

Best  
of  
EYROLLES

JEAN-FRANÇOIS BOUCHAUDY  
GILLES GOUBET

# Linux

# Administration

© Groupe Eyrolles 2007,  
ISBN : 978-2-212-12074-5

EYROLLES

# 6

- *Sauvegarde partielle, incrémentale*
- *Les commandes tar, cpio, dd*
- *Le format des bandes*
- *dump, restore*

## ***La sauvegarde***

### **Objectifs**

Après l'étude du chapitre, le lecteur connaît les stratégies de sauvegarde sous Linux et maîtrise les outils standard de sauvegarde.

### **Contenu**

Les outils de sauvegarde  
La sauvegarde par tar  
La sauvegarde par cpio, par pax  
La sauvegarde par dd  
Partimage  
Sauvegarde incrémentale  
Sauvegarde en réseau  
Sauvegarde Bare-Metal  
Les bandes

### **Références**

HOWTO : Ftape-HOWTO  
Le manuel : tar(1), dd(1), cpio(1), dump(8), restore(8)  
Livre : *Unix Backup & Recovery*, par W. Curtis Preston.  
Internet : <http://www.backupcentral.com>

# Les outils de sauvegarde

- Sauvegarde de fichiers : tar, cpio, pax (iso)
- Sauvegarde de disques ou de systèmes de fichiers : dd (iso)
- Sauvegarde d'images de FS : partimage
- Sauvegarde incrémentale de systèmes de fichiers :
  - Ext2/Ext3 *dump/restore*, XFS *xfsdump/xfsrestore*
- Sauvegarde Bare-Metal : Mondo
- Les scripts
- Les logiciels en client-serveur : Arkeia, Amanda, ...
- Sauvegarde en réseau : tar+dd+rsh ou ssh, rsync, ...

---

## Introduction

Il existe différents types de sauvegarde et le système Linux offre un ensemble de commandes dans lequel l'administrateur du système peut choisir la plus adaptée à son contexte applicatif.

---

### ATTENTION !

S'il n'y a qu'une chose à retenir en ce qui concerne les sauvegardes, c'est le conseil suivant : IL FAUT TESTER SES SAUVEGARDES !

---

## Panorama des outils de sauvegarde

### ***Sauvegarde de fichiers et d'arborescence de fichiers***

Les commandes tar, cpio et pax (ISO) permettent de sauvegarder des fichiers et des arbres d'un utilisateur ou d'une application. Bien que ces commandes ne voient que l'arborescence unifiée des systèmes de fichiers, il est possible que l'arbre sauvegardé coïncide avec le système de fichiers d'un disque si le chemin de sauvegarde est le chemin de montage, et s'il n'y a pas de systèmes de fichiers montés en dessous de celui à sauvegarder.

### ***Sauvegarde incrémentale***

La sauvegarde incrémentale consiste à sauvegarder, une première fois, la totalité d'une arborescence et de sauvegarder ensuite, d'une fois à l'autre, les seuls fichiers qui ont été modifiés. Les commandes de sauvegarde incrémentale sauvegardent les fichiers et des caractéristiques spécifiques du système de fichiers utilisé, comme par exemple les ACL. Ces commandes sont en conséquence spécifiques du système de fichier. Le système Linux utilise les commandes dump et restore pour sauvegarder des systèmes de

fichiers Ext2 ou Ext3. Il utilise les commandes `xfsdump` et `xfsrestore` pour sauvegarder des systèmes de fichiers XFS.

### **Sauvegarde physique de disques et de systèmes de fichiers**

La commande `dd` (ISO), qui réalise des copies physiques, peut parfois être utilisée pour sauvegarder des disques et des systèmes de fichiers.

### **Sauvegarde d'images de systèmes de fichiers**

La commande `partimage` réalise la sauvegarde physique d'une partition. A la différence de la commande `dd`, elle ne sauvegarde que les blocs occupés. Cette commande fonctionne avec la majorité des systèmes de fichiers Linux (Ext2/Ext3, reiserfs, XFS, ...) et Windows (vfat, ntfs, ...).

### **Sauvegarde Bare Metal**

Une sauvegarde Bare Metal consiste en la sauvegarde complète d'un système. Cette sauvegarde permet la réinstallation d'un système immédiatement opérationnel. On peut réaliser ce type de sauvegarde manuellement ou bien en s'aidant de scripts ou encore en utilisant un logiciel complet comme Mondo.

### **Les scripts**

L'administrateur d'un système Linux va souvent écrire des scripts de sauvegarde et de restauration de fichiers adaptés à son application et, quand c'est possible, automatiser leur exécution à heure fixe, quand le système est peu chargé et que les fichiers ou les systèmes de fichiers sont au repos.

### **Les logiciels client-serveur**

Quand les fichiers de données sont très volumineux, qu'ils sont distribués sur un réseau composé de systèmes hétérogènes (NT, AS400, HP-UX, SOLARIS, AIX ...), l'administrateur a sûrement intérêt à s'orienter vers un progiciel de sauvegarde en client-serveur. Cela lui assure une homogénéisation des opérations et lui procure des outils plus élaborés avec des gestions de catalogues et d'automates de sauvegarde.

La plupart des outils de sauvegarde du commerce fonctionnent sous Linux : BrightStor ARCserve (CA), Networker (Legato), Time Navigator (Atempo), Tivoli Storage Manager et Storix (IBM).

Le logiciel Arkeia (<http://www.arkeia.com>) est disponible gratuitement dans une version limitée à un serveur et deux clients.

Le logiciel libre Amanda, quoique de manipulation complexe, est également une solution.

### **Sauvegarde en réseau**

Les progiciels de sauvegarde ne sont pas les seuls outils à permettre une sauvegarde réseau. Les commandes comme `tar`, `dump/restore` ou `partimage` peuvent être utilisées en réseau. La commande `tar` peut directement fonctionner en réseau, on peut aussi l'utiliser en l'associant aux commandes `dd` et `rsh` ou `ssh`. La commande `rsync` est bien adaptée à la synchronisation de répertoires sur plusieurs machines, ce qui est une forme élémentaire de sauvegarde

### **Autres commandes de sauvegarde**

Un avantage ou un désavantage du monde libre, c'est qu'il offre souvent une pléthore de solutions. Dans le panorama précédent, de nombreux logiciels, souvent très intéressants dans certaines situations d'exploitations, ont été omis. La plupart sont décrits sur le site [linug.org](http://linug.org). En voici quelques uns parmi les plus importants : `dar`, `star`, `hostdump.sh`.

## Les autres commandes utiles pour les sauvegardes

<code>mt</code>	Permet de se positionner sur une bande magnétique.
<code>touch</code>	Met la date et l'heure de dernière modification d'un fichier à la date et à l'heure actuelle, ce qui permet de forcer la sauvegarde si elle est incrémentale.
<code>find</code>	Sélectionne les fichiers à sauvegarder. Son usage est le plus souvent joint à celui de <code>cpio</code> .
<code>du</code>	Permet de connaître la taille d'une arborescence.
<code>df</code>	Permet de connaître la taille des systèmes de fichiers.
<code>gzip</code> et <code>gunzip</code> <code>bzip2</code> et <code>bunzip2</code> <code>compress</code> et <code>uncompress</code> <code>zcat</code>	Compressent ou décompressent des fichiers. Ces commandes sont souvent utilisées avec <code>tar</code> et <code>cpio</code> . Les pages du manuel sont au format <code>gzip</code> . La commande <code>tar</code> possède des options pour sauvegarder les fichiers compressés par <code>gzip</code> , <code>bzip2</code> ou <code>compress</code> .

## Plan de sauvegarde

La sauvegarde nécessite de l'administrateur du système Linux beaucoup de discipline et une grande rigueur.

Sans prétendre donner la liste exhaustive des questions à résoudre pour construire un plan de sauvegarde, la liste qui suit peut aider l'administrateur à le définir :

- Que faut-il sauvegarder ?
- Avec quelle fréquence ?
- Combien de temps conservera-t-on les sauvegardes, à quel endroit, en combien d'exemplaires ?
- A quel endroit sera stocké l'historique des sauvegardes ?
- Quel est le support le plus approprié ?
- Quels sont les besoins, en capacité, de support de sauvegarde ?
- Combien de temps, au plus, doit durer la sauvegarde ?
- Combien de temps prévoit-on pour restaurer un fichier, un système de fichiers, un système complet, est-ce raisonnable ?
- La sauvegarde doit-elle être automatique ou manuelle ?
- Quelle est la méthode de sauvegarde la plus appropriée ?

## Références

### Livre

*Unix Backup & Recovery*, par W. Curtis Preston.

### Internet

Panorama des logiciels de sauvegarde Linux

<http://www.linux.org/apps/all/Administration/Backup.html>

Site dédié à la sauvegarde

<http://www.backupcentral.com/>

Site dédié à la sauvegarde sous Linux (exemples de commandes, conseils, ...)

<http://www.linux-backup.net/>

Un script de sauvegarde

<http://www.biochemistry.unimelb.edu.au/pscotney/backup/Backup-HOWTO.html>

Linux Backup Mini-Faq (medium, conseils, script, ...)

<http://kmsself.home.netcom.com/Linux/FAQs/backups.html>

Arkeia

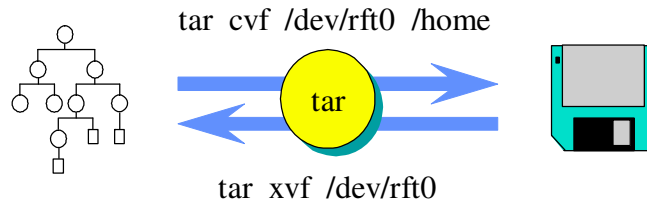
<http://www.arkeia.com>

Amanda

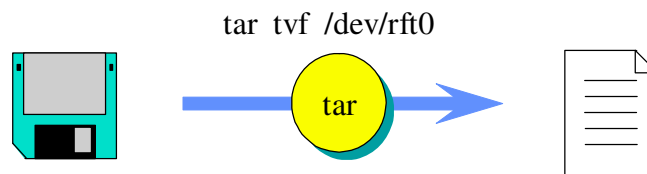
<http://www.amanda.org/>

## La sauvegarde par tar

### ■ Sauver/Restaurer



### ■ Lister l'archive



---

## La commande tar

La commande `tar` sauvegarde des fichiers, y compris une arborescence de fichiers, dans un format qui lui est propre. C'est avec la même commande qu'il est possible d'obtenir la liste des fichiers sauvegardés et de les restituer.

La commande `tar` de Linux utilise des options de forme courte, -lettre, ou longue, --mot\_clé. Pour conserver une compatibilité avec les autres systèmes UNIX, la commande `tar` permet aussi l'emploi de clés positionnelles. A la différence d'une option, une clé est une lettre qui n'est pas précédée du symbole « - » et l'association des arguments aux clés est positionnelle. La création d'une sauvegarde peut être demandée de plusieurs formes :

`tar c ...`

`tar -c ...`

`tar -create ...`

---

### Remarque

Quand on utilise des options avec argument, l'argument suit immédiatement l'option. Quand on utilise des clés, les clés sont groupées et on les fait suivre de leurs arguments.

Ainsi :

`$ tar cvkf 2000 /dev/rmt0`

`$ tar -cvk 2000 -f /dev/rmt0`

---

Dans toutes les syntaxes qui suivent, la clé `v` (verbose) génère des messages détaillés sur les opérations réalisées.

La clé `f` précise, par son argument FichierArchive, le nom du support d'archivage, souvent un périphérique (/dev/xxx). A défaut, la commande `tar` de Linux envoie la sauvegarde (c) ou la restitution (x) sur la sortie standard..

On peut toutefois explicitement utiliser, comme dans UNIX, le symbole « - » pour désigner respectivement l'entrée et la sortie standard comme support de sauvegarde ou de restitution.

La variable d'environnement `TAPE` permet d'indiquer le fichier *FichierArchive*.

La commande `tar` de Linux réalise, par défaut, des sauvegardes relatives. Elle supprime, sur l'archive, le premier caractère, « . » ou « / », des chemins d'accès aux fichiers. Dans tous les cas, il convient de se déplacer (commande `cd`) dans le répertoire qui sert de racine pour la restitution avant d'exécuter la commande `tar`.

Pour exécuter une sauvegarde et, plus tard, une restauration en absolu, il faut obligatoirement indiquer l'option « P ». A défaut de la mentionner à la restauration, la commande `tar` fonctionne en relatif.

La commande `tar` sauvegarde les fichiers au format GNU TAR. Il est compatible avec le format `ustar` de POSIX, également adopté par l'ISO. La commande `pax` de WINDOWS NT et le logiciel WINZIP peuvent aussi le lire.

---

### Remarque

Quand la commande `tar` est exécutée par un utilisateur qui n'est pas `root`, la commande utilise la logique de la commande `cp` ; pour restituer un fichier, il faut avoir le droit de le lire et l'utilisateur qui le restitue en devient le propriétaire. Quand c'est `root` qui effectue la restitution, il préserve les attributs des fichiers, dont le propriétaire et le groupe.

---

## Sauvegarde de fichiers

`tar c[v][P][f FichierArchive] Fichier ...`

Si les fichiers sont des répertoires, `tar` sauvegarde l'arborescence issue de ces répertoires.

## Liste des fichiers sauvegardés

`tar t[v][f FichierArchive]`

## Restitution de fichiers

`tar x[v][P][f FichierArchive] [fichier]...`

A défaut de dire quels sont les fichiers à restituer, `tar` restitue la totalité de l'archive.

Quand cela est nécessaire, la commande `tar` crée les répertoires manquants pour reconstituer une arborescence.

La variable d'environnement `TAPE` permet d'indiquer le fichier *FichierArchive*.

## Ce qu'il faut savoir faire

L'exemple qui suit réalise la sauvegarde de l'arborescence */home*. Le chemin de sauvegarde est donné en absolue, on note que la commande `tar` a enlevé le `/` en tête du chemin.

### Sauvegarde des fichiers

```
# tar cvf /dev/rft0 /home
```



**Liste des fichiers archivés**

```
# tar tvf /dev/rft0
drwxr-xr-x root/root                0   1999-01-21 15:35 home/
drwxr-xr-x root/root                0   1998-12-01 16:10 home/ftp/
d--x--x--x root/root                0   1998-12-01 16:09 home/ftp/bin/
---x--x--x root/root            15236 1998-04-27 18:05
home/ftp/bin/compress
drwxr-xr-x pierre/pierre            0   1999-03-16 09:52 home/pierre/
-rw-r--r-- pierre/pierre          222 1999-01-21 11:49
home/pierre/.bash_profile
```

**Restauration de l'ensemble des fichiers**

```
# cd /
# tar xvf /dev/rft0
```

**Restauration d'un fichier**

Restauration, par root, du fichier *fiche* qui se trouvait dans le répertoire */home/pierre*.

```
# rm /home/pierre/fiche
# cd /
# tar xvf /dev/rft0 home/pierre/fiche
home/pierre/fiche
# ls -l /home/pierre/fiche
-rw-r--r--  1 pierre  pierre          736 nov  7 10:58
/home/pierre/fiche
```

Pierre en est bien le propriétaire.

**Quelques options (clés) utiles**

La commande `tar` comporte de multiples clés et nous renvoyons le lecteur au manuel de référence pour mesurer l'étendue des possibilités. La liste qui suit mentionne quelques clés pratiques et utiles.

Clé	Utilisation
r,-r	Les fichiers sont ajoutés à la fin de l'archive. Cette option ne fonctionne que si le support d'archivage est à accès direct, un disque. Cela exclut l'utilisation de cette clé pour les cartouches et les bandes magnétiques.
u,-u	Le fichier est ajouté à la fin de l'archive s'il a été modifié depuis la dernière sauvegarde. La remarque précédente vaut également pour cette option.
b n,-b n	L'option <code>b</code> a comme argument le facteur de blocage qui détermine la taille du bloc à utiliser sur le support d'archivage. La valeur <code>n</code> définit le nombre de blocs de 512 octets du bloc d'archivage. Cette option est seulement utilisée avec des périphériques en mode caractère (« <i>raw device</i> »). Sa valeur par défaut est de 20 (10 Ko).
h,-h	L'option <code>h</code> indique à la commande <code>tar</code> de sauvegarder les fichiers liés et pas les liens symboliques (fichiers de type <code>l</code> ).
z,-z	Les commandes de compression <code>gzip</code> et de décompression <code>gunzip</code> sont utilisées pour la sauvegarde et la restitution.
Z,-Z	Les commandes de compression <code>compress</code> et de décompression <code>uncompress</code> sont utilisées pour la sauvegarde et la restitution.
-j, -I,	Les commandes de compression <code>bzip2</code> et de décompression <code>bunzip2</code>

Clé	Utilisation
--bzip2	sont utilisées pour la sauvegarde et la restitution.
M,-M	L'option -M est obligatoire pour que tar fonctionne en multi-volume.
-L n	L'option provoque le changement de support d'archivage au bout de n kilo octets.
-T	A l'option -T est associé le nom d'un fichier qui contient la liste des fichiers à sauvegarder. quand on utilise l'option « -T », les autres options doivent aussi être précédées de « - ». Si l'on associe l'argument « - » à l'option « -T », la commande tar utilise l'entrée et la sortie standard. Elle se comporte alors comme la commande cpio.
-P	La sauvegarde et la restitution sont réalisées en absolue.
-m	La date de dernière modification n'est pas restituée. Le fichier conserve celle qu'il a sur disque.
-w	La commande tar est interactive. On doit indiquer, pour chaque fichier, s'il doit être pris en compte.
-C <répertoire>	Le répertoire sert de répertoire racine lors d'une restauration relative.
--atime-preserve	La date de dernier accès n'est pas modifiée lors de la sauvegarde.
--exclude <fichier>	Le fichier n'est pas pris en compte au moment de la sauvegarde.

### Remarque

Les fichiers que l'on rapatrie à partir de serveurs de fichiers sur Internet sont souvent fournis au format tar et compressés (\*.tar.gz).

## Pour aller plus loin

### Exemple de sauvegarde réalisée en absolue.

Sauvegarde des fichiers de du répertoire /home/pierre.

```
# tar cvfP /dev/rft0 /home/pierre
/home/pierre/.kderc
/home/pierre/.screenrc
/home/pierre/fiche
/home/pierre/rep/
/home/pierre/rep/fic2
# tar tvf /dev/rft0
-rw-r--r-- pierre/pierre 435 2000-10-13 05:26:03 /home/pierre/.kderc
-rw-r--r-- pierre/pierre 3394 2000-10-13 05:26:03
/home/pierre/.screenrc
-rw-r--r-- pierre/pierre 736 2000-11-07 10:58:53 /home/pierre/fiche
drwxrwxr-x pierre/pierre 0 2000-11-07 10:59:38 /home/pierre/rep/
-rw-r--r-- pierre/pierre 1756 2000-11-07 10:59:13
/home/pierre/rep/fic2
```

**Restauration du fichier /home/pierre/fiche**

```
# pwd # la restauration peut être réalisée de n'importe quel point.
/root
# tar xvfP /dev/rft0 /home/pierre/fiche
# ls -l /home/pierre/fiche
-rw-r--r--  1 pierre  pierre    736 nov  7 10:58 /home/pierre/fiche
```

**Exemple d'utilisation de l'entrée et de la sortie standard comme support d'archivage pour sauvegarder avec la commande bzip2**

```
# tar cv /home/pierre | bzip2 > /dev/rft0
```

Liste des fichiers.

```
# bunzip2 < /dev/rft0 | tar tv # ou bunzip2 < /dev/rft0 | tar tv -
```

**Sauvegarde du lien symbolique**

```
# ls -l /home/pierre/passwd
lrwxrwxrwx  1 pierre  pierre    11 nov  7 12:29
/home/pierre/passwd -> /etc/passwd
# tar cvf /dev/st0 /home/pierre
# tar tvf /dev/st0
lrwxrwxrwx pierre/pierre    0 2000-11-07 12:29:13 home/pierre/passwd
-> /etc/passwd
```

**Sauvegarde du fichier lié**

```
# tar cvfh /dev/st0 /home/pierre
# tar tvf /dev/st0
-rw-r--r--  root/root        773 2000-11-07 11:13:58 home/pierre/passwd
```

**Sauvegarde sans modification de la date de dernier accès et restitution sans modification de la date de dernière modification**

```
# ls -ul fiche # attributs avant la sauvegarde
-rw-r--r--  1 pierre  pierre    736 nov  7 10:00 fiche
# ls -l fiche
-rw-r--r--  1 pierre  pierre    736 nov  7 10:00 fiche
# tar --atime-preserve -cvf /dev/st0 fiche
fiche
# ls -ul fiche
-rw-r--r--  1 pierre  pierre    736 nov  7 10:00 fiche
# ls -l fiche
-rw-r--r--  1 pierre  pierre    736 nov  7 12:58 fiche
# tar xvf /dev/st0 fiche
fiche
# ls -l fiche
-rw-r--r--  1 pierre  pierre    736 nov  7 12:59 fiche
```

**Sauvegarde à la manière de cpio**

```
# find /home | tar -T - -cvf /dev/ftape
```

**Restitution dans un autre arbre que l'arbre courant**

```
# pwd
/root
# tar -C /home/gilles -xvf /dev/st0 fiche
fiche
# ls -l /home/gilles/fiche
-rw-r--r--  1 pierre  pierre    736 nov  7 10:00 fiche
```

**Sauvegarde qui exclut certains fichiers (/home/ftp et /home/httpd)**

```
# tar --exclude /home/ftp --exclude /home/httpd -cf /dev/st0 /home
```

**Sauvegardes compressées**

```
# tar cfz /tmp/hometar.zip /home           # format zip
tar: Removing leading `/' from member names
# tar cf /tmp/hometar /home                 # sans compression
tar: Removing leading `/' from member names
# tar cfI /tmp/hometar.bzip /home           # compression bzip2
tar: Removing leading `/' from member names
# ls -l /tmp/home*                         # taille des fichiers de sauvegarde
-rw-r--r--  1 root    root      5519360 nov  7 15:51 /tmp/hometar
-rw-r--r--  1 root    root      2338109 nov  7 15:52
/tmp/hometar.bzip
-rw-r--r--  1 root    root      2428119 nov  7 15:50
/tmp/hometar.zip
# file /tmp/home*                          # type des fichiers
/tmp/hometar:      GNU tar archive
/tmp/hometar.bzip: bzip2 compressed data, block size = 900k
/tmp/hometar.zip:  gzip compressed data, deflated, last modified: Tue
Nov  7 15:
50:33 2000, os: Unix
```

**Références****Man**

tar(1)

**Info**

info tar

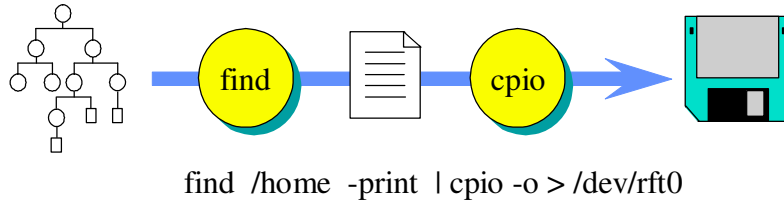
**Internet**

Le site officiel de la commande tar du GNU (documentation, ...)

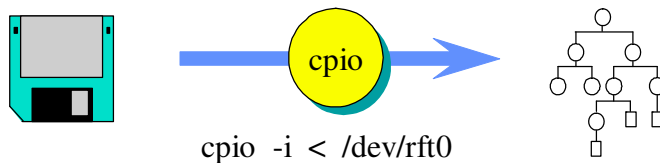
<http://www.gnu.org/software/tar>

# La sauvegarde par cpio et par pax

## ■ Sauver/Restaurer



## ■ Restaurer



## ■ Lister l'archive

```
cpio -itv < /dev/rft0
```

## La commande cpio

La commande `cpio` sauvegarde sur la sortie standard les fichiers dont on saisit les noms sur l'entrée standard, par défaut le clavier et l'écran. La redirection des entrées et des sorties permet d'automatiser la production des noms de fichiers et de désigner le support d'archivage qu'il faut réellement utiliser.

Les options `-v`, `-c` et `-B` peuvent être utilisées dans tous les cas.

L'option `-v` demande, comme dans `tar`, l'affichage d'informations détaillées.

L'option `-c` signifie à `cpio` de mémoriser les attributs des fichiers sous une forme ASCII, ce qui facilite l'échange de fichiers entre systèmes UNIX hétérogènes.

L'option `-B` augmente la vitesse d'exécution de `cpio`, qui utilise une mémoire tampon d'entrée et de sortie de 5120 octets.

La commande `cpio` est multi-volume. A chaque fois qu'une archive est pleine, la commande `cpio` demande l'installation d'une nouvelle archive. L'administrateur doit, pour continuer la sauvegarde ou la restitution, saisir le nom du fichier spécial périphérique et valider par <Entrée>. Cela permet de changer de périphérique de sauvegarde pendant l'exécution de la commande `cpio`.

### **Sauvegarde de fichiers**

```
cpio -o[L]
```

L'option `-L` demande à `cpio` de sauvegarder les fichiers liés et non les liens symboliques.

### **Liste des fichiers sauvegardés**

```
cpio -it
```

### **Restitution de fichiers**

```
cpio -i[umd] [Fichier...]
```

A défaut de dire quels sont les fichiers à restituer, `cpio` restitue la totalité de l'archive. Si les noms de fichiers que l'on désire restituer sont décrits avec des jokers (\*,?,[...]), ils doivent être protégés (\, ', ").

L'option `-d` indique à `cpio` de reconstruire les sous-répertoires manquants. C'est une option quasiment obligatoire en restauration. Quand on indique les fichiers à restaurer et que ceux-ci sont des répertoires, il faut explicitement mentionner \*, car la commande `cpio`, à la différence de `tar`, créera le répertoire mais ne restaurera pas sinon son contenu.

```
# cpio -icd 'rep/*' < /dev/xxx
```

L'option `-m` indique à `cpio` de conserver les dates de dernière modification pour les fichiers restaurés.

L'option `-u` demande à la commande `cpio` de restaurer les fichiers inconditionnellement. A défaut, les fichiers qui existent toujours sur disque ne sont pas restaurés s'ils sont plus récents ou de même âge que leurs homonymes sur disque.

### Quelques options utiles des commandes `find`, `cpio` et `touch`

Les commandes `find`, `cpio` et `touch` comportent de multiples options et nous renvoyons le lecteur au manuel de référence. La liste qui suit mentionne quelques options utiles des deux commandes.

Commande	Option	Utilisation
find	-newer fichier	Vrai si la date de dernière modification du fichier est plus récente que « fichier ». Cela permet de définir un fichier de référence dont on peut éventuellement forcer la date de dernière modification.
	-mount ou -xdev	Cette option restreint la recherche au système de fichiers auquel appartient l'arborescence.
	-H format	L'option H précise le format que la commande <code>cpio</code> doit utiliser pour les fichiers. bin      format binaire, utilisé par défaut. ustar    format POSIX (IEEE/P1003). Il reconnaît le format GNU TAR tar      format TAR. Le manuel de référence de la commande <code>cpio</code> décrit les autres formats.
touch		La commande <code>touch</code> force la date de dernière modification à la date de l'exécution de la commande. On peut préciser une date particulière. Cette commande est utile pour forcer la sauvegarde d'un fichier qui n'a pas été modifié.

### Exemples

Sauvegarder des fichiers sur disquette.

```
# cpio -ocvB > /dev/fd0
f1
/etc/group
Ctrl-D
10 blocks
```

Sauvegarder l'arborescence courante avec compression.

```
# find . -print | cpio -ocvB | compress > /dev/rft0
```

Sauvegarder tous les fichiers des utilisateurs modifiés depuis la modification de fref.

```
# find /home -type f -newer fref | cpio -ocvB > /dev/rft0
```

Sauvegarder le système de fichiers /usr.

```
# find /usr -mount | cpio -ocvB >/dev/rft0
```

Sauvegarde de l'arborescence /home dans un fichier.

```
# find /home | cpio -ocvB > /tmp/home.cpio
```

```
/home
/home/lost+found
/home/pierre
/home/pierre/.kde/Autostart/.directory
/home/gilles/fic_cron
/home/gilles/.bash_history
/home/aquota.user
4 blocks
```

Liste de l'archive.

```
# cpio -itcvB < /tmp/home.cpio
```

```
drwxr-xr-x  5 root      root              0 Mar 24 11:34 /home
drwx-----  2 root      root              0 Mar  3 11:57
/home/lost+found
drwx-----  3 pierre    pierre            0 Mar 24 11:38 /home/pierre
drwxr-xr-x  3 pierre    pierre            0 Mar 24 11:34
/home/pierre/.kde
drwxr-xr-x  2 pierre    pierre            0 Mar 24 11:34
/home/pierre/.kde/Autostart
```

Suppression du répertoire de pierre et restauration.

```
# rm -rf /home/pierre
```

```
# cpio -icvB /home/pierre < /tmp/home.cpio
```

```
/home/pierre
```

```
4 blocks
```

Le répertoire a été restauré mais il est vide.

```
# ls -l /home/pierre
```

```
total 0
```

```
# rm -rf /home/pierre
```

```
# cpio -icvB '/home/pierre/*' < /tmp/home.cpio
```

```
/home/pierre/.kde
/home/pierre/.kde/Autostart
/home/pierre/.kde/Autostart/Autorun.desktop
/home/pierre/.kde/Autostart/.directory
4 blocks
```

## La commande pax de l'ISO

La commande pax est une nouvelle commande de sauvegarde qui recouvre les commandes tar et cpio dont elle emprunte, selon les arguments, la logique de fonctionnement de l'une ou de l'autre. L'intérêt principal de la commande pax est d'autoriser la sauvegarde des fichiers dans l'un des deux formats, tar ou cpio, tar par défaut, et de détecter automatiquement le format d'une bande à restaurer.

La commande pax permet d'affiner les paramètres de restauration des fichiers quant aux droits et aux attributs (UID, ...).

Elle propose quatre possibilités :

pax -w ... Pour sauvegarder.

pax -r ... Pour restaurer.

`pax -rw ...` Pour copier un arbre (*cf. cpio -p*).

`pax ...` Pour lister le contenu de l'archive.

Si les noms de fichiers à sauvegarder ne figurent pas dans la ligne de commandes, ils sont lus depuis l'entrée standard, comme pour `cpio`. De même, l'archive peut être précisée en argument de l'option `-f`, *comme pour la commande tar* ou à défaut être respectivement la sortie standard pour une sauvegarde ou l'entrée standard pour une restauration.

Sauvegarde de l'arborescence courante.

```
# pax -w -f /dev/rft0 .
```

Restauration dans le répertoire courant.

```
# pax -r -f /dev/rft0
```

Sauvegarde de l'arborescence courante au format `cpio`.

```
# pax -w -x cpio -f /dev/rft0 .
```

Restituer les fichiers en conservant tous les attributs.

```
# pax -r -p e -f /dev/rft0
```

L'argument « e » de l'option « -p » conserve l'UID, le GID, les droits, les dates de dernier accès et de dernière modification.

Sauvegarde à la manière de `cpio`.

```
# find /home | pax -w > /tmp/sauve.pax
```

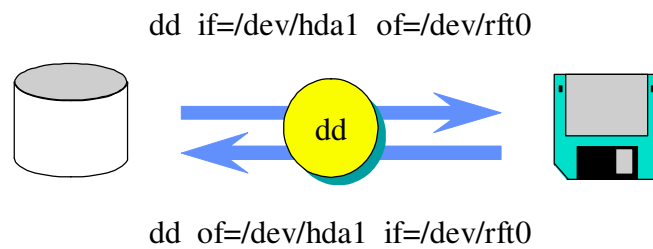
## Références

### *Man*

`cpio(1)`, `pax(1)`



# La sauvegarde par dd



if=fichier	Le fichier à copier
of=fichier	La copie
bs=n	La taille des blocs
skip=n	Nombre de blocs sautés en entrée
seek=n	Nombre de blocs sautés en sortie
count=n	Nombre de blocs copiés

## La commande dd

La commande `dd` (Device to Device) est destinée à la copie physique, bloc à bloc, d'un fichier périphérique vers un fichier périphérique. Elle a été originellement créée pour permettre la lecture ou l'écriture de bande magnétique. Sachant la logique du système UNIX, elle peut être employée pour n'importe quel fichier, spécial ou régulier, à condition que l'on dispose des droits nécessaires de lecture ou d'écriture. Elle dispose de nombreuses possibilités qui lui permettent de lire beaucoup de fichiers, à condition que l'on en connaisse le format. La commande `dd` peut être également utilisée pour réaliser la copie physique d'un disque, y compris d'un disque contenant un système de fichiers. Elle offre dans ce cas une grande facilité d'usage mais présente par contre trois inconvénients :

- La commande `dd` copie tous les blocs, y compris ceux qui ne sont pas utilisés.
- La commande `dd` ne gère pas les blocs défectueux.
- La commande `dd` n'est pas multi-volume. Cela peut être programmé dans un script.

Les arguments de la commande `dd` se présentent sous la forme :

Mot\_clé=valeur

« Mot\_clé » désigne une option de la commande `dd` et « valeur », son argument.

Argument	Utilisation
if=fichier	« fichier » désigne le fichier à copier, à défaut l'entrée standard. Cette dernière possibilité est utile quand l'entrée de <code>dd</code> vient d'une commande, de compression par exemple.
of=fichier	« fichier » désigne le fichier résultant de la copie, à défaut la sortie standard. Cette possibilité est utile pour envoyer la sortie de <code>dd</code> vers une commande, de compression par exemple.

<i>Argument</i>	<i>Utilisation</i>
<code>bs=valeur</code>	« valeur » désigne la taille commune du bloc pour les fichiers d'entrée et de sortie. Il est, par défaut, de 512 octets. A chaque fois que possible, il est conseillé de l'augmenter pour accélérer l'exécution de <code>dd</code> .
<code>ibs=valeur</code>	La taille du bloc du fichier d'entrée.
<code>obs=valeur</code>	La taille du bloc du fichier de sortie.
<code>skip=n</code>	<code>n</code> indique le nombre de blocs qu'il faut sauter au début de la copie du fichier d'entrée.
<code>seek=n</code>	<code>n</code> indique le nombre de blocs à sauter au début du fichier de sortie.
<code>count=n</code>	<code>n</code> indique le nombre de blocs à copier.
<code>conv=conversion</code> <code>[,conversion...]</code>	« conversion » indique la conversion à réaliser lors de la copie. Plusieurs conversions peuvent être groupées. Les principales valeurs possibles pour « conversion » sont : <code>lcase</code> , conversion en minuscule. <code>ucase</code> , conversion en majuscule. <code>ebcdic</code> , conversion en ebcidic (code ibm). <code>ascii</code> , conversion en ascii. <code>swab</code> , échange les octets.

## Exemples

Copie d'un disque et peut-être d'un système de fichiers.

```
# dd if=/dev/hda2 of=/dev/rft0 bs=5k
```

Restauration du disque et de son éventuel système de fichiers.

```
# dd if=/dev/rft0 of=/dev/hda2 bs=5k
```

Sauvegarde d'une arborescence, avec compression.

```
# find . -print | cpio -o | compress | dd of=/dev/rft0 bs=5k
```

Restauration de l'arborescence.

```
# dd if=/dev/rft0 bs=5k | uncompress | cpio -id
```

Sauvegarde d'un disque de 20 000 blocs sur deux cartouches.

```
# dd if=/dev/hda1 of=/dev/rft0 count=10000 #les 10000 premiers blocs
# dd if=/dev/hda1 of=/dev/rft0 skip=10000 # les 10 000 suivants
```

Restauration du disque de 20 000 blocs.

```
# dd if=/dev/rft0 of=/dev/hda1 count=10000
# dd if=/dev/rft0 of=/dev/hda1 seek=10000
```

Copie d'une disquette.

```
# dd if=/dev/fd0 of=/tmp/floppy
# le fichier de sortie est un fichier ordinaire
# dd if=/tmp/floppy of=/dev/fd0
```

Copie d'un bloc de boot d'un disque à un autre, pour un système de fichiers « s5 ».

```
# dd if=/dev/hda1 of=/dev/hda2 bs=1k count=1 skip=0
```

Lecture d'une bande au format IBM, contenant des images de cartes (80 caractères) avec dix cartes par bloc. L'option « cbs » donne la taille de l'enregistrement à l'intérieur d'un bloc.

```
# dd if=/dev/rft0 of=x ibs=800 cbs=80 conv=ascii,lcas
```

Sauvegarde du MBR

```
# dd if=/dev/hda of=/tmp/mbr count=1
```

## Références

### **Man**

dd(1)

### **Info**

Info dd

# Partimage

```

■ +-----| save partition to image file +-----+
|
| Compression level
| ( ) None (very fast + very big file)
| (*) Gzip (.gz: medium speed + small image file)
| ( ) Bzip2 (.bz2: very slow + very small image file)
|
| Options                                If finished sucessfully:
| [X] Check partition before saving      (*) Wait
| [ ] Enter description                  ( ) Halt
| [X] Overwrite without prompt           ( ) Reboot
|                                       ( ) Quit
|
| Image split mode
| ( ) Automatic split (when no space left)
| (*) Into files whose size is:..... 2086666_ Kilo-Bytes
| [ ] Wait after each volume change
|
| <Continue (F5)>      <Exit (F6)>      <Main window (F7)>
+-----+
initializing the operation.Please wait...[*to cancel,CtrlS, CtrlQ]

```

## Introduction

Partimage est un logiciel libre, sous licence GPL, qui permet la sauvegarde de l'image d'une partition. La sauvegarde résultante est de petite taille. En effet, partimage, contrairement à la commande `dd`, ne prend en compte que les blocs occupés et de plus les compresse.

Partimage supporte les systèmes de fichiers suivants : Ext2 et Ext3, Fat16, Fat32, NTFS, HPFS, ReiserFs, JFS, XFS, UFS, HFS (MacOS). Actuellement, les FS NTFS, UFS et HFS ne sont que partiellement supportés. On se rend heureusement compte du problème dès la sauvegarde.

Partimage peut sauvegarder une image dans un système de fichiers destiné à cet usage, comme celui d'une disquette ZIP ou un système de fichiers réseau comme NFS. Partimage peut également fonctionner en client serveur. Le paquetage comprend le client partimage ainsi que le serveur partimaged. La distribution Knoppix dispose de ces commandes.

## Exemple de sauvegarde locale sur ZIP

Dans l'exemple qui suit, on sauvegarde la partition `/home` sur un fichier stocké sur une disquette ZIP.

```

# mount /home
# df -h

```

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/hda1	1.4G	60M	1.2G	5%	/
/dev/hda3	1.4G	74M	1.2G	6%	/var
/dev/hda7	4.2G	200M	3.7G	5%	/usr
/dev/hda6	1.4G	33M	1.3G	3%	/home

```

# umount /home
# modprobe ppa
# mount -t vfat /dev/sda4 /mnt

```

```
# partimage -z1 -o -d save /dev/hda6 /mnt/home_image.gz
```

Après la validation de la commande précédente, un écran apparaît (*cf. la diapositive*), la sauvegarde commence après l'appui sur la touche F5.

L'argument `imginfo` permet de visualiser les caractéristiques d'une image.

```
# partimage imginfo /mnt/home_image.gz
```

L'argument `restore` provoque l'affichage d'une boîte de dialogue permettant la restauration d'une partition. Celle-ci commence après l'appui sur la touche F5.

```
# partimage restore /dev/hda6 /mnt/home_image.gz
```

## Exemple de sauvegarde réseau

### Activation du logiciel serveur

Dans l'exemple qui suit, on active le serveur `partimaged`. On spécifie en argument le répertoire où le logiciel stockera les images. Par défaut, le serveur se met à l'écoute sur le port TCP 4025.

```
herbizare:~# partimaged -D -d /usr/BCK
```

### Sauvegarde réseau d'une partition

On veut sauvegarder la partition `/dev/hda6` contenant un système de fichiers `ext3` sur le serveur `herbizarre`. La commande demande un nom et un mot de passe pour se connecter au serveur. Une boîte de dialogue s'affiche ensuite, qui permet de sélectionner des options (taux de compression, découpage en fichiers de taille fixe ...). L'appui sur la touche F5 valide ces choix. L'écran suivant demande de saisir un commentaire associé à la sauvegarde. Après validation, les paramètres du système de fichiers sont affichés. Après une dernière validation, la sauvegarde démarre.

```
carapuce:~# partimage -s herbizarre save /dev/hda6 bck_home.gz
```

### Restauration d'une partition

On veut restaurer la partition précédente.

```
carapuce :~# partimage -s herbizarre restore /dev/hda6 bck.gz.000
```

## Références

### Man

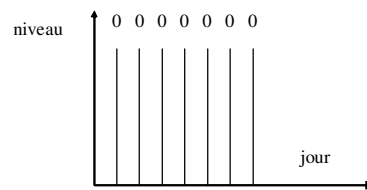
`partimage(1)`, `partimaged(8)`

### Internet

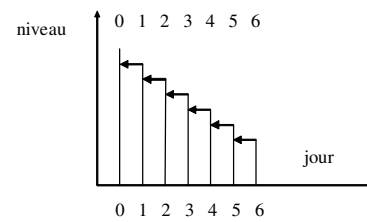
Le site officiel du logiciel Partimage (téléchargement, documentation, ...)

<http://www.partimage.org/>

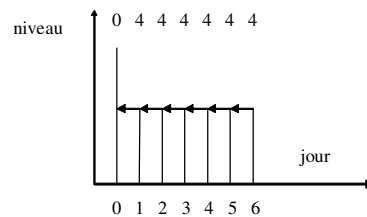
# Sauvegarde incrémentale



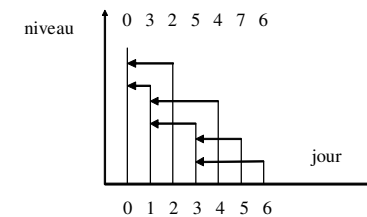
Sauvegarde non-incrémentale



Sauvegarde incrémentale



Sauvegarde différentielle



Sauvegarde « Tour de Hanoï »

## Introduction

Les commandes de sauvegarde incrémentale diffèrent profondément des commandes de sauvegarde de fichiers comme `tar` ou `cpio`. D'abord, elles ne font pas que sauvegarder les fichiers, elles sauvegardent également des attributs spécifiques du type de système de fichiers, comme par exemple les ACL. Le format de la sauvegarde est donc forcément spécifique de ce type. Il y a autant de commandes de sauvegarde incrémentale que de type de système de fichiers. Ces commandes se présentent sous forme d'un couple de commandes : l'une pour sauvegarder, l'autre pour restaurer. Enfin, ces commandes sont faites pour réaliser des sauvegardes incrémentales, c'est-à-dire qu'elles ne sauvegardent que les fichiers modifiés depuis une sauvegarde de référence.

Les sauvegardes incrémentales sont typiquement celles que l'on utilise chaque jour en automatique, via un `crontab`, pour sauvegarder son système.

### Panorama

Ext2, Ext3	<code>dump/restore</code>
XFS	<code>xfsdump/xfrestore</code>
Reiserfs, JFSPas de commandes pour le moment.	

### Caractéristiques des commandes de sauvegardes incrémentales

- Le fonctionnement incrémental de manière native.
- La sauvegarde de données spécifique du FS, comme les ACL.
- La conservation des dates de dernier accès.
- L'accès immédiat à la table du contenu de la sauvegarde (VTOC) lors d'une restauration.
- Un système de fichiers sauvegardé n'a pas besoin d'être monté durant la sauvegarde.

- Le format d'une sauvegarde est spécifique du type de FS.

## La logique des sauvegardes incrémentales

### Introduction

Une sauvegarde incrémentale consiste à ne sauvegarder que les modifications effectuées depuis une sauvegarde de référence.

Ce type de sauvegarde peut tout à fait être réalisée par des commandes comme `tar` ou `cpio`. L'intérêt d'utiliser des commandes incrémentales est quelles sont conçues pour cela. Ces commandes utilisent à la fois un niveau incrémental et un fichier jouant le rôle de journal de bord pour spécifier à chaque fois la sauvegarde de référence et par conséquent ce qu'il faut sauvegarder. L'étude sommaire de la commande `dump` et les exemples suivants vont, je l'espère, clarifier le concept.

---

### Remarque

La compréhension du principe des sauvegardes incrémentales est importante si l'on désire utiliser des logiciels en client/serveur comme Arkeia ou Amanda.

---

### Présentation sommaire de la commande `dump`

La commande `dump` sert à sauvegarder de manière incrémentale un système de fichiers (FS). Elle a essentiellement deux arguments : la clé qui spécifie les opérations effectuées et le FS. La clé est composée de plusieurs caractères notamment :

- Le niveau incrémental : c'est un nombre de 0 à 9.
- La lettre `u` entraîne la mémorisation de la sauvegarde dans le journal de bord `/etc/dumpdates` ou `/var/lib/dumpdates`.
- La lettre `f` qui stipule que l'argument suivant est, comme dans `tar`, le support d'archive.

### L'ordonnancement non incrémental

Le niveau 0 correspond à une sauvegarde non incrémentale, appelée également sauvegarde complète (full backup). Typiquement, on la réalise une fois par semaine.

```
Dimanche # dump 0uf /dev/ftape /home
```

### L'ordonnancement incrémental

L'utilisation d'un niveau différent de zéro réalise une sauvegarde incrémentale. On ne sauvegarde que les modifications effectuées depuis la dernière sauvegarde de niveau inférieur.

La stratégie la plus simple est d'augmenter chaque jour de la semaine le niveau et par conséquent de ne sauvegarder à chaque fois que les modifications de la journée.

```
Dimanche # dump 0uf /dev/ftape /home
Lundi    # dump 1uf /dev/ftape /home
Mardi    # dump 2uf /dev/ftape /home
...
Samedi   # dump 6uf /dev/ftape /home
```

En fin de semaine, le cycle de sauvegarde (Backup Cycle) est terminé. Le dimanche suivant, on fait de nouveau une sauvegarde de niveau zéro. Le cycle de rétention précise le nombre de cycles de sauvegarde que l'on conserve.

L'inconvénient majeur de la stratégie précédente est la nécessité de restaurer l'ensemble des volumes de sauvegarde dans le cas d'une restauration complète d'un FS.

**Remarque**

Si l'on ne met pas l'option `-u`, la sauvegarde n'est pas inscrite dans le journal. On peut utiliser cette astuce pour effectuer par exemple une sauvegarde complète en milieu de semaine, mais qui n'influence en rien le cycle de sauvegarde.

**L'ordonnancement différentiel**

Au lieu d'augmenter chaque jour le niveau, on peut utiliser toujours le même niveau (par exemple 4). Dans ce cas, on sauvegarde à chaque fois toutes les données modifiées depuis la sauvegarde de niveau inférieur qui n'est autre que la sauvegarde de niveau zéro. Cette variante simplifiée de sauvegarde incrémentale est souvent appelée sauvegarde différentielle.

```
Dimanche # dump 0uf /dev/ftape /home
Lundi    # dump 4uf /dev/ftape /home
Mardi    # dump 4uf /dev/ftape /home
...
Samedi   # dump 4uf /dev/ftape /home
```

L'inconvénient de cette stratégie est que chaque jour la sauvegarde prend plus de place et plus de temps. Par contre, si l'on a besoin de restaurer l'intégralité du FS, seulement deux volumes sont nécessaires, celui de la sauvegarde de niveau zéro et de la dernière sauvegarde de niveau 4.

Il faut dans tous les cas conserver toutes les sauvegardes intermédiaires pour pouvoir restaurer un fichier quelconque, par exemple un fichier créé le lundi et détruit le mardi.

**L'ordonnancement de la Tour de Hanoi**

L'ordonnancement de la Tour de Hanoi est le plus complexe. Il est utilisé par les professionnels de la sauvegarde. Il est basé sur l'algorithme du même nom « la Tour de Hanoi ». Ses objectifs sont de minimiser les temps de sauvegarde, les temps de restauration et de faire en sorte que chaque fichier soit sauvegardé au moins deux fois.

Voici la succession des niveaux :

- Pour un cycle d'une semaine : 0 3 2 5 4 7 6
- Pour un cycle de deux semaines : 0 3 2 4 3 5 4 6 5 7 6 8 7 9

**Le fonctionnement d'une commande incrémentale**

Voici les grandes étapes d'une sauvegarde incrémentale :

**Phase I**

Durant cette phase, la commande `dump` construit l'arbre des fichiers à sauvegarder. Elle se base à la fois sur le contenu du fichier `/etc/dumpdates`, du niveau incrémental et de la date de dernière modification des fichiers.

**Phase II**

La commande balaye l'arbre construit précédemment pour déterminer les répertoires à sauvegarder.

**Phase III**

La commande écrit l'entête de la sauvegarde et les inodes des répertoires à sauvegarder ainsi que les blocs de données de ces répertoires. Ces informations constituent la VTOC de la sauvegarde (Volume Table Of Content).

**Phase IV**

La commande sauvegarde ensuite les inodes et les blocs des fichiers à sauvegarder.



**Phase V**

La commande écrit une marque de fin d'archive.

**La sauvegarde de Ext2/Ext3 avec les commandes dump/restore****Syntaxe**

Voici la syntaxe simplifiée des commandes `dump` et `restore`. Nous renvoyons le lecteur au manuel (`dump(8)` et `restore(8)`) pour une étude exhaustive. L'utilisation en réseau de ces commandes est décrite dans le chapitre suivant : Sauvegarde en réseau.

**La commande dump**

`dump [-0123456789au] [-A fichier] [-B taille] [-b bs] [-F script] [-f fichier] FS`

-0123456789 C'est le niveau incrémental, par défaut 9.

-a Dump écrit jusqu'à la fin du media, c'est l'option par défaut.

-A fichier Dump stocke la VTOC dans un fichier.

-b bs Spécifie la taille des blocs, par défaut 10 Ko.

-F script Spécifie le script à exécuter à chaque changement de volume.

-f fichier Spécifie le support d'archive, par défaut le périphérique `/dev/st0`.

FS Le nom du système de fichiers à sauvegarder. C'est soit son chemin, soit le nom de la partition qui l'abrite.

La commande `dump` est sensible aux variables d'environnement suivantes :

TAPE Spécifie le support d'archive.

RMT Spécifie l'emplacement de la commande `rmt` distante.

RSH Spécifie la commande remote à utiliser lors d'une sauvegarde réseau (`rsh`, `ssh` ...).

**La commande restore**

`restore -t [-b fs] [-F script] [-f fichier]`

Liste la VTOC d'une sauvegarde

`restore -r [-b fs] [-F script] [-f fichier]`

Restaure l'intégralité d'une archive.

`restore -x [-b fs] [-F script] [-f fichier] [fichier ...]`

Restaure quelques fichiers.

`restore -i [-b fs] [-F script] [-f fichier]`

Restaure de manière interactive des fichiers.

`restore -C [-b fs] [-F script] [-f fichier] [-D FS]`

Compare le contenu d'une archive avec les données actuelles.

**Exemples****Sauvegarde complète du FS /home sur une disquette ZIP**

```
carapuce:~# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/hda1       1.4G   62M  1.2G   5% /
/dev/hda3       1.4G   75M  1.2G   6% /var
/dev/hda7       1.5G  204M  1.2G  14% /usr
/dev/hda6       1.4G   33M  1.3G   3% /home
carapuce:~# modprobe ppa
carapuce:~# dump 0uf /dev/sda4 /home
```

```

DUMP: Date of this level 0 dump: Thu Feb 12 19:17:14 2004
DUMP: Dumping /dev/hda6 (/home) to /dev/sda4
DUMP: Added inode 8 to exclude list (journal inode)
DUMP: Added inode 7 to exclude list (resize inode)
DUMP: Label: none
DUMP: mapping (Pass I) [regular files]
DUMP: mapping (Pass II) [directories]
DUMP: estimated 115 tape blocks.
DUMP: Volume 1 started with block 1 at: Thu Feb 12 19:17:15 2004
DUMP: dumping (Pass III) [directories]
DUMP: dumping (Pass IV) [regular files]
DUMP: Closing /dev/sda4
DUMP: Volume 1 completed at: Thu Feb 12 19:17:16 2004
DUMP: Volume 1 110 tape blocks (0.11MB)
DUMP: Volume 1 took 0:00:01
DUMP: Volume 1 transfer rate: 110 kB/s
DUMP: 110 tape blocks (0.11MB) on 1 volume(s)
DUMP: finished in less than a second
DUMP: Date of this level 0 dump: Thu Feb 12 19:17:14 2004
DUMP: Date this dump completed: Thu Feb 12 19:17:16 2004
DUMP: Average transfer rate: 110 kB/s
DUMP: DUMP IS DONE
carapuce:~#

```

### Lister le contenu de l'archive

```

carapuce:~# restore -tf /dev/sda4
Dump date: Thu Feb 12 19:17:14 2004
Dumped from: the epoch
Level 0 dump of /home on carapuce:/dev/hda6
Label: none
      2      .
     11      ./lost+found
    31489     ./mondo.scratch.20678
    62977     ./pierre
    62978     ./pierre/.alias
    62979     ./pierre/.bash_history
    62980     ./pierre/.bash_profile
    62981     ./pierre/.bashrc
    62982     ./pierre/.cshrc
    62983     ./pierre/f1
    62984     ./pierre/f2
    94465     ./pierre/rep
    94466     ./pierre/rep/fic

```

### Vérifier une archive

On vérifie l'archive précédente, on constate quelle est correcte. Ensuite on modifie des fichiers et on vérifie de nouveau, des erreurs apparaissent.

```

carapuce:~# restore -Cf /dev/sda4 -D /home
Dump date: Thu Feb 12 19:17:14 2004
Dumped from: the epoch
Level 0 dump of /home on carapuce:/dev/hda6
Label: none
filesys = /home
carapuce:~# echo "++++" >> /home/pierre/f1
carapuce:~# rm /home/pierre/f2
carapuce:~# cp /etc/inittab /home/pierre/rep
carapuce:~# restore -Cf /dev/sda4 -D /home

```

```
Dump   date: Thu Feb 12 19:17:14 2004
Dumped from: the epoch
Level 0 dump of /home on carapuce:/dev/hda6
Label: none
filesystem = /home
./pierre/f1: size has changed.
restore: ./pierre/f2: does not exist (-1): No such file or directory
Some files were modified!
carapuce:~#
```

### Sauvegarde incrémentale

Dans l'exemple qui suit, on sauvegarde */home* de manière incrémentale au niveau 4. On sauvegarde la VTOC dans le fichier */var/lib/home\_4*. On affiche ensuite le fichier *dumpdates*.

```
carapuce:~# dump 4uf /dev/sda4 -A /var/lib/home_4 /home
DUMP: Date of this level 4 dump: Thu Feb 12 19:35:18 2004
DUMP: Date of last level 0 dump: Thu Feb 12 19:17:14 2004
DUMP: Dumping /dev/hda6 (/home) to /dev/sda4
DUMP: Added inode 8 to exclude list (journal inode)
DUMP: Added inode 7 to exclude list (resize inode)
DUMP: Label: none
...
carapuce:~# restore -tf /var/lib/home_4
Dump   date: Thu Feb 12 19:35:18 2004
Dumped from: Thu Feb 12 19:17:14 2004
Level 4 dump of /home on carapuce:/dev/hda6
Label: none
Starting inode numbers by volume:
    Volume 1: 2
        2      .
    62977      ./pierre
    62983      ./pierre/f1
    94465      ./pierre/rep
    94467      ./pierre/rep/inittab
carapuce:~# cat /var/lib/dumpdates
/dev/hda6 0 Thu Feb 12 19:17:14 2004
/dev/hda6 4 Thu Feb 12 19:35:18 2004
carapuce:~#
```

### Restauration d'un fichier

```
carapuce:~# cd /home
carapuce:/home# restore -xf /dev/sda4 ./pierre/f2
restore: ./pierre: File exists
You have not read any volumes yet.
Unless you know which volume your file(s) are on you should start
with the last volume and work towards the first.
Specify next volume # (none if no more volumes): 1
restoring ./pierre/f2
set owner/mode for './?' [yn] n
carapuce:/home#
```

### Restauration de tout un FS

On commence par formater la partition abritant */home*, ensuite on la monte. On se déplace dans sa racine et on restaure chaque archive en commençant par le niveau zéro, puis on passe par tous les niveaux de manière croissante, ici on n'utilise que deux volumes (niveau 0 et niveau 4).

```

carapuce:~# mkfs -t ext3 /dev/hda6
mke2fs 1.27 (8-Mar-2002)
carapuce:~# mount /home
carapuce:~# cd /home
carapuce:/home# restore -rf /dev/sda4
restore: ./lost+found: File exists
carapuce:/home# eject zip
carapuce:/home# restore -rf /dev/sda4
carapuce:/home#

```

### Restauration interactive

Après avoir listé les commandes grâce à la commande `help`, on parcourt et on visualise la VTOC avec les commandes `ls`, `cd` et `pwd`. Grâce à la commande `what` on visualise l'entête. Ensuite, avec la commande `add` on positionne un drapeau sur chaque fichier à restaurer, en l'occurrence `f1` et `f2`. Avec la commande `del`, on retire le drapeau pour le fichier `f2` et enfin on démarre la restauration avec la commande `extract`. La commande `quit` met fin au programme.

```

carapuce:/home# restore -if /dev/sda4
restore > help
Available commands are:
    ls [arg] - list directory
    cd arg - change directory
    pwd - print current directory
    add [arg] - add `arg' to list of files to be extracted
    delete [arg] - delete `arg' from list of files to be extracted
    extract - extract requested files
    setmodes - set modes of requested directories
    quit - immediately exit program
    what - list dump header information
    verbose - toggle verbose flag (useful with ``ls'')
    prompt - toggle the prompt display
    help or `?' - print this list
If no `arg' is supplied, the current directory is used
restore > ls
.:
lost+found/          mondo.scratch.20678/ pierre/

restore > cd pierre
restore > ls
./pierre:
.alias          .bash_profile  .cshrc          f2
.bash_history   .bashrc        f1              rep/

restore > pwd
/pierre
restore > what
Dump   date: Thu Feb 12 19:17:14 2004
Dumped from: the epoch
Level 0 dump of /home on carapuce:/dev/hda6
Label: none
restore > add f1 f2
restore: ./pierre: File exists
restore > ls
./pierre:
.alias          .bash_profile  .cshrc          *f2
.bash_history   .bashrc        *f1             rep/

```

```
restore > del f2
restore > extract
You have not read any volumes yet.
Unless you know which volume your file(s) are on you should start
with the last volume and work towards the first.
Specify next volume # (none if no more volumes): 1
restoring ./pierre/f1
set owner/mode for '.'? [yn] n
restore > quit
carapuce:/home#
```

## La sauvegarde de XFS avec les commandes xfsdump/xfsrestore

L'utilisation des commandes xfsdump/xfsrestore est globalement équivalente à celle des commandes dump/restore.

Contrairement à la commande dump, xfsdump nécessite que le FS soit monté pour le sauvegarder.

### Syntaxe

#### xfsdump

xfsdump -h

Liste un résumé de la syntaxe.

xfsdump [options] -f dest [-f dest ...] FS

Sauvegarde le FS sur la ou les destinations introduites par les options -f.

xfsdump [options] - FS

Idem, mais la sauvegarde a lieu sur la sortie standard.

xfsdump -I

Affiche le journal des sauvegarde (/var/lib/xfsdump/inventory).

Voici les principales options :

-b bs      Spécifie la taille des blocs (blocksize), par défaut 1Mo.

-d taille    Spécifie la taille d'un volume en Mo.

-l #        Spécifie le niveau incrémental (0 à 9).

#### xfsrestore

xfsrestore -h

Liste un résumé de la syntaxe.

xfsrestore [options] -f src [-f src ...] FS

Restaure le FS à partir des sources introduites par les options -f.

xfsrestore [options] - FS

Restaure, mais l'archive est lue à partir de la sortie standard.

xfsrestore -I

Affiche le journal des sauvegarde (/var/lib/xfsdump/inventory).

Voici les principales options :

-b bs      Spécifie la taille des blocs (blocksize), par défaut 1Mo.

-i        Restauration interactive.

-r        Restauration cumulative.

- t Affiche le contenu d'une archive.
- s chemin Restaure l'arbre ou le fichier correspond au chemin.

## Exemples

### Sauvegarde d'un FS

Dans l'exemple on prend un FS situé dans un fichier.

```
# xfsdump -l 0 -f /tmp/save.dmp /mnt/loop
xfsdump: using file dump (drive_simple) strategy
xfsdump: version 2.2.13 (dump format 3.0) - Running single-threaded

===== dump label dialog
=====

please enter label for this dump session (timeout in 300 sec)
-> essai
session label entered: "essai"

----- end dialog -----
-

xfsdump: level 0 dump of herbizarre:/mnt/loop
xfsdump: dump date: Thu Feb 12 19:39:01 2004
xfsdump: session id: a4c68559-fa05-4814-8650-a9fba3b03e2e
xfsdump: session label: "essai"
xfsdump: ino map phase 1: skipping (no subtrees specified)
xfsdump: ino map phase 2: constructing initial dump list
xfsdump: ino map phase 3: skipping (no pruning necessary)
xfsdump: ino map phase 4: skipping (size estimated in phase 2)
xfsdump: ino map phase 5: skipping (only one dump stream)
xfsdump: ino map construction complete
xfsdump: estimated dump size: 34368 bytes

===== media label dialog
=====

please enter label for media in drive 0 (timeout in 300 sec)
-> ESSAI
media label entered: "ESSAI"

----- end dialog -----
-

xfsdump: creating dump session media file 0 (media 0, file 0)
xfsdump: dumping ino map
xfsdump: dumping directories
xfsdump: dumping non-directory files
xfsdump: ending media file
xfsdump: media file size 27472 bytes
xfsdump: dump size (non-dir files) : 5216 bytes
xfsdump: dump complete: 31 seconds elapsed
xfsdump: Dump Status: SUCCESS
#
```

### Visualiser le journal des sauvegardes

```
# xfsrestore -I
file system 0:
```

```
fs id:          606887f0-6ba7-4afd-97e2-c1544c23756b
session 0:
  mount point:   herbizarre:/mnt/loop
  device:        herbizarre:/tmp/disk
  time:          Thu Feb 12 19:39:01 2004
  session label: "essai"
  session id:    a4c68559-fa05-4814-8650-a9fba3b03e2e
  level:         0
  resumed:       NO
  subtree:       NO
  streams:       1
  stream 0:
    pathname:    /tmp/save.dmp
    start:       ino 131 offset 0
    end:         ino 135 offset 0
    interrupted: NO
    media files: 1
    media file 0:
      mfile index: 0
      mfile type:  data
      mfile size:  27472
      mfile start: ino 131 offset 0
      mfile end:   ino 135 offset 0
      media label: "ESSAI"
      media id:    7d217598-c3ee-4ede-
b63a-07bf0802075a
xfsrestore: Restore Status: SUCCESS
```

### Visualiser le contenu d'une sauvegarde

```
# xfsrestore -tf /tmp/save.dmp
xfsrestore: using file dump (drive_simple) strategy
xfsrestore: version 2.2.13 (dump format 3.0) - Running single-threaded
xfsrestore: searching media for dump
xfsrestore: examining media file 0
xfsrestore: dump description:
xfsrestore: hostname: herbizarre
xfsrestore: mount point: /mnt/loop
xfsrestore: volume: /tmp/disk
xfsrestore: session time: Thu Feb 12 19:39:01 2004
xfsrestore: level: 0
xfsrestore: session label: "essai"
xfsrestore: media label: "ESSAI"
xfsrestore: file system id: 606887f0-6ba7-4afd-97e2-c1544c23756b
xfsrestore: session id: a4c68559-fa05-4814-8650-a9fba3b03e2e
xfsrestore: media id: 7d217598-c3ee-4ede-b63a-07bf0802075a
xfsrestore: using online session inventory
xfsrestore: searching media for directory dump
xfsrestore: reading directories
xfsrestore: 2 directories and 4 entries processed
xfsrestore: directory post-processing
xfsrestore: reading non-directory files
f1
f2
rep/fic
xfsrestore: table of contents display complete: 0 seconds elapsed
xfsrestore: Restore Status: SUCCESS
```

**Restaurer un fichier**

```
# xfsrestore -s rep/fic -f /tmp/save.dmp /mnt/loop
xfsrestore: using file dump (drive_simple) strategy
xfsrestore: version 2.2.13 (dump format 3.0) - Running single-threaded
xfsrestore: searching media for dump
xfsrestore: examining media file 0
xfsrestore: dump description:
xfsrestore: hostname: herbizarre
xfsrestore: mount point: /mnt/loop
xfsrestore: volume: /tmp/disk
xfsrestore: session time: Thu Feb 12 19:39:01 2004
xfsrestore: level: 0
xfsrestore: session label: "essai"
xfsrestore: media label: "ESSAI"
xfsrestore: file system id: 606887f0-6ba7-4afd-97e2-c1544c23756b
xfsrestore: session id: a4c68559-fa05-4814-8650-a9fba3b03e2e
xfsrestore: media id: 7d217598-c3ee-4ede-b63a-07bf0802075a
xfsrestore: using online session inventory
xfsrestore: searching media for directory dump
xfsrestore: reading directories
xfsrestore: 2 directories and 4 entries processed
xfsrestore: directory post-processing
xfsrestore: restoring non-directory files
xfsrestore: restore complete: 0 seconds elapsed
xfsrestore: Restore Status: SUCCESS
```

**Références****Man**

dump(8), restore(8), xfsdump(8), xfsrestore(8), xfsinutil(8)

**Livre**

*Unix Backup & Recovery*, par W. Curtis Preston.



# Sauvegarde en réseau

- **tar + dd + rsh ou ssh**
  - `tar cvf - . | rsh venus dd of=s.tar.gz`
  - `rsh venus dd if=s.tar.gz | tar xvf -`
- **tar**
  - `tar cvzf venus:/dev/ftape /home/pierre`
- **dump/restore**
  - `dump Ouf venus:/dev/ftape /home`
- **rsync**
  - `rsync -avz Doc/ venus:Doc/`

---

## Introduction

Les outils commerciaux comme Arkei ou le logiciel libre Amanda sont par nature des logiciels complets de sauvegarde en client/serveur. Ils sont faits pour construire son système de sauvegarde autour d'un poste dédié à cette tâche et qui abrite les lecteurs de cartouche.

Outre ces outils complexes, les commandes de base de Linux permettent de sauvegarder ses fichiers. Ce sont celles que l'on présente dans ce chapitre.

## Utilisation de tar, dd et rsh ou ssh

La commande `tar`, grâce à l'option `-f` - utilise les entrées/sorties standards. En couplant cette commande avec la commande `dd` exécutée à distance par `rsh`, il est possible de sauvegarder ou restaurer à distance des fichiers.

Si l'on veut améliorer la sécurité, on utilise la commande `ssh` en lieu et place de la commande `rsh`. Comme ces commandes ont la même syntaxe, elles sont tout à fait interchangeables. Il est même possible d'établir un lien symbolique qui substitue automatiquement `ssh` à `rsh`. Cette approche est standard dans le système Debian.

---

### Remarque

L'utilisation et la configuration des commandes `rsh` est décrite dans le module réseau.

---

### Exemples : Sauvegarde sur le site distant

Sauvegarde distante d'une arborescence (ici « . »).

```
carapuce:/home/cathy# tar cvzf - . | rsh bulbizarre dd of=save.tar.gz
```

Lister le contenu de la sauvegarde distante.

```
carapuce:/home/cathy# rsh bulbizarre dd if=save.tar.gz | tar tvzf -
```

Restaurer les fichiers.

```
carapuce:/home/cathy# rsh bulbizarre dd if=save.tar.gz | tar xvfz -
```

### Exemples : Sauvegarde en locale de fichiers distants

Sauvegarde de fichiers distants (deux solutions).

```
bulbizarre:~# rsh carapuce tar cvzf - /home/cathy | dd of=sav2.tar.gz
bulbizarre:~# rsh carapuce tar cvzf - /home/cathy > sav4.tar.gz
```

Restaurer un fichier ;

Cette restauration aura lieu habituellement à partir du système où l'on veut restaurer.

```
carapuce:/# rsh bulbizarre dd if=sav2.tar.gz | tar xvfz -
home/cathy/f2
```

## Utiliser les commandes dd et rsh (ou ssh)

Si l'on désire sauvegarder des partitions complètes, on peut se limiter à utiliser dd et rsh (ou ssh). La sauvegarde de partitions permet de sauvegarder tout type de FS, par exemple NTFS. Grâce à un système bootable supportant le réseau, comme Knoppix, on peut sauvegarder ce que l'on veut.

Sauvegarde d'une partition contenant un FS.

```
carapuce:~# umount /mnt/dsk # préférable
carapuce:~# dd if=/dev/sda4 | rsh bulbizarre dd of=/tmp/sda4.dd
```

Restauration de la partition.

```
carapuce:~# rsh bulbizarre dd if=/tmp/sda4.dd | dd of=/dev/sda4
```

## La commande GNU tar

La commande GNU tar, la commande tar installée par défaut sous Linux, permet directement de sauvegarder en réseau. Le système distant doit avoir le service rsh (ou ssh) activé et posséder la commande rmt. Le système et le fichier distant sont spécifiés par l'option -f. Voici sa syntaxe :

```
-f hôte:chemin
-f @hôte:chemin
-f utilisateur@hôte:chemin
```

### Exemples

Sauvegarder une arborescence (ici « . »).

```
carapuce:/home/cathy# tar cvzf bulbizarre:/root/save3.tar.gz .
```

Lister le contenu de la sauvegarde.

```
carapuce:/home/cathy# tar tvzf bulbizarre:/root/save3.tar.gz
```

Restaurer un fichier.

```
carapuce:/home/cathy# tar xvfz bulbizarre:/root/save3.tar.gz ./f2
```

## Utiliser NFS

Le système NFS réalise l'accès à des fichiers distants de manière transparente. La sauvegarde ou la restauration de fichiers se fait, dans ce cas, simplement en utilisant la commande cp -rfp déjà décrite. Le système NFS est étudié dans le module « Les services réseaux ».

## Sauvegarde incrémentale

Comme dans le cas de la commande GNU tar, les commandes de sauvegarde incrémentale permettent directement de sauvegarder en réseau. La même logique s'applique : le système distant doit avoir le service rsh (ou ssh) actif et posséder la

commande `rmt`. D'autre part, le fichier distant, si ce n'est pas un fichier périphérique, doit exister.

Un des grands avantages des commandes de sauvegarde incrémentale, c'est de mémoriser les attributs Linux, les ACL et la date de dernier accès. Il est possible également de sauver un FS non monté.

Inversement, le principal désavantage des commandes de sauvegarde incrémentale, est qu'elles sont spécifiques du FS considéré. Pour chaque type de FS, il faut posséder le couple de commandes (dump/restore) associé. Le format de l'archive est également spécifique.

### **Sauvegarde incrémentale d'un volume Ext2/Ext3**

Sauvegarde complète d'un FS Ext3.

```
bulbizarre:~ # touch /root/sda4.dmp
```

```
carapuce:~# dump -0 -f bulbizarre:/root/sda4.dmp /dev/sda4
```

Restauration complète d'un FS.

```
carapuce:/mnt# mkfs -t ext3 /dev/sda4
```

```
carapuce:/mnt# mount /dev/sda4 /mnt/dsk/
```

```
carapuce:/mnt# cd /mnt/dsk
```

```
carapuce:/mnt/dsk# restore -r -f bulbizarre:/root/sda4.dmp
```

### **Sauvegarde incrémentale d'un volume reiserfs ou jfs**

Pour le moment, les commandes de sauvegarde incrémentale d'un volume reiserfs ou jfs n'existent pas. Il faut utiliser les autres techniques déjà étudiées.

### **Sauvegarde incrémentale d'un volume XFS**

Sauvegarde complète (niveau 0) d'un FS XFS à distance sans être prompté. Dans la version actuelle de `xfsdump`, le système de fichiers doit être monté.

```
root@herbizarre:/mnt# xfsdump -F -l 0 -f carapuce:/root/save_xfs /mnt/loop
```

Restauration interactive d'un fichier.

```
root@herbizarre:/mnt/loop# xfsrestore -i -f carapuce:/root/save_xfs .
...
-> add un_fichier
-> extract
----- end dialog -----
xfsrestore: restoring non-directory files
xfsrestore: restore complete: 60 seconds elapsed
xfsrestore: Restore Status: SUCCESS
root@herbizarre:/mnt/loop#
```

## **Rsync**

La commande `rsync` sert à maintenir à jour une copie de fichiers ou d'arborescence. `rsync` ne transfère que les différences des fichiers sources et des fichiers cibles.

Par défaut, `rsync` utilise `rsh` pour transférer les différences. La commande, via l'option `-e`, peut utiliser un autre shell distant, comme `ssh`. La commande `rsync` peut même utiliser son propre protocole `rsync` via le port serveur 873/tcp (cf. *rsync(1)*).

### **Quelques options**

-v Mode verbeux.

-c Vérifie les sommes de contrôle.

- a Mode archive : conserve les attributs, les droits, les liens, ...
- z Comprime les données.

### Exemples

Copier les fichiers avec l'extension .txt du répertoire courant du client mars dans le répertoire textes/ du compte homonyme sur le serveur venus.

```
mars $ rsync *.txt venus:textes/
```

La syntaxe est la même que celle de rcp. Si les fichiers ont déjà été copiés, seules les différences sont transmises.

Mettre à jour l'arborescence Documents.

```
mars $ rsync -avz Documents/ venus:Documents/
building file list ... done
./
group
./
wrote 567 bytes read 42 bytes 1218.00 bytes/sec
total size is 10836 speedup is 17.79
mars $
```

Idem, mais utilise le shell ssh.

```
mars $ rsync -avz -e ssh Documents/ venus:Documents/
```

## Références

### Man

tar(1), dd(1), rsh(1), ssh(1), dump(8), restore(8), xfsdump(8), xfsrestore(8), rsync(1)

# Sauvegarde Bare-Metal

## ■ Sauvegarde

- 1. Sauver l'architecture du système (partitionnement, ...)
- 2. Sauver les FS systèmes (`/`, `/var/`, `/usr`)
- 3. Sauver les autres FS

## ■ Restauration

- 1. Démarrer à partir d'un système Linux bootable
- 2. Repartitionner le système
- 3. Restaurer les FS systèmes
- 4. Restaurer le secteur de démarrage
- 5. Restaurer les autres FS

---

## Introduction

Un administrateur doit prévoir le pire, son système ne démarre plus. La raison en est souvent la panne du disque qui abrite le système de fichiers racine. Il faut dans ce cas changer ce disque et réinstaller le système tel qu'il était peu de temps avant l'incident. La sauvegarde qui permet de pallier à ce grave problème s'appelle une sauvegarde Bare-Metal (métal nu), car elle s'applique à un ordinateur neuf ou à l'ancien, mais avec de nouveaux disques. Non seulement cette sauvegarde permet de réinstaller le système à l'identique, mais également de le cloner.

## Principe

En cas de sinistre grave (perte d'un disque, perte de l'ordinateur), la restauration des systèmes de fichiers associés aux utilisateurs ou aux applications (`/home`, `/oracle`, `/www`, ...) ne pose pas de problèmes. Les sauvegardes classiques, étudiées dans les chapitres précédents sont faites pour cela. Il en va tout autrement pour les systèmes de fichiers essentiels : `/` (racine), `/var`, `/usr`, ... Une sauvegarde Bare-Metal consiste essentiellement à sauvegarder ceux-ci et à permettre leur restauration.

Une sauvegarde Bare-Metal peut être effectuée manuellement ou grâce à un logiciel dédié comme `mondo` ou `mkcddrec`.

### **Sauvegarde**

Voici les étapes à suivre :

1 – Sauvegarder l'architecture du système.

Les données liées à l'architecture sont spécifiques d'un système et du matériel. On peut citer quand même les éléments suivants :

- Le partitionnement du système (numéro, taille, type des partitions, la partition d'amorce)
- Les options de chargement du système (options du noyau, ...)

- La description des systèmes de fichiers montés au démarrage (/etc/fstab, df)

2 – Sauvegarder les systèmes de fichiers essentiels.

La sauvegarde des systèmes de fichiers essentiels (/ (root), /var, /usr) doit être effectuée si possible par une commande de sauvegarde incrémentale, comme `dump`, mais on peut aussi utiliser la commande `tar`. La sauvegarde a lieu sur un périphérique local (cd-rom, lecteur ZIP, clef USB) ou en réseau.

3 – Sauvegarder les autres systèmes de fichiers.

Pour les autres systèmes de fichier, on procède classiquement (*cf. chapitres précédents*).

## Restauration

1 – On démarre à partir d'un système Linux autonome bootable.

Pour se faire, on utilise par exemple le 1er cdrom d'installation du système ou une distribution « live », comme Knoppix ou System Rescue CD.

2 – On restaure l'architecture du système.

Il faut initialiser et repartitionner le disque système.

3 – On restaure les systèmes de fichiers essentiels.

On crée les systèmes de fichiers et on les monte sous l'arborescence /mnt. Le système de fichiers racine du système de dépannage étant en RAM. On utilise ensuite les commandes de restauration symétriques de celles utilisées lors de la sauvegarde (`tar`, `restore`, ...) pour restaurer les systèmes de fichiers essentiels (/ (root), /var, /usr).

4 – On restaure le secteur de boot.

Il faut ensuite installer le chargeur (lilo ou grub). Dans le cas de lilo, on peut se baser sur le fichier lilo.conf restauré pour le réinstaller.

5 – On restaure les autres systèmes de fichiers.

Après avoir restauré les systèmes de fichiers essentiels, on peut redémarrer à partir du disque dur et utiliser les commandes classiques pour restaurer le reste du système (/home, ...).

## Exemples de sauvegardes et restauration manuelles

On présente trois exemples :

- Sauvegarde et restauration du système en réseau avec `dump/restore`.
- Sauvegarde et restauration du système en locale avec `tar`.
- Sauvegarde et restauration du système en réseau avec `partimage`.

Pour la gestion des partitions, selon les cas, on utilise `parted`, `cfdisk` ou `sfdisk`.

### Sauvegarde réseau avec `dump/restore`

```
carapuce:~# fdisk -l /dev/hda
```

```
Disk /dev/hda: 16 heads, 63 sectors, 19857 cylinders
Units = cylinders of 1008 * 512 bytes
```

Device	Boot	Start	End	Blocks	Id	System
/dev/hda1	*	1	388	195520+	83	Linux
/dev/hda2		389	2191	908712	5	Extended
/dev/hda3		2192	2579	195552	83	Linux
/dev/hda5		389	640	126976+	82	Linux swap
/dev/hda6		641	1028	195520+	83	Linux
/dev/hda7		1029	2191	586120+	83	Linux

```

carapuce:~# parted /dev/hda print
Disk geometry for /dev/hda: 0.000-9773.367 megabytes
Disk label type: msdos
Minor      Start      End        Type        Filesystem  Flags
1          0.031      190.968    primary     ext3        boot
2          190.969    1078.382   extended
5          191.000    315.000    logical     linux-swaps
6          315.031    505.968    logical     ext3
7          506.000    1078.382   logical     ext3
3          1078.383    1269.351   primary     ext3

carapuce:~# cat /etc/fstab
# /etc/fstab: static file system information.
#
# <file system> <mount point> <type> <options> <dump><pass>
/dev/hda1      /          ext3     errors=remount-ro 0      1
/dev/hda5      none       swap     sw                 0      0
proc           /proc      proc     defaults           0      0
/dev/fd0       /floppy    auto     user,noauto        0      0
/dev/cdrom     /cdrom     iso9660  ro,user,noauto     0      0
/dev/hda3      /var       ext3     defaults           0      2
/dev/hda7      /usr       ext3     defaults           0      2
/dev/hda6      /home      ext3     defaults           0      2

carapuce:~# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/hda1       185M   29M  147M  17% /
/dev/hda3       185M   34M  141M  20% /var
/dev/hda7       563M  170M  365M  32% /usr
/dev/hda6       185M   4.1M  171M   3% /home

carapuce:~# mkdir bare_metal/
carapuce:~# cd bare_metal/
carapuce:~/bare_metal# fdisk -l /dev/hda > hda.fdisk
carapuce:~/bare_metal# parted /dev/hda print > hda.parted
carapuce:~/bare_metal# cp /etc/fstab .
carapuce:~/bare_metal# grep -v '^#' /etc/lilo.conf | grep .
lba32
boot=/dev/hda
root=/dev/hda1
install=/boot/boot-menu.b
map=/boot/map
delay=20
vga=normal
default=Linux
image=/vmlinuz
        label=Linux
        read-only
image=/vmlinuz.old
        label=LinuxOLD
        read-only
        optional
carapuce:~/bare_metal# cd ..
carapuce:~# tar cf herbizarre:/tmp/carapuce.meta bare_metal/
carapuce:~# dump -0 -f herbizarre:/root/carapuce.fs_root /
carapuce:~# dump -0 -f herbizarre:/root/carapuce.fs_var /var
carapuce:~# dump -0 -f herbizarre:/root/carapuce.fs_usr /usr

```

```
root@herbizarre:~# ls -lh cara*
-rw-r--r-- 1 root root 50K 2004-01-24 12:41 carapuce.fs_home
-rw-r--r-- 1 root root 32M 2004-01-24 12:34 carapuce.fs_root
-rw-r--r-- 1 root root 173M 2004-01-24 12:40 carapuce.fs_usr
-rw-r--r-- 1 root root 37M 2004-01-24 12:36 carapuce.fs_var
-rw-r--r-- 1 root root 10K 2004-01-24 12:08 carapuce.meta
root@herbizarre:~#
```

### Sauvegarde locale avec tar

Au lieu de sauvegarder en réseau, on peut utiliser un périphérique local (lecteur ZIP, clé USB, lecteur de cartouche DAT, disque dur amovible, ...). Dans l'exemple suivant, on sauvegarde avec compression, sur un seul volume ZIP l'ensemble de l'arborescence.

```
carapuce:~# tar -cvz -f /dev/sda4 /
```

### Sauvegarde réseau avec partimage

On démarre le système avec la distribution Knoppix qui possède en standard le logiciel partimage. Dans l'exemple qui suit, on sauvegarde sur le serveur 192.168.218.19 qui doit posséder un serveur partimaged actif (*cf. chapitre partimage*).

```
root@0[root]# dd if=/dev/hda bs=512 count=1 | rsh 192.168.218.19 dd
of=/usr/BCK/MBR
root@0[root]# sfdisk -d /dev/hda > /tmp/backup_hda.sf
root@0[root]# rcp /tmp/backup_hda.sf 192.168.218.19:/usr/BCK/
root@0[root]# partimage -s 192.168.218.19 save /dev/hda1 bck_root.gz
root@0[root]# partimage -s 192.168.218.19 save /dev/hda3 bck_var.gz
root@0[root]# partimage -s 192.168.218.19 save /dev/hda6 bck_home.gz
root@0[root]# partimage -s 192.168.218.19 save /dev/hda7 bck_usr.gz
```

### Crash du disque système

Il est facile de simuler la perte du disque système. Il suffit d'effacer le MBR.

```
carapuce:~# dd if=/dev/zero of=/dev/hda bs=512 count=1
1+0 records in
1+0 records out
carapuce:~#
```

### Restauration réseau avec dump/restore

1 – On démarre le système avec Knoppix.

La distribution Linux Knoppix fonctionne en autonome sur un cdrom. Elle peut jouer le rôle de système de maintenance. Après le démarrage, on peut configurer le réseau (c'est automatique si l'on dispose d'un serveur DHCP).

```
boot : knoppix 2 lang=fr
...
# ifconfig eth0 192.168.218.11
```

2 – On recrée la table des partitions.

```
root@0[root]# fdisk -l /dev/hda

Disque /dev/hda: 10.2 Go, 10248118272 octets
255 têtes, 63 secteurs/piste, 1245 cylindres
Unités = cylindres de 16065 * 512 = 8225280 octets

Disque /dev/hda ne contient pas une table de partition valide
root@0[root]# cfdisk
Pas de table de partitions ou signature inconnue dans la table de
partitions
Voulez-vous débiter avec une table à zéro [o/N]o
```



```

                                cfdisk 2.12

                                Unité de disque: /dev/hda
                                Size: 10248118272 bytes, 10.2 GB
                                Heads: 255   Sectors per Track: 63   Cylinders: 1245

    Nom   Fanions   Part Type   Type SF   Étiq.]   Size (MB)
    -----
    hda1   Amorce   Primaire   Linux     197,41
    hda5             Logique   Linux swap 197,41
    hda6             Logique   Linux     197,41
    hda7             Logique   Linux     600,45
    hda3             Primaire   Linux     197,41
                                Primaire   Espace libre 8850,41

    Êtes-vous certain de vouloir écrire la table de partitions sur le
    disque? (oui ou non) oui
    AVERTISSEMENT!! Cela pourrait détruire les données sur votre
    disque!
    root@0[root]# fdisk -l /dev/hda

    Disque /dev/hda: 10.2 Go, 10248118272 octets
    255 têtes, 63 secteurs/piste, 1245 cylindres
    Unités = cylindres de 16065 * 512 = 8225280 octets

    Périphérique Boot  Start      End        Blocks    Id  System
    /dev/hda1          *          1         24        192748+   83  Linux
    /dev/hda2                25        145        971932+   5   Extended
    /dev/hda3          146        169        192780    83  Linux
    /dev/hda5                25         48        192748+   82  Linux swap
    /dev/hda6                49         72        192748+   83  Linux
    /dev/hda7                73        145        586341    83  Linux
    root@0[root]#

```

3 – On restaure les systèmes de fichiers essentiels.

```

root@0[root]# mkfs -t ext3 /dev/hda1
root@0[root]# mkfs -t ext3 /dev/hda3
root@0[root]# mkfs -t ext3 /dev/hda6
root@0[root]# mkfs -t ext3 /dev/hda7
root@0[root]# mount /dev/hda1 /mnt
root@0[root]# cd /mnt
root@0[mnt]# restore -rf 192.168.218.19:/root/carapuce.fs_root
Connection to 192.168.218.19 established.
restore: ./lost+found: File exists
root@0[mnt]# mount /dev/hda3 /mnt/var
root@0[mnt]# mount /dev/hda7 /mnt/usr
root@0[mnt]# mount /dev/hda6 /mnt/home
root@0[mnt]# cd var
root@0[var]# restore -rf 192.168.218.19:/root/carapuce.fs_var
Connection to 192.168.218.19 established.
restore: ./lost+found: File exists
root@0[var]# cd ../usr
root@0[usr]# restore -rf 192.168.218.19:/root/carapuce.fs_usr
Connection to 192.168.218.19 established.
restore: ./lost+found: File exists

```

4 – On restaure le secteur de boot.

```

root@0[etc]# cd /mnt

```

```

root@0 [mnt]# chroot .
Knoppix:/# lilo -C /etc/lilo.conf
Added Linux *
Skipping /vmlinuz.old
Knoppix:/#

```

5 – On restaure le reste du système.

Le système est maintenant valide, on peut le redémarrer.

```

Knoppix:/# exit
root@0 [mnt]# reboot

```

Après le redémarrage, on se connecte en tant qu'administrateur et on restaure le reste du système.

```

carapuce:~# cd /home
carapuce:/home# restore -rf herbizarre:/root/carapuce.fs_home
Connection to herbizarre established.
restore: ./lost+found: File exists
carapuce:/home#

```

### Restauration locale avec tar

```

root@0 [root]# mount /dev/hda1 /mnt
root@0 [root]# cd /mnt
root@0 [mnt]# mkdir var usr home
root@0 [mnt]# mount /dev/hda3 /mnt/var
root@0 [mnt]# mount /dev/hda7 /mnt/usr
root@0 [mnt]# mount /dev/hda6 /mnt/home
root@0 [mnt]# modprobe ppa
root@0 [mnt]# tar xvzf /dev/sda4

```

### Restauration réseau avec partimage

On démarre le système avec la distribution Knoppix.

```

# rsh 192.168.218.19 dd if=/usr/BCK/MBR | dd of=/dev/hda
# rcp 192.168.218.19:/usr/BCK/backup_hda.sf /tmp
root@192.168.218.19's password:
backup_hda.sf                                100% 416      1.3MB/s
00:00
# sfdisk /dev/hda < /tmp/backup_hda.sf
Vérification qu'aucun autre n'utilise le disque en ce moment ...
OK

```

Disque /dev/hda: 1245 cylindres, 255 têtes, 63 secteurs/piste

Vieille situation:

Unités= cylindres de 8225280 octets, blocs de 1024 octets, décompte à partir de 0

Périph	Amor	Début	Fin	#cyls	#blocs	Id	Système
/dev/hda1	*	0+	187	188-	1510078+	83	Linux
/dev/hda2		0	-	0	0	0	Vide
/dev/hda3		188	375	188	1510110	83	Linux
/dev/hda4		376	1244	869	6980242+	5	Extended
/dev/hda5		376+	498	123-	987966	82	Linux swap
/dev/hda6		499+	686	188-	1510078+	83	Linux
/dev/hda7		687+	1244	558-	4482103+	83	Linux

Nouvelle situation:

Unités= secteurs de 512 octets, décompte à partir de 0

Périph	Amorce	Début	Fin	#secteurs	Id	Système
--------	--------	-------	-----	-----------	----	---------

```

/dev/hda1  *          63    3020219    3020157    83    Linux
/dev/hda2          0      -          0      0    Vide
/dev/hda3      3020220    6040439    3020220    83    Linux
/dev/hda4      6040440    20000924    13960485    5    Extended
/dev/hda5      6040503    8016434    1975932    82    Linux swap
/dev/hda6      8016498    11036654    3020157    83    Linux
/dev/hda7      11036718    20000924    8964207    83    Linux
Succès d'écriture de la nouvelle table de partitions

Relecture de la table de partitions ...

Si vous créez ou modifiez une partition DOS, /dev/foo7, par exemple,
alors utiliser dd(1) pour mettre à zéro les premiers 512 octets:
dd if=/dev/zero of=/dev/foo7 bs=512 count=1
(Consulter fdisk(8).)
#
# partimage -s 192.168.218.19 restore /dev/hda1 bck_root.gz.000
# partimage -s 192.168.218.19 restore /dev/hda3 bck_var.gz.000
# partimage -s 192.168.218.19 restore /dev/hda6 bck_home.gz.000
# partimage -s 192.168.218.19 restore /dev/hda7 bck_usr.gz.000

```

## Sauvegarde avec mondo

Mondo est un logiciel libre (GPL) très simple d'emploi qui effectue une sauvegarde Bare-Metal d'un système Linux. Il permet également de réaliser des sauvegardes classiques d'une arborescence ou d'une partition, y compris NTFS. Mondo sauvegarde sur CD-ROM, sur cartouche, sur disque dur ou en réseau via NFS. Mondo supporte les LVM, le RAID, ext2, ext3, JFS, XFS, ReiserFS, VFAT.

### Sauvegarde

Dans l'exemple qui suit, on ne dispose pas de lecteur de CD-ROM local. On réalise d'abord un montage NFS avec un serveur qui en possède un (ne pas confondre avec la sauvegarde de type NFS de Mondo). C'est sur ce serveur que seront créés les images ISO résultant de la sauvegarde.

```

carapuce:~# mkdir /mnt/nfs
carapuce:~# showmount -e herbizarre
Export list for herbizarre:
/usr/BCK carapuce
carapuce:~# mount -t nfs herbizarre:/usr/BCK /mnt/nfs
carapuce:~# mondoarchive -Oi -d /mnt/nfs -E /mnt/nfs
Initializing...
Mondo Archive v1.41 --- http://www.microwerks.net/~hugo
Switches:-
-E /mnt/nfs
-O
-d /mnt/nfs
-i
See /var/log/mondo-archive.log for details of backup run.
Checking sanity of your Linux distribution
Checking sanity of your Linux distribution
WARNING! You have very little RAM. Please upgrade to 64MB or more.
----- cat /proc/devices | grep ramdisk -----
1 ramdisk
-----end of output-----
Done.
Done.
Making catalog of files to be backed up

```

...

Ensuite, au niveau du serveur, on grave les images produites. Leur nom représente leur numéro d'ordre : 1.iso, 2.iso, etc.

```
herbizarre :~# cdrecord -v speed=4 dev=0,0,0 /usr/BCK/1.iso
```

Si le poste dispose d'un graveur, on peut créer les cdrom directement. Dans l'exemple suivant, on sauvegarde complètement son système avec un lecteur x4 et de cdrom 700 Mo.

```
herbizarre :~# mondoarchive -Oc 4 -s 700m
```

## Restauration

La restauration est très simple. Le premier CD-ROM étant bootable, on démarre avec ce CD-ROM. Un écran similaire à lilo s'affiche et propose les choix suivants :

- nuke
- interactive
- expert
- compare
- iso

Le choix interactive permet de restaurer sélectivement des fichiers. Le choix nuke correspond à une restauration Bare-Metal automatique faite pour un disque vierge. Il suffit de donner les CD-ROM au fur et à mesure que le logiciel les demande.

Après restauration, on redémarre le système. Mondo a utilisé tout l'espace disponible. Il a éventuellement retailé les partitions en tenant compte du nouvel espace disque.

```
carapuce:~# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/hda1        1.4G   60M  1.2G   5% /
/dev/hda3        1.4G   74M  1.2G   6% /var
/dev/hda7        4.2G  199M  3.7G   5% /usr
/dev/hda6        1.4G   33M  1.3G   3% /home
carapuce:~#
```

Dans le cas où l'on ne peut pas démarrer à partir de la sauvegarde (sauvegarde sur cartouche, sauvegarde réseau, ...), Mondo fournit une image ISO (mindiso.iso) qui est bootable et qui contient le logiciel mondorestore.

## Références

### Man

tar(1), dump(8), restore(8), mondoarchive(1), mondorestore(1), afio(1), mkisofs(8), cdrecord(1), dd(1)

### Howto

Linux Complete Backup and Recovery HOWTO

### Livre

*Unix Backup & Recovery*, de W. Curtis Preston

### Internet

Mondo

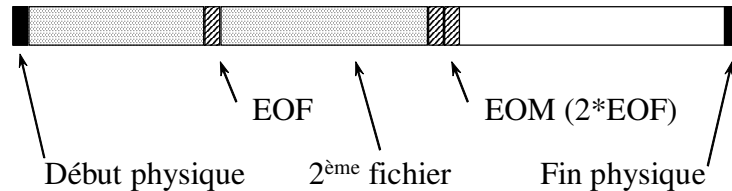
<http://www.microwerks.net/~hugo/>

Mkcdrec

<http://mkcdrec.ota.be/>

## Les bandes

### ■ Format d'une bande



### ■ La commande mt

```
# mt fsf 1  
# mt eom
```

---

## Introduction

Des fichiers, au sens UNIX, sont sauvegardés sur une cartouche magnétique sous forme d'une suite de blocs. Ils constituent un fichier, au sens de la sauvegarde. Ce fichier se termine par une marque physique de fin de fichier. On peut mettre plusieurs fichiers de sauvegarde sur la même cartouche. Le dernier se termine par deux fins de fichiers. Les cartouches présentent également un début et une fin physique.

Sur la cartouche, un élément mobile permet de la protéger en écriture.

Les cartouches « QIC » nécessitent une « retension » si :

- La cartouche est neuve.
- La cartouche est inutilisée depuis longtemps.
- Il se produit des erreurs de lecture/écriture.
- Cette retension consiste en un déroulement complet de la bande, suivi d'un rembobinage.
- Normalement, les bandes sont vendues formatées (le formatage délimite les blocs).
- Les fichiers spéciaux du système Linux permettent de préciser notamment :
  - Le lecteur.
  - Si la bande est rembobinée à la fermeture.
  - Si l'on effectue une retension de la bande (QIC).
- La densité.

## La commande mt

La commande `mt` permet de se positionner sur une bande magnétique, en se déplaçant de fichier archive en fichier archive.

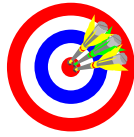
```
# mt -f /dev/nrft0 rewind
# # La variable d'environnement TAPE indique à mt le lecteur à
# utiliser.
# TAPE=/dev/nrft0 ; export TAPE
# mt rewind # on rembobine
# mt fsf 1 # on saute un fichier
# mt bsf 2 # on revient en arrière de deux fichiers
# mt eof # on écrit une marque de fin de fichier
# mt asf 3 # on va au 3ème fichier (en absolu)
# mt eom # on va en fin logique de la bande
# mt status # on affiche l'état de l'unité
# mt offline # on rembobine et on déconnecte l'unité
```

## Références

### *Howto*

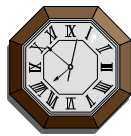
Ftape-HOWTO

## Atelier 6 : La sauvegarde



### Objectifs :

- Savoir réaliser des sauvegardes et des restaurations grâce aux commandes `tar` et `cpio`
- Savoir utiliser la commande `dd`



Durée : 60 minutes.

---

### Exercice n°1

Les exercices qui suivent illustrent l'utilisation de la commande `tar`. On utilise comme support d'archive le fichier `/tmp/pierre.sauve` et l'arborescence de fichiers créée dans l'exercice n°4 du module 3. On est connecté sous le compte de l'utilisateur `pierre`.

- Sauvegardez les fichiers de pierre (l'arborescence `/home/pierre`).
- Listez le contenu de la sauvegarde.
- Détruisez un fichier, restaurez-le.
- Détruisez l'ensemble des fichiers de pierre, restaurez-les.
- Sauvegardez les fichiers de pierre de telle manière que l'on puisse restaurer un fichier chez `cathy`. Listez le contenu de la sauvegarde.

### Exercice n°2

Sans utiliser le support, quelle est la signification de `cpio -icvdumB` ?

### Exercice n°3

Exercice identique à l'exercice n°1, mais en utilisant la commande `cpio`.

### Exercice n°4

Dupliquez la sauvegarde `cpio` (le fichier `/tmp/pierre.sauve` réalisé à l'exercice précédent), avec `dd` sur une disquette. Restaurez un fichier à partir de la disquette avec `cpio`.

### **Exercice n°5**

Dupliquez une disquette MS-DOS avec `dd`.

### **Exercice n°6**

Quelle est la taille en bloc d'une disquette ? Utilisez la commande `dd`.

### **Exercice n°7**

Listez les commandes gérant les bandes (tape en anglais) de votre système.

### **Exercice n°8**

Montez le système de fichiers créé sur la disquette (Exercice n° 15 du module 4) et réalisez une sauvegarde complète, avec la commande `dump`, dans un fichier ordinaire du disque. Affichez la liste des fichiers sauvegardés. Recréez un système de fichiers sur la disquette et restaurez les fichiers.