Mini-HOWTO Gros IDE

Ou comment utiliser ces gros hideux sous Linux et DOS

Patrick LoPresti, patl@lcs.mit.edu, version française par Hervé Mignot
v1.1, 19 Janvier 1995.

Les noyaux des versions de Linux ultérieures à la version 1.1.40 de Linux gèrent sans aucune difficulté de gros disques IDE (c'est-à-dire comportant plus de 1024 cylindres). Toutefois, de nombreuses distributions utilisent des versions de noyau plus anciennes sur leur disque d'installation, et ce Mini-HOWTO peut donc vous être utile[1].

Table des matières

1 Introduction **1**

2 Un peu de technique et de terminologie **1**

3 Le problème **2**

3.1 La mauvaise solution 2

3.2 La bonne solution 3

3.3 Exemple résumé 3

1 Introduction

Il est possible d'utiliser Linux avec de gros disques durs EIDE sans aucune restriction. Seuls les cas de DOS et Linux seront abordés dans ce document, mais je pense que l'approche présentée ici est utilisable avec tout autre système d'exploitation.

J'utilise actuellement la version 1.0.9 du noyau, mais tout ce qui est dit ici s'applique pareillement jusqu'à la version 1.1.34. Les versions ultérieures de Linux gèrent complètement les disques EIDE, ce qui réduit l'intérêt de tout ce qui suit. Je pense toutefois que cela continuera de fonctionner avec les versions de noyau à venir, même si cela devient inutile.

2 Un peu de technique et de terminologie

Les secteurs des disques ATA (IDE) comportent 512 octets. On peut adresser un secteur de deux manières : par son adresse logique (*LBA, Logical Block Address*), qui est un numéro compris entre 0 et le nombre total

1. Il servira surtout aux personnes utilisant conjointement Linux et MS-DOS, sur une même machine. En effet, si le BIOS de cette machine n'est pas un BIOS EIDE, il est nécessaire de procéder comme décrit dans ce mini-HOWTO pour permettre une cohabitation tranquille des deux systèmes d'exploitation. (*N.D.T.*)

de secteurs du disque, ou par son adresse physique qui est un triplet cylindre-tête-secteur (*CHS, Cylinder-Head-Sector*). Pour convertir des adresses logiques en adresses physiques, il est nécessaire de connaître la *géométrie du disque*, c'est-à-dire le nombre de têtes par cylindre, le nombre de secteurs par tête et le nombre de cylindres qu'il comporte. Ces informations permettent aussi de calculer la taille du disque.

Les anciens contrôleurs et les vieux BIOS ne permettent d'utiliser que l'adresse physique pour adresser un secteur. Tous les contrôleurs et tous les BIOS permettent d'utiliser l'adresse physique pour repérer un secteur. Linux manipule le plus souvent possible des adresses logiques, sauf au plus bas niveau, où il effectue une conversion de ces adresses en adresses physiques pour dialoguer avec le contrôleur. Linux n'utilise pas le BIOS pour tout cela, sauf pour déterminer la géométrie du disque.

La table des partitions contient les adresses de début et de fin des partitions sous forme logique ET physique. *fdisk* sous DOS et *fdisk* sous Linux s'attendent bien sûr à ce que les deux formes soient cohérentes entre elles pour chaque partition. *fdisk* sous DOS demande au BIOS la géométrie du disque, et *fdisk* sous Linux la demande au noyau.

C'est ici que les choses se corsent. MS-DOS et l'interface BIOS utilisent un champ de dix bits pour stocker les numéros de cylindre, ne permettant d'utiliser que des numéros compris entre 0 et 1023, ce qui est tout à fait insuffisant pour les disques modernes qui comportent habituellement 63 secteurs par tête, 16 têtes par cylindre, et un très grand nombre de cylindres (*N.D.T.* plus de 2 000 pour un disque de 1 Go). La solution pour pouvoir exploiter ces disques, en dépit de cette limitation est une bidouille : un BIOS EIDE ment au sujet de la géométrie du disque en divisant par deux (ou par quatre) le nombre de cylindres et en doublant (respectivement, en quadruplant) le nombre de têtes. Quand une requête comportant une adresse physique lui arrive, le BIOS considérera que cette requête utilise la géométrie bidon et la convertira pour dialoguer avec le contrôleur du disque. Ce mécanisme est appelé une *conversion d'adresse*.

3 Le problème

Quand le noyau Linux demande au BIOS la géométrie du disque (il se contente en fait de lire les paramètres CMOS), il obtiendra une réponse erronée indiquant qu'il y a plus de seize têtes. Mais Linux sait que cela n'est pas possible, et le noyau (source dans `hd.c`) abandonne et ignore le disque. (Notons que même si le BIOS retourne une géométrie trafiquée, les requêtes faites au contrôleur de disque doivent se faire conformément à la géométrie réelle du disque. Éliminer le test effectué dans `hd.c` ne suffit donc pas.)

3.1 La mauvaise solution

Une solution est d'utiliser le programme de paramétrage (*setup*) du BIOS pour supprimer complètement la conversion d'adresse. On peut soit fixer soi-même le nombre de cylindres/têtes/secteurs conformément à la géométrie du disque, soit simplement inhiber la conversion d'adresse selon ce que permet son programme de paramétrage du BIOS.

Vous pouvez alors repartitionner votre disque dur avec *fdisk* sous DOS et/ou sous Linux, puis installer les deux systèmes d'exploitation. Cela fonctionnera sous réserve d'observer les restrictions suivantes.

Le problème reste que le BIOS ne peut être utilisé pour accéder aux cylindres au-delà du 1024-ième. Toutes vos partitions DOS devront donc se situer avant cette limite, ainsi que toutes informations auxquelles le

BIOS aura besoin d'accéder. Ainsi, LILO utilise le BIOS pour faire le sale boulot et donc, si vous voulez utiliser LILO pour lancer Linux, vous devez vous assurer que le noyau (plus précisément toute la partition racine) se situe avant cette limite du 1024-ième secteur. Linux, lui, accèdera sans problème à l'intégralité du disque.

3.2 La bonne solution

Les limitations sont toujours gênantes, et nous allons donc nous débrouiller pour les éviter. Nous devons donc continuer à utiliser le mécanisme de conversion d'adresses et résoudre chacun des problèmes que cela engendre.

Pour résoudre le problème au niveau du noyau, nous allons passer en paramètre la véritable géométrie au noyau grâce à une ligne d'option au démarrage. Ceci peut se faire à partir de l'invite d'initialisation de LILO, en tapant `<image name> hd=<nb. cylindres>,<nb. tête>,<nb. secteurs>`. On peut aussi utiliser un ordre `append=` dans le fichier de configuration de LILO (*lilo.conf*) pour que cela se fasse automatiquement.

Le noyau peut maintenant reconnaître et utiliser le disque, mais lorsqu'un programme utilisateur (comme *fdisk* ou le programme d'installation de LILO) demande au noyau la géométrie du disque, le noyau lui renvoie la vraie géométrie et non celle obtenue après conversion. Et donc *fdisk* sous Linux (qui interroge le noyau) et *fdisk* sous DOS (qui interroge le BIOS) n'utilisent pas la même géométrie, ce qui est très gênant au niveau des informations à placer dans la table des partitions du disque dur. De même, le programme d'installation de LILO calculera des adresses physiques incompatibles avec ce dont le BIOS (donc LILO lors du démarrage) a besoin.

Le problème avec *fdisk* est simple à résoudre : lorsque vous lancez *fdisk* sous Linux, passez en mode expert, et fixez le nombre de cylindres/têtes/secteurs aux valeurs utilisables par le BIOS. Puis éditez votre table des partitions et écrivez-la, *fdisk* sous Linux et sous DOS seront ainsi d'accord.

Le problème de LILO est aussi simple à résoudre : ajouter l'ordre `linear` à votre fichier *lilo.conf*. LILO utilisera alors des adresses logiques et non des adresses physiques, l'obligeant ainsi à calculer les adresses physiques au démarrage et non lors de son installation. C'est donc la géométrie fournie par le BIOS qui sera utilisée, plutôt que celle fournie par le noyau, et ainsi tout fonctionnera.

3.3 Exemple résumé

Nous disposons d'un disque EIDE d'un giga-octet sur lequel nous allons installer Linux. Voici comment procéder.

1. Noter au niveau du programme de paramétrage (*setup*) du BIOS la géométrie trafiquée du disque. Par exemple, il est indiqué que le disque comporte 525 cylindres, 64 têtes, et 63 secteurs. Comme il n'est pas possible qu'il y ait plus de seize têtes sur ce disque, on en déduit qu'il comporte en fait 2100 cylindres, 16 têtes et 63 secteurs.

2. Démarrer sous DOS, lancer `fdisk` pour créer une partition DOS.

3. À partir d'une disquette de démarrage de la Slackware, taper `ramdisk hd=2100,16,63` avant le lancement du chargement du noyau. Puis lancer `fdisk` (sous Linux). Si on tape `p`, un tas de messages

d'erreur apparaît. Passer en mode expert `x`, fixer le nombre de cylindres à 525, le nombre de têtes à 64. Une fois revenu en mode normal, si on tape `p` pour voir la liste des partitions, il n'y a plus d'erreur. Il est alors possible créer ses partitions. Écrire la table, puis redémarrer avec la disquette de la Slackware (il est conseillé de redémarrer la machine après avoir modifié la table des partitions).

4. Procéder ensuite à l'installation de la Slackware, normalement, en créant (entre autres) un fichier `lilo.conf` classique.

5. Éditer le fichier `lilo.conf` (dans ce cas `/mnt/etc/lilo.conf`) pour y ajouter les lignes suivantes au début :

```
append="hd=2100,16,63"
linear
```

6. Lancer la commande `lilo -r /mnt`.

7. Installer, éventuellement, si c'est absolument nécessaire, DOS et/ou Windows.

Bonne chance à tous !

Patrick LoPresti

patl@lcs.mit.edu