

# Sommaire

<b>Installation d'un noyau 2.4.2</b> .....	<b>1</b>
Introduction.....	1
Procédure d'installation avec "make xconfig".....	1
Options "m".....	2
/boot et LILO.....	3
Les modules en général.....	3
Réflexions générales.....	4
<b>Copyright</b> .....	<b>6</b>

# Installation d'un noyau 2.4.2

par Daniel Moyne, mis en page et corrigé par Fred et Jicé  
Passer d'un noyau standard à un noyau compilé

---

Je me suis lancé à l'abordage de la procédure d'installation du noyau 2.4.2 en partant d'une distribution Mandrake 7.2 voulant savoir comment s'affranchir d'une pseudo-tutelle Mandrake dans le monde du logiciel libre !

## Introduction

Disons le de suite le noyau est installé, mais le fruit est amer : j'ai un nouveau moteur dans ma voiture mais l'autoradio (plus de son pas trop grave) et la boîte de vitesses (plus de Kpp) (**Note de Fred** : solution apportée au cours de l'article) ne fonctionnent plus. Malgré tout je suis satisfait de l'exercice qui m'a permis de démystifier cette procédure tant redouté de tous. J'ai même réussi à intégrer le nouveau module 1.32 de ma carte Tekram DC395 UW en le compilant séparément !

Tout ce qui est décrit ci-dessous correspond au choix de l'option "CONFIG\_MODVERSION is not set" tel que conseillé dans tout ce que j'ai lu. Qu'ai-je appris :

## Procédure d'installation avec "make xconfig"

Le fichier noyau est décompacté en "2.4.2" sous "/usr/src/" et le cher "linux" est modifié pour pointer sur "2.4.2". Avant il pointait vers "2.2.17". Même s'il reste sur ce nouveau noyau expérimental cela ne semble pas avoir de conséquence sur le retour à une utilisation normale du noyau 2.2.17-21mdk !

Quand on commence, on ne peut pas bénéficier de l'ancien fichier de configuration du noyau 2.2.17-21mdk. En ce qui me concerne je ne l'ai pas trouvé. En fait il faudrait que Mandrake recrée le fichier de configuration après installation puisque leur procédure est quelque peu différente : elle part d'une reconnaissance du matériel et prépare le noyau en conséquence.

**Note de Fred** : Pour récupérer le fichier de configuration du 2.2.17, il faut, avant de modifier les liens, lancer dans /usr/src/linux : `make xconfig` puis sélectionner : **Store Configuration To File**. Choisissez un nom absolu pour le retrouver : /root/kernel.2.2.17.cfg est un bon nom. Puis vous modifiez les liens comme indiqué précédemment et vous lancez le `make xconfig` et vous sélectionnez : **Load Configuration From File**.

Suggestion à faire ?

- CONFIG\_MODVERSION is not set
- Pour ceux qui comme moi s'interrogeaient sur le problème de Powerdown qui n'éteignait pas l'ordinateur : il faut sélectionner: CONFIG\_APM=y. Surtout que tous les commentaires sur ce point précis laissent entendre une application spécifique aux portables, ce qui semble faux.
- CONFIG\_\*\_FS : préférer "y" voir plus bas les problèmes liés aux options "m".

Conserver chaque fichier de configuration pour comprendre l'effet d'une nouvelle option activée.

L'utilisation d'un script regroupant les commandes importantes est conseillé. Chez moi le script s'appelle Install\_kernel, le voici :

```
#!/bin/bash
#
# Ce script se lance dans le source du noyau à installer en root
#
echo Phase de nettoyage
make dep && make clean
echo Phase de compilation
make bzImage && make modules && make modules_install
echo Phase installation noyau
make install
echo Phase initr
#mkinitrd -f -v /boot/initrd-2.4.2.img 2.4.2
```

Je n'ai pas inclus la dernière ligne car je fais une opération manuelle de mise à jour du module SCSI avant son exécution (voir plus bas).

Après exécution de ce script on retrouve un dossier "2.2.4" dans /lib/modules et les fichiers "vmlinuz" et "vmlinuz-2.4.2" dans /boot ainsi que "System.map" qui pointe sur "System.map2.4.2". Le fichier "System.map" se remet automatiquement à jour pour pointer sur le bon fichier au boot d'un noyau donné, l'ancien noyau 2.2.17-21mdk par exemple.

## Options "m"

Les modules décrivant le(s) file system(s) de root (situés en général dans le dossier "fs" du source) doivent avoir l'attribut "y", c'est à dire qu'ils doivent être intégrés au noyau sinon comment lire les informations de la partition root (celles décrivant les modules) !

**Note de Jicé** : pas forcément : c'est le boulot de l'initrd de stocker les modules nécessaires pour le système de fichiers de la partition root (drivers de la carte SCSI, modules du file system, etc.)

Quand on possède une carte SCSI et qu'on a opté pour l'option "CONFIG\_MODVERSION is not set", il faut absolument qu'elle soit reconnue avant montage du file system correspondant et donc là il faut lancer la commande :

```
mkinitrd -f -v /boot/initrd-2.4.2.img 2.4.2
```

qui va créer ce qu'on appelle une image initiale ramdisk de nom "initrd-2.4.2.img" sous "/boot" et qui sera lancée à condition d'ajouter la ligne suivante :

```
initrd=/boot/initrd-2.4.2.img
```

dans la section boot correspondante du fichier "lilo.conf". Aussi chez moi cette nouvelle section est ainsi :

```
# Section noyau Linux-2.4.2new
image=/boot/vmlinuz
  label=linux-2.4.2new
  root=/dev/hdb5
  initrd=/boot/initrd-2.4.2.img
```

```
vga=788
read-only
```

Que fait cette commande ? Elle utilise le fichier `/etc/modules.conf` pour y trouver chez moi l'entrée du type : `alias scsi_hostadapter dc395x_trm`. Donc gros problème : cette info est caractéristique chez moi du noyau 2.2.17-21mdk avec les modules correspondants tels que contenus dans `/lib/modules/2.2.17-21mdk`, mais quand on va lancer la commande :

```
mkinitrd -f -v /boot/initrd-2.4.2.img 2.4.2
```

le système va faire une recherche dans `/lib/modules/2.4.2` où chez moi le fichier `dc395x_trm.o` n'existe pas et cette erreur est bien signalée à l'exécution de la commande ! Comment faire ? Tout simplement recopier le fichier de votre ancien noyau 2.2.17-21mdk dans le dossier SCSI correspondant du nouveau noyau 2.4.2 et mettre à jour la liste `modules.dep` bien sûr dans `/lib/modules/2.4.2` et relancer la commande `mkinitrd`. Cette commande utilise aussi `fstab` pour y lire des informations relatives au(x) filesystem(s) installés sur le support SCSI.

## /boot et LILO

Le fichier `vmlinuz` tel que créé par la procédure de création et d'installation du nouveau noyau 2.4.2 pointe (lien) vers le fichier `vmlinuz-2.4.2` car il a écrasé (beurk....!) le fichier de même nom qui avant pointait vers le noyau `vmlinuz-2.2.17-21mdk`. Ne pas oublier de corriger LILO pour conserver la possibilité d'accès au noyau qui fonctionne : le `2.2.17-21mdk`, sinon adios ....! Ainsi chez moi :

```
# Section noyau Linux 2.2.17-21mdk
image=/boot/vmlinuz-2.2.17-21mdk
    label=linux
    root=/dev/hdb5
    initrd=/boot/initrd-2.2.17-21mdk.img
    vga=788
    read-only
```

et idem pour les sections variantes du même noyau. A noter aussi que `initrd` pointe vers `initrd-2.2.17-21mdk.img`. J'ignore si c'était comme ça à l'origine, où par l'intermédiaire d'un lien (car je l'ai peut-être modifié ?), mais in fine ça a maintenant le même effet qu'avant et c'est ça l'important. Donc pour résumer : après compilation une mise en cohérence de `lilo.conf` et du contenu de `/boot` s'impose. Toute erreur fait mal, très mal ! Une dernière chose concernant chaque section : les indentations correspondent à des TAB, ce ne sont donc pas des espaces sinon ça plante. Vous constaterez l'effet d'une suppression du premier caractère.

Ne pas oublier d'exécuter la commande `lilo` qui met à jour le chargeur de boot (bootloader) en fonction de votre `lilo.conf`, et donc fait apparaître toutes les possibilités de boot qui y sont définies. A vérifier encore et encore pour s'assurer que l'ancien noyau décrit dans sa section est parfaitement accessible avant de faire "arrêt/redémarrer".

## Les modules en général

Gros problème de dénomination : on ne retrouve pas forcément les mêmes noms. Par exemple le module `"supermount.o"` n'existe pas dans le nouveau noyau et cette option activée chez moi dans `fstab` sur zip,

floppy et cdrom ne peut être activée, ce qui en plus après boot sur le noyau 2.4.2 me fait perdre l'utilisation de ces medias.

**Note de Fred** : pour que ces médias soit activés, il faut modifier `/etc/auto.misc` de la manière suivante :

```
graveur -fstype=auto,ro,nosuid,nodev :/dev/scd0
cdrom -fstype=auto,ro,nosuid,nodev :/dev/scd1
disquette -fstype=auto,nosuid,nodev :/dev/fd0
zip -fstype=auto,ro,nosuid,nodev :/dev/sda4
```

et d'activer l'automount dans le noyau (`CONFIG_AUTOFS4_FS=y`) et au démarrage: `chkconfig --add autofs`. Les différents médias sont alors accessibles dans `/misc/cdrom`, `/misc/graveur`, `/misc/zip` et `/misc/disquette`. Pour retirer l'un de ces média il faut alors attendre au moins une minute (les cd et zip sont bloqués mais pas la disquette).

Le module "ppp" nécessaire à Kppp n'existant pas dans le nouveau noyau n'est pas chargé et donc adios les connexions Internet !

**Note de Fred** : pour activer ppp il faut (cf `/usr/src/linux/Documentation/Changes`) :

1. Créer un nouveau périphérique : `/dev/ppp`

```
mknod /dev/ppp c 108 0
```

1. Ajouter les lignes suivantes à `/etc/modules.conf` :

```
alias char-major-108 ppp_generic
alias /dev/ppp ppp_generic
alias tty-ldisc-3 ppp_async
alias tty-ldisc-14 ppp_synctty
alias ppp-compress-21 bsd_comp
alias ppp-compress-24 ppp_deflate
alias ppp-compress-26 ppp_deflate
```

1. Activer les modules correspondants :

```
PPP support
PPP support for async serial ports
PPP support for sync tty ports
PPP BSD-Compress compression
PPP Deflate compression
```

Ceci doit s'appliquer à d'autres utilitaires que je n'ai pas encore recensés.

## Réflexions générales

Ce qui est au-dessus résume l'ensemble des opérations d'installation du noyau 2.4.2 réalisées ces jours derniers. Il me reste à tenter l'option : `CONFIG_MODVERSION=y` qui elle va peut-être me permettre d'accéder aux ressources du noyau 2.2.17-21mdk sans pépins ! Je suis malgré tout convaincu qu'une description du matériel a posteriori est une erreur. Une vraie procédure d'installation doit faire le choix de ses ressources par reconnaissance matérielle préalable, finalement en :

## Kernel–kernel24–2

- prenant dans l'ancien noyau et dans le nouveau tout ce qui est nécessaire,
- ne conservant pour ce qui est des modules de même nom que les plus récents sur accord de l'utilisateur,
- en construisant ainsi /boot et lib/modules/nouveauikernel sur lancement d'une procédure de compilation et l'installation entièrement transparente,
- en conservant surtout la possibilité de lancement du noyau stable (ce qui n'est pas le cas aujourd'hui car des liens importants sont supprimés).

Vos commentaires constructifs sont évidemment les bienvenus.

Vous pouvez aussi lire cette [kernel24.php3 documentation].

Cette page est issue de la documentation 'pré-wiki' de Léa a été convertie avec HTML::WikiConverter. Elle fut créée par Daniel Moyne le 26/03/2001.

# Copyright

Copyright © 26/03/2001, Daniel Moyne



*Ce document est publié sous licence Creative Commons  
Attribution, Partage à l'identique, Contexte non commercial 2.0 :  
<http://creativecommons.org/licenses/by-nc-sa/2.0/fr/>*