

Le noyau Linux 2.6

Sources de Linux :

Les sources du noyau Linux (*) sont généralement situés dans le répertoire `/usr/src`. Avant de compiler un nouveau *noyau* (kernel) Linux nous devons nous assurer que nous sommes bien en possession des sources. Dans un premier temps nous vérifierons grâce à la commande `rpm` l'existence des *paquetages* (packages) contenant les sources. Sinon, il faudra se procurer le ou les paquetages nécessaires pour entreprendre une compilation. Pour se faire nous avons besoin des CDROM(s) de la distribution GNU/Linux concernée et le cas échéant nous les téléchargerons par internet sur le site correspondant à la version déjà installée sur la machine. Pour connaître les informations du noyau système installé, il faut utiliser la commande `uname` :

ex : `uname -sr` affiche le nom du système et la version du noyau

(*) Note : les noyaux Linux sont numérotés sur 3 nombres `x.y.z`.

Versions stables : le nombre du milieu (`y`) est pair. Par exemple : `2.4.11` ou `2.6.9`.

Versions de développement : le nombre du milieu (`y`) est impair, par exemple `2.3.35` ou `2.5.40`.

Il n'est pas recommandé à des utilisateurs peu expérimentés d'utiliser un noyau d'une version de développement, car ces noyaux ne sont pas - a priori - stables.

ATTENTION : si vous installez un package RPM : il ne faut jamais mettre à jour un noyau par l'intermédiaire de la commande `rpm -U` (upgrade). Cette commande supprimera votre noyau actuel. Si le nouveau noyau ne fonctionne pas, il ne vous restera plus qu'à réinstaller votre système :-(. C'est à dire qu'il faut installer (et non mettre à jour) le nouveau noyau par l'intermédiaire de la commande `rpm -i kernel-x.y.z.rpm`.

Configuration des modules à compiler :

On a besoin d'un nouveau noyau de système chaque fois que l'on a changé un composant matériel de l'ordinateur fonctionnant sous Linux. Bien que la plupart des matériels soient reconnus automatiquement avec les versions actuelles de GNU/Linux, il sera tout de même nécessaire de compiler un noyau si l'on veut utiliser un utilitaire, un système de fichiers etc ... pouvant satisfaire les performances de certains composants. Pour les installer, il suffira de configurer la liste des options disponibles référencant les modules. Dans toutes ces situations, il faudra réfléchir à la manière de créer un nouveau noyau et de l'utiliser. Toutes les données qui appartiennent au noyau Linux se trouvent dans le répertoire `/usr/src/linux`. On y trouve un certain nombre de sous-répertoires contenant des parties du noyau.

Toutes les parties forment le noyau du système d'exploitation :

arch composant dépendant de l'architecture matérielle

drivers pilotes de périphériques (device drivers subsystem) orientés caractères ou blocs

fs fonctions pour systèmes de fichiers (file subsystem)

ipc communication interprocessus (inter-process communication subsystem)

kernel routines pour l'interprétation des appels système (system call interface subsystem)

mm routines de gestion de la mémoire (memory management subsystem)

Placez-vous dans le répertoire des sources `/usr/src`, puis lancez les commandes suivantes :

```
[root@machine src]# cd linux
```

Pour configurer les modules de votre nouveau noyau vous avez la possibilité entre trois modes dont deux affichages graphiques et un textuel :

- configuration en mode graphique :

```
[root@machine linux]# make xconfig
```

- configuration en mode texte avec menus (ncurses) :

```
[root@machine linux]# make menuconfig
```

- configuration en mode texte pur (longue suite de questions/réponses) :

```
[root@machine linux]# make config
```

Choisir des options :

Lors d'une première expérience de compilation du noyau, passez toutes les options en revue les unes après les autres, dans l'ordre, afin de voir si elles s'appliquent à votre cas ou non. Pour en savoir plus sur chaque option, vous pouvez vous référer à la rubrique d'aide en sélectionnant soit la touche `[H]` dans la console ou le bouton `[Help]` (sous `x`). Vous pouvez sauvegarder votre configuration dans un fichier (`kernel.config-[version]`) afin de la réutiliser plus tard. Ainsi vous pourrez recompiler le noyau en rechargeant cette configuration, et n'ajuster que les paramètres nécessaires. En cas d'évolution de version du noyau, vous pourrez réutiliser ce fichier de configuration qui reste compatible (et vérifier les nouvelles options).

Compilation du noyau (*) et installation des modules :

- sauvegardez et quittez le mode de configuration. Avant de générer un nouveau noyau Linux, il y a trois points à vérifier :

- la place sur le disque dur : il faut un minimum de 25 Mo (voir plus)
- la quantité de mémoire vive : compter 32 Mo minimum (plus elle est large mieux ça compilera)
- le compilateur C et tous les outils nécessaires à la génération du noyau (programme `make` et `awk`)

- lancez la suite de commandes en une fois :

```
[root@machine linux]# make dep clean bzImage modules modules_install
```

compile et installe en une fois.

- par commande seule :

```
[root@machine linux]# make dep vérifie les dépendances
```

```
[root@machine linux]# make clean nettoie les fichiers résiduels d'un compilation antérieure
```

```
[root@machine linux]# make bzImage compile le noyau
```

```
[root@machine linux]# make modules compile et installe les modules (drivers)
```

```
[root@machine linux]# make modules_install installe les modules dans le répertoire suivant : /lib/modules/x.y.z/
```

(*) Note : si vous voulez juste recompiler le noyau, sans monter de version, déplacez le répertoire actuel `/lib/modules/x.y.z` (remplacé par votre votre version) vers `/lib/modules/old-x.y.z`, avant la phase de compilation, afin que l'installation des modules recrée ce répertoire et y place les nouveaux modules : cela vous épargnera des problèmes si la nouvelle configuration du noyau a modifié celle des modules. Si tout fonctionne correctement après le redémarrage, vous pourrez supprimer `/lib/modules/old-x.y.z`.

Configuration des variables du noyau

Configuration et installation du noyau (*)

Installation avec LILO : TP IV : UNIX et GNU/Linux : Exercices

Ensuite, placez vous dans `/usr/src/linux`, et installez le nouveau noyau par la commande suivante :

```
[root@machine linux]# make install
```

 copie le noyau et le fichier `System.map` dans le répertoire `/boot`, puis lance `lilo` afin de prendre en compte les changements que nous avons apporté à `/etc/lilo.conf`. Si tout s'est bien passé, vous n'avez plus qu'à redémarrer votre machine avec la commande `reboot`.

A l'*invite* (prompt) LILO, tapez 'linux' ou sélectionnez le dans le menu pour démarrer le nouveau noyau, et si cela ne fonctionne pas 'old' pour redémarrer l'ancien noyau.

Corrigez éventuellement ce qui ne va pas en récupérant la sauvegarde des informations de configuration du noyau et recommencez la sélection des modules, afin de modifier et de choisir les bonnes options.

Installation avec GRUB :

Vous venez de compiler un nouveau noyau et vous devrez modifier le fichier de configuration permettant de démarrer (boot) le noyau en y insérant "linux" pour le nouveau et "old" pour l'ancien. Le fichier de configuration de GRUB est `/boot/grub.lst` ou `/boot/grub/menu.lst`

Vous devez copier le noyau et `System.map` dans `/boot` (*) : déterminez le nom de l'image noyau (ici `vmlinuz`) dans le fichier de configuration de GRUB. Puis copiez le noyau :

```
ex: mv /boot/vmlinuz-2.6.x /boot/vmlinuz-2.6.x.old
```

```
cp arch/<architecture>/boot/bzImage /boot/vmlinuz-2.6.x <architecture>
```

 étant égal à `i386` pour un PC compatible Intel, `ppc` pour un Macintosh PowerPC, etc...

```
mv /boot/System.map-2.6.x /boot/System.map-2.6.x.old
```

```
cp System.map /boot/System.map-2.6.x
```

 ces instructions sont valables quelque soit le chargeur de démarrage...

Enfin, vous devez faire prendre en compte la nouvelle image noyau à GRUB :

```
ex: grub root (hdC,P) setup (hd0)
```

 en remplaçant C par le numéro du disque dur, selon la correspondance suivante : `hda:C=0`, `hdb:C=1`, `hdc:C=2`, etc... et P par le numéro de partition sur le disque, c'est à dire `hdx1:P=0`, `hdx2:P=1`, `hdx3:P=2`, etc... où x représente le périphérique.

(*) Note : la plupart des distributions récentes utilisent souvent par défaut un disque RAM initial (`initrd`) afin de démarrer le système. Ce n'est pas nécessaire en général, sauf si votre système de fichiers racine (`/`) a besoin de modules pour être chargé (par exemple, s'il se trouve sur un disque SCSI, et que le support SCSI est en modules). Si vous incluez les modules nécessaires au chargement de votre système de fichiers racine (en répondant `Y` au lieu de `M`), vous pouvez vous passer de `initrd`. Si vous utilisez `initrd`, n'oubliez pas de le régénérer avec la commande `mkinitrd` avant de lancer LILO (ou GRUB) ou de redémarrer avec `LOADLIN`.