

0.1 Thème II (noyau 2.4.x) : Installation d'un nouveau noyau

Dans cette partie nous allons créer un noyau tout neuf. Le but de cet article, n'est pas de faire une présentation exhaustive des options du noyau, loin de là. Nous souhaitons, ici, aboutir à coup sûr à un noyau fonctionnel supportant les fonctionnalités que vous désirez. Pour cela, nous essayerons de nous dégager de la structure en menu, des outils de configuration du noyau, pour présenter plutôt une approche par tâches.

0.1.1 1. Pré-requis :

Pour installer votre noyau, assurez-vous d'avoir les paquets suivants :

- la librairie ncurses-5, certaines distributions l'appellent libncurses5 et libncurses5-dev (ou libncurses5-devel)
- l'utilitaire bzip2
- l'utilitaire gzip
- tout le nécessaire de compilation (gcc, make ...)
- les sources du noyau ¹
- sous debian pensez à installer également les paquets : debhelper et modutils

0.1.2 2. Décompression des sources :

Pour décompresser les sources tapez simplement les commandes suivantes :

```
cd /usr/src/  
tar -xjvf /où_est/linux-2.4.20.tar.bz2  
mv linux linux-old  
ln -sf linux-2.4.20 linux
```

Si vous avez un patch à appliquer, vous pouvez l'appliquer ainsi :

```
cp /où_est/nouveau.gz /usr/src/linux/  
cd /usr/src/linux/  
gunzip nouveau.gz  
patch -p1 nouveau.patch
```

Pour annuler ce patch, il suffira de rappeler patch. Ceci suppose bien-sûr que le patch soit compressé avec gzip.

0.1.3 3. Le ./configure :

Rappelons que cette étape a pour but, la génération du fichier de configuration, qui indiquera au compilateur les parties à inclure dans l'exécutable. Malheureusement, il n'y a pas de script nommé configure, chargé d'inspecter votre configuration puis de générer le fichier de configuration qui convient. Précisons également, que le fichier de configuration du noyau n'est pas config.h mais .config (dans /usr/src/linux/), peu importe, ils jouent le même rôle. C'est à dire que la modification de ce fichier n'influe en rien sur l'exécutable déjà présent sur votre système. Pour changer de noyau, il faut le re-compiler et le re-installer exactement comme avec les autres logiciels. Le noyau possède plus de 500 paramètres, le but ici n'est pas de les énumérer, d'autant que leur agencement suppose quelques subtilités. Si c'est que

¹ <http://www.kernel.org/pub/linux/kernel/v2.4/>

vous cherchez, consultez la doc officielle. Nous allons utiliser une méthode que j'espère plus digeste. Je vous fournis un fichier `.config`² prêt à l'emploi, contenant le minimum vital et un peu plus, puis à l'aide de petits mémos, j'indique les cases à cocher pour lui adjoindre de nouvelles fonctionnalités.

0.1.4 3.1 Interface de configuration :

Téléchargez le fichier, puis copiez-le dans le répertoire `/usr/src/linux/` :

```
cp /ou_est/config-trust /usr/src/linux/.config
```

Vous pouvez également utiliser le fichier de config de votre noyau actuel. Ce fichier est souvent localisé dans `/boot/`, il s'appelle `/boot/config-ide-2.4.19`, `/boot/config-ide-2.4.18-bf2.4`, `/boot/config-2.4.18-3` ou `/boot/config-2.4.19-16mdk`, peu importe, copiez-le, chez moi (Slackware 9) ça donne :

```
cp /boot/config-ide-2.4.19 /usr/src/linux/.config
```

Lancez l'utilitaire : `make menuconfig`, depuis le répertoire `/usr/src/linux/`. C'est une interface ncurses, rudimentaire, mais qui conviendra pour ce qu'on souhaite faire. Elle s'organise en menus, vous pouvez naviguer grâce aux touches directionnelles. Pour valider une action, positionnez-vous sur **Select** et pressez la touche **[enter]**, pour changer une option (**[N]** pour non, **[M]** pour module, **[*]** pour yes) utilisez la touche **[espace]**.

0.1.5 3.2 Les mémos :

Ne supprimez aucune option du fichier, sauf quand ça vous est indiqué ou quand il y'a un choix exclusif à faire, ou bien-sûr quand vous savez ce que vous faites. Limitez-vous à ce qui existe sur votre PC, par exemple si vous n'avez pas de carte tv, vous pouvez ignorer le mémo correspondant. 1. Processeur et agp :³ 2. Activer le support du bus PCI :⁴ 3. Les options indispensables :⁵ 4. Activer le support du bus ISA :⁶ 5. Port série et parallèle :⁷ 6. Souris ps/2 :⁸ 7. Périphériques USB :⁹ 8. Périphériques sur Firewire :¹⁰ 9. Périphériques IDE :¹¹ 10. Périphériques SCSI :¹² 11. Mémoire flash :¹³ 12. Réseaux, modem adsl, Ethernet :¹⁴ 13. Carte son :¹⁵ 14. Carte TV :¹⁶ 15. Systèmes de fichiers :¹⁷ 16. Systèmes de fichiers Réseaux :¹⁸ 17. Encodage :¹⁹ 18. Kernel Hacking :²⁰ Quand vous êtes satisfait, choisissez

² <http://www.trustonme.net/didactels/downloads/config-trust>

³ http://www.trustonme.net/didactels/195.html#proc_agp

⁴ <http://www.trustonme.net/didactels/195.html#pci>

⁵ <http://www.trustonme.net/didactels/195.html#needed>

⁶ <http://www.trustonme.net/didactels/195.html#isa>

⁷ http://www.trustonme.net/didactels/196.html#serie_par

⁸ <http://www.trustonme.net/didactels/196.html#ps2>

⁹ <http://www.trustonme.net/didactels/196.html#usb>

¹⁰ <http://www.trustonme.net/didactels/196.html#firewire>

¹¹ <http://www.trustonme.net/didactels/197.html#ide>

¹² <http://www.trustonme.net/didactels/197.html#scsi>

¹³ <http://www.trustonme.net/didactels/197.html#flash>

¹⁴ http://www.trustonme.net/didactels/198.html#res_adsl

¹⁵ <http://www.trustonme.net/didactels/199.html#son>

¹⁶ <http://www.trustonme.net/didactels/199.html#tv>

¹⁷ <http://www.trustonme.net/didactels/199.html#files>

¹⁸ http://www.trustonme.net/didactels/199.html#res_files

¹⁹ <http://www.trustonme.net/didactels/199.html#encodage>

²⁰ <http://www.trustonme.net/didactels/199.html#hacking>

<exit>, à la question, **Do you wish to save your new kernel configuration ?** choisissez <Yes>. Voilà, votre noyau est configuré. Un petit gadget pour la route, il est possible de personnaliser le nom de son noyau, pour cela, éditez le fichier /usr/src/linux/Makefile, voici une copie du mien :

```
VERSION = 2
PATCHLEVEL = 4
SUBLEVEL = 20
EXTRAVERSION = -slack
```

Mon noyau s'appellera donc 2.4.20-slack, vous pouvez évidemment faire en sorte qu'il s'appelle : 3.2.45-king, mais je vous conseille de ne modifier que la variable EXTRAVERSION.

0.1.6 4. make :

Comme annoncé dans l'introduction, le make sert à générer un exécutable, pour ce faire, il suit les indications contenues dans le .config. Rappelons que la compilation est une traduction, au même titre que le passage de l'anglais au français. On passe d'un fichier ascii, contenant du code source C à un fichier binaire (contenant des 0 et des 1) appelé exécutable. Pour la plus part des logiciels, la commande make suffit à lancer la compilation, pour le noyau c'est :

```
make dep clean bzImage modules
```

dep : génère les dépendances **clean** : nettoie les sources, en supprimant les fichiers objets des précédentes compilations **bzImage** : compile le noyau et génère l'exécutable qui vous servira de noyau **modules** : compile les modules

0.1.7 5. make install :

Rien de bien compliqué ici non plus, puisqu'il s'agit d'installer le noyau et les modules. On commence par le noyau :

```
cd /usr/src/linux/
cp arch/i386/boot/bzImage /boot/vmlinuz.2420
cp System.map /boot/System.map-2420
```

Le .2420 c'est parce que j'installe un noyau 2.4.20, mais le nom n'a pas d'importance, évitez néanmoins de l'appeler vmlinuz tout court. A ce stade votre noyau est installé, mais pour être totalement en règle avec la syntaxe Linux, exécutez également les commandes suivantes :

```
cd /boot/
mv System.map System.map.old
mv vmlinuz vmlinuz.old
ln -s vmlinuz.2420 vmlinuz
ln -s System.map-2420 System.map
```

Il ne vous reste plus qu'à installer les modules :

```
cd /usr/src/linux/
make modules_install
```

0.1.8 6. Mise à jour du chargeur d'OS :

Maintenant que votre noyau est installé, il vous reste à mettre à jour votre chargeur d'OS, il peut s'agir de lilo ou grub.

0.1.9 6.1 Si vous utilisez Lilo :

Editez votre fichier `/etc/lilo.conf`, focalisez sur la zone qui démarre Linux (`label=linux`), chez moi elle ressemble à ce qui suit, il peut y avoir quelques variations chez vous, cela n'a aucune importance :

```
image = /boot/vmlinuz
root = /dev/hda4
append = "hdc=ide-scsi hdd=ide-scsi"
label = Linux
vga = 773
read-only
```

Vous devez la dédoubler et modifier chacun des deux blocs, concrètement voici en gras les modifications à apporter :

```
# Ancien noyau :
image = /boot/vmlinuz.old
root = /dev/hda4
append = "hdc=ide-scsi hdd=ide-scsi"
label = mylinux-old
vga = 773
read-only
# Nouveau noyau :
image = /boot/vmlinuz
root = /dev/hda4
append = "hdc=ide-scsi hdd=ide-scsi"
label = Linux
vga = 773
read-only
```

Note : si vous aviez une ligne comme : `initrd=/boot/initrd.img`, je pense à RedHat et Mandriva, vous pouvez l'effacer, des paramètres du nouveau noyau. Le reste doit rester inchangé. Quand vous êtes satisfait, validez vos changements en tapant : `[fixed]/sbin/lilo -v[fixed]` Voilà vous pouvez redémarrer, au boot vous choisirez Linux, en cas de difficultés vous pourrez toujours revenir à l'ancienne version en choisissant `mylinux-old` et recommencer plus consciencieusement.

0.1.10 6.2 Si vous utilisez Grub :

Pour grub c'est le fichier `/boot/grub/menu.lst`, qu'il faut éditer, la section linux (title Linux) ressemble à :

```
title Linux
root (hd0,3)
kernel /boot/vmlinuz root=/dev/hda4 hdc=ide-scsi hdd=ide-scsi
vga=773
```

Vous devez la dédoubler et modifier chacun des deux blocs, concrètement voici en gras les modifications à apporter :

```
# Ancien noyau :
title mylinux-old
root (hd0,3)
kernel /boot/vmlinuz.old root=/dev/hda4 hdc=ide-scsi hdd=ide-scsi
vga=773
# Nouveau noyau
title Linux
root (hd0,3)
kernel /boot/vmlinuz root=/dev/hda4 hdc=ide-scsi hdd=ide-scsi
vga=773
```

Note : si vous aviez une ligne comme : `initrd (hd0,5)/boot/initrd.img`, je pense à RedHat et Mandriva, vous pouvez l'effacer, des paramètres du nouveau noyau. Le reste doit rester inchangé. Voilà vous pouvez redémarrer, au boot vous choisirez Linux, en cas de difficultés vous pourrez toujours revenir à l'ancienne version en choisissant `mylinux-old` et recommencer plus consciencieusement. Introduction²¹ Thème I : Update d'un noyau fonctionnel²²

²¹<http://www.trustonme.net/didactels/192.html>

²²<http://www.trustonme.net/didactels/193.html>