Démarrage

Ok, maintenant que vous avez installé Slackware sur votre système, vous devriez savoir exactement ce qui contrôle la séquence de démarrage de votre machine, et comment remédier à tout problème de démarrer si vous vous retrouver avec un système bloqué. Si vous utilisez Linux assez longtemps, tôt ou tard, vous ferez une erreur qui rendra votre bootloader non opérationnel. Heureusement, la résolution du problème ne nécessite pas une réinstallation du système. Contrairement à de nombreux autres systèmes d'exploitation qui cachent les détails sous-jacents de leur fonctionnement, Linux (et en particulier, Slackware) vous donne plein contrôle sur le processus de démarrage. En modifiant simplement un fichier de configuration ou deux et en ré-exécutant le programme d'installation du bootloader, vous pouvez réparer (ou détruire) rapidement et facilement votre système. Slackware simplifie aussi l'option de dual-boot plusieurs systèmes d'exploitation, tels que d'autres distributions Linux ou Microsoft Windows.

mkinitrd

Avant d'aller plus loin, une brève discussion sur le kernel Linux s'impose. Slackware Linux comprend au moins deux, voire parfois plus, kernels différents. Bien qu'ils soient tous compilés à partir du même code source, et sont donc les *“mêmes”*, ils ne sont pas pour autant identiques. Selon votre architecture et la version Slackware, l'installateur peut avoir installé votre système avec plusieurs kernels. Il existe des kernels pour systèmes mono-processeur et des kernels pour systèmes multi-processeurs (sur Slackware 32 bits). Autrefois, il y avait une multitude de kernels destinés à différents types de contrôleurs de disques durs. Dans le cadre de notre discussion, nous nous attarderons sur les kernels de type “*huge*” et ceux de type “*generic*” .

Si vous jetez un coup d’œil à votre répertoire `/boot`, vous y trouverez les différents kernels installés sur votre système.

```
darkstar:~# ls -1 /boot/vmlinuz*
/boot/vmlinuz-huge-2.6.29.4
/boot/vmlinuz-generic-2.6.29.4
```

Vous pouvez remarquer ici la présence de deux kernels, `vmlinuz-huge-2.6.29.4` et `vmlinuz-generic-2.6.29.4`. Chaque version de Slackware comprend différentes versions du kernel et parfois même des noms un peu différents. Ne vous inquiétez donc pas si ce que vous voyez ne correspond pas exactement à ce que je viens d'énumérer ici.

Les Huge kernels sont exactement ce que vous pourriez penser; ils sont de grande taille. Cependant, cela ne signifie pas qu'ils intègrent tous les pilotes possibles. Au contraire, ces kernels sont faits pour démarrer (et s'executer) sur tout ordinateur supportant Slackware (il peut toutefois en exister qui ne démarreront pas ou ne fonctionneront pas avec le kernel). Ces kernels supportent certainement des matériels pouvant ne pas être présent sur votre machine (et ne le sera jamais), mais cela ne doit pas vous inquiéter. Les Huge kernels sont inclus pour plusieurs raisons, mais probablement la raison la plus importante est leur utilisation par l'installateur de Slackware - l'installateur de Slackware exécute ces kernels. Si vous avez choisi de laisser l'installateur configurer votre bootloader, il choisira d'utiliser ces kernels en raison de l'incroyable variété de matériel qu'ils supportent. En revanche, les generic kernels supportent un nombre très réduit de matériel a moins d'utiliser des modules externes. Si vous

souhaitez utiliser l'un des generic kernels, vous aurez besoin de faire usage de ce qu'on appelle un initrd créé en utilisant l'utilitaire ***mkinitrd*** (8).

Alors, pourquoi devriez-vous utiliser un generic kernel ? Actuellement, l'équipe de développement Slackware recommande l'utilisation d'un generic kernel pour diverses raisons. Peut-être la plus évidente est la taille. Les huges kernels sont de grande taille, environ deux fois la taille des generic kernel avant décompression et mise en mémoire. Si vous utilisez une vielle machine, ou une machine avec une RAM réduite, vous apprécierez l'économie offerte par les generic kernels. Les autres raisons sont un peu plus difficile à quantifier. Des conflits apparaissent de temps en temps entre les pilotes inclus dans les huge kernels, et de manière générale, les huge kernels peuvent ne pas fonctionner aussi bien que les generic kernels. En outre, l'utilisation des generic kernels offrent la possibilité de passer séparément des arguments spéciaux aux pilotes de matériel, plutôt que d'exiger que ces options soient passées sur la ligne de commande du kernel. Certains utilitaires inclus dans Slackware fonctionneront mieux si votre kernel utilise certains pilotes en tant que modules au lieu de les compiler statiquement dans le kernel. Si vous avez des difficultés à comprendre cela, ne soyez pas inquiet : il suffit de penser *“huge kernel = bon, generic kernel = meilleur”*.

Malheureusement, utiliser les generic kernels n'est pas aussi simple que d'utiliser les huge kernels. Afin que le generic kernel puisse démarrer votre système, vous devez également inclure quelques modules de base dans un initird. Les modules sont des morceaux de code compilés du kernel qui peuvent être insérés ou retirés d'un kernel en cours d’exécution (idéalement en utilisant ***modprobe*** (8)). Ceci rend le système un peu plus souple au prix d'un peu de complexité ajoutée. Vous trouverez peut-être plus facile de penser aux modules comme des pilotes de périphériques, du moins pour cette section. En général, vous devez ajouter le module pour le système de fichiers que vous avez choisi d'utiliser pour votre partition racine lors de l'installation. Et si votre partition racine se trouve sur un disque SCSI ou un contrôleur RAID, vous aurez besoin d'ajouter les modules correspondants. Enfin, si vous utilisez le RAID logiciel, le chiffrement du disque, ou LVM, vous aurez également besoin de créer un initrd indépendamment du fait que vous utilisez le generic kernel ou non.

Un initrd est une archive ***cpio***(1) compressée. En créer un n'est donc pas si simple. Heureusement pour vous, Slackware inclut un utilitaire qui rend la tache vraiment très simple : ***mkinitrd***. Une présentation complète de ***mkinitrd*** sort du cadre de ce livre, mais nous vous en montrerons tous les points saillants. Pour une description beaucoup plus complète, consultez le page du manuel ou exécutez ***mkinitrd*** avec l'argument *–help*.

```
darkstar:~# mkinitrd --help
mkinitrd crée une ramdisk initiale (plus précisément une archive initramfs
cpio+gzip) utilisée pour charger les modules du kernel qui sont nécessaires
pour monter le système de fichiers, ou les autres modules qui seraient
nécessaires avant que le système de fichier racine ne soit disponible.
D'autres codes binaires peuvent être ajoutés à l'initrd, et le script est
facile à modifier. Soyez créatif. :-)
.... many more lines deleted ....
```

Quand vous utilisez ***mkinitrd***, vous aurez besoin d'un certain nombre d'information: votre partition racine, le système de fichiers de votre partition racine, tous les contrôleurs de disques que vous utilisez, et l'usage ou la non-utilisation de LVM, RAID logiciel, ou le chiffrement de disque. A moins que vous utilisez une sorte de contrôleur SCSI (et votre partition racine est sur le contrôleur SCSI), vous auriez seulement besoin de savoir le système de fichiers de votre partition racine et le type de partition. En supposant que vous avez utilisé le huge kernel pour démarrer Slackware, vous pouvez facilement trouver cette information avec la commande ***mount*** ou en vérifiant le contenu de

/proc/mounts.

```
darkstar:~# mount
/dev/sda1 on / type ext4 (rw,barrier=1,data=ordered)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
usbfs on /proc/bus/usb type usbfs (rw)
/dev/sda2 on /home type jfs (rw)
tmpfs on /dev/shm type tmpfs (rw)
```

Dans cet exemple, le système de fichier est sur /dev/sda1 et il s'agit d'une partition de type *ext4*. Si l'on souhaite créer un initrd pour ce système, il suffit de donner cette information à ***mkinitrd***.

```
darkstar:~# mkinitrd -f ext4 -r /dev/sda1
```

Notons que dans la plupart des cas, ***mkinitrd*** est assez intelligent pour déterminer cette information par lui même, mais cela ne nuit pas de la spécifier manuellement. Maintenant que nous avons créé une initrd, nous avons simplement besoin d'indiquer à LILO où le trouver. La section suivante traitera de cela.

Rechercher toutes ces différentes options de ***mkinitrd*** ou pire, les mémoriser, peut être difficile, plus particulièrement si vous essayez différents kernels régulièrement. Cela est devenu fastidieux pour l'équipe de développement Slackware, et ils ont ainsi développé un fichier de configuration simple, mkinitrd.conf(5). Vous pouvez trouver un fichier qui peut être personnalisé pour votre système dans le répertoire /etc/mkinitrd.conf.sample. Voici le mien.

```
darkstar:~# >/prompt>cat /etc/mkinitrd.conf.sample
# See "man mkinitrd.conf" for details on the syntax of this file
#
SOURCE_TREE="/boot/initrd-tree"
CLEAR_TREE="0"
OUTPUT_IMAGE="/boot/initrd.gz"
KERNEL_VERSION="$(uname -r)"
#KEYMAP="us"
MODULE_LIST="ext3:ext4:jfs"
#LUKSDEV="/dev/hda1"
ROOTDEV="/dev/sda1
ROOTFS="ext4"
#RESUMEDEV="/dev/hda2"
#RAID="0"
LVM="1"
#WAIT="1"
```

Pour une description complète de chacune de ces lignes et ce qu'elles font, vous aurez besoin de consulter la page de manuel de mkinitrd.conf. Copiez le fichier exemple dans /etc/mkinitrd.conf et éditez le comme bon vous semble. Une fois correctement configuré, vous aurez juste besoin d'exécuter ***mkinitrd*** avec l'argument *-F*. Un fichier initrd sera généré et installé pour vous sans que vous ayez à vous souvenir de toutes ces options obscures.

Si vous n'êtes pas certain des options de configuration à spécifier dans le fichier ou dans la ligne de commande, il existe une dernière solution. Slackware comprend un petit utilitaire qui permet

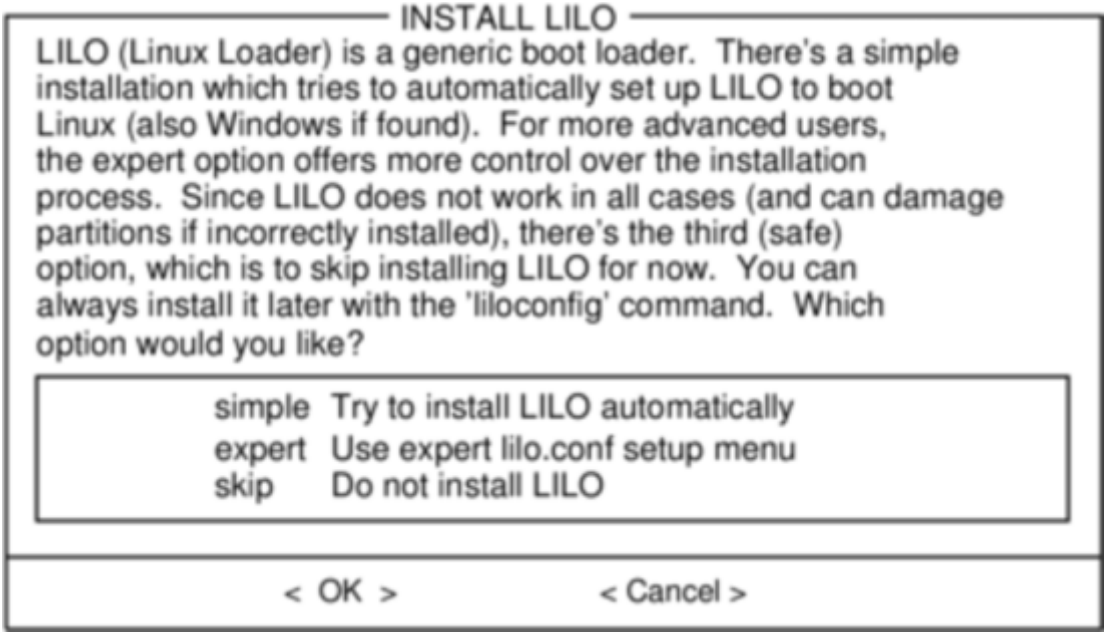
d'indiquer les options requises pour votre kernel en cours d'exécution ***/usr/share/mkinitrd/mkinitrd_command_generator.sh***. Quand vous exécutez ce script, il générera une ligne de commande pour ***mkinitrd*** qui devrait marcher pour votre ordinateur, mais vous pouvez vérifier tout de toute façon.

```
darkstar:~# /usr/share/mkinitrd/mkinitrd_command_generator.sh
mkinitrd -c -k 2.6.33.4 -f ext3 -r /dev/sda3 -m \
  usbhid:ehci-hcd:uhci-hcd:ext3 -o /boot/initrd.gz
```

LILO

LILO est un Linux Loader et est le bootloader installé par défaut dans Slackware Linux. Si vous avez auparavant utilisé d'autres distributions Linux, vous serez peut être plus familier avec GRUB. Si vous préférez utiliser GRUB, vous pouvez facilement le trouver dans le répertoire `extra/` sur un des CD ou DVD d'installation de Slackware. Toutefois, puisque LILO est le bootloader par défaut de Slackware, nous nous concentrerons exclusivement sur LILO.

Configurer LILO peut être un peu intimidant pour les nouveaux utilisateurs. Slackware contient ainsi un utilitaire de configuration appelé ***liloconfig***. Normalement, ***liloconfig*** est executé initialement par l'installateur, mais vous pouvez l'exécuter à tout moment a partir d'un terminal.



liloconfig a deux modes d'opérations : simple et expert. Le mode "*simple*" essaie de configurer automatiquement LILO pour vous. Si Slackware est le seul système d'exploitation installé sur votre ordinateur, le mode ""*simple*" saura presque toujours configurer votre système rapidement et facilement. Il est très bon pour détecter également les installations Windows et les ajouter à `/etc/lilo.conf` afin que vous puissiez choisir le système d'exploitation à démarrer lorsque vous allumez votre ordinateur.

Afin d'utiliser le mode *"expert"*, vous aurez besoin de connaitre la partition racine de Slackware. Vous pouvez aussi configurer d'autres systèmes d'exploitation si vous connaissez leur partition racine. Toutefois cela peut ne pas marcher comme souhaité. ***liloconfig*** essaiera de démarrer chaque système d'exploitation avec un kernel Slackware, et ceci n'est probablement pas ce que vous désirez.

Heureusement, la mise en place des partitions Windows en mode expert est triviale. Un conseil lorsque vous utilisez le mode expert : vous devriez presque toujours installer LILO sur le Master Boot Record (MBR). Autrefois, il était recommandé d'installer le boot loader sur la partition racine et de marquer cette partition comme amorçable. Aujourd'hui, LILO a beaucoup mûri et il n'y a aucun risque à installer sur le MBR. En fait, vous rencontrerez moins de problèmes si vous procédez de cette façon.

liloconfig est une excellente façon de configurer rapidement votre bootloader, mais si vous avez vraiment besoin de savoir ce qui se passe, vous aurez besoin de consulter le fichier de configuration de LILO : `lilo.conf` (5) dans le répertoire `/etc`. `/etc/lilo.conf` est séparé en plusieurs sections. Au sommet, vous trouverez une section *`global`* où vous précisez les options telles que l'endroit où installer LILO (généralement le MBR), toute image spéciale ou écran à afficher au demarrage, et le délai au terme duquel LILO lancera le système d'exploitation par défaut. Voici à quoi ressemble la section *`global`* de mon fichier lilo.conf.

```
# LILO configuration file

boot = /dev/sda
  bitmap = /boot/slack.bmp
  bmp-colors = 255,0,255,0,255,0
  bmp-table = 60,6,1,16
  bmp-timer = 65,27,0,255

append=" vt.default_utf8=0"
prompt
timeout = 50

# VESA framebuffer console @ 1024x768x256
vga = 773
.... many more lines ommitted ....
```

Pour une liste complète de toutes les options de LILO, vous devriez consulter le page de manuel de `lilo.conf`. Nous discuterons brièvement dans ce document des options les plus courantes.

La première chose qui devrait attirer votre attention est la ligne *“boot”*. Cette ligne précise l'endroit où le bootloader est installé. Pour installer sur le Master Boot Record (MBR) de votre disque dur, il vous suffit d'indiquer sur cette ligne l'entrée de périphérique correspondante au disque dur. Dans mon cas, j'utilise un disque dur SATA qui apparaît comme un périphérique SCSI `/dev/sda`. Pour installer sur le bloc d'amorçage d'une partition, vous aurez à indiquer l'entrée de périphérique correspondante à la partition. Par exemple, si vous effectuez l'installation sur la première partition de l'unique disque dur SATA de votre ordinateur, vous devrez probablement utiliser `/dev/sda1`.

L'option *“prompt”* indique simplement à LILO de vous demander quelle système d'exploitation démarrer. Les systèmes d'exploitation sont chacun énumérés dans leur section propre, un peu plus loin dans le fichier. Nous y reviendrons dans un instant. L'option *“timeout”* indique à LILO combien de temps à attendre (en dixièmes de secondes) avant de démarrer le système d'exploitation par défaut. Dans mon cas, c'est 5 secondes. Certains systèmes semblent mettre un temps très long pour afficher l'écran de démarrage, de sorte que vous devrez peut-être utiliser une plus grande valeur de temporisation que ce que j'ai utilisée. Ceci est en partie la raison pour laquelle la méthode d'installation simple utilise une très longue timeout (quelque part autour de deux minutes). La ligne *append* dans mon cas a été mise en place par ***liloconfig*** . Vous pouvez (et devrez probablement) voir quelque chose de similaire dans votre propre fichier de configuration `/etc/lilo.conf`. Je ne vais

pas entrer dans les détails des raisons pour lesquelles cette ligne est nécessaire. Vous allez donc juste devoir me faire confiance lorsque je vous dis que les choses fonctionneront mieux si cette ligne est présente. : ^)

Maintenant que nous avons vu la section "*global*", nous allons jeter un coup d'œil à la section des systèmes d'exploitation. Chaque section Linux débute avec une ligne *"image"*. Les systèmes d'exploitation Microsoft Windows sont spécifiés avec une ligne *"other"*. Regardons un échantillon de `/etc/lilo.conf` qui démarre Slackware et Microsoft Windows.

```
# LILO configuration file
... global section ommitted ....
# Linux bootable partition config begins
image = /boot/vmlinuz-generic-2.6.29.4
  root = /dev/sda1
  initrd = /boot/initrd.gz
  label = Slackware64
  read-only
# Linux bootable partition config ends
# Windows bootable partition config begins
other = /dev/sda3
  label = Windows
  table = /dev/sda
# Windows bootable partition config ends
```

Pour les systèmes d'exploitation Linux comme Slackware, la ligne *"image"* spécifie quel kernel démarrer. Dans ce cas, nous lancerons `/boot/vmlinuz-generic-2.6.29.4`. Les autres sections sont assez explicites. Ils indiquent à LILO où se trouve le système de fichiers racine, quel initrd (s'il y en a) utiliser , et de monter initialement le système de fichiers racine en lecture seule. La ligne initrd est très importante pour toute personne exécutant un generic kernel ou utilisant LVM ou un RAID logiciel. Elle montre à LILO (et au kernel) où se trouve l'initrd créé à l'aide de ***mkinitrd***.

Une fois que vous avez configure `/etc/lilo.conf` pour votre ordinateur, exécutez tout simplement ***lilo***(8) pour l'installer. Contrairement à GRUB et à d'autres bootloaders, LILO nécessite que vous ré-executez ***lilo*** chaque fois que vous faites une modification à ses fichiers de configuration, sinon la nouvelle image du bootloader ne sera pas installée, et les modifications ne seront pas appliquées.

```
darkstar:~# lilo
Warning: LBA32 addressing assumed
Added Slackware *
Added Backup
6 warnings were issued.
```

Ne soyez pas trop effrayé par les nombreux avertissements que vous pouvez voir lors de l'exécution de ***lilo***. A moins que vous voyez une erreur fatale, tout devrait bien se passer. En particulier, l'avertissement concernant l'adressage LBA32 est monnaie courante.

Démarrer plusieurs systèmes d'exploitation

Un bootloader (comme LILO) est un outil très flexible, vu qu'il existe uniquement pour déterminer quel

disque dur, partition, voire kernel spécifique sur une partition, démarrer. Ceci suggère en soi un choix lors du démarrage. Ainsi, l'idée d'avoir plus d'un système d'exploitation sur un ordinateur vient très naturellement à un utilisateur de LILO ou GRUB.

Les gens *“dual boot”* pour diverses raisons; certaines personnes veulent avoir une installation de Slackware stable sur une partition et un système de développement sur une autre. D'autres personnes pourraient désirer avoir Slackware sur une partition et une autre distribution Linux ou BSD sur une autre, et encore d'autres personnes peuvent avoir Slackware sur une partition et un système d'exploitation propriétaire (pour le travail ou pour cette application non disponible pour Linux) sur l'autre.

Le dual boot ne devrait pas cependant être pris à la légère, car cela implique l'existence de deux systèmes d'exploitation différents qui tenteront de gérer le bootloader. Si vous utilisez le dual boot, la probabilité qu'un système d'exploitation écrase ou mette à jour les entrées du bootloader sans votre intervention directe est grande. Si cela se produit, vous serez obligé de modifier manuellement GRUB ou LILO de sorte que vous puissiez accéder à chaque système d'exploitation.

Il y a deux façons de dual (ou multi) boot. Vous pouvez mettre chaque système d'exploitation sur son propre disque dur (fréquent pour un ordinateur de bureau, car il est facile d'ajouter un disque) ou mettre chaque système d'exploitation sur sa propre partition (ce qui est fréquent pour un ordinateur portable où un seul disque physique est présent).

Dual Boot avec des partitions

Afin de mettre en place un système dual-boot avec chaque système d'exploitation sur sa propre partition, vous devez d'abord créer des partitions. Ceci est plus facile lorsqu’on le fait avant d'installer le premier système d'exploitation. Dans ce cas de figure, il s'agit d'une simple pré-planification et dépeçage de votre disque dur aussi nécessaire que vous le sentez. Consultez la section intitulée "Partitioning" pour les infos portant sur l'utilisation des outils de partitionnement ***fdisk*** ou ***cfdisk***.

Si vous avez un dual boot entre deux distributions Linux, il est déconseillé de tenter de partager le répertoire /home entre les deux systèmes. Bien que cela soit techniquement possible, cela augmente également les chances de voir vos configurations personnelles malmenées ou modifiées par des environnements de bureau ou versions concurrentes.

Il est toutefois raisonnable d'utiliser une partition swap commune.

Vous devriez partitionner votre disque dur en au moins trois parties :

- Une partition pour Slackware
- Une partition pour le système d'exploitation secondaire
- Une partition swap

Premièrement, installez Slackware Linux sur la première partition du disque dur comme décrit dans le chapitre Installation.

Une fois Slackware installé, démarré, et que vous avez confirmé que tout fonctionne comme prévu,

redémarrez le programme d'installation pour le deuxième système d'exploitation. Ce système d'exploitation vous proposera d'utiliser la totalité du disque dur ; il faut évidemment **ne pas** faire cela. Restreignez donc l'installateur à seulement la seconde partition. En outre, le système d'exploitation essayera d'installer un bootloader au début du disque dur, écrasant ainsi LILO.

Vous avez quelques cours d'action possibles en ce qui concerne le bootloader:

Scenari Possibles pour le boot Loader

Si le système d'exploitation secondaire est Linux, interdisez lui d'installer un bootloader.
Si vous faites un dual boot avec une autre distribution Linux, le programme d'installation de cette distribution demandera généralement si vous souhaitez installer un bootloader. Vous êtes bien sur libre de ne pas installer le bootloader de l'autre distribution, et de manuellement gérer Slackware et l'autre distribution avec LILO.

Selon la distribution, vous pourriez vous retrouver à éditer LILO plus que vous ne le feriez si vous exécutiez seulement Slackware ; certaines distributions sont connues pour les mises à jour fréquentes du kernel, ce qui signifie que vous aurez besoin de modifier LILO afin de refléter la nouvelle configuration après une telle mise à jour. Mais si vous ne voulez pas modifier les fichiers de configuration de temps en temps, vous n'aurez probablement pas choisi Slackware.

Si le système d'exploitation secondaire est de type Linux, laissez le écraser LILO avec GRUB.

Si vous faites un dual boot avec une autre distribution Linux, vous êtes parfaitement capable de simplement utiliser GRUB plutôt que LILO, ou d'installer Slackware en dernier et utiliser LILO pour les deux distributions. LILO et GRUB ont tous les deux de bonnes capacités d'auto-détection, et par conséquent, celui qui est installé en dernier devrait pouvoir détecter la présence de l'autre distribution et créer une entrée pour elle.

Comme les autres distributions essaient souvent de procéder à des mises à jour automatiques du menu GRUB, il y a toujours une chance qu'un dysfonctionnement apparaisse lors de la mise à jour et que vous soyez soudainement incapable de démarrer votre système Slackware. Si cela arrive, ne paniquez pas : démarrez l'autre distribution et éditez manuellement GRUB de façon à ce qu'il pointe sur la bonne partition, et initrd (si besoin), pour Slackware.

Permettre à la distribution secondaire d'écraser LILO et plus tard ré-installer manuellement et reconfigurer LILO.

Ceci n'est pas un mauvais choix, plus particulièrement quand Windows est le système d'exploitation secondaire. Toutefois, des problèmes potentiels peuvent apparaitre quand Windows procède à sa mise à jour automatique. Windows pourrait essayer d'écraser à nouveau le MBR (*Master Boot Record*), et vous devriez à nouveau ré-installer manuellement LILO.

Pour rétablir LILO après qu'un autre système d'exploitation l'ait effacé, vous pouvez démarrer votre système a partir de votre média d'installation de Slackware et entrer dans la phase setup. <emphasis>Ne re-partitionnez pas</emphasis> votre disque dur ou ne réinstallez pas Slackware ; Allez immédiatement à la section Configuration.

Même Lorsque vous utilisez l'option *“simple”* pour installer, LILO devrait détecter les deux systèmes d'exploitation et configurer automatiquement un menu compréhensible pour vous. Si LILO échoue, ajoutez alors les entrées vous-même.

Dual boot à partir de plusieurs disques durs

Il est en général plus facile de dual boot à partir de disques durs différents que de dual boot à partir de partitions, cela parce que le BIOS de l'ordinateur ou EFI possède presque invariablement un sélecteur de périphérique de démarrage qui vous permet d'interrompre le processus d'initialisation immédiatement après le POST et choisir le lecteur devant avoir la priorité.

La façon d'accéder au sélecteur de disque de démarrage varie d'une carte mère à l'autre ; consultez le manuel de la carte mère ou lisez les instructions sur l'écran de démarrage pour savoir quelle combinaison de touches utiliser. Les combinaisons de touches les plus fréquentes sont F1, F12, DEL. Pour les ordinateurs Apple, c'est toujours la touche clavier ⌥ (Alt).

Si vous gérez la priorité d'amorçage via le BIOS ou EFI, alors chaque bootloader sur chaque disque dur gère seulement son propre disque dur et ne peut donc jamais interférer avec un autre bootloader. Ceci est toutefois contraire à l'esprit de fonctionnement d'un boot loader, mais peut être un palliatif utile lorsqu'il s'agit de systèmes d'exploitation propriétaires qui exigent d'être le seul système d'exploitation sur le système, au détriment des préférences de l'utilisateur.

Navigation

Chapitre précédent : Installation

Chapitre suivant : L'invite de commandes (shell)

Sources

- Source originale : http://www.slackbook.org/beta

- Publication initiale d'Alan Hicks, Chris Lumens, David Cantrell, Logan Johnson
- Traduction initiale de escaflown

slackware, booting, mkinitrd, lilo, dual-boot