | grep -i \"$device\"\n";

close (OUTPUT);

# These scripts are dangerous & should only be visible to root.

chmod 0700, "${outputfilepath}make.$outputfilename";
chmod 0700, "${outputfilepath}mount.$outputfilename";
chmod 0600, "${outputfilepath}$outputfilename";
```

11.1.2. make.dev.hda

Ce script est un exemple de la production de `make.fdisk`, plus haut. Il fait appel à des fichiers de données comme `dev.hda`, plus bas. Il crée des partitions et des systèmes de fichiers sur certains d'entre elles. C'est le premier script lancé à la restauration.

Si vous êtes suffisamment courageux pour éditer `dev.hda` (voir plus haut), pour, disons, ajouter une nouvelle partition, il se peut que vous deviez éditer ce script.

Si vous voulez que `make.dev.hda` vérifie les secteurs défectueux quand il installe un système de fichiers sur les partitions, saisissez en ligne de commande l'option `"-c"`.

```
#!/bin/sh

# A script to restore the partition data of a hard drive and format
# the partitions. Created at bare metal backup time by the Perl script
# make.fdisk.

# Copyright 2001 through the last date of modification Charles Curley.

# This program is free software; you can redistribute it and/or modify it
# under the terms of the GNU General Public License as published by the
# Free Software Foundation; either version 2 of the License, or (at your
# option) any later version.

# This program is distributed in the hope that it will be useful, but
# WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
# General Public License for more details.
```

```
# You should have received a copy of the GNU General Public License along
# with this program; if not, write to the Free Software Foundation, Inc.,
# 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

# You can also contact the Free Software Foundation at http://www.fsf.org/

# For more information contact the author, Charles Curley, at
# http://www.charlescurley.com/.

export blockcheck=$1;

if [ "$blockcheck" != "-c" ] && [ -n "$blockcheck" ]
then
    echo "${0}: automated restore with no human interaction."
    echo "${0}: -c: block check during file system making."
    exit 1;
fi

dd if=/dev/zero of=/dev/hda bs=512 count=2

sync

fdisk /dev/hda < dev.hda

sync

echo
echo formatting /dev/hda1
mkdosfs $blockcheck /dev/hda1
# restore FAT boot sector.
dd if=dev.hda1 of=/dev/hda1 bs=512 count=1

echo
echo formatting /dev/hda2
mke2fs -j $blockcheck -L /boot /dev/hda2

echo
echo formatting /dev/hda3
mke2fs -j $blockcheck -L / /dev/hda3

echo Making /dev/hda5 a swap partition.
mkswap $blockcheck /dev/hda5

fdisk -l "/dev/hda"
```

11.1.3. mount.dev.hda

Ce script est un exemple de la production de `make.fdisk`, plus haut. Il crée des points de montage et monte les partitions, préparant le système de fichiers cible à la restauration des fichiers. C'est le second script lancé à la restauration.

Si vous êtes suffisamment courageux pour éditer `dev.hda` (voir plus haut), pour, disons, ajouter une nouvelle partition, il se peut que vous deviez éditer ce script.

```
#!/bin/sh

# A script to create a minimal directory tree on the target hard drive
# and mount the partitions on it. Created at bare metal backup time by
# the Perl script make.fdisk.

# Copyright 2001 through the last date of modification Charles Curley.
```

```
# This program is free software; you can redistribute it and/or modify it
# under the terms of the GNU General Public License as published by the
# Free Software Foundation; either version 2 of the License, or (at your
# option) any later version.

# This program is distributed in the hope that it will be useful, but
# WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
# General Public License for more details.

# You should have received a copy of the GNU General Public License along
# with this program; if not, write to the Free Software Foundation, Inc.,
# 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

# You can also contact the Free Software Foundation at http://www.fsf.org/

# For more information contact the author, Charles Curley, at
# http://www.charlescurley.com/.

# WARNING: If your Linux system mount partitions across hard drive
# boundaries, you will have multiple "mount.dev.*" scripts. You must
# ensure that they run in the proper order. The root partition should
# be mounted first, then the rest in the order they cascade. If they
# cross mount, you'll have to handle that manually.

# / is the mountpoint for /dev/hda3.
mkdir /target/
mount /dev/hda3 /target/

# /boot is the mountpoint for /dev/hda2.
mkdir /target/boot
mount /dev/hda2 /target/boot

mount | grep -i "/dev/hda"
```

11.1.4. dev.hda

Ce fichier de données est utilisé au moment de la restauration. Il est utilisé par **fdisk** et est alimenté par `make .dev.hda`. Il est produit pendant la sauvegarde par `make .fdisk`. Ceux qui connaissent bien **fdisk** reconnaîtront que chaque ligne est une commande ou une valeur de **fdisk**, tel qu'un numéro de cylindre. Donc, il est possible de changer les tailles des partitions et d'ajouter de nouvelles partitions en éditant ce fichier. C'est pourquoi la pénultième commande est **v**, qui vérifiera la table des partitions avant qu'elle soit écrite.

```
n
p
1
1
29
t
1
6
a
1
n
p
2
30
44
n
e
3
```

```
45
1023
n
l
45
944
n
l
945
1023
t
6
82
v
w
```

11.1.5. save.metadata

C'est le premier script lancé dans le processus de sauvegarde. Il appelle `make.fdisk`, plus haut. Si vous devez sauvegarder un disque dur SCSI ou plusieurs disques durs, faites de sorte que l'appel à `make.fdisk` soit effectué de façon correcte.

```
#!/bin/sh

# A script to save certain meta-data off to the boot partition. Useful for
# restoration.

# Time-stamp: <2004-04-29 15:36:52 root save.metadata>

# Copyright 2000 through the last date of modification, Charles Curley.

# This program is free software; you can redistribute it and/or modify it
# under the terms of the GNU General Public License as published by the
# Free Software Foundation; either version 2 of the License, or (at your
# option) any later version.

# This program is distributed in the hope that it will be useful, but
# WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
# General Public License for more details.

# You should have received a copy of the GNU General Public License along
# with this program; if not, write to the Free Software Foundation, Inc.,
# 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

# You can also contact the Free Software Foundation at http://www.fsf.org/

# 2003 01 08: We now age the output from rpm -VA to make back
# comparisons easier.

# The loop that creates directories now has the -p option for mkdir,
# which means you can create parents on the fly if they don't already
# exist.

# initrd is now in the list of directories to create automatically.

# We now exclude more stuff when building the tarballs.

# 2002 07 01: Went to bzip2 to compress the archives, for smaller
# results. This is important in a 100MB ZIP disk. Also some general
# code cleanup.

# 2002 07 01: The function crunch will tar and BZIP2 the
```



```
# archives. This is cleaner than the old code, and has better safety
# checking.

# For more information contact the author, Charles Curley, at
# http://www.charlescurley.com/.

# Crunch: A function to compress the contents of a directory and put
# the archive onto the ZIP disk.

# The first parameter is the name of the archive file to be
# created. The backup location, $zip, will be prepended and the
# extension, "tar.bz2" will be appended.

# All following parameters will be taken as additional directories or
# files to be put into the archive.

function crunch {
if [ -z "$1" ] || [ -z "$2" ] # Checks if parameter #1 or #2 is zero length.
then
    echo "-Parameter #1 or #2 is missing.-" # Also if no parameter is passed
    return 1
else
    local file=$1 # The archive file to create
    shift # Discard the file name
    local dirs=$@ # The director[y]ies] to archive
    local tarcmd="tar -cjf" # The tar command.

    $tarcmd $zip/$file.tar.bz2 $dirs # do it!!

    error=$? # Preserve the exit code

    if [ $error != 0 ] # Did we fail?
    then # Yes
        echo "Tar failed with error $error"
        echo $tarcmd $zip/$file.tar.bz2 $dirs
        exit $error # return tar's exit code as ours
    fi

    return 0 # For error testing if needed.
fi
}

# Begin the main line code
export zip="/mnt/zip"; # Where we will put archives
# export save="/mnt/save";

RPMVABACKS=/etc # where we keep our backups
RPMVAROOT=rpmVa # The root name of the pg backups
ANC=${RPMVABACKS}/${RPMVAROOT}.anc # name for the oldest (ancient) backup
OLD=${RPMVABACKS}/${RPMVAROOT}.old # name for the middling oldest backup
NEW=${RPMVABACKS}/${RPMVAROOT}.txt # name for the newest backup

if [ -f ${ANC} ]; then
echo "Deleting ${ANC}"
rm ${ANC}
fi

if [ -f ${OLD} ]; then
echo "Aging ${OLD}"
mv ${OLD} ${ANC}
fi

if [ -f ${NEW} ]; then
echo "Aging ${NEW}"
mv ${NEW} ${OLD}
fi
fi
```

```
# Now we save hard drive information. Run make.fdisk on each hard
# drive in the order in which it mounted from the root partition. That
# is, run it first on the hard drive with your root partition, then
# any hard drives that mount to the first hard drive, then any hard
# drives that mount to those. For example, if your root partition is
# on /dev/sdc, run "make.fdisk /dev/sdc" first.

# The reason for this is that make.fdisk produces a script to make
# mount points and then mount the appropriate partition to them during
# first stage restore. Mount points must be created on the partition
# where they will reside. The partitions must be mounted in this
# order. For example, if your /var and /var/ftp are both separate
# partitions, then you must mount /, create /var, then mount /var,
# then create /var/ftp. The order in which the script "first.stage"
# runs the mounting scripts is based on their time of creation.

# If necessary, put a line, "sleep 1" between calls to make.fdisk.

echo "Saving hard drive info"
make.fdisk /dev/hda

# back up RPM metadata

echo "Verifying RPMs."

rpm -Va | sort +2 -t ' ' > ${NEW}

echo "Finished verifying RPMs; now mounting the ZIP drive."

# Make sure we have the ZIP drive mounted.
umount $zip
modprobe ppa                # Driver for 100MB parallel port ZIP disk
mount $zip                  # It should have ext2fs on partition 1.

# clean it all out
rm -r $zip/*
mkdir $zip/lost+found

echo "`hostname` bare metal ZIP disk, created `date`. `uname -a`" > $zip/README.txt

echo "Building the ZIP drive backups."

# These are in case we need to refer to them while rebuilding. The
# rebuilding process should be mostly automated, but you never
# know....

fdisk -l /dev/hda > $zip/fdisk.hda

ls -al /mnt > $zip/ls.mnt.txt
ls -al / > $zip/ls.root.txt

cd /

# Build our minimal archives on the ZIP disk. These appear to be
# required so we can restore later on.

crunch boot boot
crunch root root --exclude root/.cpan --exclude root/.mozilla
crunch etc etc --exclude etc/samba --exclude X11
crunch lib lib

crunch usr.sbin usr/sbin
crunch usr.bin usr/bin --exclude usr/bin/emacs --exclude usr/bin/emacs-21.2 --excl
crunch sbin sbin
crunch bin bin
crunch dev dev
crunch kerberos usr/kerberos/lib/
```

```
# Now optional saves.

# arkeia specific:
# crunch arkeia usr/knox

# save these so we can use ssh for restore. *crack* for RH 7.0 login
# authentication.
crunch usr.lib usr/lib/*crack* usr/lib/libz* usr/lib/libssl* usr/lib/libcrypt*

# save the scripts we used to create the ZIP disk and the ones we will
# use to restore it.
mkdir $zip/root.bin
cp -p /root/bin/* $zip/root.bin
rm $zip/root.bin/*~ $zip/root.bin/###

echo "Testing our results."
find $zip -iname "*.bz2" | xargs bunzip2 -t

# Not a normal part of the process: we duplicate the ZIP disk onto an
# NFS mount elsewhere.

# echo "Backing the ZIP drive to the NFS mount."

# umount $save
# mount $save

# rm -r $save/zip
# mkdir $save/zip
# cp -pr $zip $save

du -hs ${zip}*
df -m
```

11.1.6. restore.metadata

Ce script restaure les métadonnées du disque ZIP à la première étape de la restauration.

```
#!/bin/sh

# A script to restore the meta-data from the ZIP disk. This runs under
# tomsrtbt only after partitions have been rebuilt, file systems made,
# and mounted. It also assumes the ZIP disk has already been
# mounted. Mounting the ZIP disk read only is probably a good idea.

# Time-stamp: <2003-08-23 10:09:16 ccurley restore.metadata>

# Copyright 2000 through the last date of modification Charles Curley.

# This program is free software; you can redistribute it and/or modify it
# under the terms of the GNU General Public License as published by the
# Free Software Foundation; either version 2 of the License, or (at your
# option) any later version.

# This program is distributed in the hope that it will be useful, but
# WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
# General Public License for more details.

# You should have received a copy of the GNU General Public License along
# with this program; if not, write to the Free Software Foundation, Inc.,
# 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

# You can also contact the Free Software Foundation at http://www.fsf.org/
```

```
# 2003 08 23: Oops: tar on tomsrtbt does not respect -p. Try setting
# umask to 0000 instead.

# 2003 02 13: Tar was not preserving permissions on restore. Fixed
# that.

# 2002 07 01: Went to bzip2 to compress the archives, for smaller
# results. This is important in a 100MB ZIP disk. Also some general
# code cleanup.

# For more information contact the author, Charles Curley, at
# http://www.charlescurley.com/.

umask 0000

zip="/mnt";           # Where we mount the zip drive.
target="/target";     # Where the hard drive to restore is mounted.

ls -lt $zip           # Warm fuzzies for the user.

cd $target

# Restore the archived metadata files.
for archive in $( ls $zip/*.bz2 ); do
echo $archive
ls -al $archive
bzip2 -dc $archive | tar -xf -
done

# Build the mount points for our second stage restoration and other
# things.

# If you boot via an initrd, make sure you build a directory here so
# the kernel can mount the initrd at boot.

for dir in mnt/zip mnt/cdrom mnt/floppy mnt/imports proc initrd; do
mkdir -p $target/$dir
done

chmod a-w $target/proc      # Restore /proc's read-only permissions

# Restore the scripts we used to create the ZIP disk and the ones we will
# use to restore it. These should be the latest & greatest in case we had
# to do any editing during 1st stage restore.
cp -p $zip/root.bin/* $target/root/bin

# Now install the boot sector.
chroot $target /sbin/lilo -C /etc/lilo.conf

df -m
```

11.1.7. first.stage

Ce script effectue complètement la première étape de la restauration sans intervention humaine.

Si vous désirez vérifier les secteurs défectueux pendant la création des systèmes de fichiers des partitions, utilisez l'option de ligne de commande « -c ».

```
#!/bin/sh

# A master script to run the other, detailed scripts. Use this script
# only if you want no human intervention in the restore process. The
# only option is -c, which forces bad block checking during formatting
```

```
# of the partitions.

# Time-stamp: <2003-04-24 10:03:07 ccurley first.stage>

# Copyright 2002 through the last date of modification Charles Curley.

# This program is free software; you can redistribute it and/or modify it
# under the terms of the GNU General Public License as published by the
# Free Software Foundation; either version 2 of the License, or (at your
# option) any later version.

# This program is distributed in the hope that it will be useful, but
# WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
# General Public License for more details.

# You should have received a copy of the GNU General Public License along
# with this program; if not, write to the Free Software Foundation, Inc.,
# 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

# You can also contact the Free Software Foundation at http://www.fsf.org/

# For more information contact the author, Charles Curley, at
# http://www.charlescurley.com/.

export blockcheck=$1;

if [ "$blockcheck" != "-c" ] && [ -n "$blockcheck" ]
then
    echo "${0}: automated restore with no human interaction."
    echo "${0}: -c: block check during file system making."
    exit 1;
fi

cd /mnt/root.bin          # just in case we aren't already where we should be

for drive in $( ls make.dev.* ); do
    ./${drive} $blockcheck;
done

# WARNING: If your Linux system mount partitions across hard drive
# boundaries, you will have multiple "mount.dev.*" scripts. You must
# ensure that they run in the proper order, which the loop below may
# not do. The root partition should be mounted first, then the rest in
# the order they cascade. If they cross mount, you'll have to handle
# that manually.

# The "ls -tr" will list the scripts in the order they are created, so
# it might be a good idea to create them (in the script save.metadata)
# in the order in which you should run them.

for drive in $( ls -tr mount.dev.* ); do
    ./${drive};
done

./restore.metadata

# People who are really confident may comment this line in.
# reboot
```

11.2. Deuxième étape

Ces scripts sont lancés sur l'ordinateur à sauvegarder ou restaurer.

11.2.1. back.up.all

Ce script effectue une sauvegarde vers un autre ordinateur via un montage NFS. Vous pouvez l'adapter pour effectuer vos sauvegardes vers une bande ou d'autres supports.

```
#!/bin/sh

# Back up the entire system to another computer's drive. To make this
# work, we need a convenient chunk of disk space on the remote computer we
# can nfs mount as /mnt/save.

# Time-stamp: <2003-04-24 09:56:05 ccurley back.up.all>

# Copyright 2000 through the last date of modification Charles Curley.

# This program is free software; you can redistribute it and/or modify it
# under the terms of the GNU General Public License as published by the
# Free Software Foundation; either version 2 of the License, or (at your
# option) any later version.

# This program is distributed in the hope that it will be useful, but
# WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
# General Public License for more details.

# You should have received a copy of the GNU General Public License along
# with this program; if not, write to the Free Software Foundation, Inc.,
# 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

# You can also contact the Free Software Foundation at http://www.fsf.org/

# For more information contact the author, Charles Curley, at
# http://www.charlescurley.com/.

save="/mnt/save"

# Make sure it's there
umount $save
mount $save

cd /

rm $save/tester.tar.old.gz
mv $save/tester.tar.gz $save/tester.tar.old.gz

# save everything except /mnt, /proc, and nfs mounted directories.

time tar cf - / --exclude /mnt --exclude /proc --exclude $save\
| gzip -c > $save/tester.tar.gz
```

11.2.2. back.up.all.ssh

Ce script fait exactement la même chose que `back.up.all` mais il utilise `ssh` à la place de `nfs`.

```
#!/bin/sh

# Back up the entire system to another computer's drive. To make this
# work, we need a convenient chunk of disk space on the remote
# computer. This version uses ssh to do its transfer, and compresses
# using bz2. This means this script has to know more about the other
# computer, which does not make for good modularization.
```

```
# Time-stamp: <2003-04-24 09:56:52 ccurley back.up.all.ssh>
# Copyright 2000 through the last date of modification Charles Curley.

# This program is free software; you can redistribute it and/or modify it
# under the terms of the GNU General Public License as published by the
# Free Software Foundation; either version 2 of the License, or (at your
# option) any later version.

# This program is distributed in the hope that it will be useful, but
# WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
# General Public License for more details.

# You should have received a copy of the GNU General Public License along
# with this program; if not, write to the Free Software Foundation, Inc.,
# 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

# You can also contact the Free Software Foundation at http://www.fsf.org/

# For more information contact the author, Charles Curley, at
# http://www.charlescurley.com/.

save="/backs/tester"
backup_server="charlesc"

# rotate the old backups. Do it all in one line to minimize authentication over
ssh $backup_server "rm $save/tester.tar.old.bz2; mv $save/tester.tar.bz2 \
    $save/tester.tar.old.bz2"

# save everything except /mnt, /proc, and squid directories.

time tar cf - / --exclude /mnt --exclude /proc --exclude /var/spool/squid\
    | ssh $backup_server "bzip2 -9 > $save/tester.tar.bz2"
```

11.2.3. restore.all

Vous utiliserez ce script de restauration si votre sauvegarde a été effectuée avec `back.up.all`.

```
#!/bin/sh

# A script to restore all of the data from an nfs mount. This is our final
# stage restore.

# Time-stamp: <2003-04-24 09:58:51 ccurley restore.all>
# Copyright 2000 through the last date of modification Charles Curley.

# This program is free software; you can redistribute it and/or modify it
# under the terms of the GNU General Public License as published by the
# Free Software Foundation; either version 2 of the License, or (at your
# option) any later version.

# This program is distributed in the hope that it will be useful, but
# WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
# General Public License for more details.

# You should have received a copy of the GNU General Public License along
# with this program; if not, write to the Free Software Foundation, Inc.,
# 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

# You can also contact the Free Software Foundation at http://www.fsf.org/
```

```
# For more information contact the author, Charles Curley, at
# http://www.charlescurley.com/.

export save="/mnt/save"

mount $save

cd /
gunzip -dc $save/tester.tar.gz | tar -xpkf -

rm /var/run/*.pid

lilo
```

11.2.4. restore.all.ssh

Vous utiliserez ce script de restauration si votre sauvegarde a été effectuée avec back.up.all.ssh.

```
#!/bin/sh

# A script to restore all of the data using ssh and bunzip2. This is
# our final stage restore.

# Copyright 2000 through the last date of modification Charles Curley.
# Time-stamp: <2003-04-24 09:59:10 ccurley restore.all.ssh>

# This program is free software; you can redistribute it and/or modify it
# under the terms of the GNU General Public License as published by the
# Free Software Foundation; either version 2 of the License, or (at your
# option) any later version.

# This program is distributed in the hope that it will be useful, but
# WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
# General Public License for more details.

# You should have received a copy of the GNU General Public License along
# with this program; if not, write to the Free Software Foundation, Inc.,
# 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

# You can also contact the Free Software Foundation at http://www.fsf.org/

# For more information contact the author, Charles Curley, at
# http://www.charlescurley.com/.

save="/backs/tester/"
backup_server="charlesc"

cd /

ssh $backup_server "cat $save/tester.tar.bz2" | bunzip2 | tar -xpkf -

rm /var/run/*.pid

lilo
```

11.3. Scripts de sauvegarde du serveur

Les scripts ssh décrits ci-dessus sont susceptibles de poser un problème de sécurité. Si vous les lancez derrière un pare-feu, le pare-feu doit permettre à ssh d'accéder au serveur de sauvegarde. Dans ce cas, il se peut qu'un pirate intelligent soit aussi capable de pirater le serveur de sauvegarde. Il serait plus sûr d'exécuter les scripts de sauvegarde et de restauration sur le serveur de sauvegarde, et de laisser le serveur de sauvegarde accéder au pare-feu. Ces scripts sont conçus pour fonctionner ainsi. Renommez-les en `get.x` et en `restore.x` où `x` est le nom de l'ordinateur cible. Éditez-les (la variable `$target` d'initialisation définissant la cible) pour qu'ils utilisent le nom d'hôte de l'ordinateur cible, ou réécrivez-les pour qu'ils utilisent un argument de ligne de commande.

Ces scripts effectuent complètement la sauvegarde et la restauration de la cible, et pas uniquement la première étape de la sauvegarde et de la restauration. Remarquez aussi que `get.test` sauvegarde aussi le disque ZIP, au cas où vous auriez besoin de remplacer un disque ZIP défectueux.

J'utilise couramment ces scripts.

11.3.1. `get.test`

```
#!/bin/sh

# Back up another computer's drive to this system. To make this work, we
# need a convenient chunk of disk space on this computer. This version
# uses ssh to do its transfer, and compresses using bz2. This version was
# developed so that the system to be backed up won't be authenticated to
# log onto the backup computer. This script is intended to be used on a
# firewall. You don't want the firewall to be authenticated to the backup
# system in case the firewall is cracked.

# Time-stamp: <2004-04-03 12:24:12 ccurley get.test>

# Copyright 2000 through the last date of modification Charles Curley.

# This program is free software; you can redistribute it and/or modify it
# under the terms of the GNU General Public License as published by the
# Free Software Foundation; either version 2 of the License, or (at your
# option) any later version.

# This program is distributed in the hope that it will be useful, but
# WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
# General Public License for more details.

# You should have received a copy of the GNU General Public License along
# with this program; if not, write to the Free Software Foundation, Inc.,
# 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

# You can also contact the Free Software Foundation at http://www.fsf.org/

# For more information contact the author, Charles Curley, at
# http://www.charlescurley.com/.

# 2004 04 03: added /sys to the list of excludes. It is a read-only
# pseudo-file system like /proc.

# 2002 07 01: We now set the path on the target to the zip drive with
# a variable. This fixes a bug in the command to eject the zip disk.

# 2002 07 01: The zip disk archives are now in bzip2 format, so this
# script has been changed to reflect that.

# The host name of the computer to be backed up.
target=tester
zip=/mnt/zip

echo Backing up $target
```

```
echo Aging the ZIP disk backups.
rm -r $target.old.zip
mv $target.zip $target.old.zip
ssh $target "modprobe ppa ; mount -r $zip"
echo Copying the ZIP disk.

# -r for recursive copy, -p to preserve times and permissions, -q for
# quiet: no progress meter.

scp -qpr $target:$zip $target.zip
du -hs $target.zip

echo Aging the archives
rm $target.tar.old.bz2
mv $target.tar.bz2 $target.tar.old.bz2

echo Backing up $target to the backup server.

ssh $target tar -cf - / --exclude /sys --exclude /mnt --exclude /proc\
    --exclude /var/spool/squid\
    | bzip2 -9 | cat > $target.tar.bz2

echo Testing the results.
find . -iname "*.bz2" | xargs bunzip2 -t

ssh $target "eject $zip"
```

11.3.2. restore.testster

```
#!/bin/sh

# A script to restore all of the data to tester via ssh. This is our final
# stage restore.

# Time-stamp: <2003-04-24 09:59:45 ccurley restore.testster>

# Copyright 2000 through the last date of modification Charles Curley.

# This program is free software; you can redistribute it and/or modify it
# under the terms of the GNU General Public License as published by the
# Free Software Foundation; either version 2 of the License, or (at your
# option) any later version.

# This program is distributed in the hope that it will be useful, but
# WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
# General Public License for more details.

# You should have received a copy of the GNU General Public License along
# with this program; if not, write to the Free Software Foundation, Inc.,
# 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

# You can also contact the Free Software Foundation at http://www.fsf.org/
```

```
# For more information contact the author, Charles Curley, at
# http://www.charlescurley.com/.

# The host name of the computer to be restored.

target=tester

bunzip2 -dc $target.tar.bz2 | ssh $target "cd / ; tar -xpkf - "
ssh $target lilo
```

12. Ressources

Dans le désordre. Il y a des points que vous avez peut-être envie d'approfondir. Une présence dans cette liste ne doit pas être comprise comme une approbation. En fait, bien souvent, je n'ai pas utilisé le produit et ne peux pas le commenter.

- W. Curtis Preston [<http://www.oreilly.com/catalog/unixbr/author.html>] et son excellent *Unix Backup & Recovery* [<http://www.oreilly.com/catalog/unixbr/>]. Voici le livre qui m'a fait démarrer ce travail de restauration intégrale de systèmes. Je le recommande chaleureusement ; lire mon bilan [<http://www2.linuxjournal.com/lj-issues/issue78/3839.html>].
- Une vieille (datant de 2000) liste de petites distributions Linux. [<http://www.fokus.gmd.de/linux/linux-distrib-small.html>]
- tomsrtbt [<http://www.toms.net/rb>], « l'essentiel de Linux sur une disquette. » Tom est aussi lié à d'autres petites distributions.
- Le projet de documentation Linux [<http://www.tldp.org/>]. En particulier, jetez un coup d'œil à « LILO, Linux Crash Rescue HOW-TO ».
- Le programme de la Free Software Foundation parted [<http://www.gnu.org/software/parted>] pour éditer (agrandir, diminuer, déplacer) les partitions.
- QtParted [<http://qtparted.sourceforge.net/>] semble faire la même chose mais avec une interface graphique.
- Partition Image [<http://www.partimage.org/>] pour les sauvegardes des partitions.

De sa page eb : « Partition Image est un utilitaire Linux/UNIX qui sauvegarde des partitions d'un grand nombre de formats (voir plus bas) dans un fichier image. Le fichier image peut être comprimé au format GZIP/BZIP2 pour économiser de l'espace, et être découpé en plusieurs fichiers destinés à être copiés sur des supports amovibles (des disques ZIP par exemple) La partition peut être sauvegardée sur le réseau depuis la version 0.6.0. »

- Bacula [<http://sourceforge.net/projects/bacula>] est un produit de sauvegarde sous licence GPL comprenant du code pour effectuer une restauration intégrale de système, qui a été inspiré en partie par ce guide pratique.
- « g4u ghost pour unix ('ghost for unix') [<http://www.feyrer.de/g4u/>] est une disquette/CDROM de démarrage basée sur NetBSD qui permet de cloner facilement des disques durs de PC devant être déployés via FTP avec une configuration commune sur un certain nombre de PC. La disquette (ou le CDROM) a deux fonctions. La première est de transférer l'image comprimée d'un disque dur local vers un serveur FTP. L'autre est de restaurer cette image par FTP, la décompresser et la restaurer sur un disque; la configuration réseau est obtenu par DHCP. Le disque dur étant traité comme une image, n'importe quel système d'exploitation ou de fichiers peut être déployé à l'aide de g4u. »
- « Nous présentons Frisbee [<http://www.cs.utah.edu/flux/papers/frisbee-usenix03-base.html>], qui

est un système destiné à sauvegarder, transférer et installer les images de disques entiers, avec comme objectifs d'être rapide et extensible dans un environnement de réseau local. Techniquement, Frisbee utilise une méthode particulière de compression qui tient compte des systèmes de fichiers, un protocole maison de diffusion sélective de niveau application et un découpage de trames flexible de niveau application. De cette conception, il en résulte un système capable de distribuer rapidement et de façon fiable l'image d'un disque à de nombreux clients simultanés. Par exemple, Frisbee est capable d'écrire un total de 50 giga-octets de données sur 80 disques en 34 secondes, ceci sur des PC standards. Nous décrivons la conception et la mise en œuvre de Frisbee, examinons les décisions importantes de conception et évaluons ses performances. »

- Il y a un certain nombre de distributions disponibles sous forme de clés USB. Rendez-vous sur le site de DistroWatch [<http://www.distrowatch.com/>] pour plus de détails.
- Kits de secours basés sur CDROM. Ceci n'est pas une liste exhaustive. Si vous en connaissez une (ou du moins qui prétend en être une), faites-le moi savoir. [charlescurley CHEZ charlescurley POINT com] Vous trouverez des informations plus récentes sur DistroWatch [<http://www.distrowatch.com/>].
- Le Mondo [<http://www.microwerks.net/~hugo/>] de Hugo Rabson « ... crée un ou plusieurs CDROM (ou disquettes + bandes) amorçables de secours contenant tout ou partie de votre système de fichiers. En cas de perte catastrophique de données, vous pourrez restaurer intégralement. »
- Le kit de récupération après une panne irrécupérable (Crash Recovery Kit for Linux) [<http://crashrecovery.org/>]
- « Récupération de système avec Knoppix (System recovery with Knoppix) » [<http://www-106.ibm.com/developerworks/linux/library/l-knopp.html?ca=dgr-ln-xw04Knoppix>] est une bonne introduction à la récupération de système en général et dispose de liens Knoppix [<http://www.knoppix.org/>] utiles.
- « Cool Linux CD [<http://emergencycd2.sourceforge.net/>] est une version de Linux sur CDROM. Il est basé sur le noyau 2.4 et on y trouve des logiciels libres et de démonstration. »
- SystemRescueCd [<http://www.sysresccd.org/index.en.php>] « est un système linux monté sur un CDROM amorçable, destiné à réparer votre système et vos données après une panne irrécupérable. Il a aussi pour but de faciliter les tâches d'administration de votre ordinateur, telles que créer et éditer les partitions du disque dur. Il contient un bon nombre d'utilitaires système (parted, partimage, fstools, ...) et basiques (éditeurs, midnight commander, outils réseau). Il a pour objectif d'être facile à utiliser : démarrez simplement avec le CDROM et vous pourrez tout faire. Son noyau supporte les principaux systèmes de fichiers (ext2/ext3, reiserfs, xfs, jfs, vfat, ntfs, iso9660) et réseaux (samba et nfs). »
- Syslinux [<http://syslinux.zytor.com/>] crée le code de démarrage pour des images de disquettes, de CDROM et de PXE (Environnement de pré-exécution) Intel. Il ne dépend pas d'une image de disquette. Vous pourrez créer vos propres CDROM à l'aide de certains outils, tels que tomsrtbt [<http://www.toms.net/rb>].
- Au cas où voudriez vous débrouiller tout seul : « Linux Live [<http://www.linux-live.org/>] est un ensemble de scripts bash qui vous permet de créer votre propre LiveCD à partir de n'importe quelle distribution Linux. Installez simplement votre distribution favorite, enlevez tous les fichiers superflus (les pages de manuel, par exemple, et tous les autres fichiers que vous n'estimez pas importants). »
- « Le CDROM PPART [<http://www.linbox.com/en/ppart.html>] vous permet de générer un CD amorçable de récupération de système à partir de disques durs précédemment sauvegardés. »
- Timo's Rescue CD Set (Le CD de secours de Timo) [<http://rescuecd.sourceforge.net/>] : « Cette boîte à outils constitue mon approche pour générer facilement un CD amorçable de sauvegarde de système, qui peut aisément être adapté à vos besoins. Le projet est en train de se transformer en une distribution "debian sur CDROM", ce qui fait qu'il n'est pas seulement

utilisable comme CD de secours mais que l'on peut aussi installer une debian complète sur le CD. »

- La liste des distributions basées sur CDROM
[<http://www.frozentech.com/content/livecd.php>] comprend plus de distributions basées sur CDROM.

A. License GNU Free Documentation

Version 1.1, March 2000

Copyright (C) 2000 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBULE

The purpose of this License is to make a manual, textbook, or other written document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you".

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (For example, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, whose contents can be viewed and edited directly and straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup has been designed to thwart or discourage subsequent modification by readers is not Transparent. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML designed for human modification. Opaque formats include PostScript, PDF, proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies of the Document numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a publicly-accessible computer-network location containing a complete Transparent copy of the Document, free of added material, which the general network-using public has access to download anonymously at no charge using public-standard network protocols. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has less than five).
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section entitled "History", and its title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. In any section entitled "Acknowledgements" or "Dedications", preserve the section's title, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section as "Endorsements" or to conflict in title with any Invariant Section.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the

Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections entitled "History" in the various original documents, forming one section entitled "History"; likewise combine any sections entitled "Acknowledgements", and any sections entitled "Dedications". You must delete all sections entitled "Endorsements."

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, does not as a whole count as a Modified Version of the Document, provided no compilation copyright is claimed for the compilation. Such a compilation is called an "aggregate", and this License does not apply to the other self-contained works thus compiled with the Document, on account of their being thus compiled, if they are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one quarter of the entire aggregate, the Document's Cover Texts may be pla-

ced on covers that surround only the Document within the aggregate. Otherwise they must appear on covers around the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License provided that you also include the original English version of this License. In case of a disagreement between the translation and the original English version of this License, the original English version will prevail.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/> [<http://www.gnu.org/copyleft/>].

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

11. How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright (c) YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have no Invariant Sections, write "with no Invariant Sections" instead of saying which ones are invariant. If you have no Front-Cover Texts, write "no Front-Cover Texts" instead of "Front-Cover Texts being LIST"; likewise for Back-Cover Texts.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.
