

Petit manuel de gravure CD sous LINUX ou FreeBSD

Jean-Marc LICHTLE

13 octobre 2005

Table des matières

I	La gravure sous LINUX	2
1	Introduction	2
2	Logiciels disponibles	3
3	Etat des lieux	3
4	Création de CD de données	4
4.1	Création de l'image	4
4.2	Test de l'image nouvellement créée	5
4.3	Graver l'image	5
5	Copie d'un CD de données ou d'un CD bootable	6
5.1	Image de CD bootable avec mkisofs	6
5.2	Création de l'image iso avec readcd	6
5.3	Gravure de l'image	7
6	Copie directe de CD de données	7
7	Rippage de pistes audio	7
8	Création de CD audio	8
9	Informations annexes	9
9.1	Emploi de CD-RW	9
9.2	Précautions	9
10	CD multi-sessions	9
10.1	CD Extra	9
10.1.1	Récupération des pistes audios	9
10.1.2	Récupération des fichiers vidéos	9

10.1.3 Graver les fichiers audios	10
10.1.4 Récupérer les informations de limites de sessions	10
10.1.5 Créer l'image iso	10
10.1.6 Graver l'image	10
II La gravure sous FreeBSD	10
11 Introduction	11
11.1 L'émulation SCSI	11
12 Compilation du noyau pour émulation SCSI	11
12.1 Les droits d'accès	11
13 Création de CD de données	12
13.1 Création de l'image	12
13.2 Test de l'image créée	12
13.3 Graver l'image	13
14 Copie d'un CD de données ou d'un cd bootable	13
14.1 Emploi de dd	13
14.2 Emploi de readcd	13
15 Copie directe de CD de données	14
16 Rippage et gravure de pistes audio	14
16.1 Rippage	14
16.2 Gravure	14
17 Remerciements	15

Première partie

La gravure sous LINUX

1 Introduction

Mon objectif est de décrire les différentes techniques qui conduisent à copier ou à créer des CD. Je n'aborderai ici ni l'installation d'un lecteur ou d'un graveur de CD, ni la configuration de LINUX. Je m'attacherai à décrire l'emploi des commandes en mode texte, cdrecord, cdrdao, mkisofs, readcd, cdda2wav, cdparanoia etc. La description est relative à une configuration assez classique, LINUX Mandrake 9.1, un lecteur CD IDE et un lecteur / graveur IDE lui aussi mais reconnu par le système comme un appareil SCSI. L'objectif étant essentiellement didactique je

n'aborderai pas non plus l'utilisation des interfaces graphiques, cet aspect n'apportant rien à l'apprentissage de la technique et à l'acquisition des connaissances nécessaires à la gravure des CD.

J'engage vivement le lecteur à faire l'acquisition de quelques CD-RW qui seront très utiles lors des essais, à défaut l'utilisation exclusive de CD-R risque de conduire à la production d'un certain nombre de rondelles de plastique qui n'auront d'autre utilisation possible que la confection de mobiles (pas vraiment très beaux) d'épouvantails (très efficaces s'il y a un peu de vent et un peu de soleil) ou la constitution d'une collection de sous verres.

2 Logiciels disponibles

Logiciel	User	Source	Cible	Usage
mkisofs	user	fichiers	iso	création d'images iso
cdrecord	root	iso, raw, wav	scsi	gravure
cdrdao ->	user	scsi	raw+toc	lecture
cdrdao <-	user	raw+toc	scsi	gravure
readcd	root	scsi	raw	lecture
dd	user	ide	raw	lecture
cdparanoia	user	ide	wav	ripage son
cdda2wav	root	ide	wav+inf	ripage son

TAB. 1 – Tableau des principaux logiciels mode texte

Le tableau 1 montre la liste des principaux logiciels disponibles en mode texte sous LINUX. Sauf erreur ils sont tous inclus dans la distribution LINUX Mandrake 9.1 "de chez mon marchand de journaux" en 3 CD. Vous disposez donc très facilement d'une gamme aux possibilités multiples. Tous ces logiciels sont utilisables par l'utilisateur de base, sauf cdrecord, readcd et cdda2wav. A cet endroit je dois dire que, concernant cdda2wav, la logique ne me semble pas fulgurante dans la mesure où cdda2mp3 par exemple est accessible à l'utilisateur de base.

A noter que dd est une des fonctions de copie standard de LINUX et qu'elle est utile à bien d'autres tâches que la copie de CD.

3 Etat des lieux

La première chose à faire est de vérifier à quelle adresse SCSI se trouve le graveur. Pour cela vous pouvez utiliser la commande :

```
\$ cdrecord -scanbus
```

```
Linux sg driver version: 3.1.24
```

```
Cdrecord 2.0 (i586-mandrake-linux-gnu) Copyright (C) 1995-2002 Jörg Schilling
```

```
Using libscg version 'schily-0.7'
```

```
scsibus0:
```

```
0,0,0 0) 'SAMSUNG ' 'CD-R/RW SW-240B ' 'R403' Removable CD-ROM
0,1,0 1) *
0,2,0 2) *
0,3,0 3) *
0,4,0 4) *
0,5,0 5) *
0,6,0 6) *
0,7,0 7) *
```

Cet exemple montre que le lecteur graveur est situé à l'adresse 0,0,0 du bus SCSI. On peut aussi utiliser `cdrdao` pour faire cet état des lieux, la syntaxe est semblable à ceci près que le tiret disparaît.

```
\$ cdrdao scanbus
```

```
Cdrdao version 1.1.7 - (C) Andreas Mueller <andreas@daneb.de>
```

```
SCSI interface library - (C) Joerg Schilling
```

```
Paranoia DAE library - (C) Monty
```

Check <http://cdrdao.sourceforge.net/drives.html#dt> for current driver tables.

```
Using libscg version 'andreas-0.5-UNIXWARE_Patch'
```

```
0,0,0: SAMSUNG, CD-R/RW SW-240B, R403
```

Nota : Ne négligez jamais de faire un petit état des lieux avant de démarrer une séance de gravure. Un jour ou l'autre vous brancherez un appareil phot ou une clef USB sur votre linuxette et là bonjour les déménagements. Selon que les accessoires auront été connectés avant le boot ou après les adresses SCSI des différents appareils vont avoir des valeurs différentes ! Croyez moi, au début ça surprend un peu.

4 Création de CD de données

Cette section décrit comment créer un CD de données à partir de fichiers présents sur le disque dur ou sur un autre CD. Il s'agit des cas classiques de la sauvegarde de données sur CD. Dans un premier temps nous examinerons le cas de la sauvegarde de fichiers.

4.1 Création de l'image

L'objectif est ici de créer une image ISO des données à sauvegarder. L'image ISO va contenir les données, mais aussi une table d'allocation des fichiers, les informations de droits d'accès etc. La commande `mkisofs` va ici nous être très utile :

4.2 Test de l'image nouvellement créée

```
\$ mkisofs -r -J -o image.iso /home/copie
```

Options utilisées :

- r : option Rock Ridge et droit de lecture à tous pour les fichiers qui seront gravés sur le CD
- J : option Joliet pour lecture des noms longs par les systèmes windows
- o : output, nomme le fichier de sortie

Autre option possible :

- T : “transtable”, création d’une table de conversion des noms de fichiers, utile pour des systèmes non Rock Ridge.

Le nom du sous répertoire à sauvegarder est donné en argument à la ligne de commande. Les fichiers contenu dans ce répertoire seront contenus à la racine de l’image (et donc du futur CD).

Quelques explications sur les options.

- L’option Rock Ridge permet de conserver les noms longs pour une lecture ultérieure par LINUX. L’oubli de cette option conduit à des noms de fichiers en format 8.3 classique DOS. Ces noms sont obtenus par amputation du nom d’origine après le 8ème caractère. Si cette amputation conduit à une ambiguïté (deux ou plusieurs noms de fichiers identiques) alors l’amputation est encore plus brutale, après le 5ème caractère et complétée par des chiffres 000, 001 etc. Il s’agit donc d’une option qu’il vaut mieux ne jamais oublier. Elle existe d’ailleurs en deux variantes, -r pour Rock Ridge ET droit de lecture attribué à tous les utilisateurs du futur disque ou -R qui ne change pas les droits de lecture d’origine.
- L’option Joliet est, très en gros, l’équivalent de l’option Rock Ridge, mais cette fois pour Windows. Sans cette option les noms sont vu par windows compactés au format 8.3.

Nous verrons plus loin d’autres options intéressantes, notamment celles qui permettent de créer des disques multi-sessions.

4.2 Test de l'image nouvellement créée

LINUX donne la possibilité de tester l’image ainsi créée en la “montant” comme s’il s’agissait d’un disque amovible. A la différence de mkisofs qui peut être utilisée par l’utilisateur de base, le “montage” d’une image iso ne peut être fait que par l’administrateur système :

```
# mount image.iso -t iso9660 -o loop,ro /mnt/test
```

Cette commande “monte” l’image dans /mnt/test/. Il suffit alors de faire cd /mnt/test/ pour découvrir le contenu de la nouvelle archive et vérifier qu’elle est effectivement lisible, que les noms de fichiers sont conservés proprement etc.

4.3 Graver l’image

Nous y voilà, nous allons graver notre premier CD. Prenez une grande inspiration, mettez un CD vierge dans le graveur et allons y. La commande

```
\$ cdrecord -v -dao dev=0,0 image.iso
```

déclenche le processus de gravure. Vous voyez s'afficher un message sur plusieurs lignes, puis un chronomètre attaque un compte à rebours vous donnant 10 sec de réflexion supplémentaire. Ce laps de temps étant écoulé la gravure commence.

Quelques explications sur les options.

- L'option -v rend le processus bavard, les messages sont plus nombreux et donc plus détaillés.
- L'option -dao signifie Disk At Once, gravure du disque d'un seul tenant.

Une fois la gravure terminée vous avez un beau CD contenant les données que vous vouliez sauvegarder.

Le processus décrit ici est ce qui peut se faire de plus simple. Nous avons par exemple passé sous silence le sujet de la vitesse de gravure et laissé la machine se débrouiller pour trouver une valeur par défaut qui convienne. Si vous le souhaitez vous pouvez régler la vitesse de gravure en précisant -speed=8 (par exemple).

5 Copie d'un CD de données ou d'un CD bootable

5.1 Image de CD bootable avec mkisofs

Le cas classique est celui de la copie d'un CD de distribution linux. L'objectif est ici, partant d'un CD qui est bootable, d'obtenir une image qui permette à son tour de graver un CD bootable. La syntaxe décrite ci-dessus est insuffisante pour obtenir ce résultat. Elle conduit à une image ne bootant pas.

La démarche la plus simple consiste à se placer à la racine du CD à copier, par exemple dans /mnt/cdrom/ puis à taper :

```
# mkisofs -v -r -b KNOPPIX/boot.img -o /home/jml/knoppix33.iso .
```

L'exemple qui précède est, vous l'aurez deviné, relatif à la création d'une image d'un CD de la distribution KNOPPIX version 3.3. Options utilisées (en plus des options précédentes) :

- b : définit l'emplacement de l'image de boot, c'est à dire le petit bout de logiciel qui va charger en mémoire et lancer le boot depuis le CD.
- "." : important ici, donne la consigne d'utiliser comme source le répertoire courant (et ses sous-répertoire), donc le contenu du CD.

5.2 Création de l'image iso avec readcd

La commande utile est ici readcd à employer sous compte administrateur :

```
# readcd dev=0,0 f=image.iso
```

Cette commande va créer dans le répertoire dans lequel vous l'avez lancé un fichier image du CD introduit dans votre graveur. Vous noterez que readcd nécessite les droits d'administrateur et ne travaille qu'avec le graveur, non le lecteur de CD.

Vous aurez certainement remarqué que cette commande fait double emploi avec mkisofs ce qui est assez juste. En fait les deux commandes sont équivalentes à ceci près que la syntaxe de

readcd est plus simple et ne change pas selon que le disque est un simple disque de données ou un disque bootable. Il vous restera ensuite à graver l'image exactement comme si cette image avait été créée par mkisofs.

Une autre possibilité consiste à utiliser la commande dd qui est l'une des commandes de copie standard de LINUX. Avantage, dd est capable de lire depuis un lecteur CD ide et ne nécessite pas d'être logué comme administrateur. Inconvénient : alors que readcd essaye de relire un secteur qui aurait été mal reçu, dd continue sans correction. Il en résulte que le résultat est moins fiable. Syntaxe :

```
\$ dd if=/dev/hdc of=image.iso
```

Pour connaître "l'adresse" de votre lecteur cd je vous invite à lire le contenu du fichier /etc/fstab.

5.3 Gravure de l'image

L'image iso étant créée il reste à la graver en suivant le processus décrit plus haut.

Remarque importante : J'ai exposé plus haut la méthode pour créer un CD bootable à partir d'un CD existant, bootable lui aussi, au moyen de mkisofs. Readcd présente l'avantage de proposer une méthode beaucoup plus simple que mkisofs puisque l'indication de l'emplacement de l'image de boot et de son catalogue n'est pas nécessaire.

6 Copie directe de CD de données

Attention, la méthode décrite ci-après n'est pas valable pour la copie de CD audio. Elle est par contre très efficace pour effectuer des copies 1 :1 de CD de données à condition de disposer sur la machine utilisée d'un lecteur et d'un graveur séparés.

A la réflexion d'ailleurs la syntaxe est assez évidente :

```
# cdrecord -v dev=0,0 -isosize /dev/hdc
```

Le principe de base consiste à remplacer le nom de l'image iso par le nom du lecteur, ici /dev/hdc. L'indication -isosize est nécessaire dans ce cas car elle permet de se débarrasser des deux blocs de fin de piste ajoutés au CD source par le graveur en mode TAO. J'ai essayé cette méthode et vérifié qu'elle est également applicable au cas des CD bootables. J'ai ainsi fait une copie de KNOPPIX avec un minimum d'efforts.

7 Rippage de pistes audio

L'objectif est ici de récupérer toutes les pistes d'un CD ou alors de ripper une piste audio en particulier pour en faire un fichier .wav. La commande cdparanoia s'applique dans ce cas :

```
\$ cdparanoia -v -B -d /dev/scd0
```

ou encore

```
\$ cdparanoia -v ``2'' -d /dev/scd0 pénitencier.wav
```

Cdparanoia accepte un certain nombre d'options. Explication de celles qui sont utilisées ici :

- -v, mode bavard
- -B, mode batch, les fichiers se verront attribuer un nom très explicite en track*.cdda.wav.
- -d, device, par exemple /dev/hdc sur ma machine pour le lecteur CD ou /dev/scd0 pour le lecteur-graveur
- "x", permet de désigner la piste à ripper.

L'avantage de la seconde syntaxe est qu'elle permet de renommer le morceau avec un nom qui soit plus explicite que le track*.cdda.wav standard.

8 Création de CD audio

Le schéma de création est loin d'être aussi simple que dans le cas des données. Le format d'enregistrement est spécifique à l'audio et permet de relire le CD aussi bien sur PC que dans un lecteur de salon ou encore un baladeur. Attention : les baladeurs ne supportent pas les CD-RW. N'employez donc que des CD-R si vous envisagez la possibilité d'utiliser le CD dans un tel appareil.

L'objectif ici est de créer de toutes pièces un cd audio en partant de fichiers .wav préalablement enregistrés sur le disque dur. L'origine de ces fichiers peut être multiple, soit ces fichiers ont été téléchargés, soit ils ont été créés par copie préalable depuis un CD audio, soit même ils ont été fabriqués de toute pièce par un logiciel de musique comme Brahms. Comme pour les données la commande cdrecord peut être employée ici, par exemple :

```
\$ cdrecord -v -dao dev=0,0 audio*.wav
```

ou encore

```
\$ cdrecord -v -dao -pad dev=0,0 audio*.wav
```

La première syntaxe correspond au transfert vers le CD de fichiers .wav dont le nom commence par audio. La seconde syntaxe est précieuse pour copier des fichiers de synthèse générés par un logiciel de création musicale. L'option -pad permet d'éviter le rejet par cdrecord de fichiers dont la longueur ne correspond pas à un nombre entier de blocs de 1/75ème de seconde. Attention lors de la création de ces fichiers, cdrecord n'accepte que les fichiers 16 bits stéréo échantillonnés à 44.1 kHz. Il est donc illusoire d'espérer gagner de la place en trichant sur ces paramètres.

Notez que le format des fichiers audio .wav n'est pas directement copiable sur CD. Le logiciel cdrecord va donc faire appel à une 'moulinette' de conversion appelée sox pour transformer le format .wav en format audio CD. Le lecteur se reportera utilement à la page de manuel de sox, un logiciel dont les possibilités vont bien au delà de la simple conversion décrite ici. Grâce à sox cdrecord est en mesure de traiter directement les formats .wav et .au (SUN). Concernant les fichiers mp3 de plus en plus démocratisés il convient de mettre en oeuvre un convertisseur nommé mpg123 (mpg123 -cdr - track1.mp3 > track1.cdr). Voir l'aide ou la manpage pour plus de détails.

9 Informations annexes

9.1 Emploi de CD-RW

Le CD-RW réagit exactement comme un CD-R normal mais présente l'avantage de pouvoir être effacé, alors que le CD-R est à écriture unique. L'effacement est un processus voisin d'une écriture et s'obtient avec :

```
\$ cdrecord blank=fast
```

Une variante est d'utiliser `blank=all`, la différence réside dans le fait que `all` efface tout le disque alors que `fast` n'efface que la table d'allocation des fichiers.

9.2 Précautions

La gravure d'un CD est une opération qui doit être continue. Elle ne supporte aucune interruption. Il en découle qu'une activité débordante du PC risque de planter la gravure. Il reste que le risque n'est tout de même pas énorme et que sur une machine moderne vous ne devriez pas avoir de soucis. Pour ma part j'ai déjà scanné des photos alors que je gravais, sans le moindre inconvénient.

10 CD multi-sessions

10.1 CD Extra

Le CD extra, également appelé enhanced CD, combine le son et les données, mais il le fait de telle sorte :

- que les fichiers audios continuent à pouvoir être lus avec un lecteur CD standard, baladeur, lecteur de salon etc. y compris un lecteur CD monté sur un PC et managé par un logiciel de reproduction audio, `kscd` ou autre.
- que les fichiers binaires puissent être lus sur tout lecteur de CD de PC.

La technique consiste, bien évidemment à enregistrer les deux types de données. L'astuce consiste toutefois à le faire dans le bon ordre. Le son devra être enregistré avant les fichiers binaires faute de quoi nous aurons un CD qui ne sera pas extra (je n'allais tout de même pas la rater celle-là non ?) mais qui correspondra à une autre architecture de CD nommée ****.

10.1.1 Récupération des pistes audios

`Cdparanoia` est l'instrument adapté pour récupérer les pistes audios. Voir plus haut la syntaxe se cette commande.

10.1.2 Récupération des fichiers vidéos

Là l'affaire est plus simple, il suffit de copier les fichiers avec `cp`.

10.1.3 Graver les fichiers audios

Ici c'est à la commande `cdrecord` que nous allons faire appel avec la syntaxe suivante :

```
# cdrecord -v -dao -multi -pad dev=0,0,0 track*.wav
```

La syntaxe est relativement claire, avec toutefois quelques nouveautés :

- `multi` : indique à `cdrecord` de ne pas clore l'inscription sur le disque et de laisser la possibilité d'ajouter des informations.
- `pad` : arrondi éventuellement le longueur du fichier `.wav` en nombre entier de blocs de 1/75ème de secondes, faute de quoi, si l'arrondi n'est pas correct, `cdrecord` va refuser de graver.

10.1.4 Récupérer les informations de limites de sessions

La première session, contenant les fichiers audios, étant en place il convient maintenant d'ajouter les fichiers vidéos. Cet ajout nécessite la création d'une image iso, laquelle doit impérativement tenir compte des dimensions de la session précédente de façon à compléter la table d'allocation des fichiers. La récupération des informations de sessions est confiée à `cdrecord` :

```
# cdrecord -msinfo dev=0,0,0
```

commande qui va déclencher l'affichage d'une réponse du genre :

```
# 0,124682
```

Les deux valeurs qui s'affichent en réponse correspondent aux limites de la première session.

10.1.5 Créer l'image iso

Restons dans les grands classiques, c'est encore `mkisofs` qui va nous rendre ce service, ici avec l'option `-C`.

```
# mkisofs -R -o sessions2.raw -C=0,124682 -M dev/scd0 /home/session2/
```

Les nouveautés de cette syntaxe sont les suivantes :

- `C` : Préfixe les limites de sessions lues plus haut.
- `M` : Définit l'emplacement définitif de l'image.

10.1.6 Graver l'image

`Cdrecord` permet simplement de graver cette image à la suite des fichiers audio.

Deuxième partie

La gravure sous FreeBSD

11 Introduction

11.1 L'émulation SCSI

Bien évidemment, FreeBSD étant de la même famille que LINUX, on va retrouver sensiblement les mêmes logiciels, les mêmes syntaxes, et donc à priori l'essentiel de ce qui a été écrit plus haut s'appliquera encore. Toutefois les choses ne sont pas tout à fait aussi simple. Il convient en effet de souligner que, pour prendre en charge l'émulation SCSI qui permettra d'employer cdrecord, readcd (et, par dessus cdrecord les interfaces graphiques xcdroast, gcombust et K3b par ex.), une compilation du noyau est nécessaire. Cette information est la seule mauvaise nouvelle de cette partie. Voilà maintenant les bonnes nouvelles :

- La compilation du noyau est vraiment une opération très simple et à la portée de tout utilisateur un tant soit peu méthodique. Dans la suite nous distinguerons deux les cas, noyau standard et utilisation d'un périphérique ATAPI ou noyau recompilé et emploi de périphérique SCSI ou émulant SCSI.
- Il n'est pas obligatoire de passer la l'émulation SCSI pour graver sous FreeBSD, burncd est une application mode texte assez semblable à cdrecord, mais qui est utilisera directement le graveur ATAPI qui équipe votre machine !

Admettez que ces deux nouvelles, surtout la dernière, sont tout de même agréables non ?
Le tableau étant brossé voyons maintenant les détails.

12 Compilation du noyau pour émulation SCSI

Pour ce qui est de la technique de compilation je vous renvoie à mon article "Compilation du noyau FreeBSD, application à la mise en place d'un firewall" sur le site du Mirabellug (<http://www.mirabellug.org>)
Le noyau FreeBSD 5.2.1 sera correctement configuré quand vous aurez ajouté la ligne :

```
device atapicam # ajout jml pour émulation SCSI
```

dans le paragraphe #SCSI peripherals. N'oubliez pas de commenter correctement de façon à retrouver ultérieurement ou ont été faites mes modifications.

12.1 Les droits d'accès

Dans la première partie consacrée à LINUX j'ai fait un tableau des logiciels couramment utilisés et des droits nécessaires à les utiliser. Dans le cas de FreeBSD un tel tableau ne s'impose pas, les commandes qui mettent en oeuvre les lecteurs / graveurs de CD imposent des droits

d'administrateurs. On pourra bien entendu travailler sur cette affirmation et faire que l'utilisateur lambda soit abonné au groupe "operator" ce qui étendrait ces droits, cette approche dépasserait toutefois le cadre de cette présentation.

13 Création de CD de données

Ce chapitre s'applique aussi bien au graveur ATAPI qu'au graveur SCSI.

13.1 Création de l'image

Vous utiliserez ici mkisofs exactement comme il a été exposé plus haut, paragraphe 4.1 page 4, pour LINUX.

13.2 Test de l'image créée

Là les choses changent un peu sans se compliquer vraiment. La première étape va consister à créer un nouveau "device" correspondant au fichier iso que l'on souhaite monter. Cette notion de device est assez familière aux utilisateurs de LINUX et autres UNIX, donc rien de bien compliqué ici. La syntaxe à employer sous compte root est la suivante :

```
# mdconfig -a -f fichier.iso
```

En retour vous obtiendrez un numéro de device du genre md0, md1 etc.. N'hésitez pas à vérifier que le sous répertoire /dev/ contient maintenant une nouvelle entrée /dev/md0, /dev/md1 etc..

Les options ont les significations suivantes

- a signifie "attach", cette option crée donc l'entrée dans /dev
- f fichier.iso, désigne simplement le fichier sur lequel doit porter l'opération.

Arrivé à ce stade vous aurez deviné qu'il nous reste à "monter" effectivement le fichier, cette fois avec une commande tout à fait classique sous FreeBSD :

```
# mount_cd9660 /dev/md0 /mnt/test
```

et le tour est joué.

En variante à cette syntaxe typiquement BSDienne vous pourrez tester :

```
# mount -t cd9660 /dev/md0 /mnt/test
```

laquelle est très proche de celle qui est employée sous LINUX à ceci près que le iso9660 classique est remplacé ici par cd9660.

Contrairement au cas du montage d'une clef USB sous FreeBSD les droits d'accès au montage ne sont pas ceux d'origine du sous répertoire sur lequel l'opération a été réalisée mais sont forcés en 755.

13.3 Graver l'image

L'image ISO est simplement gravée au moyen de la syntaxe :

```
# burncd -f /dev/acd1 -s 20 data fichier.iso fixate
```

dans laquelle :

- -f désigne la cible de la gravure, ici le second appareil (le premier est un lecteur simple et est désigné par /dev/acd0).
- -s 20 dicte la vitesse à laquelle doit s'effectuer la gravure, ici 20.
- data annonce le nom du fichier image iso.
- fixate donne l'ordre de "fixer" le cd, en clair de clore la session ouverte.

La gravure se déroule ensuite sans histoire majeure. Il n'est pas obligatoire de fixer la vitesse de gravure. Sans indication explicite le graveur va faire son travail à un pas de sénateur, 4x ! En clair on oublie une fois, deux fois mais rarement plus !

14 Copie d'un CD de données ou d'un cd bootable

14.1 Emploi de dd

Le plus simple ici est d'utiliser la commande dd laquelle vous permet de copier directement le contenu d'un CD sur un fichier. Attention, le contenu du CD étant au format ISO le fichier obtenu sera une image ISO. De fait nous sommes ici dans une logique assez semblable à celle décrite pour LINUX avec la récupération du contenu du CD par readcd. L'image ainsi obtenue peut ensuite être directement gravée au moyen de burncd comme décrit ci-dessus.

La syntaxe de cette commande est la suivante :

```
# dd if=/dev/acd0 of=image.iso bs=2048
```

dans laquelle :

- if= désigne le périphérique d'entrée, ici de premier lecteur ATAPI de la nappe.
- of= désigne le périphérique de sortie, ici le fichier image.iso.
- bs fixe la taille des blocs à transférer, transfert par paquets de 2048 octets dans le cas d'un CD.

Vous noterez, qu'à la différence de LINUX, du moins les dernières Mandrake, dd ne peut être employé que par root.

14.2 Emploi de readcd

La commande readcd ne peut être employée que si l'émulation SCSI est en place. La syntaxe est très proche de celle de readcd sous LINUX à ceci près que l'indication de périphérique d'entrée doit obligatoirement être complète, par exemple dev=1,0,1 la forme simplifiée à deux arguments utilisable sous LINUX conduit à une erreur sous FreeBSD. Pour plus de détails se reporter au paragraphe [5.2](#) page [6](#).

15 Copie directe de CD de données

Voir à ce sujet la paragraphe 6 page 7. La commande `cdrecord` s'utilise ici avec la syntaxe :

```
# cdrecord -v dev=1,1,0 -isozsize /dev/acd0
```

dans laquelle `dev=1,1,0` représente le périphérique de gravure (syntaxe à trois arguments) et `/dev/acd0` le périphérique source.

Ici aussi la syntaxe LINUX à deux arguments (`dev=1,0`) conduit à une erreur. Vous adapterez bien entendu les arguments à votre cas, éventuellement en faisant un petit `cdrecord -scanbus` pour obtenir un état des lieux.

16 Rippage et gravure de pistes audio

16.1 Rippage

Vous vous reporterez à nouveau au paragraphe correspondant dans la section relative à LINUX à savoir 7 page 7.

La syntaxe devient ici :

```
# cdparanoia -v -B -d /dev/acd0
```

ou alors :

```
# cdparanoia -v "2" -d /dev/acd0
```

En peu de mots la syntaxe est la même que la syntaxe LINUX à ceci près que la désignation du lecteur doit être adaptée à la désignation standard sous FreeBSD.

Une autre possibilité offerte ici est d'utiliser la commande `dd`, dans une syntaxe assez similaire à celle qui a été employée pour la copie de CD de donnée :

```
# dd if=/dev/acd0t12 of=track12.cdr bs=2352
```

L'argument de `if` précise cette fois le numéro de piste, ici `t12` pour piste 12, le fichier de sortie est ici au format `.cdr`. Notez que la taille des blocs est cette fois de 2352 octets et non 2048 comme lors de la copie de données.

16.2 Gravure

La commande `burncd` est la première possibilité offerte pour graver des pistes audio préalablement "rippées". La syntaxe est très proche de celle qui a été vue plus haut paragraphe 13.3 page 13.

```
# burncd -f /dev/acd1 audio track* fixate
```

Pour résumer disons qu'il suffit de remplacer l'indication "data" par "audio" pour adapter la syntaxe.

17 Remerciements

J'apprécie tout particulièrement qu'un lecteur me fasse part de son intérêt pour l'un de mes articles. Dans certains cas le lecteur en profite pour demander un renseignement, ce qui donne parfois naissance à des échanges amicaux. Dans d'autres cas c'est l'horreur "pourriez vous m'expliquer comment installer un serveur LAMP complet en 10 lignes". Ce genre de mail reste bien entendu sans réponse de ma part, contribuer ok, être pris pour le larbin de service non.

Dans le cas de cet article je tiens à distinguer M. Jonathan SAVIN lequel m'a fort gentiment remercié d'avoir partagé mes (quelques) connaissances sur le sujet et m'a adressé une liste des erreurs d'orthographe, grammaire et typographies relevées dans une version plus ancienne. En écho à la formulation délicate du mail qu'il a pris le temps de m'adresser j'ai entrepris la correction de mon texte lequel est maintenant un peu plus conforme aux règles de notre belle langue française.

Je salue ici cette initiative et remercie avec plaisir M. SAVIN de sa contribution à ce document.

jml