Les étapes pour compiler Linux

1. Télécharger le code source du noyau

Pour pouvoir compiler un nouveau noyau, il faut en télécharger le code source sur le site [www.kernel.org](http://www.kernel.org/). Toutes les versions y sont disponibles. Prenons un exemple : nous voulions compiler la version 4.4.1 du noyau Linux. Nous devons télécharger le code source depuis cette addresse

*https://www.kernel.org/pub/linux/kernel/v4.x/linux-4.4.1.tar.xz*

Il vaut mieux télécharger la version compressée par bzip. Voici à quoi ressemblera un wget sur la ligne de commande :

*wget https://www.kernel.org/pub/linux/kernel/v4.x/linux-4.4.1.tar.xz*

Une fois le source de la version de noyau requise téléchargé, il faut le décompresser avec bunzip et le désarchiver avec untar de la façon suivante :

*tar xvjf linux-2.6.9.tar.bz2*

L'option x désigne l'e'x'traction de l'archive , v signifie « verbeux », j spécifie que nous employons la commande bunzip avant de désarchiver et f indique le nom du fichier en entrée.

Le fichier sera désarchivé dans le répertoire linux-4.4.1. Dès que vous avez fini de désarchiver, déplacez-vous avec cd dans linux-4.4.1.

Nous pouvon aussi télécharger le code source de torvalds avec la commande git clone

*git clone git://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git*

1. Configurer le noyau

Soyez sûr d'être dans le repertoire des sources du noyau.

1. Nettoyage

Cette étape peut s'appliquer dans le cas où vous avez déjà des sources compilées dans le dossier /usr/src/linux/. Elle s'utilise principalement lorsque vous avez modifier plusieurs fois votre configuration et que vous avez passé certains modules dans le kernel.

En effet, il peut être parfois utile de supprimer les fichiers compilés de l'arborescence des sources avant de commencer la procédure de compilation.

Il existe plusieurs commande pour effectuer cette étape de “nettoyage” :

*#* *make clean* ou *# make mrproper*

*make clean* supprime tous les fichiers compilés.

*make mrproper* supprime tous les fichiers compilés ainsi que le fichier de configuration .config qui contient les options de configuration du kernel.

1. Récupération de la configuration

Cette étape optionnelle permet de récuperer les paramètres déjà définis dans le noyau courant, ce qui est intéressant si vous avez déjà personnalisé votre noyau courant et que vous désirez l'utiliser comme base pour le nouveau noyau. Pour effectuer ceci, récutez les commandes suivantes :

*# zcat /proc/config.gz > .config*

*# make oldconfig.*

1. Configuration

Démarrez ensuite la configuration :

*# make menuconfig*

Notez que vous pouvez aussi utiliser les commandes suivantes :

*make config* propose une longue suite de questions-réponses (y/n/m).

*make xconfig* propose une configuration en mode graphique (Qt).

*make gconfig* propose une configuration en mode graphique (GTK).

Toutes les options de configuration du noyau ne peuvent malheureusement pas être décrites ici en détail. Utilisez les nombreux textes d'aide intégrés pour la configuration. La documentation mise à jour du noyau est également placée dans /usr/src/linux/Documentation.

1. Créer l'image finale

Il est possible de construire divers types d'images binaires de noyau. Nous pouvons compiler une image de noyau complète ou une version compressée de celle-ci; habituellement, on choisit la version compressée ou l'image bzImage, que l'on peut créer en saisissant:

*make bzImage*

1. Compiler et installer les modules

Si, dans la section configuration, nous avons sélectionné des composants devant être intégrés en tant que modules du noyau, il faut maintenant les compiler. Pour compiler les modules, exécutons la commande :

*make modules*

Cette commande compilera les composants (qui sont sélectionnés pour la compilation des modules) sous forme de modules.

Après la compilation des modules, il est à présent temps de les installer. pour ce faire, exécutez la commande :

*make modules\_install*

en tant que super-utilisateur (root). Cette commande permet d'installer les modules et les autres fichiers nécessaires dans le répertoire /lib/modules/4.4.1.

1. Amorcer à partir du nouveau noyau

L'installation des modules terminée, nous pouvons entreprendre une procédure d'installation automatique du noyau pour l'exécutable du noyau. Saisissons simplement :

*make install*

Cette commande permet de mettre à jour l'image du noyau dans le répertoire /boot, d'actualiser le fichier de configuration du chargeur d'amorçage, puis d'effectuer les actions nécessaires pour rendre le nouveau noyau amorçable.

1. Réinitialisation du secteur de démarrage

Il faut maintenant recréer le ram disk initial pour le nouveau noyau, sinon le système ne sera certainement pas capable de redémarrer le nouveau noyau.

*# mkinitrd ou #mkinitramfs*

*mkinitrd* (basée sur Redhat) et *mkinitramfs* (basée sur Debian) créera un initrd pour le processus de démarrage, et ils utiliseront le «vieux» ou «nouvelle» méthode en fonction de la version.

Notons que les 2 commandes recréer un nouveau ram disk pour chacun des noyaux en présence détectés dans /boot/.

1. Redémarrage

Enfin, redémarrez.

*# reboot*