



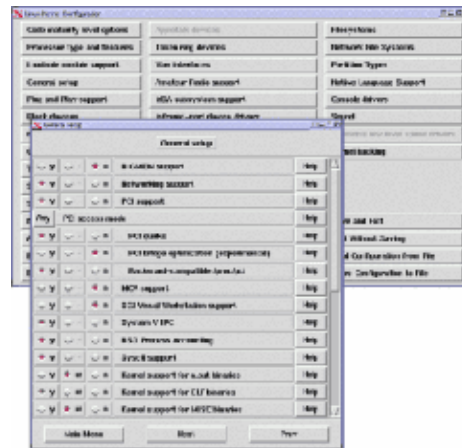
par Philip de Groot
 <philip/at/authors.linuxfocus.org>

L'auteur:

J'espère obtenir mon Doctorat de [l'Université de Nijmegen](#) cette année. Ma thèse porte sur ce que l'on appelle [la chimométrie](#). Je travaille actuellement sur l'informatique appliquée à la biologie, au Centre Médical Académique d'Amsterdam, en Hollande. Je maintiens également mon propre [site pour les débutants sous Linux \(en Hollandais\)](#) : c'est une des nombreuses initiatives par et pour la communauté Linux. J'aime travailler avec Linux et je rends compte fréquemment de mes expériences.

Traduit en Français par:
 Iznogood
 <iznogood/at/iznogood-factory.org>

Compiler votre propre noyau Linux



Résumé:

Vous allez découvrir que vous pouvez, vous aussi, obtenir, configurer, compiler et installer votre noyau personnel.

Introduction

Pourquoi voudriez-vous compiler et installer votre propre noyau ? Les raisons possibles sont :

- Le nouveau noyau possède un meilleur support matériel.
- Le nouveau noyau offre certains avantages, comme un meilleur support pour les machines multi-processeurs (SMP) ou le support pour l'USB. Ceci s'applique aux noyaux 2.4.x.
- Le nouveau noyau résout d'anciens bogues.
- Votre noyau personnalisé ne contient pas d'éléments inutiles et par conséquent il est plus rapide et plus stable.

Le problème, c'est que compiler votre propre noyau réclame de bonnes connaissances sur les ordinateurs. C'est la raison pour laquelle un nouvel utilisateur Linux n'ira pas facilement vers la compilation de noyaux. Cet article présente des photos d'écran sur la manière de compiler en utilisant la commande 'make xconfig'. Avec cette commande, l'utilisateur gère le noyau par le biais d'une GUI, une Interface Graphique Utilisateur et la souris. Vous disposez d'environ 40 copies d'écran, pour clarifier les raisons pour lesquelles vous devez choisir ou non certaines options dans des situations particulières. Détailler ces 40 copies d'écran peut sembler excessif mais c'est la meilleure manière d'éclaircir le fonctionnement interne du noyau et le pourquoi ou le comment de certaines options. Les copies d'écran sont basées sur le noyau 2.4.6. Le plus récent est le 2.4.19 (au moment d'écrire ces lignes), mais à part quelques entrées supplémentaires dans le menu (i.e. le support pour le nouveau matériel), les captures sont les mêmes, idem pour le processus de compilation. Un mot d'avertissement : imprimez cette page avant de démarrer, de manière à toujours avoir toutes les informations nécessaires !

L'article est structuré comme suit. Il indique d'abord où trouver le code source sur internet et comment l'installer, pour ensuite montrer la configuration graphique du noyau avec des copies d'écran. Une fois le noyau configuré, il doit être compilé mais même un noyau fraîchement compilé n'est pas utilisable tel quel. Le nouveau noyau doit d'abord être installé avec le gestionnaire de démarrage « lilo » et avant de l'utiliser, vous devez construire le fichier « /etc/lilo.conf ». Vous pouvez aussi copier le noyau compilé vers une partition depuis laquelle il est possible de démarrer Linux avec un programme DOS/Windows appelé « loadlin ». De plus, il existe quelques points spécifiques qui doivent être pris en compte, tels que le support PCMCIA nécessaire pour les portables. Les cartes PCMCIA, cartouches qui permettent de gérer le réseau et qui ressemblent à d'épaisses cartes de crédit, ne sont supportées par le noyau que depuis les versions 2.4.x. Les noyaux plus anciens peuvent supporter le PCMCIA grâce à une compilation et une installation séparée. SuSE a un autre problème pour le support du son par les pilotes ALSA. Ces pilotes ne font pas partie du noyau et doivent être compilés et installés séparément car les pilotes originaux ne fonctionnent plus. Pour rendre les choses plus complexes, passer d'une série de noyau à une autre, disons du 2.2 vers le 2.4 peut provoquer d'autres problèmes avec certains utilitaires du noyau, « modutils », par exemple. Ils contiennent le code nécessaire pour charger un module de noyau : la figure 3 explique ce qu'est un module. Il arrive parfois que le nouveau noyau ne sache pas quoi faire avec l'ancien « modutils », vous devez alors compiler et installer une version de modutils plus récente. Ce type de problème est rare mais il se produit, il est donc préférable de le signaler par avance.

Mais, si vous suivez consciencieusement les procédures décrites dans cet article, il n'y a pratiquement aucune chance que les choses se passent mal. Le nouveau noyau est ajouté à « lilo » ou copié dans la partition « loadlin ». Ainsi, en cas de difficulté, vous pouvez toujours redémarrer le noyau original. Puis, en travaillant avec votre noyau original, vous pouvez résoudre les problèmes du nouveau noyau. Même si vous avez des soucis avec le « modutils » du nouveau noyau, il est encore possible de redémarrer l'ancien noyau et de résoudre l'anomalie en recompilant et en installant, par exemple, séparément : toutes les nouvelles versions de « modutils » ont une compatibilité descendante avec les anciens noyaux, de telle manière que le nouveau « modutils » fonctionne correctement avec les anciens noyaux.

Installer le code source du noyau

Tout ce que vous allez faire ici demande des privilèges root, vous devez donc vous connecter en tant que root. Tout d'abord, vous devez installer le code source du noyau, i.e., depuis le CD d'installation. Dans SuSE, le source est situé dans la partie « d » (pour « développement ») comme le paquet 'lx_kernel'. Il est recommandé d'installer le code source qui est fourni avec votre distribution car les différentes "GUI" sont aussi installées automatiquement. Une fois ceci terminé, l'archive contenant le noyau le plus récent, i.e., le fichier « linux-2.4.6.tar.bz2 » peut être téléchargée depuis (<http://www.kernel.org/pub/linux/kernel/v2.4/>) et installée. Le modutils correspondant se trouve sur <http://www.kernel.org/pub/linux/Utils/kernel/modutils/v2.4/>. Notez

que les numéros de versions de « modutils » ne correspondent pas avec ceux du noyau : téléchargez simplement la version la plus récente. Compiler et installer modutils est détaillé plus loin, sous le titre « Installer modutils ». Nous allons d'abord nous occuper du noyau proprement dit.

Le code source pour le noyau est déjà sur votre machine dans le répertoire « /usr/src/linux/ ». Il est sage de garder ce code source en sécurité, en renommant, par exemple, le répertoire Linux de la manière suivante :

```
cd /usr/src
mv linux linux-2.2.19 (si le code source original est pour le 2.2.19).
```

C'est seulement après que vous ayez mis en sécurité le noyau original que vous pouvez décompresser le nouveau noyau : vous verrez que le fichier linux-2.4.6.tar.bz2 se décompresse dans le répertoire « linux » par défaut. Il écrase tout si le répertoire existe déjà et vous avez alors des problèmes : vous ne pouvez plus recompiler le noyau original, vous perdez la configuration originale, etc. Dans cet exemple, j'ai renommé le répertoire « linux » juste après la décompression du code source du noyau « linux-2.4.6 » et j'ai créé un lien symbolique avec le nom « linux » vers le répertoire « linux-2.4.6 ». L'avantage de cette procédure est que vous pouvez voir la version du noyau présent sur le système immédiatement. En plus, il est plus facile d'installer une mise à jour noyau. Les commandes sont : (rappelez-vous que vous devez être root) :

```
cd /usr/src
cp ~/linux-2.4.6.tar.bz2 (en supposant que l'archive a été téléchargée)
                        (dans votre répertoire de travail ('~') )
bzip2 -d linux-2.4.6.tar.bz2 (ceci peut prendre du temps )
tar -xvf linux-2.4.6.tar
mv linux linux-2.4.6
ln -s /usr/src/linux-2.4.6 /usr/src/linux
```

Une fois que ceci est fait, allez dans le répertoire du noyau et tapez :

```
cd /usr/src/linux
make xconfig (voir Figure 1)
```

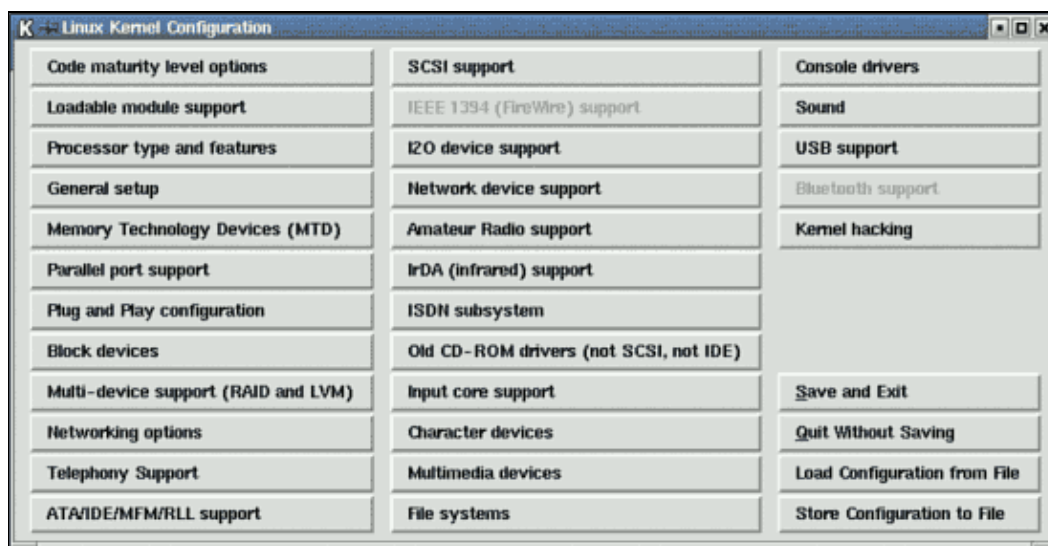


Figure 1 :

L'interface graphique pour définir votre noyau Linux après la commande « make xconfig ».

C'est le menu principal utilisé pour définir le noyau. Pour ce faire, vous devez cliquer sur les différentes options. Cliquer sur « Save and Exit » sauvegarde vos choix sur le disque et une fois terminé, vous pouvez enfin compiler et installer le noyau (comme dans la Figure 40). Mais nous n'en sommes pas encore là.

Configurer le noyau

Ci-dessous, je reproduis les diverses captures d'écran avec mes propres sélections. Chaque image est accompagnée d'une explication sur la raison du choix des options spécifiques. Lisez attentivement ces exemples et vous comprendrez les raisons de mes choix et ainsi il sera plus facile de savoir quelle option choisir dans votre propre cas. La fonction « help » procure une information similaire. Vous pouvez visualiser l'aide en créant le « make xconfig » de votre distribution Linux. Cliquez alors sur « Help ». Le texte qui apparaît recommande habituellement les options à utiliser.

Les exemples, bien sûr, ne détaillent pas tous les matériels disponibles. Ils peuvent néanmoins clarifier la manière de gérer votre propre matériel et indiquer comment regarder dans le noyau lui-même pour trouver si votre matériel est supporté.

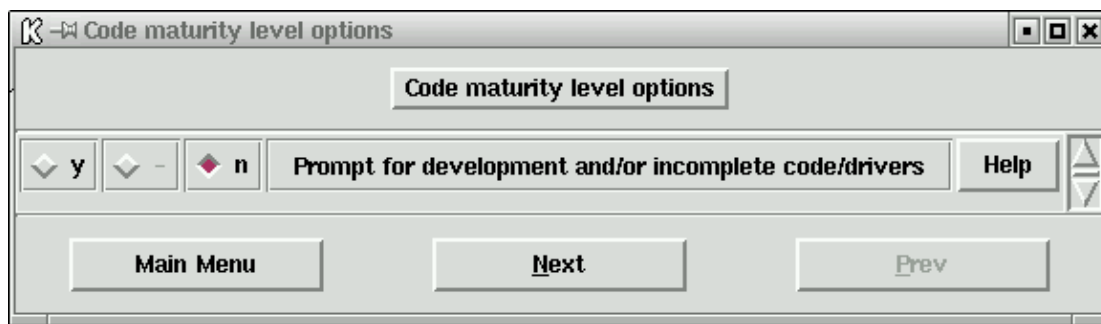


Figure 2: Sélection de « code maturity level options ».

Dans cette section, vous pouvez permettre l'utilisation des options expérimentales du noyau. Elles sont parfois nécessaires, par exemple pour les nouveaux types de cartes. Néanmoins, dans la plupart des cas, vous désélectionnez cette option car les codes expérimentaux donnent des noyaux moins stables. Dans la Figure 1, vous voyez des options grisées comme le « support de l'IEEE 1394 (FireWire) » et « Bluetooth ». A l'heure actuelle, vous ne pouvez pas encore sélectionner ces deux options car le code est encore expérimental.

Figure 3: Support pour les modules chargeables.

(à partir d'ici, les captures d'écran peuvent être ouvertes dans une nouvelle fenêtre pour que vous puissiez les visualiser)

Les modules sont des parties de code, i.e., les pilotes, qui sont compilés séparément mais idéalement, au moment de la compilation du noyau. Ce code ne fait donc pas partie du noyau mais il peut être chargé et devenir disponible lorsque vous en avez besoin. En général, la recommandation est de compiler le code du noyau comme module lorsque cela est possible car cela permet d'obtenir un noyau plus petit et plus stable. Un avertissement : ne compilez jamais le système de fichiers comme un module, voyez la figure 32. Si vous faites cette erreur et que le système de fichiers est un module, le noyau ne pourra pas lire son propre système de fichiers. Il ne pourra alors même pas charger ses propres fichiers de configuration, qui forment le prérequis pour démarrer correctement Linux. Vous verrez que j'utilise très rarement les modules : j'aime que mon noyau puisse dialoguer directement avec tout le matériel sans avoir à charger de modules mais cela ne représente que mon opinion.

Figure 4: Sélectionner le type de processeur et ses fonctionnalités.

Vous choisissez le type de processeur que vous possédez et vous indiquez les diverses options à appliquer. En général, les options « /dev/cpu » sont assez spécialisées et ne devraient pas être sélectionnées par la plupart des utilisateurs. Le « High Memory Support » ne devient nécessaire que si votre ordinateur possède plus d'un gigaoctet de RAM (il ne s'agit pas de l'espace disque). Le plupart des ordinateurs embarquent de 64 à 512 méga-octets de RAM (et de 8 à 60 Go sur le disque dur), « High Memory Support » est donc désélectionné. Vous devez activer l'option « Math Emulation » si vous utilisez Linux sur des systèmes 386 ou 486SX. Ces vieux systèmes ne possèdent pas le coprocesseur mathématique nécessaire à Linux, c'est la raison pour laquelle il faut sélectionner « Math Emulation ». Normalement tous les nouveaux processeurs embarquent le coprocesseur et vous pouvez donc oublier cette option. L'option « MTRR » permet des communications plus rapides entre un bus PCI ou AGP. Comme tous les systèmes modernes possèdent leur carte vidéo sur un bus PCI ou AGP, vous choisirez normalement « MTRR » : dans tous les cas, il est toujours plus sûr d'activer cette option même si votre système n'utilise pas le bus PCI ou AGP pour les cartes vidéo. Le support pour le multi-traitement symétrique (SMP) s'applique seulement pour les cartes mères qui possèdent plus d'un processeur, i.e., les cartes mères avec deux Pentium II. SMP s'assure que le noyau optimise le chargement des deux processeurs. La dernière option (APIC) s'applique aussi habituellement aux systèmes multi-processeurs et elle est donc désactivée.

Figure 5: Options Générales du Noyau .

Vous spécifiez ici certaines options générales du noyau. Tout le monde sélectionne « Networking support » car vous en avez besoin, par exemple, pour Internet. Linux est particulièrement orienté Internet et ne peut pas fonctionner correctement sans réseau. De plus, le support réseau est aussi nécessaire pour tous les autres types d'actions qui ne semblent pourtant pas avoir grand chose à faire avec le réseau. Il est même possible que le noyau ne se compile pas sans le support réseau. Pour résumer : il faut inclure le support réseau. Tous les systèmes modernes utilisent les bus PCI, vous devez donc aussi sélectionner ces options. Le texte en grisé « PCMCIA/CardBus support » indique que cette option n'est pas disponible car vous avez indiqué plus haut que vous ne vouliez pas utiliser de code expérimental (voir Figure 2.). Si vous utilisez un portable, vous avez besoin du support PCMCIA/CardBus dans le noyau pour permettre l'utilisation d'un réseau ou d'un modem (voir aussi l'en-tête « pcmcia support (laptops) » plus loin). « System V IPC » permet aux programmes de communiquer et de se synchroniser, « BSD process accounting » garde, par exemple, le code d'erreur lorsque le processus s'arrête et « Sysctl support » permet à des programmes de modifier certaines options du noyau sans recompiler ou redémarrer le système. Ces options sont habituellement activées. Les distributions modernes de Linux ont leur « kernel core (/proc/kcore/) format » sélectionné comme « ELF » : c'est le format standard de diverses bibliothèques système, i.e., les morceaux de code qui sont disponibles sur le système et utilisés par les programmes. « ELF » est le successeur du format obsolète « a.out » et similaire aux fichiers .dll windows. Tous les programmes modernes de Linux utilisent les bibliothèques ELF mais, malheureusement, quelques anciens programmes ont besoin du support du format « a.out ». Un exemple est « Word Perfect 8 pour X Window System » : c'est une application native X Window System/Linux qui n'est seulement disponible qu'au format « a.out », de même que « xwp » ne fonctionne tout simplement pas sans le support du format « a.out ». Incluez « a.out » comme module si vous voulez utiliser « xwp ». J'utilise aussi « MISC » comme module. En principe, je ne l'utilise pas mais il est très utile d'avoir le code à portée de main si vous faites partie des utilisateurs fréquents de java, python ou de l'émulateur DOS, DOSEMU. J'ai sélectionné « Power Management support » et « Advanced Power Management BIOS support » (ils n'apparaissent pas dans la Figure 5). Ces deux options représentent le minimum nécessaire avec les cartes mère ATX modernes pour permettre au noyau d'éteindre l'ordinateur automatiquement lors de la fermeture de la session Linux. Les autres fonctions de gestion de l'alimentation sont désactivées car, normalement, elles ne fonctionnent pas sous X Window System (qui est le standard lors de l'utilisation de Linux). KDE et Gnome possèdent leur propre standard de gestion de l'énergie dans lequel vous pouvez faire des choix.

Figure 6: Configurer la Technologie des Composants de Mémoire.

Vous avez besoin de cette option pour que Linux puisse, par exemple, lire les cartes flash. Celles-ci sont fréquemment utilisées dans les appareils photo numériques. Avec cette option, Linux peut lire les cartes flash

(à partir du matériel adéquat) et copier les photographies comme des fichiers .jpg vers le disque dur. À moins qu'elle vous soit nécessaire, je désactiverais cette option : si vous pensez en avoir besoin vous pourrez toujours l'ajouter ultérieurement.

Figure 7: Configurer le Port Parallèle.

Avant l'apparition de la technologie USB, le port parallèle était fréquemment utilisé pour connecter l'ordinateur aux imprimantes et aux scanners. Mon imprimante utilise le port parallèle, je souhaite donc que ce port reste disponible sous Linux. Notez que configurer le port parallèle est différent de configurer l'imprimante : cela se fera plus tard dans la Figure 28.

Figure 8: Configurer le Plug & Play.

Pratiquement tout le monde possède le système « Plug & Play » et veut donc que cette option soit supportée. L'activer permet au noyau de configurer les périphériques « Plug & Play » et de les rendre disponibles pour le système. Il est quelquefois nécessaire d'activer l'option « Plug & Play OS » dans le BIOS sinon Linux (mais aussi Windows) ne peut pas configurer les périphériques « Plug & Play ». L'option « ISA Plug & Play support » concerne les cartes ISA qui sont « Plug & Play » tout en utilisant le bus ISA. Un des exemples est la AWE64 sound blaster. Le bus ISA n'a jamais eu de standard « Plug & Play » ce qui rend difficile la configuration de ces cartes. Il y a quelques temps, avant les noyaux 2.4.x, les utilisateurs Linux devaient appeler le programme « isapnp » (le paquetage isapnptools, rpm -qil isapnptools pour voir tous les fichiers) pendant le processus de démarrage. « isapnp » lit le fichier « /etc/isapnp.conf ». Ce fichier contient tous les ports, adresses et interruptions utilisés par les diverses cartes. Si les informations dans « /etc/isapnp.conf » n'étaient pas correctes ou si « isapnp » n'était pas appelé, la carte n'était pas accessible à Linux et le modem, les cartes réseau et son ne fonctionnaient pas. Sélectionner l'option « ISA Plug & Play support » remplace l'ancienne procédure : le fichier « /etc/isapnp.conf » n'est plus utilisé. En échange, les configurations sont détectées automatiquement. Sous SuSE 7.1, j'ai renommé le fichier « /etc/isapnp.conf » en i.e., « /etc/isapnp.conf.old » après la compilation du 2.4.x car le noyau et « isapnp » réclamaient les mêmes ressources avec des conséquences désastreuses. Le problème est que SuSE 7.1 (et les versions antérieures) activent automatiquement « isapnp » au démarrage même si le noyau contient déjà le support nécessaire. Ce n'est néanmoins valable que pour les anciens systèmes Linux, les nouveaux n'utilisant pas isapnp par défaut.

Figure 9: Configurer les Périphériques Blocks.

Tout le monde, virtuellement, veut utiliser une disquette, donc l'option principale est activée (ou, dans mon cas, sélectionnée comme module). Lors d'une requête pour l'accès à la disquette, le noyau charge automatiquement le module nécessaire, pourvu que le fichier « /etc/modules.conf » ou « /etc/conf.modules » soit correctement configuré dans votre distribution comme cela est normalement le cas. En tant qu'utilisateur, vous ne devriez pas avoir de problèmes si vous avez sélectionné les bonnes options dans la Figure 3. Pour accéder à la disquette, le noyau doit bien sûr être capable de lire le système de fichiers de la disquette. Vous devez donc copier la Figure 32 correctement. Les autres options peuvent être importantes si vous utilisez un support de stockage IDE par le port parallèle mais elles sont normalement désactivées. Une des exceptions possible est le « loopback device support ». Avant de graver des CD sous Linux, vous faites d'abord une image du CD et le « loopback device » est alors nécessaire pour visualiser le contenu de l'image. J'ai sélectionné cette option (5^{ème} ligne à partir du bas) comme module (cela n'apparaît pas dans la Figure 9).

Figure 10: Configurer des périphériques multiples.

Les petits utilisateurs de Linux n'ont pas besoin normalement du support raid ou LVM . « Raid » implique que le système utilise deux disques durs ou plus pour stocker les informations en parallèle. Si un des disques tombe en panne, le second continue à fonctionner et le système continue de fournir le service. LVM rend possible l'ajout d'un disque dur de telle manière qu'une partition existante semble plus grande. En pratique, cela signifie que vous n'avez pas à repartitionner ou à copier une petite partition vers une plus grande. Les noms de chemins restent aussi les mêmes. Cette possibilité est vraiment très utile mais n'est habituellement pas nécessaire pour les petits utilisateurs.

Figure 11: Configurer les options de réseau.

Vous avez besoin de l'option « Packet Socket » pour communiquer avec les éléments du réseau sans implémenter de protocole réseau dans le noyau. Je peux, ici, être très bref : vous devez toujours la sélectionner. La plupart des autres options sont désactivées à moins que vous n'ayez besoin de leur support spécifique. Par exemple, j'ai sélectionné « Network packet filtering (replaces ipchains) » car j'utilise le pare-feu standard de SuSE. Un pare-feu protège votre ordinateur contre les attaques externes, i.e. depuis internet, au moins lorsque vous avez correctement configuré le pare-feu. La protection par pare-feu au niveau du noyau est évidemment très avantageuse. Les choix suivants dans la configuration du « network packet filtering » sont expliqués dans la Figure 12. Vous avez besoin du « Unix domain sockets » pour établir des connexions réseau mais aussi ailleurs : X Window System utilise automatiquement les sockets Unix, de telle manière que vous ne pouvez pas utiliser X Window System sans cette option. Vous devez toujours l'activer. « TCP/IP networking » contient les protocoles nécessaires pour internet et aussi pour les réseaux internes. Normalement, vous activerez le support TCP/IP. En cas de doute sur la sélection d'une option particulière, essayez les textes d'aide. Si vous hésitez encore, il est toujours possible d'inclure le support et de le retirer ensuite pendant la phase de test. Compiler certaines options comme modules peut aussi être une alternative intéressante.

Figure 12: Configurer le IP netfilter (pare-feu).

Pour un fonctionnement correct de son pare-feu, SuSE Linux réclame le support d'ipchains. J'ai donc sélectionné cette option pour SuSE. Si vous utilisez un pare-feu dans d'autres distributions ou installations, consultez leur manuel.

Figure 13: Configurer le support téléphonique.

Cela n'est nécessaire que si vous avez une carte téléphone dans votre ordinateur, i.e., pour faire des appels téléphoniques par Internet. La plupart des petits utilisateurs n'ont pas cette carte et n'ont pas besoin de cette option.

Figure 14: Configurer les supports ATA, IDE, MFM et RLL (protocoles de communication pour les disques durs).

Quasiment tout le monde a besoin de ces protocoles, avec pour seule exception ceux dont les systèmes n'ont que des disques SCSI et rien d'autre. C'est la raison pour laquelle la plupart des utilisateurs sélectionnent cette option. Cliquer sur la ligne juste en dessous affiche un sous-menu avec un autre jeu d'options. Elles sont détaillées ci-dessous. À cause de leur importance, il n'y a pas une mais trois captures d'écran. Remplissez-les avec précaution : elles sont extrêmement importantes.

Figure 15: Configurer le support ATA, IDE, MFM et RLL : capture d'écran 1.

L'option du haut est nécessaire à tous ceux qui communiquent avec le matériel par l'interface IDE/ATAPI. Ceci inclut les disques durs mais aussi les lecteurs de bandes et les disques ZIP, les lecteurs et les graveurs de CD. Tous les ordinateurs modernes utilisent l'interface IDE/ATAPI, cette option est donc activée. L'option « include IDE/ATA-2 DISK support » est nécessaire pour supporter les disques dur du système. Cette option doit donc aussi être activée, sauf si vous n'avez que du SCSI.

Figure 16: Configurer le support ATA, IDE, MFM et RLL : capture d'écran 2.

L'option « include IDE/ATAPI CDROM support » est habituellement sélectionnée si vous avez un lecteur de CDROM ATAPI. Néanmoins, les graveurs de CD ATAPI doivent être gérés par une émulation SCSI. L'émulation SCSI peut être utilisée pour accéder à la fois au lecteur de CD et au graveur. Vous pouvez néanmoins rencontrer des problèmes si vous montez vos CD via votre émulation SCSI, tels que des messages d'erreur lors du montage de CD ou lors de l'utilisation du lecteur pour écouter des CD audio. La meilleure solution consiste à sélectionner « include IDE/ATAPI CDROM support » et « SCSI emulation support » comme indiqué dans la Figure 16. Le périphérique qui a besoin de l'émulation SCSI, habituellement le graveur de CD, peut être défini dans « /etc/lilo.conf » en ajoutant la ligne « append="hdd=ide-scsi": » ; ce qui est développé plus loin sous le titre « Configuration de Lilo. ». Comme je possède un pilote de ZIP interne qui

communiqué avec la carte mère par l'interface ATAPI, j'ai sélectionné l'option « include IDE/ATAPI FLOPPY support. ». Vous avez besoin de la même option pour accéder aux autres périphériques à disquettes, tels que le LS120. La plupart des cartes mères utilisent « PCI IDE » pour accéder aux disques durs, CDROM et disquettes, cette option est donc généralement activée. De même pour les deux manières d'activer le DMA. DMA donne à votre matériel un accès direct à la mémoire interne de votre ordinateur, sans l'intervention de votre processeur. Le résultat est un accès plus rapide aux disques IDE. C'est ce que nous voulons. L'option « sharing PCI IDE interrupts support » est désactivée car vous n'en avez généralement pas besoin. Quelques contrôleurs peuvent vraiment partager les interruptions avec d'autres périphériques, par exemple une carte réseau exotique. Malheureusement, partager les interruptions IDE diminue la performance des disques partagés, vous ne partagerez donc les interruptions que lorsque ce sera la seule manière de résoudre certains problèmes matériel sérieux.

Figure 17: Configurer le support ATA, IDE, MFM et RLL : capture d'écran 3.

Ma carte mère possède un Pentium II et un chipset Intel, j'ai donc bien sûr voulu utiliser le support spécifique de ce chipset particulier. Lorsque vous remplissez vos propres options noyau, vous allez voir d'autres chipsets qui n'apparaissent pas dans la Figure 17.

Figure 18: Configurer le support SCSI.

Si vous avez une carte SCSI, vous devez bien sûr sélectionner l'option dont vous avez besoin. La capture d'écran ne montre que les options nécessaires pour votre graveur CD ATAPI si vous avez sélectionné « SCSI emulation support » (Figure 16).

Figure 19: Configurer le support de périphérique I2O.

Vous devez sélectionner cette option si vous avez une interface I2O dans votre ordinateur. La plupart des personnes ne l'ont pas et dans ce cas, vous la désactivez.

Figure 20: Configurer le support de périphérique réseau.

Je n'ai jamais réussi à compiler un noyau sans le support de périphérique réseau. Vous devez donc toujours sélectionner cette option. Vous devriez aussi sélectionner le pilote factice, soit dans le noyau, soit comme module. Linux réclame un tel pilote même lorsque le réseau physique réel est absent, comme c'est le cas pour la plupart des utilisateurs. Même lorsqu'il y a un réseau, Linux utilise le pilote factice fréquemment. Dans ce menu, vous pouvez sélectionner votre type de réseau ainsi que la carte, comme montré dans l'exemple de la Figure 21. Notez qu'il reste du travail si vous voulez accéder à Internet par un modem : vous devez activer le support ppp en choisissant soit « PPP support for async serial ports » (pour les ports COM) ou « PPP support for sync tty ports » (pour les connexions rapides avec un adaptateur i.e., un SyncLink). Si vous oubliez de le faire, le noyau vous indiquera que le module ppp n'existe pas, même si vous l'avez sélectionné, un message d'erreur qui n'aide pas à trouver le problème réel. Vous pouvez aussi choisir les méthodes de compression sans problèmes : si le noyau en a besoin, il les utilise, sinon il les ignore.

Figure 21: Configurer le périphérique ethernet.

Ma carte ethernet est une 3COM /100 MBit qui utilise le chipset 3c509/3c529. Comme je n'ai pas de connexion physique à un réseau (j'ai une carte réseau mais je suis connecté au réseau par le modem), j'ai compilé ce pilote comme un module, juste dans le cas où j'aurais besoin de la carte dans le futur. Vous sélectionnez bien évidemment le type de réseau et la carte réseau de votre machine. De plus, vous devez configurer le réseau avec un programme de configuration Linux, comme « yast2 » sous SuSE.

Figure 22: Configurer le support radio amateur.

Vous sélectionnez cette option si vous voulez utiliser le support radio amateur et vous activez le pilote indispensable. La plupart des personnes n'utilisent pas cette option.

Figure 23: Configurer le support pour la communication infrarouge (sans fil).

Vous validez l'option de communication infrarouge si vous avez un périphérique sans fil, i.e., une souris ou un

clavier sans fil. La plupart des systèmes de bureaux ne l'ont pas et n'ont pas besoin de cette option.

Figure 24: Configurer le support ISDN.

Vous sélectionnez le support pour une carte ISDN si vous en possédez une dans votre système. Il est important de savoir quelle carte vous avez, chipset inclus : vous avez besoin de cette information pour choisir le bon pilote.

Figure 25: Configurer les anciens pilotes de CDROM.

Dans les anciens systèmes 486 et même 386, le CDROM n'est pas connecté via le contrôleur de disque dur IDE (ATAPI) mais par une carte son ou une carte spéciale. Utiliser ces anciens CD réclame une sélection du pilote correspondant. Cette option est dépassée dans les systèmes modernes et elle est donc superflue.

Figure 26: Configurer le support du coeur d'entrée.

Ceci fait référence à une des additions les plus importantes des noyaux 2.4.x : le support USB. Le support du coeur d'entrée est une couche entre le noyau et les périphériques USB. La Figure 38 montre les divers périphériques USB que vous pouvez sélectionner et le texte d'aide indique ceux qui ont besoin du « input core support » : voir la Figure 38. Vous devez activer ici le « input core support » si un des périphériques USB en a besoin. Toutes les cartes mères modernes ont une connexion USB, vous devez avoir pour règle de l'activer. Mais, pour être honnête, je sais que je n'aurais pas besoin du support USB dans mon système et je ne l'ai pas activé.

Figure 27: Configurer les périphériques caractères : capture d'écran 1.

L'option (« virtual terminal ») permet d'ouvrir un xterm (en utilisant X Window System) ou d'utiliser le mode texte pour l'ouverture de session. Normalement, cette option est toujours activée. La seconde option (« support for console on virtual terminal ») indique au noyau où il doit envoyer les messages, tels que les avertissements sur l'absence de module ou si ceux-ci ne fonctionnent pas correctement, les problèmes avec le noyau lui-même et les messages de démarrage. Sous X Window System, vous avez souvent une fenêtre dédiée pour les messages du noyau mais en mode texte, ils vont généralement dans le premier terminal virtuel (« CTRL+ALT+F1 »). Laissez cette option activée. Vous pouvez aussi choisir d'envoyer ces messages vers le port série, i.e., une imprimante ou un autre terminal (la quatrième option). Pour envoyer des messages à l'imprimante, vous devez aussi activer le port avec la troisième option. De même, vous devez activer ce port si vous voulez l'utiliser pour une « souris série ». Encore une fois, habituellement, la troisième option (« standard/generic (8250/16550 and compatible UARTs) serial support ») est activée. Sur mon système, j'ai choisi de le compiler comme module. La raison est que pendant le démarrage, SuSE se plaint de l'absence du module « serial support » et inclure le support comme module est une manière élégante d'éviter les messages en s'assurant que le module existe. Configurer « character devices » est très important. Si vous ne le faites pas correctement, vous pouvez vous retrouver avec un système inopérant. D'où les Figures 28 à 30 pour détailler quelques options supplémentaires.

Figure 28: Configurer « character devices » : capture d'écran 2.

Si vous voulez utiliser un xterm sur votre propre machine depuis un site déporté, par « telnet » ou « ssh », par exemple, vous devez activer l'option « unix98 PTY support ». Il semblerait qu'une station seule n'a pas besoin de cette option mais certains processus d'arrière-plan utilisent aussi cette option. C'est donc une bonne idée d'activer cette option dans tous les cas, ne serait-ce que pour éviter les messages d'erreur (au moins pour SuSE) pendant le démarrage. Quiconque connecte une imprimante à un port parallèle nécessite bien sûr « Parallel printer support ». Cela dit, tout le monde n'a pas besoin d'un port parallèle : c'est le cas des imprimantes USB modernes. Les messages du noyau peuvent aller vers l'imprimante en activant « Support for console on line printer » : habituellement, vous ne le choisissez pas. Vous devez sélectionner l'option « support for user-space parallel port device drivers » si vous avez certains éléments d'équipement qui utilisent le port parallèle mais ce n'est habituellement pas le cas. De même, habituellement, vous n'avez pas besoin de « I2C support » : il est nécessaire pour certaines cartes qui gèrent la vidéo mais si vous en avez besoin, il peut toujours être ajouté plus tard, une fois que le noyau fonctionne bien. Vous sélectionnez le support pour la

souris et le joystick lorsque vous les utilisez mais toutes les souris n'utilisent pas ce pilote (voir ci-dessous, sur la Figure 29). Les graveurs de CD actuels ont rendu obsolètes les pilotes de périphériques qui nécessitent le « QIC-02 Tape support », cette option est donc habituellement désactivée.

Figure 29: Configurer « character devices » : Souris.

Vous n'avez besoin de rien de cette option si vous avez une souris série mais pour les autres types de souris, vous devez configurer ici certains paramètres. Si vous utilisez un bus de souris ORIGINAL, vous devez sélectionner l'option du haut, avec au-dessous le type ou la marque du bus souris. De nombreux ordinateurs, de nos jours, possèdent d'autres types de souris, habituellement (et par erreur) appelé « busmouse » ou « PS/2 mouse ». Ces souris sont souvent connectées à « /dev/aux » et branchées par un petit connecteur comme ceux utilisés pour les claviers. Ce type de souris utilise souvent le clavier pour se connecter à l'ordinateur. Pour faire fonctionner correctement ces souris, vous devez sélectionner les options montrées sur la Figure 29, « mouse support (not serial and bus mice)» et « PS/2 mouse (aka "auxiliary device" support) ».

Figure 30: Configurer « character devices » : capture d'écran 3.

Les options pour configurer le noyau entre celles de la Figure 28 et de la Figure 30, ne sont pas détaillées ici. Elles sont normalement désactivées. L'option « Ftape, the floppy tape device driver » fait référence au support pour les pilotes de bandes connectés sur le contrôleur de disquette. Même si vous avez un tel pilote, il n'est pas essentiel de compiler le support, au moins pour la première approche. Les autres options se rapportent aux cartes vidéo 3D modernes. Si vous avez une carte vidéo qui est connectée sur un bus AGP, vous pouvez activer le support AGP et aussi le pilote spécifique pour votre carte vidéo (sous « /dev/agpgart (AGP support) »). Notez qu'il est possible d'avoir un noyau fonctionnant correctement sans ces options mais pas nécessairement ! Les gens qui possèdent une carte vidéo intégrée sur le carte mère, tel que le chipset intel i815, DOIVENT utiliser le pilote du noyau ! Si ce n'est pas le cas, XFree 4.0 ou mieux (utilisé avec les distributions les plus récentes) ne fonctionne pas. Mon système possède une carte AGP, une NVidia TNT2 mais cette carte n'est pas supportée par un module de noyau spécifique (NVidia refuse de partager les spécifications matérielles, nécessaires au développement de ces pilotes). Malheureusement, dans mon cas, il est inutile d'activer le support AGP. Malgré ce problème particulier, je peux utiliser XFree 4.0 sans le pilote du noyau. « Direct rendering support » est une option permettant d'accélérer les performances graphiques par le biais du noyau, disponible à partir de la version 4.0 de XFree. Pour utiliser cette option, votre carte vidéo spécifique doit être supportée et vous devez utiliser XFree86 4.0 ou mieux. De plus, vous devez aussi activer le « AGP support ». Vous pouvez malgré tout désactiver ces options en toute sécurité et obtenir un noyau Linux fonctionnant parfaitement.

Figure 31: Configurer « multimedia devices ».

Cette option est activée si vous avez une carte qui gère la vidéo ou la radio. Comme précédemment, cette option n'est pas essentielle au fonctionnement correct du noyau.

Figure 32: Configurer « file systems » : capture d'écran 1.

Vous spécifiez ici les systèmes de fichiers pouvant être lus par le noyau Linux. Vous pouvez vouloir un noyau capable de lire des partitions ou des disquettes Windows mais vous devez vous assurer que le noyau puisse lire au moins le système de fichier ext2 propre à Linux ou bien le nouveau système de fichiers ReiserFS. Linux ne peut même pas démarrer si vous l'oubliez car le noyau ne peut même plus lire son propre disque de démarrage (comme cela a été détaillé dans la Figure 3). Pour lire des disquettes et des disques DOS/Windows, vous devez activer l'option « DOS FAT support » : néanmoins, pour lire les disques Windows NT/Windows 2000, il vous faut un pilote à lecture seule qui peut être sélectionné plus tard dans ce menu. Pour lire et aussi écrire sur les disques et les disquettes DOS/Windows, vous avez besoin de l'option « MSDOS fs support ». Virtuellement tout le monde veut cela et donc la plupart des gens activent ces options. « VFAT » fait référence au support de l'utilisation des noms de fichiers longs sous Windows 95 et 98. Mon système est un de ceux que l'on appelle à double démarrage avec lequel je peux démarrer Windows 98 ou linux (en utilisant le gestionnaire de démarrage de Linux lilo, voir « Configuring lilo »). J'ai donc activé « VFAT ». Vous devez inclure le support ISO 9660 pour lire les CD dans le format standard. En dessous, vous avez l'option « Joliet

extensions » qui autorise des noms de fichiers plus longs que ceux de MS-DOS 8.3 qui est la limite du standard ISO 9660. Pratiquement tout le monde veut lire les CD actuels et c'est pourquoi ces options sont toujours activées. La Figure 33 clarifie certaines options additionnelles, parmi lesquelles se trouve le système ext2.

Figure 33: Configurer les systèmes de fichiers : capture d'écran 2.

Les fichiers dans le répertoire « /proc » contiennent des informations sur le statut du système, i.e., quelles sont les interruptions utilisées. Vous devez normalement activer cette option. « Second extended fs support » se réfère au système de fichier (encore) standard de Linux. Vous DEVEZ absolument le compiler en interne (PAS comme un module) ! Les Figures 32 et 33 ne montrent pas l'option « ReiserFS » qui peut être sélectionnée ici aussi : successeur désigné de ext2, ReiserFS est plus apte à traiter les dommages causés à un système de fichiers par des coupures d'alimentation et autres problèmes similaires. Pour l'instant, ReiserFS est encore en développement et par conséquent considéré comme code expérimental. Mais même dans ce cas, les distributions les plus récentes proposent déjà le support de ReiserFS, mais même si ReiserFS est supposé remplacer « ext2 » dans le futur, je ne vous le recommande pas comme système de fichiers pour toutes les partitions. Vous avez besoin de « UDF file system support » si vous utilisez (sous Windows) le programme « packetCD », qui vous permet de copier des fichiers de CD à la volée comme depuis un disque dur lent. Il est très pratique d'échanger des données avec d'autres PC. Lire ces CD peut aussi se faire sous Linux, en les montant avec le système de fichiers « udf », i.e., par une commande comme « mount -t udf /dev/scd0 /cdrom ». Cette partie contient aussi des utilitaires comme « Network file systems », « partition types » et « Native language support ». Vous n'avez pas besoin de « Network File Systems » à moins que votre ordinateur fasse partie d'un grand réseau, auquel cas, vous avez besoin d'activer « NFS File System Support » et peut être aussi « SMB file support » mais pour un ordinateur isolé, vous n'avez pas besoin de ces options. L'option « Partition Types » est assez complexe mais pas nécessaire pour l'utilisation effective du noyau Linux. Il est préférable de la désactiver. Les Figures 34 et 35 détaillent plus avant le « Native Language Support ».

Figure 34: Configurer « native language support » : capture d'écran 1.

Dans ce menu, vous choisissez la table de code qui pourra être utilisée par Linux pour gérer les noms de fichiers sous DOS et Windows. Les tables des codes de la Figure 34 sont pour les noms de fichiers DOS courants. Les tables NLS de la Figure 35 sont nécessaires pour utiliser des noms de fichiers longs. L'option en haut de la Figure 34, « Default NLS option » détermine les symboles qui seront en standard sous Linux. La Figure 35 décrit et explique l'option « iso8859-15 ».

Figure 35: Configurer « native language support » : capture d'écran 2.

Vous avez besoin de l'option « NLS ISO 8859-15 » pour reproduire correctement la FAT de Windows et les extensions Joliet du système de fichiers des CD c'est donc une bonne idée de l'activer. La sélection de « NLS ISO 8859-15 » est appropriée pour les langues Occidentales et elle inclut le symbole Euro. C'est la raison pour laquelle cette table est presque toujours compilée en interne. « NLS ISO 8859-1 » est l'ancienne table pour les langues Occidentales mais sans le symbole Euro.

Figure 36: Configurer les pilotes de console.

La « VGA text console » autorise le mode texte avec une résolution VGA. Presque tout le monde en a besoin, cette option est donc presque toujours activée. Seuls quelques vieux ordinateurs 386 n'ont pas de carte compatible VGA alors que les ordinateurs modernes n'ont pas le plus petit problème avec ce choix. La seconde option, « video mode selection support », rend possible la sélection de la résolution en mode texte lors du processus de démarrage. Cela peut être quelquefois très utile si vous voulez avoir plus de lettres dans une ligne mais cette option n'est généralement pas activée. Les deux dernières options sont expérimentales et je vous conseille de les désactiver.

Figure 37 : Configuration du son.

Dans cette section, vous configurez le son. Si votre distribution utilise les pilotes de son ALSA (comme SuSE 6.3 et au-dessus), il suffit de sélectionner « sound card support » comme MODULE. Les pilotes ALSA sont

compilés et liés plus tard (voir plus loin sous l'en-tête « SuSE and the ALSA sound drivers »). Si votre distribution utilise les pilotes de son standard du noyau, vous devez alors sélectionner celui correspondant à votre carte son. Virtuellement toutes les marques de cartes sont présentes ici, la sélection du bon pilote ne doit donc pas être un problème. Si votre carte son fonctionne bien avec le noyau standard de votre distribution, vous pouvez aussi utiliser les programmes de configuration (comme le « yast2 de SuSE) pour trouver les pilotes requis par votre carte son. Il est rassurant de savoir que le son n'est pas critique : il vous manquera le son si quelque chose se passe mal mais le noyau lui-même fonctionnera correctement.

Figure 38: Configurer « USB support ».

Ma carte mère utilise un port USB mais je ne l'utilise pas. Néanmoins, si je devais désactiver tous les supports USB, SuSE m'afficherait un message d'erreur pendant le démarrage. SuSE, bien sûr, supporte l'USB et tente donc de charger les modules nécessaires, d'où ma sélection du « Support for USB » comme module. Même si ce message d'erreur ne m'importe pas, je l'ai résolu d'une manière élégante en compilant le pilote nécessaire aux ports USB sur ma carte mère. Pour ce faire, le minimum est de mettre « Preliminary USB device filesystem » à « y » et de charger un pilote USB spécifique. Comme ma carte mère Pentium II est assez ancienne, j'ai pris le pilote « UHCI (Intel PIIX4, VIA, ...) » comme module. Mais vous devez sélectionner le module « UHCI Alternate Driver (JE) support » si vous avez une carte mère récente avec le chipset Intel, alors que pour les Compaq, par exemple, vous devriez choisir « OHCI support » comme module. En principe, vous avez seulement besoin de l'un des trois modules mais en cas de doute, vous pouvez choisir les trois. Votre distribution Linux a déjà trouvé celui qui est nécessaire et charge automatiquement le bon.

Simplement valider les ports USB de votre carte mère n'est pas suffisant, vous devez aussi spécifier les pilotes (modules) de l'équipement USB qui est connecté à votre ordinateur. La liste qui figure sous « USB Device Class drivers » possède de nombreux choix. Tout ceci est assez simple et peu de choses peuvent aller mal : une fois encore, dans le doute, lisez les textes d'aide.

Figure 39: Configurer « kernel hacking ».

Celui-ci est facile : NE le sélectionnez PAS ! C'est une option utile pour les programmeurs qui veulent trouver la raison d'un arrêt du noyau ou pour lire le cache du disque dur : l'option est totalement inutile pour les utilisateurs normaux.



Figure 40: Sauvegarder et Sortir.

Pffft, nous avons réussi. Tout ce qui reste à faire est de compiler et d'installer le noyau comme expliqué ci-dessous.

Compiler le noyau

En cliquant sur « Save and Exit », vous stockez vos choix de configuration dans le fichier « `./config` » (ou « `/usr/src/linux/.config` » si vous compilez dans `/usr/src/linux`). Parallèlement, il est pratique de sauvegarder ce fichier et de le copier dans le nouveau répertoire source du noyau si vous ne faites qu'une mise à jour mineure, du noyau 2.4.5 jusqu'au 2.4.6 par exemple. De cette manière vous pouvez (habituellement) garder tous vos anciens choix de configuration, ce qui vous permet d'économiser beaucoup de temps. D'une manière similaire, vous pouvez commencer en utilisant le fichier de configuration du noyau « standard » de votre distribution, qui est appelé dans certaines distributions dans `/boot/config` (copiez-le dans `./config` pour commencer à l'utiliser). Mais si vous mettez à jour le source du noyau et que des problèmes bizarres apparaissent lors de la compilation, c'est bien sûr le premier fichier à enlever ! Car vous avez bien sûr sauvegardé prudemment les anciens fichiers de configuration du noyau qui fonctionne correctement. La procédure de compilation du noyau est la suivante :

```
make dep  
make clean (pour les anciens noyaux)  
make bzImage  
make modules  
make modules_install
```

La Figure 40 indiquait déjà le besoin de la commande « `make dep` ». Vous devez bien sûr exécuter ces commandes dans le répertoire source de Linux « `/usr/src/linux` ». Les noyaux des séries 2.0.x ou plus anciennes ont aussi besoin de la commande « `make clean` », qui supprime les fichiers plus anciens avant la compilation du nouveau noyau. La commande « `make clean` » protégeait des messages d'erreurs qui étaient difficiles à résoudre mais qui provenaient probablement de fichiers objets (.o) qui n'avaient pas été écrasés. La commande « `make bzImage` » compile le nouveau noyau mais ne l'installe pas encore. Vous pouvez aussi compiler le noyau avec d'autres commandes « `make` », i.e. « `make bzlilo` » ou « `make zImage` » mais ces commandes peuvent générer des problèmes. La plupart des noyaux sont trop gros pour permettre un « `make zImage` » proprement exécuté : vous obtenez un message d'erreur pendant la compilation et vous terminez sans noyau. Avec la commande « `make bzlilo` », tout doit être configuré proprement dans des fichiers comme « `/etc/lilo.conf` » mais ce n'est pas toujours le cas. Il est donc plus sûr d'éviter ces dernière commandes. La commande « `make modules` » compile les modules : ils sont installés avec la commande « `make modules_install` ». Cette commande met les modules dans le répertoire « `/lib/modules/2.4.6/` » si la version de noyau actuelle est la 2.4.6 : il change lorsque vous compilez une autre version de noyau. De cette manière les modules correspondants à un noyau particulier sont automatiquement dans un répertoire séparé, ce qui évite les conflits avec des modules obsolètes et autres problèmes similaires. Pendant le démarrage, le noyau Linux sait maintenant dans quel répertoire il peut trouver les bons modules. Mais les fichiers dans « `/lib/modules/2.4.6/` » sont écrasés et les anciens modules restent si vous avez déjà compilé le noyau 2.4.6 et que vous voulez le recompiler. Les anciens modules peuvent alors être laissés de côté même s'ils ne sont plus utilisés par le nouveau noyau. Ce n'est normalement pas un problème mais c'est toujours une bonne idée de prendre le temps de retirer les anciens modules avant d'installer les nouveaux.

Pour éviter les problèmes à l'installation du noyau, vous devez aussi vous assurer que la configuration de lilo « `/etc/lilo.conf` » est correcte et vous devez copier le noyau et le fichier « `System.map` » au bon endroit. Après tout cela, vous devez aussi exécuter la commande « `lilo` ». Une alternative consiste à utiliser « `loadlin` », qui permet le démarrage d'un noyau Linux sous Dos/Windows. Les deux options sont détaillées plus loin.

Configurer lilo

Vous pouvez trouver le fichier de configuration dans le répertoire « /etc » sous le nom « /etc/lilo.conf ». Ouvrez-le avec un simple éditeur de texte ASCII : vous devez avoir installé un éditeur complet comme XEmacs (utilisez alors « xemacs /etc/lilo.conf & »), un éditeur plein écran simple comme kedit ou gedit (utilisez « k(g)edit /etc/lilo.conf ») ou un éditeur de ligne simple tel que pico ou nano (utilisez alors « pico /etc/lilo.conf »). Le fichier lilo.conf ressemblera à quelque chose comme ceci :

```
boot      = /dev/hda
vga       = normal
read-only
menu-scheme = Wg:kw:Wg:Wg
lba32
prompt
timeout = 300
message = /boot/message

    other = /dev/hda1
    label = win98

image = /boot/bzImage
label = linux-2.4.6
root = /dev/hda3
append = "parport=0x378,7 hdd=ide-scsi"

image = /boot/vmlinuz.suse
label = suse
root = /dev/hda3
append = "hdd=ide-scsi"
initrd = /boot/initrd.suse
```

Le contenu détaillé du fichier lilo.conf peut différer de celui ci-dessus pour chaque système et selon les distributions. C'est la raison pour laquelle je vais détailler ce fichier. Les 8 lignes du haut ont déjà le bon format et vous n'avez pas, habituellement, besoin de les changer. La commande « boot » dans la première ligne indique le disque dur physique depuis lequel le système démarre, ce qui signifie que « boot » pointe vers l'endroit du « master boot record ». Dans mon cas, le démarrage se fait depuis /dev/hda, le premier disque dur physique. L'option « vga » indique que le démarrage utilise un mode texte VGA standard de 80x25 caractères. L'option « read-only » signifie que le processus de démarrage monte d'abord la partition Linux en lecture seule. Pendant le démarrage de Linux, un contrôle d'erreur est fait sur les partitions : ce n'est qu'après qu'elles sont remontées avec les permissions de lecture et écriture. La ligne « menu-scheme » indique les couleurs du menu de démarrage « lilo » en mode texte. Avec « lba32 », il est possible de démarrer le système d'exploitation après le 1024^{ième} cylindre, si c'est supporté par le BIOS. Tous les systèmes modernes supportent « lba32 ». Si vous avez des problèmes avec ceci, vous pouvez les régler avec une mise à jour du BIOS, quelque chose qui devient une nécessité avec les gros disques durs qui sont maintenant disponibles. La commande « prompt » force « lilo » à fournir un prompt qui permet à l'utilisateur de choisir le système d'exploitation. L'option « timeout » donne le temps en millisecondes pendant lequel « lilo » attend une entrée au prompt avant de démarrer le système d'exploitation standard. Si « lilo.conf » ne donne pas un système d'exploitation par défaut, comme dans l'exemple donné ici, le processus de démarrage commence avec le premier système d'exploitation rencontré. Dans mon cas, c'est Windows98, donc les personnes qui ne connaissent pas Linux se retrouveront dans un environnement Windows. L'option « message » montre un message alors que « lilo » est exécuté. Sous SuSE, c'est Tux le pingouin, la jolie mascotte de Linux, avec (bien sûr) le texte « SuSE Linux 7.1 ». Vous pouvez voir ce message en tapant « xv /boot/message » ou « gv /boot/message » (quelques fois même « gimp /boot/message ») : « xv » et « gv » (ghostview) sont des programmes shareware avec lesquels vous pouvez visualiser divers types de formats d'images. Veuillez noter que le fichier /boot/message n'existe pas sur des systèmes sans écran de connexion graphique (i.e. les anciennes distributions), dans ce cas, le « message de démarrage » est un simple message texte. Il est en principe possible de montrer votre image préférée pendant le démarrage mais je n'ai pas encore tenté cette

expérience. Toutes les options de « lilo » sont bien sûr documentées dans les pages de manuel auxquelles vous accédez par les commandes « man lilo » et « man lilo.conf ».

Les autres options gèrent le démarrage des divers autres systèmes d'exploitation. Au mieux, vous pouvez démarrer seize différents systèmes d'exploitation ou noyaux. C'est en principe amplement suffisant. Vous définissez le système d'exploitation avec la ligne « label= ». Par défaut, Windows98 (et donc aussi les anciennes versions de Windows et de DOS mais pas Windows NT ou Windows 2000) doit être sur la première partition primaire. Ces systèmes d'exploitation ont donc seulement besoin d'une ligne « other » et d'une ligne « label ». La seconde section, démarré avec « image=/boot/bzImage », démarre le nouveau noyau avec le label « linux-2.4.6 ». Mon répertoire racine Linux est « /dev/hda3 ». La ligne « append = "parport=0x378,7 hdd=ide-scsi" » indique au noyau l'adresse et l'interruption pour le port parallèle (port 0x378, interrupt 7) et spécifie que mon graveur de CD « hdd » doit être adressé par une émulation SCSI. Le nom pour le lecteur de CD dépend du système : dans le mien, c'est « hdd » mais il peut avoir un autre nom dans le votre. L'utilisation d'une interruption est plus une question de préférence personnelle. Une interruption accélère l'impression mais vous pouvez oublier cette commande si vous n'avez pas d'interruption disponible pour le port parallèle (imprimante). Par défaut, Linux utilise le « polling » qui permet au noyau d'utiliser le port parallèle sans une interruption. La dernière section, commençant par « image = /boot/vmlinuz.suse », contient la configuration de lilo faite pendant le processus de configuration de SuSE : j'ai ajouté manuellement la ligne append="hdd=ide-scsi". Le fichier « boot/vmlinuz.suse » est le noyau standard de la distribution. Il est préférable de TOUJOURS sauvegarder ce noyau en cas de problème. La ligne « initrd = /boot/initrd.suse » ne s'applique que pour le noyau installé en standard : elle spécifie le chargement d'une image « ramdisk », un disque virtuel qui est chargé dans la mémoire (ou « random access memory », RAM). Le « ramdisk » contient les modules nécessaires au démarrage correct de Linux : un noyau de distribution doit bien sûr être capable d'accéder à un grand choix de matériel, chose qui n'est réalisable qu'en utilisant de nombreux modules.

L'endroit où stocker le nouveau noyau Linux avant l'exécution de lilo devrait maintenant être évident. Dans cet exemple, les bonnes commandes sont les suivantes :

```
cp /usr/src/linux/arch/i386/boot/bzImage /boot
cp /usr/src/linux/System.map /boot/System.map-2.4.6
lilo
```

Sous SuSE 7.3, vous pouvez aussi faire une seconde copie comme ceci :

```
cp /usr/src/linux/System.map /boot
```

(le System.map original a déjà été renommé)

Si vous avez déjà un noyau avec le nom « bzImage » et que vous voulez garder ce noyau, vous pouvez copier le nouveau noyau dans « /boot/bzImage-2.4.6 » et modifier /etc/lilo.conf : changez /boot/bzImage en /boot/bzImage-2.4.6. La compilation recrée toujours le fichier System.map, qui contient les noms et la configuration des variables importantes du noyau. La commande « depmod -a » crée un fichier qui contient toutes les dépendances des modules du noyau, c'est-à-dire les différentes relations entre le noyau et les modules, entre les modules eux-mêmes, ainsi que les informations du fichier « /etc/modules.conf ». La plupart des distributions Linux, SuSE compris, s'assurent que « depmod -a » est exécuté pendant le démarrage mais il est plus sûr de vérifier que le fichier /boot/System.map existe et qu'il correspond à la bonne version de noyau. La commande « lilo » installe les nouvelles configurations du vieux noyau ou du nouveau noyau. Si vous avez déjà un fichier « /boot/bzImage » que vous écrasez avec un noyau fraîchement compilé mais sans exécuter lilo, vous allez vous retrouver avec un nouveau noyau qui ne peut pas démarrer. L'ancien noyau original de la distribution que vous avez conservé en lieu sûr, fonctionnera toujours car il n'a pas été écrasé. Garder l'ancien noyau permet la compilation et le test de nouveaux noyaux d'une manière responsable.

Utiliser loadlin

Si vous utilisez « loadlin » pour démarrer Linux, vous savez déjà sans doute comment trouver ce programme très pratique, i.e., dans C:\loadlin. Virtuellement toutes les distributions Linux possèdent « loadlin » sur le premier CD dans le répertoire « dosutils ». Vous devez aussi copier le nouveau noyau Linux sur le disque dur dans 'C:\loadlin', avec peut-être un nom unique. En démarrant Windows98 en mode DOS, vous pouvez alors démarrer Linux avec la commande :

loadlin bzImage

Dans des circonstances normales, la plupart des configurations qui sont stockées dans le noyau lui-même, telles que la localisation de la partition root, sont correctes si vous avez compilé votre propre noyau et, dans ce cas, la commande ci-dessus est suffisante pour démarrer Linux sans problèmes. En saisisant (dans DOS) la commande « loadlin | more », vous obtiendrez un écran d'aide incluant un lien vers un « loadlin-HowTo » sur Internet. Avec « loadlin », vous pouvez tester un noyau nouvellement compilé même si vous rechignez à tripoter « lilo » : de toute manière, « loadlin » et « bzImage » rentrent dans une disquette de 1.44 Mo. Il n'y a rien qui puisse aller mal si vous démarrez d'abord DOS avec une disquette de démarrage (avec le support pour EMM386) et lancez ensuite Linux depuis la disquette avec « loadlin » et « bzImage ». Mais vous devez bien sûr avoir une disquette de démarrage DOS.

SuSE et les pilotes de son ALSA

SuSE utilise les pilotes de son standard d'ALSA (Advanced Linux Sound Architecture). Ces pilotes sont de meilleure qualité que ceux du projet OSS, en standard dans le noyau. Si vous avez sérieusement besoin du son sous Linux, vous devez absolument utiliser les pilotes ALSA. Ces pilotes ne font pas partie du code source des (vieux) noyaux, ce qui implique que les pilotes doivent être compilés et installés séparément. Le code source pour les pilotes ALSA est situé dans la partie « zq » de la distribution SuSE : renseignez-vous pour savoir ce que cela signifie. Installez ALSA avec YaST ou YaST2 de manière à ce que le code source soit disponible dans le répertoire « /usr/src/packages/ ». Pour compiler et installer les pilotes ALSA, vous devez faire ce qui suit :

```
rpm -bb /usr/src/packages/SPECS/alsa.spec  
cd /usr/src/packages/BUILD/alsa/alsa-driver-<version number>/  
./configure  
make install
```

La première ligne installe le code source, incluant les pilotes, dans le répertoire « /usr/src/packages/BUILD/ ». De plus, les bibliothèques ALSA et leurs utilitaires sont directement compilés en tant que fichiers rpm. Malheureusement, les pilotes ALSA ne sont pas compilés par défaut. Vous devez les compiler et les installer séparément à la main en utilisant les deux commandes du bas. La commande « ./configure » trouve les configurations et les fichiers nécessaires dans votre système et les intègre dans un fichier de configuration. La commande « make install » compile tous les pilotes ALSA et les installe dans le répertoire « /lib/modules/2.4.6/misc/ » afin qu'ils soient disponibles pour le noyau. Maintenant, lors du démarrage de SuSE, le pilote son désiré est installé automatiquement. J'admets que la procédure suggérée ici est assez lourde mais vous devez savoir exactement ce que vous faites pour trouver les pilotes nécessaires et inclure le support du son dans le nouveau noyau d'une manière plus directe.

Si vous n'utilisez pas SuSE ou si vous voulez utiliser une version plus récente des pilotes ALSA, vous pouvez télécharger ces pilotes, les bibliothèques correspondantes et les utilitaires sur <http://www.alsa-project.org>. La page de démarrage du site donne les dernières nouvelles du projet ALSA (i.e., l'intégration en février 2002 des pilotes ALSA dans l'arbre source du noyau de la série 2.5) et donne le lien vers les divers fichiers à

télécharger. Ci-dessous, je vais expliquer comment compiler les pilotes ALSA : vous pouvez suivre les mêmes étapes pour les bibliothèques et les utilitaires. Décompressez les pilotes dans un répertoire adapté, i.e., « /usr/local/ ». Déplacez-vous dans ce répertoire qui est dans ce cas « /usr/local/alsa-driver-<version-number>/ » et exécutez les commandes suivantes en démarrant par « ./configure ». Il peut être nécessaire de procéder à des étapes additionnelles pour que les pilotes fonctionnent si votre distribution n'utilise pas les pilotes ALSA en standard. Malheureusement, ces problèmes sont hors du cadre de cet article déjà assez long, mais vous pouvez obtenir plus ample information dans la FAQ d'ALSA (« Frequently Asked Questions » ; questions les plus fréquemment posées) que vous pouvez aussi télécharger.

Support de pcmcia (portables)

Le support de pcmcia est standard dans les noyaux depuis les 2.4.x. Le HOWTO PCMCIA, indique néanmoins qu'il faut éviter d'utiliser la version PCMCIA du noyau. Aujourd'hui, le source pcmcia, incluant les scripts et les pilotes, peut fonctionner avec toutes les branches de noyaux : 2.0, 2.2, et 2.4. Selon mon expérience, j'ai constaté que les pilotes pcmcia cessent de fonctionner après une recompilation de noyau et qu'ils doivent aussi être recompilés. Il existe deux solutions en fonction de votre distribution. La première consiste à utiliser le code source de votre distribution. En installant et en compilant le code source vers un fichier rpm, vous obtenez un fichier que vous pouvez installer. L'autre solution consiste à télécharger, décompresser, compiler et installer la version pcmcia la plus récente sur <http://sourceforge.net/projects/pcmcia-cs/>, comme ceci :

```
cp /etc/rc.d/pcmcia /etc/rc.d/pcmcia.SuSE
cp ~/pcmcia-cs-3.1.?.tar.gz /usr/src
cd /usr/src
tar -zxf ./pcmcia-cs-3.1.?.tar.gz
make config
make all
make install
cp /etc/rc.d/pcmcia.SuSE /etc/rc.d/pcmcia
```

La première et la dernière ligne résolvent des problèmes spécifiques à SuSE. Le script d'initialisation pcmcia de SuSE « /etc/rc.d/pcmcia » est écrasé par la commande « make install », ce qui fait échouer le script sous SuSE. Le problème est résolu en recopiant le script original après le « make install ». Si vous avez écrasé le script SuSE original par erreur, vous devez réinstaller le paquetage pcmcia en commençant par la partie « a1 », copier le script original dans un autre fichier, exécuter « make install » à nouveau et enfin recopier le script original.

Le nouveau fichier rpm pour le support pcmcia obtenu avec SuSE est comme suit :

```
rpm -i /cdrom/suse/zq1/pcmcia-3.1.?.spm
cd /usr/src/packages
rpm -bb ./SPECS/pcmcia-3.1.?.spec
cd /RPMS/i386/
rpm -i --force ./pcmcia-3.1.?.rpm
SuSEconfig
```

Dans la première ligne, je suppose que vous installez les pilotes pcmcia depuis les 6ièmes et 7ièmes CD et que le lecteur de CD-ROM est déjà monté sur /cdrom. La commande « rpm -i » installe le code source et la commande « rpm -bb » compile le fichier rpm pcmcia. Vous installez alors ce fichier rpm particulier de la même manière que vous le faites avec tous les autres. Notez que vous devez utiliser l'option « --force » sinon, le programme rpm vous indiquera (à juste titre) que « pcmcia » est déjà installé et ignorera le nouveau

fichier. Comme toujours, vous devez exécuter le programme SuSEconfig (notez les majuscules et les minuscules) lorsque vous avez installé les fichiers rpm à la main sous SuSE, afin d'activer les changements de configuration. Cette dernière chose est faite automatiquement par les programmes d'initialisation de SuSE YaST ou YaST2 après installation ou modification de nouveaux paquetages. Il n'est alors plus nécessaire de les activer à la main.

Pour utiliser le support pcmcia correctement, vous devez laisser le « network support » pendant la compilation mais vous devez désactiver tous les autres pilotes pour cartes réseau. Et, comme déjà dit avec la Figure 11, vous devez bien sûr aussi activer « TCP/IP support » si vous voulez utiliser Internet.

Installer « modutils »

Comme déjà mentionné, le noyau utilise les petits programmes de « modutils » pour gérer les modules du noyau. Ces programmes incluent :

```
insmod (qui installe un module),  
rmmod (pour enlever un module)  
lsmod (pour montrer tous les modules utilisés),
```

parmi d'autres. Avec des commandes comme « man lsmod », vous pouvez trouver leur manière de fonctionner, ce que je ne vais pas détailler ici.

Compiler et installer « modutils » est évident. Tapez simplement :

```
cd /usr/src  
cp ~/modutils-2.4.6.tar.bz2 . (en supposant que le fichier est dans votre)  
                          (répertoire de travail, '~')  
bzip2 -d modutils-2.4.6.tar.bz2 (unzip : ceci peut prendre du temps)  
tar -xvf modutils-2.4.6.tar  
cd modutils-2.4.6 (allez dans le répertoire dans lequel « modutils »  
                  vient d'être décompressé)  
./configure (trouve les configurations spécifiques au système)  
make ( compile « modutils » : comme c'est petit, la compilation peut être)  
      (étonnamment rapide)  
make install (installe « modutils » dans le répertoire « /sbin/ »)
```

C'est tout ce que vous avez à faire pour rendre « modutils » prêt à l'emploi. Notez encore que dans cet exemple « modutils » doit avoir le même numéro de version que le noyau mais cela n'est pas toujours le cas.

Le noyau fonctionne-t-il correctement ?

Le nouveau noyau est configuré, compilé et probablement aussi installé avec lilo. Vous redémarrez le système et vous vous demandez : comment puis-je savoir si le nouveau noyau fonctionnera correctement ? Vous trouverez facilement du matériel qui ne fonctionne pas lorsque vous faites des essais mais en plus, le noyau donne beaucoup d'informations utiles à l'écran pendant le démarrage, avant même que vous ne démarriez la connexion graphique. Cette information contient des choses telles que les configurations qui ont été automatiquement détectées (les ports, IRQs, etc.) mais aussi les messages d'erreur dans le cas où des pilotes ne peuvent pas être initialisés directement, que ce soit en tant que module ou en tant que partie du noyau. Lire

ces messages donne un premier avertissement sur certains problèmes. Pour Linux SuSE, c'est très facile : chaque partie du noyau qui démarre met sur la droite de l'écran le message (vert) « done » si tout fonctionne bien ou « failed » quelque chose se passe mal. Comme les messages peuvent s'afficher à grande vitesse, tous les messages d'erreur sont résumés au-dessus de l'invite de connexion. Vous trouvez l'invite de connexion sous « <Ctrl>+<Alt>+F1 » si vous obtenez l'écran de connexion graphique automatiquement. De cette manière, vous avez au moins une idée d'où proviennent les problèmes si vous comprenez le message d'erreur. Les messages d'erreurs peuvent ressembler à « Cannot find module » ou « Cannot load module », indiquant habituellement que vous n'avez pas inclus certaines parties dont vous avez besoin. Reconfigurer le noyau et recompiler résoud généralement le problème. Notez qu'il n'est pas nécessaire de recompiler tout le noyau. Si vous avez juste oublié quelques modules, il suffit de ré-exécuter « make modules » et « make modules_install ». Recompiler tout le noyau n'est même pas un problème. La plupart du code noyau est déjà compilé et les seules parties qui ont besoin d'être recompilées sont celles qui manquent. En résumé, modifier simplement un peu le noyau et recompiler va très vite, par rapport à la première compilation de noyau. Vous pouvez alors prendre un café en toute quiétude.

Une autre manière de lire les messages de démarrage du noyau est fournie par la commande « dmesg ». Simplement exécuter cette commande appelle les messages qui ont défilé plus tôt à l'écran. Les mettre dans un fichier avec « dmesg > temp » vous permet de lire les messages d'erreur à votre rythme (avec « more temp » ou votre éditeur favori).

Conclusion

Avec ce guide en main, vous pouvez maintenant débiter vos expérimentations avec le noyau d'une manière un peu plus raisonnable. Heureusement le niveau pour commencer à jouer avec le noyau est suffisamment bas pour que vous ayez envie de débiter. La majeure partie de votre temps sera de décider de la bonne configuration du noyau, alors que pendant la compilation, vous pourrez jouer tranquillement avec « freecell » ou faire autre chose.

Si vous avez encore des problèmes et que vous ne vous en sortez pas, vous pourrez utiliser les différentes listes de diffusion Linux et sites internet où vous pourrez poser vos questions. Ceux-ci existent dans différentes langues, pas seulement en anglais. Dans un délai relativement court, vous devriez obtenir une réponse qui vous permettra de résoudre votre problème. La meilleure manière de trouver ces listes et les sites internet est d'utiliser un moteur de recherche.

Site Web maintenu par l'équipe d'édition LinuxFocus
© Philip de Groot
"some rights reserved" see linuxfocus.org/license/
<http://www.LinuxFocus.org>

Translation information:

nl --> -- : Philip de Groot <philipg@authors.linuxfocus.org>

nl --> en: Nino R. Pereira <pereira@speakeasy.org>

en --> fr: Iznogood <iznogood@iznogood-factory.org>