

Master Ingénierie du Logiciel Libre – 2^{ème} Année (M2I2L)

Module système, réseaux et sécurité

Année 2009-2010

TP N°2 : [Re-]Compilation d'un noyau de Linux Debian

D. DUVIVIER (duvivier@lil.univ-littoral.fr) – LIL/ULCO

Ce document est une liste de commandes, extraites des transparents « de cours ». Je suppose pour la suite que vous êtes logués sous « root ». Toutes les recommandations/vérifications reprises sur les transparents de cours ne figurent pas dans ce document. Avant de commencer, plusieurs vérifications doivent être effectuées :

- `uname -r` donne la version du noyau actuel (par exemple 2.6.30-1-686)
- `uname -v` donne la version/révision du noyau actuel (notamment la date de création)

Vérifiez si les sources du noyau ne sont pas déjà dans un sous-répertoire de `/usr/src` ainsi que la place disponible (`df -h /usr/src`) car il faut prévoir 47Mo à plus de 720Mo !!!

Faites le ménage dans les paquets via `aptitude clean`

Vérifiez que les outils de base nécessaires sont disponibles :

- `dpkg -l binutils` doit indiquer la version installée (...) sinon téléchargez le via `apt-get install binutils`
- `dpkg -l bzip2` doit indiquer la version installée (...) sinon téléchargez le via `apt-get install bzip2`

Si c'est la première compilation d'un noyau sur votre machine, ne lisez pas la suite de ce paragraphe... En revanche, si un noyau a déjà été recompilé à partir de ses sources sur votre machine, alors avant d'effectuer une nouvelle compilation du noyau, assurez-vous que le lien vers `/usr/src/linux` existe, que vous vous trouvez dans le répertoire `/usr/src/linux`, que vous avez effectué une sauvegarde de votre fichier de configuration du noyau (`/usr/src/linux/.config`) et que l'arborescence a été nettoyée (via `make mrproper`).

Détermination de la configuration de la machine cible...

Avant de vous lancer dans la compilation du noyau, il est préférable de bien connaître les constituants de votre machine préférée si vous voulez un noyau performant, en adéquation avec votre machine... C'est maintenant qu'il faut sortir toutes les documentations (techniques si possible) de votre matériel si vous voulez vraiment « customiser » votre noyau. Attention aux surprises parfois ! C'est comme dans d'autres domaines, le « vrai » tuning est un art ☺ !

De nombreuses commandes/outils peuvent vous aider pour configurer correctement votre système, parmi lesquelles :

<code>dmesg less</code>	<code>lsmod</code>
<code>cat /proc/meminfo</code>	<code>free -mt</code>
<code>cat /proc/cpuinfo</code>	<code>lspci</code>
<code>cat /proc/ioports</code>	<code>lspci -v</code>
<code>cat /proc/iomem</code>	<code>lspci -vv</code>
<code>cat /proc/dma</code>	<code>lsusb</code>
<code>cat /proc/filesystems</code>	<code>lsusb -v</code>
<code>ls -r /proc/irq/*</code>	<code>lsscsi</code>
<code>cat /sys/devices/pnp0/00:00/resources</code>	➔ fouillez dans <code>/proc</code> et <code>/sys</code>
<code>fdisk -l</code>	<code>dmidecode</code>

Il existe de nombreux paquets dédiés à la détection des cartes réseau (+/-wifi) des périphériques usb, des périphériques scsi, des cartes vidéo... Juste un exemple au travers du paquet « hal » qui provoque l'installation de plusieurs paquets dont il dépend :

Tapez « `aptitude install hal` »

Ce paquet provoque l'installation de divers outils (gestion de l'énergie « `apm` »...) et installe un daemon « `hald` ». Il fournit notamment la commande :

`lshal` ➔ avec ça vous avez toutes les infos ... en vrac !

Téléchargement et préparation des sources pour la recompilation du noyau

- 0) apt-cache search ^linux-source donne la liste des versions disponibles
- 1) apt-get install linux-source-2.6.30 transfère les sources dans le fichier /usr/src/linux-source-2.6.30.tar.bz2
- 2) cd /usr/src **NE QUITTEZ PAS CE REPERTOIRE POUR LES COMMANDES SUIVANTES**
- 3) tar jxf linux-source-2.6.30.tar.bz2 dépaquete les sources
- 4) chmod 700 linux-source-2.6.30 protège le répertoire des regards indiscrets ☺
- 5) apt-get remove linux-source-2.6.30 puis apt-get clean fait le ménage proprement (en utilisant le système de paquets)
- 6) ln -s /usr/src/linux-source-2.6.30 linux crée un lien « standard » sur les sources
- 7) apt-get install gcc make binutils util-linux module-init-tools oprofile oprofile-gui e2fsprogs procs udev libcurses5-dev kernel-package initramfs-tools cramfsprogs debconf-utils build-essential transfère des paquets utiles pour la suite.
- 8) apt-get clean pour faire un peu de ménage avant de poursuivre...
- 9) cp -i /boot/config-2.6.30-1-686 /usr/src/linux/.config copie la configuration courante comme base de la nouvelle
- 10) cd /usr/src/linux **NE QUITTEZ PAS CE REPERTOIRE POUR LES COMMANDES SUIVANTES**
- 11) make menuconfig configure le noyau via des menus textuels, utilisez la touche « Entrée » pour les sous-menus ...
- 12) Sauvegardez le fichier /usr/src/linux/.config que vous venez de générer en utilisant une commande du genre :
cp /usr/src/linux/.config /boot/config-2.6.30-i2ldebian1

Quelques options à connaître lors de la compilation du noyau :

- General setup
 - Kernel log buffer size → Réduisez la taille du fichier log du noyau sur les machines possédant peu de RAM
 - Cpuset support → Inutile si un seul cœur/processeur → [*] → []
 - Initial RAM filesystem (...) → Surtout ne pas désactiver si votre config. utilise initrd sinon plantage ☹ !
 - Optimize for size → Si c'est actif [*] on réduit la taille du noyau en mémoire, sinon [] il va plus vite !
 - Profiling support → Utile [*] tant que vous faites du tuning, à supprimer [] ensuite !
 - Oprofile → Sous forme de <M>odule tant que vous faites du tuning, à supprimer <> ensuite !
- Enable loadable module support → Config classique → Ne modifiez rien là, sinon plus de modules → Plantage !
- Enable the block layer
 - Support for Large Block Devices [*] → [] sauf si vous avez au moins 2To de disque(s) ☺ !
 - Support for tracing block io actions [*] → [] sauf pour faire du tuning sur les disques !
 - Support for Large Single Files [*] → [] sauf si vous avez des fichiers de plus de 2To !
 - ...
- Processor type and features
 - Symmetric multi-processing support [*] → [] si vous avez un seul processeur/cœur
 - Subarchitecture Type (PC-compatible) → Ne pas modifier en principe !
 - Paravirtualized guest support → laissez actif [*] si vous voulez utiliser un hyperviseur
 - Processor family → Choisissez en fonction de VOTRE processeur !!!
 - Generic x86 support → Pas utile [] sauf si le même noyau est destiné à plusieurs processeurs x86
 - Preemption Model → Là c'est vraiment du tuning, ça dépend de ce que vous faites sur la machine !
 - High Memory Support → ça dépend de votre machine, sélectionnez « (X) off » si pas utile !
 - Timer frequency → Privilégier réactivité ou puissance de calcul, là est la question ...
- Bus options → Choisissez ce qui est nécessaire pour VOTRE machine
- File systems → Choisissez vos systèmes de fichiers préférés (ça prend +/- du temps et de la place sur disque)
ATTENTION : le système de fichiers racine ne doit pas être un module <M> mais inclus dans la code (<*>) du noyau
→ il s'agit de ext2fs ou ext3fs selon la configuration (ou les deux).
- Kernel hacking → Sélectionnez certaines options via [*] pour faire du tuning, sinon désélectionnez []
 - Magic SysRq key → **FAITES TRES ATTENTION A CETTE OPTION !!!**
SUR UNE MACHINE DE PRODUCTION ET/OU ACCESSIBLE LIBREMENT CETTE OPTION DOIT IMPERATIVEMENT ETRE DESACTIVEE !!!
- Virtualization → Exclusivement utile sur les processeurs qui le permettent !!!

SURTOUT N'OUBLIEZ PAS DE SAUVEGARDER VOTRE CONFIGURATION AVANT DE QUITTER !!!

Mise en garde avant la recompilation du noyau...

Les sources du noyau doivent être placées dans un répertoire caché aux yeux des utilisateurs. Ce répertoire, votre répertoire d'accueil (/home/<nom_de_login>) et le répertoire d'accueil de root (/root) doivent être a priori uniquement visibles et accessibles par leur propriétaire (drwx-----), ceci est obtenu via la commande suivante :

```
chmod 700 <nom_du_répertoire>
```

La recompilation du noyau peut prendre énormément de temps et d'espace disque... Sur les ordinateurs possédant peu de mémoire ou peu d'espace disque ou tout simplement si vous voulez recompiler rapidement, ne sélectionnez que le minimum nécessaire et suffisant dans les systèmes de fichiers et les drivers à recompiler !!!

Avant de lire la suite de ce paragraphe, je précise que je vous déconseille vivement d'utiliser la commande présentée ci-dessous sur les machines en salle de TP car elles ont suffisamment de puissance de calcul et de mémoire pour permettre la compilation du noyau tout en restant en mode multi-utilisateurs/graphique ! Sur une machine en réseau ou sur une machine possédant peu de mémoire en revanche, il peut être souhaitable de passer en mode « single user ». Attention dans ce cas, un seul utilisateur peut se connecter sur une seule console locale (par défaut), il n'y a plus de réseau, plus d'interface graphique ... La commande permettant de passer en mode single est la suivante, ne l'invoquez pas dans une console graphique/X sinon vous allez vous faire « éjecter » : telinit 1

Cette commande force le passage du processus « init » (processus ancêtre de tous les processus du système) en mode/niveau « 1 » en stoppant/tuant tous les processus (daemons) n'appartenant pas à ce niveau.

Sauvegardez votre fichier /usr/src/linux/.config ailleurs en utilisant un nom qui vous permette de retrouver la date, la machine sur laquelle le noyau a été compilé et éventuellement d'autres informations...

Par exemple :

```
cp /usr/src/linux/.config /root/config_i2ldebian_`hostname`_`date +%Y%m%d%H%M%S`_`
```

ATTENTION : dans la commande ci-dessus, ne confondez pas quote « ' » et backquote « ` » !!!

Recompilation du noyau selon la « méthode Debian » (i.e. via un paquet .deb)

13) make-kpkg --initrd --revision=1 --append-to-version=i2ldebian kernel-image compile le noyau et le place dans un paquet debian.

Remarque : par sécurité, si vous tapez uniquement make-kpkg --initrd kernel-image, une chaîne de caractères sera automatiquement ajoutée au nom du noyau pour indiquer qu'il s'agit d'un noyau customisé/personnel et non d'un noyau officiel. Ceci évite de remplacer accidentellement un noyau officiel par votre noyau « expérimental ».

14) cd /usr/src puis dpkg --install linux-image-2.6.30-i2ldebian_1_i386.deb installe le noyau et met à jour Grub

15) cd /usr/src/linux puis make-kpkg clean fait le ménage dans les sources

16) Rebootez la machine, testez le nouveau noyau en vérifiant la version grâce à la commande uname -a.

17) Rebootez la machine et relancez l'ancien noyau (vérifiez avec uname -a et uname -v)

ATTENTION : pour la suite vous devez impérativement avoir démarré linux à partir de l'ancien noyau

18) apt-get remove --purge linux-image-2.6.30-i2ldebian désinstalle le nouveau noyau

19) `rm /usr/src/linux-image-2.6.30-i2ldebian_1_i386.deb` supprime le paquet contenant le noyau

Recompilation du noyau « à la main » (*i.e.* via la commande make)

Pour les commandes suivantes, j'ai supposé que <tuning/custom_string> était initialisé à « -i2lmain »

- 20) cd /usr/src/linux puis make mrproper fait le ménage dans les sources (y-compris le fichier .config !)
- 21) Recréez un fichier de configuration /usr/src/linux/.config à partir d'une sauvegarde antérieure ou en récupérant une configuration de base dans /boot et en la modifiant via make menuconfig
- 22) Editez le fichier Makefile pour remplacer « EXTRAVERSION= » par « EXTRAVERSION=-i2lmain »
- 23) make lance la compilation
- 24) make modules_install installe les modules dans /lib/modules/2.6.30-i2lmain
- 25) Copie des fichiers générés par la compilation du noyau dans /boot (vérifiez avant si les noms de fichiers n'existent pas déjà) :
cd /usr/src/linux
cp -i arch/x86/boot/bzImage /boot/vmlinuz-2.6.30-i2lmain
cp -i System.map /boot/System.map-2.6.30-i2lmain
cp -i .config /boot/config-2.6.30-i2lmain
- 26) Création de l'image RamDisk (si initrd est nécessaire) :
cd /usr/src/linux
mkinitramfs -o /boot/initrd.img-2.6.30-i2lmain /lib/modules/2.6.30-i2lmain
- 27) Création des liens dans / utile uniquement pour lilo a priori (vérifiez avant les liens qui existent déjà...) :
cd /
mv vmlinuz vmlinuz.old
ln -si boot/vmlinuz-2.6.30-i2lmain vmlinuz
mv initrd.img initrd.img.old
ln -si boot/initrd.img-2.6.30-i2lmain initrd.img
mv System.map System.map.old
ln -si boot/System.map-2.6.30-i2lmain System.map
- 28) cd /boot/grub puis cp menu.lst menu.lst.orig crée une copie de sécurité de la configuration de Grub
- 29) Editez menu.lst et ajoutez une entrée pour le nouveau noyau (*cf.* « cours » pour explications détaillées ou basez-vous sur le détail donné lors de l'explication du fichier menu.lst automatiquement généré lors de l'utilisation de la recompilation du noyau via un paquet Debian)
- 30) update-grub met à jour Grub et crée automatiquement un mode « single-user » pour votre version de noyau
- 31) Rebootez la machine, testez le nouveau noyau en vérifiant la version grâce à la commande uname -a.

ATTENTION : NE FAITES PAS LE MENAGE DANS /usr/src/linux
ET NE REBOOTEZ PAS SUR L'ANCIENNE VERSION DU NOYAU
POUR EFFECTUER LA SUITE DES TP SRS/KERNEL !!!

Voilà, c'est terminé !