

# Créer une disquette de démarrage GRUB pour Knoppix

Gazette Linux n°108 — Novembre 2004

Anonymous

Copyright © 2004 Anonymous

Copyright © 2004 Gauthier Savart

Copyright © 2004 Joëlle Cornavin

Article paru dans le n°108 de la Gazette Linux de novembre 2004.

Traduction française par Gauthier Savart <gauthier POINT savart CHEZ laposte POINT net>.

Relecture de la traduction française par Joëlle Cornavin <jcornavi CHEZ club TIRET internet POINT fr>.

Article publié sous Open Publication License (<http://linuxgazette.net/copying.html>). La Linux Gazette n'est ni produite, ni sponsorisée, ni avalisée par notre hébergeur principal, SSC, Inc.

## Table des matières

- 1. Cas général : disque visible par le BIOS .....1
- 2. Cas de figure anormal : disque non visible par le BIOS .....2

GRUB ne peut pas démarrer le cédérom de Knoppix si l'option Démarrage à partir du CD-ROM est désactivée. C'est une fonctionnalité que les développeurs de GRUB devraient penser à ajouter. Lisez mon article précédent (<http://ftp.traduc.org/doc-vf/gazette-linux/html/2004/107/lg107-C.html>) dans la Gazette Linux d'octobre 2004.

Néanmoins, GRUB (version actuelle 0.95) peut générer une disquette permettant d'amorcer Knoppix après une installation complète sur disque dur — tâche qui semble poser quelques problèmes à LILO en raison de son incapacité à chercher des fichiers et du fait que le noyau, ainsi que le disque virtual initial ont une taille croissante qui dépasse la capacité d'une disquette.

Supposons que Knoppix soit installé sur le disque dur mais que sa partition ne soit pas amorçable, pour quelque raison que ce soit. Vous avez alors besoin d'une disquette de démarrage Knoppix que nous allons créer avec GRUB.

## 1. Cas général : disque visible par le BIOS

Vous pouvez utiliser une simple disquette GRUB qui démarre sur une ligne de commande GRUB. À partir de cette ligne de commande, vous pouvez lancer des commandes pour démarrer votre Knoppix sur

la partition non amorçable.

Cette simple disquette simple de démarrage GRUB peut être préparée avec le script `grub-floppy`. Elle est très rapide à créer mais elle ne contient pas de système de fichiers, ce qui en limite considérablement l'utilisation. Vous ne pouvez pas y écrire et vous devrez saisir au clavier quelque temps après avoir amorcé, à l'invite de GRUB.

Le script `grub-floppy`, qui nécessite une disquette formatée, offre une solution plus confortable. Une disquette au format DOS conviendra et vous pourrez la modifier à partir de DOS — un avantage non négligeable. Montez la disquette, par exemple sur `/floppy` et saisissez la commande :

Ce script va créer le répertoire `/floppy/boot/grub` et y copier les fichiers images requis pour amorcer. Vous pouvez en supprimer certains, puisque vous n'avez pas besoin de la prise en charge d'une demi-douzaine de systèmes de fichiers.

Vous avez à présent l'avantage de pouvoir écrire dans le répertoire `/floppy/boot/grub` un fichier texte nommé `menu.lst`, qui est ici le fichier de configuration de GRUB. Vous y placerez toutes les instructions que vous souhaitez pour démarrer votre Knoppix et c'est terminé ! (Il se peut que la séquence de fin de ligne soit de style DOS ou Linux). La disquette démarre assez rapidement — ou mieux encore, elle démarre très rapidement si vous réduisez le temps d'attente à 1 seconde et que vous supprimez le menu.

## 2. Cas de figure anormal : disque non visible par le BIOS

Occupons-nous maintenant d'une petite astuce qui semble un peu étrange.

Le disque dur sur lequel est installé Knoppix n'est pas visible par le BIOS. Pour quelle raison ? Parce que le disque est explicitement configuré en `none` dans le BIOS, pour quelque raison particulière. Ou bien parce que les BIOS anciens ne peuvent détecter les partitions supérieures à 8 Go.

Linux peut maintenant démarrer, voir le disque même s'il est invisible par le BIOS et y installer son répertoire racine.

La préparation des disquettes GRUB pour réaliser cette astuce suit la même méthode que précédemment, mais quelques ajustements complémentaires sont nécessaires.

Concentrons-nous sur le cas où une disquette de démarrage au format DOS a été préparée avec `grub-install`.

Premièrement, copiez votre noyau Linux (appelez-le `vmlinuz`) sur un disque dur visible par le BIOS, par exemple le premier disque dur, la première partition, que GRUB appelle `(hd0, 0)`.

Deuxièmement, vous devez choisir la désignation Linux appropriée pour la partition Knoppix, quand Linux tourne. Souvenez-vous que Linux appelle les disques SCSI `/dev/sda`, `/dev/sdb`, etc. et les disques IDE `/dev/hda`, `/dev/hdb`, etc. Les partitions y ont ensuite accolées sous forme de numéros, par exemple `/dev/hda1`, `/dev/hda2`, etc. Cela peut devenir complexe, selon les périphériques que vous avez dans votre PC. En tout état de cause, vous devez trouver la désignation appropriée pour la partition Knoppix puisqu'elle est invisible pour le BIOS et pour GRUB. Supposez, pour l'exemple, que la désignation Linux correcte est `/dev/hdb1`.

Maintenant, écrivez **menu.lst** sur la disquette, dans le répertoire `/boot/grub`, comme suit :

Nous n'avons pas encore terminé. GRUB contient sur la disquette un fichier auto-généré `device.map`, qui répertorie tous les périphériques qu'il reconnaît. Dans ce cas « anormal », il est judicieux de vérifier

le fichier `map` ; il se peut nécessiter une modification manuelle. En effet, il faut qu'il contienne ce périphérique inconnu du BIOS mais destiné à contenir la racine. Notez que le fichier `map` est simplement une traduction entre les noms de périphériques GRUB en noms de périphériques Linux.

Voilà qui est fait ! Amorcez la disquette GRUB et le système va aller chercher le fichier `vmlinuz` sur votre premier disque dur : `vmlinuz` s'exécute, puis reconnaît et utilise la partition `/dev/hdb1` bien que `/dev/hdb` soit inconnu du BIOS.

Quelqu'un aurait-il une explication ? Si le noyau peut voir des partitions invisibles pour le BIOS, pourquoi GRUB n'y parvient-il pas ? GRUB dispose dans le secteur d'amorçage un petit programme appelé `stage1` qui n'a aucune autre fonction que d'amorcer `stage2`. Du fait que `stage2` fait plus de 100 ko, ne devrait-il pas être capable de détecter les périphériques disponibles de la même façon que le noyau le fait ?