Le HOWTO Windows 95 + Windows 3.x + Linux

Robert Goodwin, <Robert.Goodwin@mcc.ac.uk>, traduit par Olivier Tharan, <Olivier.Tharan@int-evry.fr>
Août 1996

1 Introduction

Ce document a d'abord été écrit en janvier 1996. J'y ai inclus divers commentaires, informations et questions reçus depuis lors. Ce document est aussi disponible en japonais ; voyez l'ensemble de documentations Linux en japonais à (`http://epsenewsc.kyoto-u.ac.jp/JF/JF.html`).

2 Démarrer plusieurs systèmes d'exploitation

Si vous voulez démarrer plusieurs systèmes d'exploitation (et que vous ne voulez pas les démarrer à partir d'une disquette !), vous avez besoin d'utiliser une sorte de *gestionnaire de démarrage*.

Windows 95 n'a pas *vraiment* de gestionnaire de démarrage – il a des options de démarrage, mais pour moi on peut configurer un "gestionnaire de démarrage" pour démarrer n'importe quoi.

On peut configurer Lilo pour démarrer quasiment n'importe quoi, comme le peuvent les gestionnaires de démarrage d'OS/2 et de Windows NT. Lequel vous utilisez dépend entièrement de vous ; cela dépend de ce que vous voulez sur votre système. Si vous utilisez le gestionnaire de démarrage d'OS/2, par exemple, il vous "cache" les "autres" partitions de genre DOS.

3 Autres endroits à regarder

Si vous voulez juste Windows 95 et Linux, vous aurez sûrement besoin du HOWTO Windows 95 + Linux. On peut trouver la dernière version à :

```
http://www.in.net/~jkatz/win95/Linux-HOWTO.html
```

On peut installer Windows 95 et Linux sur un même disque ; les problèmes couverts dans le reste de ce document tournent autour du fait que DOS ne permet pas d'avoir plus d'*une* partition DOS primaire par lecteur.

Si vous voulez utiliser le gestionnaire de démarrage de NT, jetez un coup d'oeil à (`http://www.bcpl.lib.md.us/ dbryan/directboot.html`). Pour ajouter Linux (ou d'autres, comme OS/2), il y a un petit programme sympa qui fabrique des fichiers de secteurs de démarrage pour les utiliser avec le chargeur NT. Vous pouvez trouver de l'information là-dessus à (`http://www.winimage.com/bootpart.htm`).

Vous pouvez aussi fouiller dans `LOADLIN.EXE` qui vous permet de "démarrer" Linux en démarrant d'abord sous DOS.

4 Pourquoi ce HOWTO?

Ce document traite des problèmes suivants:

- Comment arriver à faire cohabiter Windows 3.x sur la même machine que Windows 95 sans problèmes (et que pourraient être ces problèmes de toute façon);
- Comment éviter les problèmes sur une machine sous Windows 95 qui peut aussi être démarrée sous DOS à distance;
- Comment installer Windows 95 avec Linux sans devoir réinstaller Lilo (avec le démarrage fastidieux de Linux sur disquette).

Bien que le système de fichiers de Windows 95 vive au-dessus de la FAT standard DOS, il lui fait certaines choses déplaisantes. Démarrez votre machine Windows 95 à partir d'une disquette DOS bootable et lancez Norton pour vérifier le disque (mais *ne* le laissez tenter aucune réparation ou vous mettrez en l'air quelques noms de fichiers "longs").

On me demande en ce moment de supporter des applications à la fois dans l'environnement Windows 95 et Windows 3.x (les deux tournent soit en local soit démarrent à partir du réseau). J'ai développé la procédure décrite dans ce document pour me permettre de faire tout ceci avec un seul PC.

Ne demandez pas pourquoi une personne d'Unix a terminé dans cette position `:-)`

5 Eléments nécessaires

Si vous êtes préparé à vous plonger dans les sources de Lilo (Lilo 1.7 ou plus, je crois), il est possible de faire tout ceci avec *un* disque dur. Ceci fonctionne en changeant le contenu de la table des partitions au moment du démarrage du système; si vous n'êtes pas confiant pour essayer ça, ne le faites pas!

Sinon, vous aurez besoin de *deux* disques durs. Ceci est dû à certaines limitations de DOS/Windows concernant le démarrage et l'allocation des lettres de lecteurs. Croyez-moi, j'ai essayé de le faire fonctionner avec un seul (mais je ne voulais pas changer les sources de Lilo).

6 Ce qu'il en ressortira

Un mot sur les noms de périphériques. J'ai vu des systèmes qui utilisent `/dev/hdc` pour nommer le troisième disque IDE (premier disque IDE sur le deuxième contrôleur) et `/dev/hdd` pour le quatrième. J'ai aussi vu des systèmes qui utilisent `/dev/hd1a` et `/dev/hd1b` (ce qui donne des partitions comme `/dev/hd1a3`, etc.). Mon système utilise la deuxième méthode de nommage, mais j'ai changé les noms en `/dev/hdc` et `/dev/hdd` pour limiter les confusions.

Voici une brève description de ce que j'ai maintenant – regardez les lettres de lecteurs parce qu'elles changent...

Si vous utilisez un contrôleur IDE secondaire, vous aurez peut-être besoin de créer les entrées dans `/dev` vous-même (`/dev/hdc*` et `/dev/hdd*`). Ceci peut être le cas si vous ajoutez un deuxième disque à une machine qui a déjà un disque IDE et un CD-ROM IDE ; votre deuxième disque dur serait `/dev/hdc`. J'ai essayé cet arrangement sans problèmes.

J'ai :

```
/dev/hda - premier disque dur /dev/hdb - lecteur CD-ROM /dev/hdc - deuxième disque dur
```

Option 1 :

En allumant la machine, je peux permettre à la *boot ROM* de la carte Ethernet de démarrer DOS à distance. Le lecteur `C:` est la première partition DOS sur le *premier* disque IDE (dans mon cas `/dev/hda1`). Le lecteur `D:` est la première partition DOS sur le *deuxième* disque IDE (dans mon cas `/dev/hdc1`), et le lecteur `F:` est la deuxième partition DOS sur le *deuxième* disque IDE (`/dev/hdc2`). Le CD-ROM devient `F:`.

Option 2 :

Permettre à Lilo de démarrer le système par défaut (Linux, naturellement).

Option 3 :

Interrompre Lilo et lui demander une option que j'appelle `dos`. Ceci démarre DOS à partir de `/dev/hda1` et, comme avec l'option 1, le disque `C:` est `/dev/hda1`, le disque `D:` est `/dev/hdc1` et le disque `E:` est `/dev/hdc2`. Le CD-ROM devient `F:`.

Option 4 :

Interrompre Lilo et demander une option que j'appelle `win95`. Ceci démarre Windows 95 à partir de la première partition DOS sur le *deuxième* disque IDE (dans mon cas `/dev/hdc1`). Suivez ceci avec attention : le disque `C:` est maintenant la première partition DOS sur le *deuxième* disque IDE (`/dev/hdc1`), le disque `D:` est maintenant la première partition du *premier* disque IDE (`/dev/hda1`), et le disque `E:` reste la deuxième partition DOS sur le *deuxième* disque IDE (`/dev/hdc2`). Le CD-ROM devient `F:`.

Remarquez que le disque `C:` change selon votre méthode de démarrage. Ceci veut dire que quand vous installez Windows 95, vous l'installez dans `C:`
`WINDOWS`, et quand vous installez Windows 3.x, vous l'installez dans `C:`
`WINDOWS` mais ce n'est pas le même endroit `:-)`

Remarquez aussi que la troisième partition DOS (que j'utilise comme disque de données général) est `E:` quelle que soit la manière de démarrer, et que le CD-ROM reste aussi constant.

7 Comment le faire

En premier, installez Linux ; le disque sur lequel vous l'installez n'a pas d'importance, mais comme vous utilisez deux disques il est sensé de créer une partition de swap sur les deux.

Créez une partition DOS primaire sur le deuxième disque. Malheureusement, le programme DOS `FDISK` ne vous laissera pas faire ceci, vous devez donc utiliser le programme Linux `fdisk` pour créer la partition, mettre le type (6 pour `DOS 16 BIT FAT >32Mb`), positionner la partition bootable, et suivre *attentivement* les conseils de la page de manuel de `fdisk`. Celle-ci décrit comment forcer DOS à reconnaître une partition créée de cette manière en utilisant la commande `dd` pour mettre à zéro les 512 premiers octets de la partition. (En gros, vous faites `dd if=/dev/zero of=/dev/XXXX bs=512 count=1` où `XXXX` est le périphérique mais soyez *très* prudent puisque c'est une bonne méthode pour crasher un disque – par exemple en mettant `/dev/hda` au lieu de `/dev/hda1` !)

Créer la partition DOS primaire sur le premier disque dur peut être fait avec le programme DOS `FDISK`. Toute autre partition DOS nécessaire peut être créée de la même manière.

Les deux partitions primaires doivent être formatées comme des partitions DOS bootables ; utilisez `FORMAT C: /S` et `FORMAT D: /S` en ayant démarré à partir d'une disquette. Pour éviter la confusion, donnez aux partitions des noms de volume reconnaissables !

Editez `/etc/lilo.conf` pour vous donner l'option de démarrer à partir de l'une des deux partitions DOS primaires. J'ai mis en annexe un exemple. Notez l'utilisation de la ligne `loader` dans ce fichier d'exemple. C'est vous qui choisissez sur quelle partition vous installez Windows 95 et Windows 3.x ; j'ai utilisé le deuxième disque dur pour Windows 95 puisque le démarrage par le réseau de la machine donne alors la lettre `C:` pour le disque DOS normal. De même (c'est utile), installer Windows 95 sur le deuxième disque dur évite d'avoir à démarrer Linux à partir d'une disquette de secours et de réinstaller Lilo (pourquoi vous devriez faire ceci est expliqué plus bas). Rappelez-vous de lancer `lilo` pour installer les options.

Si vous pensez (à juste titre) installer Windows 95 à partir d'un CD-ROM, vous devrez inclure les pilotes correspondants sur la partition Windows 95 pour que quand vous démarrez dessus, le lecteur de CD-ROM soit accessible.

Maintenant démarrez, en utilisant Lilo, sur le disque que vous voulez utiliser pour DOS/Windows 3.x et installez le reste de DOS et de Windows 3.x – l'installation de Windows devrait aller dans le répertoire `C:`
`WINDOWS` comme d'habitude.

Une fois ceci fait, redémarrez la machine et, en utilisant Lilo, démarrez sur la partition Windows 95. Lancez l'installation de Windows 95. La procédure d'installation peut suggérer `D:`
`WINDOWS` pour installer Windows 95 parce qu'il recherche sur la machine les versions existantes de Windows – *n'acceptez pas ceci* – installez Windows 95 sur `C:`
`WINDOWS`.

Maintenant, la bonne nouvelle ! Windows 95 est un système plutôt arrogant – quand vous l'installez, il suppose que c'est le seul système d'exploitation sur la machine et continue en écrivant son propre MBR (Master Boot Record) sur le disque dur. Voici pourquoi vous devez en général réinstaller Lilo. Si vous avez installé Windows 95 sur le deuxième disque dur, vous avez fait quelque chose que les programmeurs de Microsoft(tm) n'ont pas pris en compte. Sur une machine où j'ai fait cette opération, Windows 95 a écrit son MBR de remplacement sur le MBR du *deuxième* disque dur. Sur une autre, je n'ai pas trouvé de preuve du tout qu'il l'avait fait. La conséquence pratique de ceci est que le MBR qui compte, celui du premier disque dur, n'est pas changé. Donc quand vous redémarrerez la machine, vous serez accueilli par le prompt amical et familier de Lilo.

Exemple de liste de partitions :

```
/dev/hda1       *       partition DOS                   (C: ou D: selon le demarrage)
/dev/hda2               partition etendue
/dev/hda5               /
/dev/hda6               swap
/dev/hda7               /home

/dev/hdc1       *       partition Windows 95    (C: ou D: selon le demarrage)
/dev/hdc2               partition DOS                   (toujours E:)
/dev/hdc3               swap
```

(Les partitions avec une `*` sont positionnées bootables (ou *actives*) par `fdisk`.)

Exemple de `lilo.conf` :

```
# /etc/lilo.conf
install = /boot/boot.b
compact
delay = 20              # optionnel, pour les systemes qui demarrent tres rapidement
#prompt                 # a utiliser a la place de delay pour forcer une reponse au prompt
#vga = normal   # forcer un etat sain
#ramdisk = 0    # parametre de paranoiaque
#root = current # utiliser la racine "courante"
boot = /dev/hda
image = /boot/vmlinuz
  read-only
  label = linux
other = /dev/hdc1
  label = win95
  loader = /boot/any_d.b
other = /dev/hda1
  table = /dev/hda
  label = dos
image = /boot/vmlinuz.old
  label = linux.old
  optional
  read-only
```

8 Quelques questions et réponses

Est-ce que cette méthode fonctionne avec les disques SCSI ?

On m'a dit que ça fonctionnait, mais je n'ai pas pu l'essayer moi-même.

Est-ce que cette méthode fonctionne si Linux est complètement sur un disque, et DOS et Windows 95 sont sur des partitions sur l'autre disque ?

Non. Les parties DOS et Windows 95 doivent être toutes les deux sur des partitions primaires *DOS* d'un disque. Il est possible de contourner ceci en recompilant Lilo.

J'ai téléchargé une FAQ sur Windows 95 qui parle de la possibilité de démarrer soit DOS soit Windows 95. Si j'installe Windows 3.11 dans un autre répertoire que `windows` elle dit que je peux faire tourner Windows 95 et Windows 3.11 en harmonie. Il y a une partition.

Oui, on peut le faire. Ceci peut cependant causer des problèmes. Windows 95 fait des choses bizarres sur le disque FAT et certaines opérations que vous effectuerez avec vos anciens DOS et Windows (3.x) peuvent facilement détruire les informations sur les noms de fichiers longs. Par exemple, défragmenter le lecteur en utilisant un utilitaire DOS/Windows 3.x le fera. Vous vous créez aussi des difficultés quand ça ne fonctionne pas ; vous devez vous soucier des fichiers `INI` sur chaque système *et* de la *registry* Windows 95.

Est-ce que le LBA est important ?

Oui. Houla ! Les BIOS postérieurs à environ 1994 supportent le LBA pour contourner une limite quelque part dans le DOS qui empêche DOS de pouvoir se débrouiller avec des numéros de cylindres supérieurs à 1024 (ceci est couvert en plus grands détails dans les FAQs sur le matériel PC). Le LBA joue avec la géométrie des disques , en multipliant le nombre de têtes par 2 ou 4 (etc.) pour diviser le nombre apparent de cylindres par 2 ou 4 (etc.) et obtenir un nombre de cylindres inférieur à 1024. Ceci contourne la limite.

Linux peut manipuler plus de 1024 cylindres (à condition que la partition à partir de laquelle vous démarrez soit entièrement en-dessous de 1024 cylindres), et peut donc manipuler des grands disques (supérieurs à 504 Mo) même sur des vieilles machines (BIOS antérieur à 1994). Il peut aussi manipuler des grands disques avec les BIOS qui supportent le LBA, que le LBA soit activé ou pas.

Il est vital que *tous* les systèmes d'exploitation voient chaque disque sous la même géométrie – ceci parce que les numéros de la table des partitions sont les numéros de cylindres *perçus*, et non les numéros réels. Ainsi, changer les paramètres du BIOS pour activer le LBA invalidera le contenu existant d'un disque.

Si votre système Linux ne *voit* pas la géométrie *correcte* (c'est-à-dire la même que *voit* DOS), vous devrez ajouter une ligne dans `lilo.conf` : `append="hd=x,y,z"` où `x`, `y`, `z` représentent la géométrie du disque (voyez les pages de manuel correspondantes).

Mon BIOS ne contient des informations que sur deux disques durs et non quatre. Est-ce que ça a de l'importance ?

Sûrement ! Avoir quatre disques durs avec des vieux BIOS, sous DOS demandait l'utilisation d'un logiciel pilote. Les BIOS plus récents contiennent les informations sur les quatre disques.

Linux peut sans problèmes utiliser quatre disques durs même avec la plupart de ces BIOS plus anciens, mais si vous mettez une partition DOS sur le disque 3 vous ne serez capable d'y accéder que par Linux.

Ceci est pertinent puisque beaucoup de machines possèdent un disque IDE, un CD-ROM IDE et, si vous voulez mettre ce document en pratique, un autre disque IDE.

Si vous avez trouvé ce document utile, merci de me le faire savoir.