

## 0.1 Thème I (noyau 2.4.x) : Update d'un noyau fonctionnel

Dans cette partie nous allons supposer que vous avez déjà un noyau fonctionnel, mais auquel il manque une ou deux fonctionnalités. Par exemple, le module `n_hdlc`. Il est donc, dans ce cas, inutile de passer en revue les quelques 500 options possibles du noyau.

### 0.1.1 1. Pré-requis :

Pour upgrader 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` ...)

### 0.1.2 2. Installer les sources :

Votre distribution est livrée avec les sources du noyau qui anime votre machine. Vous pouvez bien-sûr prendre les sources du noyau correspondant sur [kernel.org](http://kernel.org)<sup>1</sup>, mais le noyau de votre distribution contient des patches et des corrections de bugs, il est préférable d'en profiter. Concrètement :

- Pour les utilisateurs de Redhat : scanner vos cdroms à la recherche du paquet `kernel-source`, et installez le par :

```
rpm -Uvh kernel-source-*.rpm
```

- Pour les utilisateurs de Mandriva : c'est plus simple, il suffit de taper :

```
urpmi kernel-headers kernel-source
```

- Pour les utilisateurs de Debian : très simple également :

```
apt-get install debhelper modutils kernel-source-2.4.18 kernel-package
```

- Pour les utilisateurs de Slackware : scannez votre cdrom (`slackware/k/`) et installez `kernel-source` ainsi :

```
installpkg kernel-source-2.4.19-noarch-1.tgz
```

- Pour les autres : Regardez vos cdroms, votre distribution propose probablement un paquet `kernel-source`, sinon téléchargez les sources depuis [kernel.org](http://kernel.org)<sup>2</sup>. Pour les installer :

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

```
tar -xjvf /où_est/linux-2.4.19.tar.bz2
```

Terminez en faisant le lien suivant, par exemple, chez moi sur une Slackware 9, le paquet `kernel-source` a créé le répertoire `/usr/src/linux-2.4.19/`, j'ai donc fait le lien suivant :

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

```
mv linux linux-old
```

```
ln -sf linux-2.4.19 linux
```

---

<sup>1</sup><http://kernel.org/>

<sup>2</sup><http://www.kernel.org/pub/linux/kernel/v2.4/>

### 0.1.3 3. Patcher son noyau :

Cette étape, concerne les personnes qui souhaitent rajouter un patch à leur noyau, si vous n'avez pas de patch à appliquer passer au 3. Sinon, commencez par copier le patch dans le répertoire des sources du noyau :

```
cp /où_est/nouveau.gz /usr/src/linux/
```

Puis appliquez-le ainsi :

```
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.

### 0.1.4 4. 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. Heureusement pour nous, les distributions sont livrées avec le fichier de config du noyau qu'elles utilisent. 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
```

Si vous avez l'expérience des commandes Unix, vous avez naturellement déduit que le fichier de configuration du noyau n'est pas config.h mais .config, 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. A ce stade, si vous décidez de lancer la compilation, vous obtiendrez exactement l'exécutable que vous avez déjà sur votre système, pour rajouter une option à ce fichier, 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]**. Mettons que je souhaite, activer n\_hdlc en module, il me suffit d'aller dans **Character devices** → et de m'assurer d'avoir :

```
<M> HDLC line discipline support
```

Quand vous êtes satisfait, choisissez **<exit>**, à la question, **Do you wish to save your new kernel configuration ?** choisissez **<Yes>**. Voilà, votre noyau est configuré, il ne vous reste plus qu'à le compiler où l'installer.

### 0.1.5 5. 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

### 0.1.6 6. 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.2419  
cp System.map /boot/System.map-2419
```

Le .2419 c'est parce que j'installe un noyau 2.4.19, 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.2419 vmlinuz  
ln -s System.map-2419 System.map
```

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

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

Si votre distribution utilise un disque ram initial (initial ramdisk = `initrd`) pour démarrer, je pense à Mandriva et RedHat, tapez également :

```
mkinitrd -o /boot/initrd.2419 /lib/modules/votre_nouveau_noyau
```

### 0.1.7 7. 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`.

### 0.1.8 7.1 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
```

Le reste doit rester inchangé. Si vous utilisez une distribution avec initrd, votre `/etc/lilo.conf` ressemblera plutôt à ceci :

```
image=/boot/vmlinuz
label=linux
root=/dev/hdc6
initrd=/boot/initrd.img
append="quiet devfs=mount"
vga=788
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
label = mylinux-old
root=/dev/hdc6
initrd=/boot/initrd.img
append="quiet devfs=mount"
vga=788
read-only
# Nouveau noyau
```

```

image=/boot/vmlinuz
label=Linux
root=/dev/hdc6
initrd=/boot/initrd.2419
append="quiet devfs=mount"
vga=788
read-only

```

Le reste doit demeurer inchangé. Quand vous êtes satisfait, validez vos changements en tapant :

```
/sbin/lilo -v
```

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.

### 0.1.9 7.2 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

```

Le reste doit rester inchangé. Si vous utilisez une distribution avec initrd, votre /boot/grub/menu.lst ressemblera plutôt à ceci :

```

title Linux
kernel (hd0,5)/boot/vmlinuz root=/dev/hdc6 quiet devfs=mount
vga=788
initrd (hd0,5)/boot/initrd.img

```

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

```
# Ancien noyau :  
title mylinux-old  
kernel (hd0,5)/boot/vmlinuz.old root=/dev/hdc6 quiet devfs=mount  
vga=788  
initrd (hd0,5)/boot/initrd.img  
# Nouveau noyau  
title Linux  
kernel (hd0,5)/boot/vmlinuz root=/dev/hdc6 quiet devfs=mount  
vga=788  
initrd (hd0,5)/boot/initrd.2419
```

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. Introduction<sup>3</sup> Thème II : Installation d'un nouveau noyau<sup>4</sup>

---

<sup>3</sup><http://www.trustonme.net/didactels/192.html>

<sup>4</sup><http://www.trustonme.net/didactels/194.html>