Mini-HOWTO LILO

Cameron Spitzer, cls@truffula.sj.ca.us

Table des matières

1 Introduction

Traduction française : Eric Dumas (dumas@blaise.ibp.fr) .

Ce fichier décrit quelques installations typiques de LILO. Ce document peut être considéré comme un complément du guide de l'utilisateur de LILO. Je pense que ces exemples sont instructifs, même si votre configuration est différente de la mienne. J'espère que cela vous sera utile.

2 Exemple 1

J'ai deux disques `IDE` et un disque `SCSI`. Ce dernier `SCSI` ne peut être vu par le BIOS. Le chargeur de **Linux**, LILO, utilise des appels du BIOS et ne peux voir que les disques vus par le BIOS. Mon stupide BIOS AMI ne peut donc démarrer que sur "A:" ou "C:". La racine de mon système de fichiers se trouve sur le disque SCSI.

La solution : Mettre le noyau, le fichier mappé et le chargeur dans une partition **Linux** sur le premier disque IDE. Remarquez qu'il n'est pas nécessaire de mettre le noyau sur la partition racine.

La deuxième partition de mon premier disque IDE `/dev/hda2` est montée sur `/u2`. Voici le fichier de configuration que j'utilise :

```
#  Installe le programme de lancement LILO sur le secteur de demarrage
#  du premier disque IDE
#
boot = /dev/hda
```

```
#  /sbin/lilo (programme d'installation) copie le fichier de demarrage LILO
install = /u2/etc/lilo/boot.b

#  J'ai ecrit un texte pour le menu de demarrage. LILO le trouve ici.
message = /u2/etc/lilo/message

#  Le programme d'installation va construire le fichier suivant.
#  Il indique au chargeur ou se trouvent les blocs du noyau.
map = /u2/etc/lilo/map
compact
prompt

#  Attendre 10 secondes, puis lancer le noyau 1.2.1 par defaut.
timeout = 100

#  Le noyau est enregistre a un emplacement visible pour le BIOS
#  en executant :
#      cp -p /usr/src/linux/arch/i386/boot/zImage /u2/z1.2.1
image = /u2/z1.2.1
label = 1.2.1

#  LILO indique au noyau de monter la premiere partition SCSI comme
#  racine.
#  Le BIOS n'aurait pas ete capable de la voir
root = /dev/sda1

#  Cette partition va etre verifiee puis remontee par /etc/rc.d/rc.S
read-only

#  On peut faire monter comme racine plusieurs partitions differentes par
#  un meme noyau. J'ai une configuration en cas d'imprevu sur /u2
image = /u2/z1.2.1
label = 1.2.1-hda2
root = /dev/hda2
read-only

#  J'ai garde une vieille version d'un noyau de la Slackware au cas ou je
#  construise un noyau qui ne fonctionne pas. Je ne m'en suis servi
#  qu'une seule fois.
image = /u2/z1.0.9
label = 1.0.9
root = /dev/sda1
read-only
```

```
#  Ma partition DR-DOS
other = /dev/hda1
        loader=/u2/etc/lilo/chain.b
        label = dos
        alias = m
```

3 Exemple 2

La machine de mon bureau possède un disque IDE d'1 Giga-octets. Le BIOS ne peut voir que les 504 premiers Mo du disque IDE (Mo signifie 2^10 octets et non 10^6 octets).

Donc, j'avais MS-DOS sur une partition /dev/hda1 de taille 350 Mo, et la racine de la partition **Linux** sur /dev/hda2, 120 Mo.

MS-DOS était incapable de s'installer correctement tout seul sur un disque neuf.

Le DOS 7 de Novell avait les mêmes problèmes. Malheureusement pour moi, à la suite d'un oubli d'IBM, je ne disposais pas de la disquette "OnTrack". Le disque était supposé arriver avec un produit appellé "OnTrack Disk Manager". Si vous n'avez que MS-DOS, je vous souhaite d'en disposer.

Donc, j'ai bâti une table de partition avec le fdisk de **Linux**. MSDOS-6.2 a refusé de s'installer dans /dev/hda1, prétextant quelque chose du genre :

'Cette version de MS-DOS est dédiée aux nouvelles installations. MS-DOS est déjà installé sur votre ordinateur ***(ce qui était faux : disque neuf)*** *donc, vous avez besoin d'obtenir une version de mise à jour chez votre vendeur.'*

Quel ignare ! Donc, je relance le fdisk **Linux** et détruis la première partition de la table. Cela convient à MS-DOS 6.2 qui peut alors créer la même partition que celle que je viens de détruire et s'installer. MS-DOS 6.2 écrit alors dans le secteur de lancement du disque, mais impossible de démarrer.

Par chance, j'avais un noyau de la Slackware sur disquette (réalisé par le programme `setup` d'installation), et j'ai donc lancé **Linux** puis écrasé le secteur de démarrage par celui de LILO... et tout marche !

Voici le fichier `/etc/lilo.conf` utilisé :

```
boot = /dev/hda
map = /lilo-map
delay = 100
ramdisk = 0           # Ne cree pas le disque virtuel du noyau Slackware
timeout = 100
prompt

disk = /dev/hda=09      # le BIOS ne voit que les 500 premiers Mo.
   bios=0x80          # indique le premier IDE.
   sectors = 63       # prendre ces chiffres dans la documentation du disque
   heads = 16
   cylinders = 2100
```

```
image = /vmlinuz
  append = "hd=2100,16,63"  # Cela peut etre redondant
  root = /dev/hda2
  label = linux
  read-only
  vga = extended

other = /dev/hda1
  label = msdos
  table = /dev/hda
  loader = /boot/chain.b
```

Après avoir installé ces systèmes, j'ai vérifié que les partitions contenant les fichiers zImage, boot.b, map, chain.b, et messages peuvent utiliser le système de fichiers MS-DOS, tant que ni Stacker, ni Doublespace ne sont installés. Donc, j'aurais pu faire une partition DOS sur /dev/hda1 de 500 Mo.

J'ai également appris que *OnTrack* aurait écrit une table de partitions commençant à quelques douzaines d'octets sur le disque au lieu de l'écrire au début. Il est possible de modifier le gestionnaire de périphérique IDE de **Linux** pour contourner ce problème. Mais l'installation aurait été impossible avec le noyau précompilé de la Slackware. En fin de compte, IBM m'a envoyé une disquette *OnTrack*. J'ai alors téléphoné au support technique de *OnTrack*. Ils m'ont dit que **Linux** était buggé car il n'utilisait pas le BIOS. J'ai renvoyé la disquette.

4 Exemple 3

Ensuite, j'ai installé *Windows-95* sur mon ordinateur au bureau. Il a détruit mon joli secteur de démarrage LILO, mais n'a pas touché à mes partitions **Linux**. Les noyaux sont très lents à charger à partir des lecteurs de disquettes. J'ai donc fait une disquette avec un fichier de configuration pour LILO qui me permet de lancer le noyau se trouvant sur le disque IDE. J'ai fait la disquette de cette manière :

1. `fdformat /dev/fd0H1440`: formatage de la disquette vierge
2. `mkfs /dev/fd0 1440` : disquette au format *Minix*
3. `mkdir /3` : montage
4. `mount /dev/fd0 /3`
5. `cp -p /boot/chain.b /3` : copie du chargeur
6. `lilo -C /etc/lilo.flop` : installation de LILO
7. `umount /3`

Notez que la disquette DOIT ETRE MONTEE LORSQUE VOUS LANCEZ L'INSTALLATION pour que LILO puisse écrire proprement le fichier de configuration.

Ce fichier est `/etc/lilo.flop`. Il ressemble à celui-ci.

```
#  Cree une disquette qui puisse lancer des noyaux sur disque dur
boot = /dev/fd0
map = /3/lilo-map
delay = 100
ramdisk = 0
timeout = 100
prompt

disk = /dev/hda        # 1 Go IDE, BIOS ne voit que les premiers 500 Mo.
   bios=0x80
   sectors = 63
   heads = 16
   cylinders = 2100

image = /vmlinuz
  append = "hd=2100,16,63"
  root = /dev/hda2
  label = linux
  read-only
  vga = extended

other = /dev/hda1
  label = msdos
  table = /dev/hda
  loader = /3/chain.b
```

Enfin, j'avais besoin de MS-DOS 6.2 sur mon ordinateur du bureau, mais je ne voulais pas toucher au premier disque. J'ai donc ajouté un contr=F4leur SCSI et un disque, créé une partition au format MS-DOS avec le programme **`mkdosfs`** de **Linux**, et Windows-95 l'a reconnu comme disque "D:". Mais, bien sûr, MS-DOS ne démarrera pas sur "D:". Ce n'est pas un problème avec LILO. J'ai ajouté :

```
other = /dev/sda1
  label = d6.2
  table = /dev/sda
  loader = /boot/any_d.b
```

au fichier `lilo.conf` de l'exemple 2. MSDOS-6.2 fonctionne en pensant être sur C:, et Windows 95 est en D:.