Compiler un noyau

- Objet : Compiler un noyau
- Niveau requis :
  avisé
- Commentaires : *Compiler un noyau signifie de traduire le code d'un noyau (kernel).*
- Suivi :
  à-tester
  - Création par neo le 17-05-2010
  - Testé par <...> le <...>
- Commentaires sur le forum : Lien vers le forum concernant ce tuto[1)]

Vous pouvez aussi tenter, si cela ne marche pas, ce tutoriel Compiler un kernel sous debian

Introduction

Compiler un noyau, cela signifie de traduire le code d'un noyau (kernel) dans un langage interprétable par le processeur.

Si vous compilez et installez un nouveau noyau pour la branche `Stable`, votre distribution restera `Stable` et une ligne supplémentaire dans le `Grub` concernant le nouveau noyau se mettra en place. Ainsi, vous pourrez démarrer sur ce nouveau noyau, ou bien sur le(s) ancien(s) noyaux à votre guise.

Important !

Comme pour toute modification du noyau, les `modules` qui ont été compilés devront l'être de nouveau pour le nouveau `noyau` !

Par exemple : ceux des `Cartes Graphiques` ou de `Virtualbox`.

Compilation pas à pas

Il existe plusieurs manières de compiler un noyau sous Linux, j'utilise la méthode ci-dessous sous Debian.

Il est inutile de compiler en mode root, comme on le voit assez souvent dans les fils de discussion sur les forums ou dans des tutos, les seules opérations à effectuer en root sont l'installation des utilitaires nécessaires à la compilation, l'installation du paquet linux-source puis à la fin de la compilation, l'installation des paquets .deb créés.

Il n'est pas non plus indispensable de compiler un noyau dans le répertoire /usr/src, il est plutôt conseillé de créer un répertoire dans son répertoire /home.

- http://linuxmafia.com/faq/Kernel/usr-src-linux-symlink.html

Ceci dit allons-y.

Installer le nécessaire à la compilation du noyau.

```
apt-get update
```

puis

```
apt-get install build-essential fakeroot kernel-package libncurses5-dev
```

Il faut maintenant installer le paquet du noyau que l'on veut compiler, si vous avez plusieurs dépôts, utiliser apt-cache search linux-source, pour voir les différentes version disponible, puis installer.

L'archive linux-source s'installe par défaut dans /usr/src.

Compilation - Préparation

A partir de maintenant tout se fait en simple utilisateur.

Le nécessaire est installé, on va maintenant créer un répertoire nommé compil dans /home et s'y placer pour décompresser l'archive du noyau.

```
mkdir compil && cd compil
```

Puis, une fois dans le répertoire compil, décompressez l'archive se trouvant dans /usr/src

```
tar xvf /usr/src/linux-source-2.6.24-rc5.tar.bz2
```

Puis, rendez-vous dans le dossier décompressé

```
cd linux-source-2.6.24-rc5
```

Nous allons maintenant copier le fichier .config de votre noyau actuel se trouvant dans le répertoire /boot

```
cp /boot/config-$(uname -r) .config
```

Saisissez la commande:

```
make oldconfig
```

Cette commande permet d'adapter une configuration d'une précédente version du noyau à une nouvelle en vous posant des questions sur les nouvelles fonctionnalités à configurer au besoin.

Répondez aux questions, les lettres en majuscules sont les réponses conseillées, si vous hésitez, répondez par y pour yes vous pourrez toujours revenir à ces options.

Configuration du noyau

Une fois terminé, vous pouvez lancer l'utilitaire de configuration du noyau :

```
make menuconfig
```

Vous pouvez faire le tour des options à configurer, vous pouvez vous aidez de ce site

- Options du noyau

Pour personnaliser votre noyau. L'activation ou désactivation d'options se fait via la touche Espace de votre clavier et le déplacement avec les flèches de direction.

1. si < > –> l'option sera désactivée.
2. si <*> –> l'option sera compilée en dur.
3. si <M> –> l'option sera configurée en module.

Si vous connaissez bien votre matériel, mettez en dur les options le concernant, il suffit ensuite de sélectionner :

```
 Save an Alternate Configuration File
 Validez --> Validez ''OK'' --> Validez ''exit''
```

Créations des paquets kernel_image et kernel_headers

Pour rappel, `kernel_headers` (ou, une fois installé, `linux-headers-2.6.24`) permet de construire des modules pour le pilote graphique ou autres pilotes via module-assistant.

```
fakeroot make-kpkg --initrd --append-to-version=-perso kernel_image
kernel_headers
```

Vous pouvez changer `-perso` par autre chose, cette option se rajoute à la fin du nom du noyau –>

```
 linux-image-2.6.24-rc5-perso
```

Compilation - Détail

Gain de compilation

Si vous utilisez un puissant processeur `double coeur` ou plus, on peut lui demander de lancer plusieurs processus ainsi :

Editer le fichier `/etc/kernel-pkg.conf` :

```
nano /etc/kernel-pkg.conf
```

Et ajouter à la fin :

```
ROOT_CMD=fakeroot
CONCURRENCY_LEVEL=2
```

- Pour `CONCURRENCY_LEVEL=2` en fait on propose le nombre de “coeur” du processeur.

Exemple pour un processeur à 3 coeur, ce sera :

```
CONCURRENCY_LEVEL=3
```

- Pour `ROOT_CMD=fakeroot`, cela permet de ne plus ajouter la commande `fakeroot`, elle sera automatiquement utilisée par la commande.

Compilation

Il est temps de lancer la compilation. Si vous n'avez pas ajouter `ROOT_CMD=fakeroot`, saisissez la commande :

```
fakeroot make-kpkg --us --uc --initrd --append-to-version=-perso
buildpackage
```

Si vous avez ajoutez ROOT_CMD=fakeroot, saisissez la commande :

```
make-kpkg --us --uc --initrd --append-to-version=-perso buildpackage
```

Et validez.

Suivant les capacités de votre Pc, cela peut-être long, alors un p'tit café s'impose.

Installation des paquets .deb

Lorsque la compilation est terminée, vous pouvez installer les paquets `.deb` créés dans le répertoire `~/compil`

Pour revenir dans le répertoire `compil`, saisissez :

```
cd ..
```

Vous y trouverez les paquets “.deb” créés. Vous pouvez maintenant passer en `root` pour les installer avec la commande `dpkg -i` .

Nettoyage avant une seconde compilation

Si vous recompilez une deuxième fois le noyau, pour nettoyer le répertoire et effacer les fichiers créés par la précédente compilation, pensez à exécuter cette commande (avec ou sans `fakeroot` selon l'astuce précédente) :

```
fakeroot make-kpkg clean
```

N'oubliez pas le `man` en ligne de commandes pour voir les différentes options :

```
man make-kpkg
```

Option --initrd

L'option `–initrd` peut être enlevée, je ne l'utilise plus, mais je vous le déconseille fortement, car :

cela implique de compiler en dur tout ce qui concerne votre matèriel

comme le support `IDE/SATA/SCSI` qui sont par défaut compilés en modules, ou encore les systèmes de fichiers que l'on souhaite utiliser, à moins de bien connaître son matériel, laisser cette option, sinon au démarrage de votre nouveau noyau un `kernel panic` risque fortement de faire apparition.

Il y aussi moyen de faire la totale, c'est à dire de créer :

```
kernel_image, kernel_headers, kernel_docs, kernel_source.......
```

Ce n'est pas vraiment utile.

Remerciements :

Tuto tout de `neos`*, alias* `n3os` !

Merci à :

`engil` *pour sa relecture et ses suggestions/rectifications généreuses, ainsi que pour le procédé de configuration afin d'accélérer la compilation finale !*

`juddix` *pour ses suggestions avisées,*

`captnfab` *pour son aide toujours claire !*

[1)]
N'hésitez pas à y faire part de vos remarques, succès, améliorations ou échecs !

From:
http://debian-facile.org/ - **Documentation - Wiki**

Permanent link:
http://debian-facile.org/doc:systeme:noyau:compiler

Last update: **18/05/2017 21:10**