

<https://clx.asso.fr/spip/?Recompiler-son-noyau-Linux>



Etude de cas

Recompiler son noyau Linux Mandrake

- Documentations - Installation / Administration de base de Linux -



Date de mise en ligne : mercredi 25 février 2004

Copyright © Club LinuX Nord-Pas de Calais - Tous droits réservés

Dans [l'épisode précédent](#), nous avons installé la Mandrake 9.2 sur un portable *Acer TravelMate 243*. Celui-ci refusait obstinément les noyaux *Mandrake*. Le noyau de la *Fedora* nous a sorti du pétrin. Néanmoins, le fonctionnement d'une distribution avec le noyau d'une autre n'est pas satisfaisant. Nous avons perdu le confortable « supermount ». La lecture des forums permet de cibler une cause probable : LAPIC (Local APIC avec APIC acronyme de Advanced Programmable Interrupt Controller). Même si saisir l'option `nolapic` au démarrage du système, lors de l'apparition du prompt `boot :`, n'a rien donné. *Mandrake* l'inclut dans son noyau alors que la *Fedora* l'exclut. Compilons donc un noyau sans LAPIC...

Pré-requis

Il faut bien sûr avoir installé une [mandrake 9.2 \[1\]](#), connaître le mot de passe de l'administrateur (root) et avoir accès à une console (ou une fenêtre xterm). Pour illustrer nos propos, nous partons d'une session graphique KDE dans laquelle nous lançons une console fenêtrée (pour utiliser une console non graphique, appuyez simultanément sur les touches `CTRL+ALT+F2`). Dans le menu K, le sous-menu « Terminaux » vous donnera la liste des outils disponibles (Konsole, `eterm` ou `xterm`...). Le lancement de l'un d'eux ouvre une fenêtre dans laquelle nous allons passer administrateur en tapant `su` - et en répondant à la demande du mot de passe administrateur.

Vérification des paquetages nécessaires :

Il faut les paquetages `gcc`, `kernel-source` et leurs dépendances. Dans la version 9.2 de la mandrake, `gcc` est fourni dans sa version 3, la ligne suivante permet de s'assurer de l'installation de `gcc` :

```
urpmi gcc
```

De deux choses, l'une : où le résultat est une liste de paquetages suivie d'une demande de confirmation ou bien il sera écrit « tout est déjà installé ». Dans le premier cas, il faudra donc répondre par l'affirmative et insérer les CD réclamés. Dans le second, vous n'avez rien à faire de plus.

Même manipulation pour les sources du noyau :

```
urpmi kernel-source
```

Si vous avez une version « download edition » de la Mandrake 9.2, `urpmi` ne trouvera aucun paquetage nommé `kernel-source`, il faudra donc rechercher sur Internet et télécharger les paquetages `kernel-source-2.4.22-10mdk.i586.rpm` [2] Pour cela, le meilleur moyen est encore de consulter le site [rpmfind](#) et de lancer une recherche sur `kernel-source`, puis de préciser *Mandrake* dans le champ `System`, dès la première recherche terminée. Une fois téléchargé, il vous faudra encore installer le paquetage. Dans la console graphique, tapez par exemple :

```
urpmi /home/user/kernel-source-2.4.22-10mdk.i586.rpm
```

Remplacez `/home/user` par le chemin menant au fichier `/home/user/kernel-source-2.4.22-10mdk.i586.rpm`.

L'utilitaire `urpmi` récupérera éventuellement les paquetages nécessaires à la compilation du noyau : `automake`, `autoconf` et consors.

Préparation et Configuration

Allez dans le répertoire `/usr/src/linux` en tapant :

```
cd /usr/src/linux
```

En réalité, vous serez dans :

```
/usr/src/linux-2.4.22-10mdk... Là où tout se passe.
```

En effet, `/usr/src/linux` est un lien pointant vers `/usr/src/linux-2.4.22-10mdk`.

Trois possibilités vous sont offertes pour configurer notre futur noyau. « `make config` » est le moyen la plus rustique : mode texte, sans couleur, avec une succession de questions. Une version texte (interface ncurses) mais avec des couleurs et un menu déroulant se lancera via la commande « `make menuconfig` ». Enfin, une interface graphique est lancée par la commande « `make xconfig` » (nécessite les bibliothèques graphiques tk). Nous utiliserons ici le deuxième moyen (texte avec menus) qui est un bon compromis entre aridité et fiabilité. Donc :

```
make menuconfig
```

[https://clx.asso.fr/spip/local/cache-vignettes/L400xH270/compil_noyo_400x200-dbffa.png]

Compilation du noyau - choix des options de compilation via `make menuconfig`

Les options de compilations du noyau spécifiques à l'architecture sont disponibles dans cette fenêtre.

On parcourt les menus à l'aide de la touche `Tab` (pour tabulation, à côté de la touche `A` et généralement représenté par deux flèches opposées). La barre d'espace sert à valider son choix ou à cocher une case. Dans les menus, la séquence « `--->` » signifie un sous-menu. Un appui sur la barre d'espace permet de l'ouvrir. Pour chaque option (ou presque, vous aurez le choix entre 3 possibilités, correspondantes à trois caractères " ", * et m :

- [] qui signifie « non sélectionné » ou décoché,
- [*] pour une option sélectionnée donc intégrée au noyau,
- [m] pour une option disponible en module (donc « activable » via la commande `modprobe`).

En parcourant les menus et sous-menus, à la recherche de notre **LAPIC** que l'on soupçonne de bloquer notre noyau, nous rencontrons des options que l'on sait inutiles pour notre PC portable : **old cdrom driver**, **IEEE 1394** (pas de connecteur firewire sur cette machine), **IRDA** (pas d'infrarouge non plus)...

Nous les décochons. Assurez-vous que vous n'avez pas besoin (ou que l'un des périphériques dont est équipé votre PC n'en a pas besoin) de l'option que vous venez de désactiver. Dans le doute, laissez l'option telle quelle (surtout lorsqu'il s'agit d'un module). En effet, le système ne le chargera en mémoire à la demande.

Enfin, nous trouvons l'option tant recherchée, **LAPIC**, intégrée au noyau. Décochons-la.

[https://clx.asso.fr/spip/local/cache-vignettes/L400xH72/compil_noyo_details-20a6d.png]

Détail des options de compilation du noyau Linux

Voici la fameuse option Local APIC qui bloque le lancement du systÉme avec l'ordinateur portable TravailMate 243.

Résumé de la configuration :

- **Processor Type and Feature** -Â» Local APIC (on décoche pour la compatibilité avec l'ACER) ;
- **General Type** -Â» aucun changement ;
- **Memory Tech. Device** -Â» aucun changement ;
- **Parallel Port** -Â» aucun changement ;
- **Plug and Play configuration** -Â» aucun changement ;
- **Block Device** -Â» aucun changement ;
- **RAID** -Â» aucun changement ;
- **Networking** -Â» aucun changement ;
- **Telephony** -Â» aucun changement ;
- **ATA/IDE** -Â» aucun changement ;
- **SCSI** -Â» aucun changement (Ne l'enlevez pas. Certains périphériques IDE peuvent être émulsés en SCSI. C'est le cas du combo graveur CD/ lecteur DVD de ce portable) ;
- **Fusion MPT** -Â» aucun changement ;
- **IEEE 1394** -Â» décoché car ce portable n'a pas de connecteur firewire ;
- **IRDA** -Â» décoché car ce portable n'a pas de port infrarouge ;
- **old CDROM no ide/scsi** -Â» décoché.

Par défaut, le noyau sera nommé 2.4.22-10.mdkcustom. Vous pouvez modifier le nom du noyau en éditant le fichier *Makefile* du répertoire */usr/src/linux*. Dans les toutes premières lignes de ce fichier, vous remarquerez la ligne *EXTRAVERSION=* qui désigne le numéro de version personnel du noyau. Dans notre exemple, *EXTRAVERSION* est égale à *-10.mdkcustom*.

La compilation

Elle peut se faire en une ou plusieurs commandes.

- En une seule fois :

```
make dep clean bzImage modules modules_install install
```

- En 6 commandes :

```
make dep
make clean
make bzImage
make modules
make modules_install
make install
```

(Attention au *i* de *bzImage*, il est en majuscule !)

Vous pouvez bien sûr choisir une possibilité intermédiaire, le tout étant de respecter l'ordre de compilation.

`dep` recherche des dépendances est très court

`clean` nettoie les résidus éventuels d'une précédente compilation (rapide aussi)

`bzImage` compile le noyau (comptez un peu plus de temps)

`modules` compile les modules (étape la plus longue)

`modules_install` stocke les modules dans le répertoire `/lib/modules` (ils étaient stockés dans un répertoire temporaire de vos sources)

`install` copie les fichiers, dont le noyau, dans le répertoire `/boot`, met à jour les liens symboliques de ce même répertoire, crée le fichier `initrd` et ajoute l'entrée correspondant au noyau dans `/etc/lilo.conf`.

Contrôlez que la copie et la mise à jour des liens dans le répertoire `/boot` ait été correctement effectuée. Vous devez en effet y retrouver les fichiers suivants : `vmlinuz`, `initrd (.img)`, `config`, `system.map`. Tous ces noms doivent être suivis du numéro de version personnalisé : 2.4.22-10mdkcustom.

Il faut ensuite vérifier le contenu du chargeur de démarrage Lilo (ou Grub si vous l'avez préféré) :

```
cat /etc/lilo.conf
```

Vérifiez que vous avez bien une entrée 2.4.22-10mdkcustom identique à l'entrée d'origine (2.4.22-10mdk), à l'exception du numéro de version pour les lignes suivantes : `image`, `label` et `initrd`.

Ne supprimez pas les lignes correspondantes à l'ancienne version du noyau. Ce dernier sera nécessaire si votre noyau personnalisé ne vous permet pas de lancer Linux. C'est le cas par exemple si vous avez oublié de compiler un module indispensable au démarrage de votre machine ou si le noyau est défaillant.

Testez, c'est fini.

Rétablir l'ancien noyau

Dans le cas où cela ne fonctionnerait pas, vous devez remettre par défaut l'ancien noyau au démarrage :

```
cd /boot
ls -la
```

Vous voyez des liens de `vmlinuz` vers `vmlinuz-votreversion`, `system.map` vers `system.map-votreversion`, etc. Il faut rétablir les liens vers l'ancien noyau. Saisissez alors :

```
ln -s initrd-2.4.22-10mdk.img initrd.img
ln -s system.map-2.4.22-10mdk system.map
ln -s vmlinuz-2.4.22-10mdk vmlinuz
```

Les liens `kernel.h` et `config` ne devraient pas avoir été changés mais vérifiez quand même et le cas échéant, modifiez-les.

Pour aller plus loin...

Recompiler son noyau Linux Mandrake

On pourrait en profiter pour patcher le noyau 2.4.22 en 2.4.24 ou même passer sur un noyau 2.6, après avoir récupéré les sources sur kernel.org. Pour le moment, j'y ai renoncé : utiliser les patches 2.4, impose de répondre à plein de questions plus ou moins complexes (Mandrake propose déjà des kernel hyperpatchés). Quant au noyau 2.6.1, il aboutit dans mon cas à un *kernel panic* (message d'erreur du noyau) parce qu'il n'a pas su monter les partitions ext3 [3].

Quelques liens :

- Un article du Linux Journal consacré à la compilation du noyau Linux <http://www.linuxfocus.org/Francais/...>,
- des échanges très intéressants sur APIC et LAPIC <http://linuxfr.org/~redfish/2724.html>.

[1] NDLR : Le principe général de la compilation s'applique à toutes les distributions linux ; la méthode est donc facilement transposable.

[2] ou [kernel-source-2.4.22-26mdk](#), voir un paquetage plus récent, si vous souhaitez bénéficier des derniers correctifs de sécurité (NDLR : cette dernière option est chaudement recommandée).

[3] NDLR : il ne suffit pas d'installer les sources du noyau 2.6, il faut également installer de nouveaux outils pour la gestion des scripts d'initialisation et la gestion des modules. Pour tester un noyau 2.6 sur une Mandrake 9.2, mieux vaut passer par Cooker grâce à [urpmi](#) et [easyurpmi](#).