

Configuration des fonctions réseau & compilation du noyau Linux

Philippe Latu
philippe.latu(at)inetdoc.net

<http://www.inetdoc.net>

Dans ce support de travaux pratiques, on se propose de préparer un système GNU/Linux pour être utilisé comme équipement d'interconnexion réseau. Après avoir passé en revue les fonctions réseau utiles du noyau Linux et sélectionné les pilotes des périphériques effectivement présents sur la plateforme matérielle, on construit un paquet de noyau Linux à partir de ses sources.

Table des matières

1. Copyright et Licence	1
1.1. Méta-information	1
1.2. Conventions typographiques	2
2. Le noyau courant et son arborescence	2
3. Les sources du noyau Linux	4
4. La configuration du noyau Linux	6
5. La compilation & l'installation du nouveau noyau Linux	11
6. Documents de référence	12

1. Copyright et Licence

Copyright (c) 2000,2012 Philippe Latu.
Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Copyright (c) 2000,2012 Philippe Latu.
Permission est accordée de copier, distribuer et/ou modifier ce document selon les termes de la Licence de Documentation Libre GNU (GNU Free Documentation License), version 1.3 ou toute version ultérieure publiée par la Free Software Foundation ; sans Sections Invariables ; sans Texte de Première de Couverture, et sans Texte de Quatrième de Couverture. Une copie de la présente Licence est incluse dans la section intitulée « Licence de Documentation Libre GNU ».

1.1. Méta-information

Ce document est écrit avec *DocBook*¹ XML sur un système *Debian GNU/Linux*². Il est disponible en version imprimable au format PDF : [interco.kernel.qa.pdf](http://www.inetdoc.net/pdf/interco.kernel.qa.pdf)³.

Toutes les commandes utilisées dans ce document ne sont pas spécifiques à une version particulière des systèmes UNIX ou GNU/Linux. C'est la distribution *Debian GNU/Linux* qui est utilisée pour les tests présentés. Voici une liste des paquets contenant les commandes :

- coreutils - The GNU core utilities
- procps - The /proc file system utilities
- pciutils - Linux PCI Utilities
- module-init-tools - tools for managing Linux kernel modules
- make - The GNU version of the "make" utility
- libncurses5 - shared libraries for terminal handling
- libncurses5-dev - developer's libraries and docs for ncurses
- kernel-package - A utility for building Linux kernel related Debian packages
- fakeroot - Gives a fake root environment

¹ <http://www.docbook.org>

² <http://www.debian.org>

³ <http://www.inetdoc.net/pdf/interco.kernel.qa.pdf>

1.2. Conventions typographiques

Tous les exemples d'exécution des commandes sont précédés d'une invite utilisateur ou *prompt* spécifique au niveau des droits utilisateurs nécessaires sur le système.

- Toute commande précédée de l'invite \$ ne nécessite aucun privilège particulier et peut être utilisée au niveau utilisateur simple.
- Toute commande précédée de l'invite # nécessite les privilèges du super-utilisateur.

2. Le noyau courant et son arborescence

Avant d'attaquer la compilation d'un nouveau noyau à partir de ses sources, on doit identifier et localiser les différents composants du noyau en cours d'exécution sur le système.

Le jeu de questions ci-dessous suppose que la configuration système est directement issue de l'installation de la distribution *Debian GNU/Linux*. Le noyau courant exécuté est fourni via un paquet de la distribution.

1. Quelle est la commande UNIX usuelle qui identifie le noyau et sa version ?

Effectuer une recherche dans les pages de manuels des commandes installées sur le système avec une requête du type : **apropos informations, système**.

C'est la commande **uname** qui identifie le noyau courant. Pour interroger les pages de manuels à l'aide de la commande **apropos**, il faut que les paquets correspondant soient installés et que l'index de recherche soit construit.

Pour interroger les pages de manuels, on contrôle la liste des paquets correspondants installés et on lance manuellement la construction de l'index de recherche :

```
$ aptitude search ~imanpages
i  manpages          - Manual pages about using a GNU/Linux system
i  manpages-fr       - French version of the manual pages about using GNU/Linux
i  manpages-fr-extra - French version of the manual pages

<snip/>
# /etc/cron.daily/man-db

<snip/>
$ apropos informations, système | grep uname
uname (1)          - Afficher des informations sur le système
```

Pour obtenir la version courante du noyau exécuté :

```
$ uname -a
Linux vm0 3.0.0-1-amd64 #1 SMP Sat Aug 27 16:21:11 UTC 2011 x86_64 GNU/Linux
```

2. Où est placée l'image de la partie monolithique du noyau courant ?

Repérer le paquet *Debian* correspondant au noyau et retrouver l'image dans la liste des fichiers de ce paquet.

Une fois la version courante du noyau identifiée à l'aide de la commande **uname**, on peut faire la correspondance avec les paquets de noyau installés.

```
$ aptitude search ~ilinux-image
i  linux-image-2.6-amd64 - Linux for 64-bit PCs (dummy package)
i A linux-image-3.0.0-1-amd64 - Linux 3.0.0 for 64-bit PCs
i A linux-image-amd64    - Linux for 64-bit PCs (meta-package)
```

Connaissant le nom du paquet de noyau installé on peut lister les fichiers qu'il contient. À partir de cette liste on peut localiser la partie monolithique du noyau ainsi que ses modules dans l'arborescence du système de fichiers.

C'est dans le répertoire */boot* que sont placées les images des noyaux disponibles sur un système GNU/Linux.

```
$ ls -Al /boot/
config-3.0.0-1-amd64 ❶
grub
initrd.img-3.0.0-1-amd64 ❷
lost+found
System.map-3.0.0-1-amd64 ❸
vmlinuz-3.0.0-1-amd64 ❹
```

- ❶ Fichier de configuration du noyau de la distribution. Il contient l'ensemble des options qui ont été sélectionnées par le responsable du paquet. C'est une configuration très complète dans la mesure où un noyau publié dans une distribution doit supporter le maximum de matériel.
- ❷ Image compressée du disque RAM d'initialisation contenant une arborescence racine simplifiée, des outils et l'ensemble des modules du noyau. Cette technique d'initialisation est la seule qui puisse fonctionner sur des systèmes sans disque dur où sur lesquels aucun système GNU/Linux n'a encore été installé.

- ③ Fichier de cartographie des appels de fonctions du noyau. Cette cartographie est une aide à la mise au point pour les développeurs. On y trouve une identification nominative des fonctions en cas de problème au lieu d'adresses numériques en hexadécimal.
- ④ Fichier image de la partie monolithique du noyau. C'est ce fichier qui est utilisé par le gestionnaire de démarrage pour lancer le système d'exploitation. Le gestionnaire de démarrage y accède directement à l'aide d'un appel BIOS.

3. Où sont placés les fichiers des modules correspondant au noyau courant ?

Comme dans le cas précédent, la liste des fichiers du paquet permet de retrouver l'arborescence de stockage des modules.

On peut parcourir la liste des fichiers contenus dans le paquet de noyau et effectuer des recherches par mots clés en utilisant la commande suivante :

```
$ dpkg -L linux-image-3.0.0-1-amd64 | egrep -e 'kernel$'  
/lib/modules/3.0.0-1-amd64/kernel  
/lib/modules/3.0.0-1-amd64/kernel/arch/x86/kernel
```

La liste ci-dessus montre que les modules du noyau sont placés dans le répertoire `/lib/modules/3.0.0-1-amd64/kernel/`.

4. Dans quels cas de figure utilise-t-on l'arborescence ou le disque RAM ?

Il faut bien différencier l'utilisation du disque RAM `initrd-*` de l'arborescence installée sur le disque du système.

Le fichier image du disque RAM d'initialisation a déjà été identifié ci-dessus.

Ce fichier est utilisé lors du lancement du système d'exploitation. Il est reconnu par le gestionnaire de démarrage de la même façon que la partie monolithique du noyau. Une fois le système complètement initialisé, les opérations de (chargement|déchargement) des modules utilisent l'arborescence du disque dur : `/lib/modules/uname -r/`.

5. Que contiennent les arborescences `/proc` et `/sys` ?

Consulter les documents ressource [sysfs](#)⁴ et [Linux Filesystem Hierarchy](#)⁵

L'arborescence `/sys` est une représentation visible de l'arbre des périphériques physiques vus par le noyau. Cette arborescence a été introduite avec les noyaux de la série 2.6.xx. Elle est construite dynamiquement en fonction des branchements «à chaud» effectués sur les différents bus de la machine. Les informations répertoriées dans cette arborescence sont du type : nom de périphérique, canal DMA, vecteur d'interruption, tensions d'alimentation, etc.

L'arborescence `/proc` comprend l'ensemble des paramètres du noyau en cours d'exécution. Ces paramètres sont modifiables en cours de fonctionnement. L'exemple emblématique, vis-à-vis de ces travaux pratiques est donné par l'ensemble des «réglages» possibles sur les machines d'états de la pile des protocoles réseau. La commande `ls /proc/sys/net/ipv4/` en donne un aperçu.

6. Quelle est la commande qui permet de lister les modules chargés en mémoire ? À quel paquet appartient elle ?

Rechercher dans la base de données des paquets de la distribution les informations relatives aux manipulations sur les modules à l'aide d'une interrogation du type : `aptitude search ~imodule`.

La commande **lsmod** :

```
$ lsmod  
Module                Size  Used by  
ext2                   63732  1  
loop                   22711  0  
joydev                 17262  0  
snd_pcm                68104  0  
evdev                  17558  2  
snd_timer              22581  1 snd_pcm  
snd                    52823  2 snd_pcm,snd_timer  
soundcore              13152  1 snd  
<snip/>
```

Cette commande appartient au paquet `module-init-tools`. En listant le contenu de ce paquet on obtient les noms des commandes associées et les pages de manuels correspondantes.

```
$ dpkg -S `which lsmod`  
module-init-tools: /sbin/lsmod  
  
$ dpkg -L module-init-tools | grep sbin/  
/sbin/depmod  
/sbin/modinfo  
/sbin/modprobe
```

⁴ <http://en.wikipedia.org/wiki/Sysfs>
⁵ <http://tldp.org/LDP/Linux-Filesystem-Hierarchy/html/>

```
/sbin/insmod  
/sbin/rmmod  
/sbin/lsmmod
```

7. Quelles sont les commandes qui permettent de charger un module en mémoire «manuellement» ? Identifier celle qui traite automatiquement les dépendances entre modules.

Rechercher les informations dans la liste des fichiers du paquet ainsi que dans les pages de manuels des commandes.

On dispose de deux commandes : **insmod** et **modprobe**. Seule la commande **modprobe** traite les dépendances au (chargement|déchargement) d'un module. Illustration avec un pilote d'interface RNIS :

```
# modprobe -v hfcpci  
insmod /lib/modules/3.0.0-1-amd64/kernel/drivers/isdn/mISDN/mISDN_core.ko  
insmod /lib/modules/3.0.0-1-amd64/kernel/drivers/isdn/hardware/mISDN/hfcpci.ko
```

8. Quelles sont les commandes qui permettent de retirer un module de la mémoire «manuellement» ? Identifier les options de la commande qui traite automatiquement les dépendances entre modules.

Rechercher les informations dans les pages de manuels des commandes.

Comme dans le cas précédent, c'est la commande **modprobe** qui retire de la mémoire les modules associés au déchargement. Toujours avec le pilote d'interface RNIS :

```
# modprobe -rv hfcpci  
rmmod /lib/modules/3.0.0-1-amd64/kernel/drivers/isdn/hardware/mISDN/hfcpci.ko  
rmmod /lib/modules/3.0.0-1-amd64/kernel/drivers/isdn/mISDN/mISDN_core.ko
```

3. Les sources du noyau Linux

Dans cette partie, on s'appuie pas sur le gestionnaire de paquets de la distribution et on télécharge directement les sources du noyau Linux à partir du dépôt défini dans la liste des sources (fichier `/etc/apt/sources.list`).

Il faut bien reconnaître que s'attaquer à toutes les options de configuration du noyau Linux en partant de zéro est une tâche particulièrement ardue. Pour rendre la démarche plus aisée, on se propose de partir de la configuration fournie avec le paquet de la distribution. En procédant par modifications élémentaires à partir de cette configuration réputée sûre puisque permettant le fonctionnement du système actuel, on limite ainsi les possibilités d'erreurs.

Les versions stables du noyau évoluent fréquemment. Les questions ci-dessous sont basées sur la version courante de la série 2.6.xx.

1. Quels sont les principaux canaux de diffusion des sources du noyau Linux ?

Rechercher un site web, un dépôt de code en ligne et le nom du paquet de la distribution.

- Le site principal de publication des sources du noyau Linux est à l'adresse <http://www.kernel.org/>.
- Le développement du système de contrôle de version git a été initié par les développeurs du noyau Linux. Depuis, des services en lignes ont été bâtis à partir de git. Les branches de développement du noyau sont disponibles sur le site [GitHub](https://github.com/torvalds/linux)⁶ à l'adresse <https://github.com/torvalds/linux>.
- La distribution Debian GNU/Linux propose des paquets contenant les sources qui on servi à construire les paquets de noyau. Pour identifier ces paquets, on effectue une recherche dans le catalogue de la distribution.

```
$ aptitude search linux-source  
p linux-source - Linux kernel source (meta-package)  
p linux-source-2.6 - Linux kernel source (dummy package)  
p linux-source-2.6.32 - Linux kernel source for version 2.6.32 with Debian patches  
p linux-source-3.0.0 - Linux kernel source for version 3.0.0 with Debian patches
```

2. Quels sont les modes de téléchargement des sources qui permettent de s'affranchir d'une interface graphique ?

La grande majorité des téléchargements se font via le protocole HTTP. Pour trouver l'outil permettant de lancer un téléchargement HTTP, on peut faire une requête par mot clé dans les pages de manuels des outils installés sur le système : **apropos "network download"**.

Lorsque l'on utilise des serveurs *rack*, ceux-ci ne possèdent ni écran ni clavier. Il est donc nécessaire d'effectuer les opérations à distance sans recours à une interface graphique. On dispose de deux protocoles pour les transferts : FTP et HTTP. Les outils correspondant sont `ncftp` et `wget`.

Compte tenu des réponses à la question précédente, on peut utiliser les trois ressources suivantes.

- **Téléchargement** à partir du site principal de publication *kernel.org*.

⁶ <https://github.com/>

<http://www.eu.kernel.org/>

The Linux Kernel Archives

Linux Kernel Archives. This is the primary site for the Linux kernel source, but it has much more than just Linux kernels. [Frequently Asl](#)

Protocol	Location
HTTP	http://www.kernel.org/pub/
FTP	ftp://ftp.kernel.org/pub/
RSYNC	rsync://rsync.kernel.org/pub/

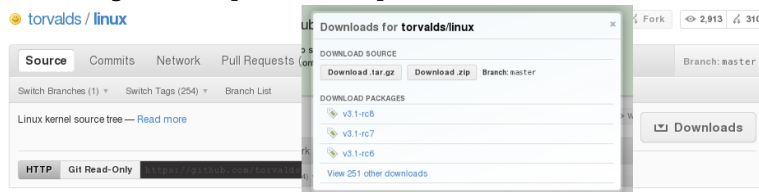
The latest stable version of the Linux kernel is: 2.6.27	2008-10-10 00:00 UTC	V V C Changelog
The latest 2.4 version of the Linux kernel is: 2.4.36.7	2008-09-07 10:22 UTC	Open Link in New Window
The latest prepatch for the 2.4 Linux kernel tree is: 2.4.37-rc1	2008-09-07 16:41 UTC	Open Link in New Tab
The latest 2.2 version of the Linux kernel is: 2.2.26	2004-02-25 00:28 UTC	Bookmark This Link
The latest prepatch for the 2.2 Linux kernel tree is: 2.2.27-rc2	2005-01-12 23:55 UTC	Save Link Target
The latest _mm patch to the stable Linux kernels is: 2.6.27-rc5-mm1	2008-09-05 05:03 UTC	Copy Link Location
F = full source, B = patch baseline, V = view patch, VI = view incremental, C = current changeset Change logs are provided by the kernel authors directly. Please don't write the webmaster about the Customize the patch viewer		Bookmark This Page
		Save Page

Téléchargement des sources du noyau Linux - vue complète⁷

```
$ wget http://www.eu.kernel.org/pub/linux/kernel/v2.6/linux-2.6.32.tar.bz2
-- http://www.eu.kernel.org/pub/linux/kernel/v2.6/linux-2.6.32.tar.bz2
Résolution de www.eu.kernel.org... 130.239.17.4, 199.6.1.164
Connexion vers www.eu.kernel.org|130.239.17.4|:80...connecté.
requête HTTP transmise, en attente de la réponse...200 OK
Longueur: 50355835 (48M) [application/x-bzip2]
Saving to: `linux-2.6.32.tar.bz2'

18% [=====] 9 500 807 217K/s eta 6m 46s
```

- Téléchargement à partir du dépôt **GitHub**⁸.



Téléchargement des sources du noyau Linux sur GitHub - vue complète⁹

```
$ wget https://github.com/torvalds/linux/tarball/v3.1-rc8
--2011-10-04 00:43:19-- https://github.com/torvalds/linux/tarball/v3.1-rc8
Résolution de github.com (github.com)... 207.97.227.239
Connexion vers github.com (github.com)|207.97.227.239|:443...connecté.
requête HTTP transmise, en attente de la réponse...302 Found
Emplacement: https://nodeload.github.com/torvalds/linux/tarball/v3.1-rc8 [suivant]
--2011-10-04 00:43:20-- https://nodeload.github.com/torvalds/linux/tarball/v3.1-rc8
Résolution de nodeload.github.com (nodeload.github.com)... 207.97.227.252
Connexion vers nodeload.github.com (nodeload.github.com)|207.97.227.252|:443...connecté.
requête HTTP transmise, en attente de la réponse...200 OK
Longueur: 98197772 (94M) [application/octet-stream]
Sauvegarde en : «v3.1-rc8»

100%[=====] 98 197 772 6,64M/s ds 18s

2011-10-04 00:43:39 (5,14 MB/s) - «v3.1-rc8» sauvegardé [98197772/98197772]

$ mv v3.1-rc8 linux-3.1-rc8.tar.bz2
$ tar tvf linux-3.1-rc8.tar.bz2
```

- Téléchargement à partir du gestionnaire de paquets de la distribution.

```
$ # aptitude install linux-source-3.0.0
Les NOUVEAUX paquets suivants vont être installés :
binutils{a} bzip2{a} cpp{a} cpp-4.6{a} gcc{a} gcc-4.6{a} libc-dev-bin{a}
libc6-dev{a} libcloog-pp10{a} libgmp10{a} libgmpxx4ldbl{a} libgomp1{a}
libmpc2{a} libmpfr4{a} libppl-c4{a} libppl9{a} libpwl5{a} libquadmath0{a}
linux-libc-dev{a} linux-source-3.0.0 make{a} manpages-dev{a}
0 paquets mis à jour, 22 nouvellement installés, 0 à enlever et 0 non mis à jour.
Il est nécessaire de télécharger 102 Mo d'archives. Après dépaquetage, 148 Mo seront utilisés.
Voulez-vous continuer ? [Y/n/?]
```

3. À quel groupe doit appartenir l'utilisateur normal pour pouvoir effectuer les opérations de compilation de modules ou du noyau ?

Rechercher dans la liste des groupes système, celui consacré à la manipulation des sources.

On cherche la chaîne `src` dans le fichier `/etc/group` et on ajoute l'utilisateur normal dans ce groupe.

⁷ http://www.inetdoc.net/travaux_pratiques/interco.kernel.qa/images/kerneldnld.png

⁸ <https://github.com/>

⁹ http://www.inetdoc.net/travaux_pratiques/interco.kernel.qa/images/github-kerneldnld.png

```
# grep src /etc/group
src:x:40:

# adduser etu src
Ajout de l'utilisateur « etu » au groupe « src »...
Ajout de l'utilisateur etu au groupe src
Fait.

# id etu
uid=1000(etu) gid=1000(etu) groupes=1000(etu),24(cdrom),25(floppy),
29(audio),30(dip),40(src),44(video),46(plugdev)
```

4. Quel est le répertoire de l'arborescence système dédié au stockage des sources du noyau Linux ?
Faire une recherche dans le document *Linux Filesystem Hierarchy*¹⁰.
C'est le répertoire `/usr/src` qui doit accueillir les sources du noyau.
On vérifie que les membres du groupe système `src` ont bien accès à ce répertoire.

```
# chgrp -R src /usr/src
# chmod 2775 /usr/src
```

5. Quelles sont les commandes «rituelles» d'installation des sources du noyau Linux ?
Pour chaque commande, expliquer les opérations réalisées et justifier le choix des options.
Il faut consulter les ressources suivantes : *Debian Linux Kernel Handbook*¹¹ et *Manuel de référence Debian - Chapitre 9*¹².

Pour traiter cette question, on utilise les fichiers sources obtenus à l'aide du gestionnaire de paquets. D'après les documents de référence on doit utiliser la séquence de commandes suivante.

```
$ cd /usr/src/
$ tar xf linux-source-3.0.0.tar.bz21
$ ln -s linux-source-3.0.0 linux2
$ cd linux
$ cp /boot/config-3.0.0-1-amd64 .config3
$ make menuconfig4
```

- ① Extraction de l'arborescence des sources du noyau.
- ② Création d'un lien symbolique sur l'arborescence de travail. L'utilisation de ce lien permet de conserver plusieurs arborescences de sources. De cette façon, on peut travailler sur plusieurs versions de noyau.
- ③ Copie du fichier de configuration fourni avec le paquet de noyau. Ce fichier est réputé fiable puisqu'il correspond au noyau en cours d'exécution et que le système est opérationnel.
- ④ Lancement de l'interface des menus de configuration des options du noyau Linux. C'est à ce niveau que les «choses sérieuses» commencent.

La dernière commande n'est utilisable que si le paquet de bibliothèques de développement `ncurses` est installé. `aptitude install libncurses-dev`.

4. La configuration du noyau Linux

On se propose de configurer un système d'interconnexion. Le noyau correspondant doit donc comprendre les éléments suivants.

- Un coeur système monolithique : microprocesseur, périphériques non réseau et système de fichiers,
- Le support des fonctions réseau nécessaires au routage.
- Le support du filtrage netfilter sous forme modulaire.
- Un pilote d'interface réseau Ethernet sous forme modulaire,
- Les fonctions de l'ancien sous-système RNIS sous forme modulaire,
- Un pilote d'interface RNIS sous forme modulaire,

1. Quelle est la commande utilisée pour les opérations de configuration et de compilation ?

Toutes les opérations de compilation du noyau étant basées sur des Makefiles, c'est la commande **make** qui sert aussi pour la configuration.

2. Comment obtenir la liste des options de cette commande ?

¹⁰ <http://tldp.org/LDP/Linux-Filesystem-Hierarchy/html/>

¹¹ <http://kernel-handbook.alioth.debian.org/>

¹² http://www.debian.org/doc/manuals/debian-reference/ch09.fr.html#_the_kernel

La commande `make help` donne la liste des options disponibles.

3. Quelles sont les 3 options de configuration du noyau ?

Préciser les différences entre ces 3 options.

Les 3 commandes sont `make config`, `make menuconfig` et `make xconfig`.

Il est préférable d'utiliser la commande `make menuconfig`. C'est le meilleur compromis entre facilité de navigation et administration distante. Les bibliothèques de développement ncurses ne consomment que très peu de ressources CPU et l'utilisation d'une interface graphique sur un serveur est à proscrire.

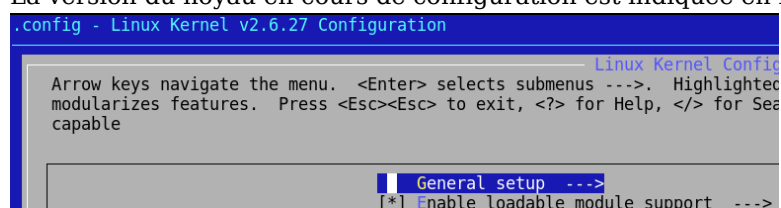
4. Sans opération préalable, quel est le fichier contenant les options de configuration du noyau utilisé ?

C'est le fichier texte `.config` qui contient l'ensemble des options de configuration du noyau Linux courant. Il est placé à la racine de l'arborescence des sources du noyau ; soit le répertoire `/usr/src/linux` dans notre cas.

Le fichier «patron» de configuration pour ces travaux pratiques doit donc être copié dans le répertoire `/usr/src/linux` et renommé `.config`. L'opération a déjà été effectuée à la Q : 5

5. Une fois la commande de configuration exécutée, comment identifier la version du noyau à compiler ?

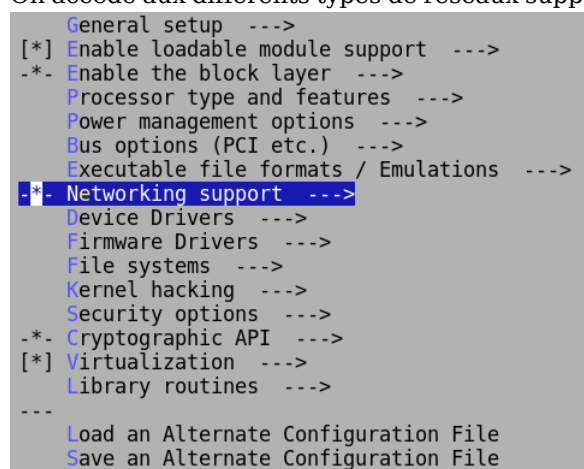
La version du noyau en cours de configuration est indiquée en haut à gauche de l'écran.



Identification version noyau Linux - vue complète¹³

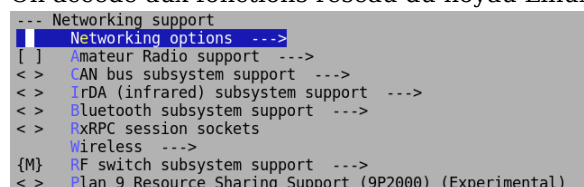
6. Quelles sont les options indispensables et facultatives des rubriques Networking Support puis Networking options ?

On accède aux différents types de réseaux supportés par le noyau Linux via l'item Networking Support.



Accès aux types de réseaux supportés - vue complète¹⁴

On accède aux fonctions réseau du noyau Linux via l'item Networking options.



Accès aux fonctions réseau du noyau Linux - vue complète¹⁵

À partir du support *Fonctions réseau du noyau Linux*¹⁶ et de l'organisation des menus, on distingue les options génériques, telles que le support des *sockets*, des options spécifiques telles que celles relatives au filtrage.

¹³ http://www.inetdoc.net/travaux_pratiques/interco.kernel.qa/images/kernelcfgver.png

¹⁴ http://www.inetdoc.net/travaux_pratiques/interco.kernel.qa/images/kernelcfgnet0.png

¹⁵ http://www.inetdoc.net/travaux_pratiques/interco.kernel.qa/images/kernelcfgnet1.png

¹⁶ <http://www.inetdoc.net/guides/linux.networking/>

```

[*] Packet socket
[*] Packet socket: mmaped IO
<-> Unix domain sockets
<-> Transformation user configuration interface
[ ] Transformation sub policy support (EXPERIMENTAL)
[ ] Transformation migrate database (EXPERIMENTAL)
[ ] Transformation statistics (EXPERIMENTAL)
<-> PF_KEY sockets
[ ] PF_KEY MIGRATE (EXPERIMENTAL)
[*] TCP/IP networking
[*] IP: multicasting
[*] IP: advanced router
    Choose IP: FIB lookup algorithm (choose FIB_HASH if unsure) (FIB_HASH)
[*] IP: policy routing
[*] IP: equal cost multipath
[*] IP: verbose route monitoring
[ ] IP: kernel level autoconfiguration
<-> IP: tunneling
<-> IP: GRE tunnels over IP
[*] IP: broadcast GRE over IP
[*] IP: multicast routing
[*] IP: PIM-SM version 1 support
[*] IP: PIM-SM version 2 support
[ ] IP: ARP daemon support (EXPERIMENTAL)
[*] IP: TCP syncookie support (disabled per default)
<-> IP: AH transformation
<-> IP: ESP transformation
<-> IP: IPComp transformation
<-> IP: IPsec transport mode
<-> IP: IPsec tunnel mode
<-> IP: IPsec BEET mode
<-> Large Receive Offload (ipv4/tcp)
<-> INET: socket monitoring interface
[*] TCP: advanced congestion control --->
[*] TCP: MD5 Signature Option support (RFC2385) (EXPERIMENTAL)
<-> IP virtual server support (EXPERIMENTAL) --->
<-> The IPv6 protocol --->
v(1)
<-Select> <-Exit> <-Help>
    
```

Fonctions TCP/IP du noyau Linux - vue complète¹⁷

Le menu principal de la partie filtrage *netfilter* se présente comme une longue liste de fonctions. Sachant que les modules dédiés à une fonction du filtrage se chargent dynamiquement à la demande lors de l'application des règles de filtrage, on sélectionne généralement la totalité de ces fonctions sous forme modulaire. Seuls les modules effectivement utilisés seront chargés en mémoire.

```

[-] Network packet filtering framework (Netfilter)
[ ] Network packet filtering debugging
[ ] Advanced netfilter configuration
Core Netfilter Configuration --->
IP: Netfilter Configuration --->
IPv6: Netfilter Configuration --->
    
```

Fonctions de filtrage réseau du noyau Linux - vue complète¹⁸

7. Quelles sont les options indispensables et facultatives des rubriques Device Drivers puis Network device support ?

Voir le support *Fonctions réseau du noyau Linux*¹⁹ pour s'orienter dans les options à sélectionner.

Pour accéder au catalogue des interfaces réseau supportées par le noyau il faut passer par la catégorie des pilotes de périphériques ou *Device Drivers* pour accéder à l'item Network device support.

```

Device Drivers
Enter> selects submenus --->. Highlighted letters are hotkeys. Press:
<-> <-Esc> to exit, <-?> for Help, <-/?> for Search. Legend: [*] built-in

Generic Driver Options --->
<-> Connector - unified userspace <-> kernelspace linker --->
<-> Memory Technology Device (MTD) support --->
<-> Parallel port support --->
-* Plug and Play support --->
[*] Block devices --->
[*] Misc devices --->
<-> ATA/ATAPI/MFM/RLL support --->
    SCSI device support --->
<-> Serial ATA (prod) and Parallel ATA (experimental) drivers --->
[*] Multiple devices driver support (RAID and LVM) --->
[*] Fusion MPT device support --->
IEEE 1394 (FireWire) support --->
<-> I2O device support --->
[*] Macintosh device drivers --->
[*] Network device support --->
[*] ISDN support --->
<-> Telephony support --->
    Input device support --->
    Character devices --->
{M} I2C support --->
[*] SPI support --->
[ ] GPIO Support --->
{M} Dallas's 1-wire support --->
-* Power supply class support --->
{H} Hardware Monitoring support --->
{M} Generic Thermal sysfs driver --->
[*] Watchdog Timer Support --->
[*] Sonics Silicon Backplane --->
    Multifunction device drivers --->
    Multimedia devices --->
    Graphics support --->
<-> Sound card support --->
[*] HID Devices --->
[*] USB support --->
<-> MMC/SD card support --->
<-> Sony MemoryStick card support (EXPERIMENTAL) --->
    
```

¹⁷ http://www.inetdoc.net/travaux_pratiques/interco.kernel.qa/images/kernelcfgnet2.png

¹⁸ http://www.inetdoc.net/travaux_pratiques/interco.kernel.qa/images/kernelcfgnet3.png

¹⁹ <http://www.inetdoc.net/guides/linux.networking/>

Accès aux interfaces réseau supportées - vue complète²⁰

Le catalogue recense tous les types d'interfaces réseau. Dans notre cas, il faut choisir le bon modèle d'interface Ethernet.

```

Network device support
<Enter> selects submenus ---. Highlighted letters are hotk
sc<<Esc> to exit, <?> for Help, </> for Search. Legend: [*]

--- Network device support
<M> Intermediate Functional Block support
<M> Dummy net driver support
<M> Bonding driver support
<M> MAC-VLAN support (EXPERIMENTAL)
<M> RQL (serial line load balancing) support
<M> Universal TUN/TAP device driver support
<M> Virtual ethernet pair device
< > General Instruments Surfboard 1000
< > ARCnet support --->
<M> PHY Device support and infrastructure --->
[*] Ethernet (10 or 100Mbit) --->
[*] Ethernet (1000 Mbit) --->
[ ] Ethernet (10000 Mbit) --->
[ ] Token Ring driver support --->
Wireless LAN --->
USB Network Adapters --->
[ ] Wan interfaces support --->
[ ] FDDI driver support
[ ] HIPPI driver support (EXPERIMENTAL)
< > PLIP (parallel port) support
<M> PPP (point-to-point protocol) support
[*] PPP multilink support (EXPERIMENTAL)
[*] PPP filtering
<M> PPP support for async serial ports
<M> PPP support for sync tty ports
<M> PPP Deflate compression
<M> PPP BSD-Compress compression
<M> PPP MPPE compression (encryption) (EXPERIMENTAL)
<M> PPP over Ethernet (EXPERIMENTAL)
<M> PPP over L2TP (EXPERIMENTAL)
<M> SLIP (serial line) support
[*] CSLIP compressed headers
[*] Keepalive and linefill
[*] Six bit SLIP encapsulation
[ ] Fibre Channel driver support
<M> Network console logging support (EXPERIMENTAL)

```

Catalogue des types d'interfaces réseau - vue complète²¹

8. Quelles sont les options indispensables et facultatives de la rubrique ISDN subsystem ?

À partir de la liste des pilotes de périphériques du noyau, on accède aux paramétrage du sous-système RNIS/ISDN.

```

[ ] Macintosh device drivers --->
[*] Network device support --->
[*] ISDN support --->
< > Telephony support --->
Input device support --->

```

Accès au sous-système RNIS/ISDN - vue complète²²

Il existe trois types d'utilisation des connexions RNIS/ISDN dans le noyau Linux.

- Le premier, le plus récent, utilise un mécanisme de sockets adapté aux reste des fonctions réseau du noyau.
- Le second est hérité des noyaux de la série 2.2.xx. Il comprend une machine d'état logicielle autonome de gestion de l'établissement, du maintien et de la libération des connexions. C'est ce type de connexion que l'on utilise dans la suite des travaux pratiques de la série.
- Le troisième utilise le standard CAPI. Il s'agit d'une interface logicielle normalisée entre le noyau et le périphérique matériel.

²⁰ http://www.inetdoc.net/travaux_pratiques/interco.kernel.qa/images/kernelcfgnet4.png

²¹ http://www.inetdoc.net/travaux_pratiques/interco.kernel.qa/images/kernelcfgnet5.png

²² http://www.inetdoc.net/travaux_pratiques/interco.kernel.qa/images/kernelcfgisdn0.png

```
ISDN support
lects submenus --->. Highlighted letters are ho
for Help, </> for Search. Legend: [*] built-in

--- ISDN support
< > Modular ISDN driver --->
<M> Old ISDN4Linux (deprecated) --->
< > CAPI 2.0 subsystem --->
```

Types de connexions RNIS/ISDN - vue complète²³

Le catalogue des paramètres utilisables avec le protocole PPP associé au sous-système RNIS/ISDN historique du noyau Linux est donné ci-dessous.

```
Old ISDN4Linux (deprecated)
lects submenus --->. Highlighted letters are hotkeys. F
for Help, </> for Search. Legend: [*] built-in [ ] excl

-[-] Old ISDN4Linux (deprecated)
[*] Support synchronous PPP
[*] Use VJ-compression with synchronous PPP
[*] Support generic MP (RFC 1717)
[*] Filtering for synchronous PPP
<M> Support BSD compression
[ ] Support audio via ISDN
[ ] X.25 PLP on top of ISDN
ISDN feature submodules --->
*** ISDN4Linux hardware drivers ***
Passive cards --->
Active cards --->
< > Siemens Gigaset support (isdn) --->
```

Paramètres PPP du sous-système RNIS/ISDN - vue complète²⁴

La liste des modèles de cartes RNIS/ISDN supportés par le sous-système RNIS/ISDN historique du noyau Linux est donnée ci-dessous. Cette organisation est liée au fait que les mêmes composants SiemensTM ont été utilisés sur de nombreux modèles de cartes de marques différentes.

²³ http://www.inetdoc.net/travaux_pratiques/interco.kernel.qa/images/kernelcfgisdn1.png

²⁴ http://www.inetdoc.net/travaux_pratiques/interco.kernel.qa/images/kernelcfgisdn2.png

```

Passive cards
selects submenus -->. Highlighted letters are hotkey
> for Help, </> for Search. Legend: [*] built-in [ ]

< > HiSax SiemensChipSet driver support
*** D-channel protocol features ***
[ ] HiSax Support for EURO/DSS1
[ ] HiSax Support for german ITR6
[ ] HiSax Support for US NI1
(8) Maximum number of cards supported by HiSax
*** HiSax supported cards ***
[ ] Teles 16.3 or PNP or PCMCIA
[ ] Teles PCI
[ ] Teles S0Box
[ ] AVM PnP/PCI (Fritz!PnP/PCI)
[ ] AVM A1 PCMCIA (Fritz)
[ ] Elsa cards
[ ] Eicon.Diehl Diva cards
[ ] Sedlbauer cards
[ ] NETjet card
[ ] NETspider U card
[ ] Niccy PnP/PCI card
[ ] Telekom A4T card
[ ] Scitel Quadro card
[ ] Gazel cards
[ ] HFC PCI-Bus cards
[ ] Winbond W6692 based cards
[ ] HFC-S+, HFC-SP, HFC-PCMCIA cards
[ ] HiSax debugging
*** HiSax PCMCIA card service modules ***
< > AVM A1 PCMCIA cards
*** HiSax sub driver modules ***
< > ST5481 USB ISDN modem (EXPERIMENTAL)
< > HFC USB based ISDN modems (EXPERIMENTAL)

```

Modèles de cartes utilisant les mêmes composants Siemens - vue complète²⁵

Le modèle des cartes implantées dans les postes de travaux pratiques est de type AVM Fritz/PCI 2.0.

5. La compilation & l'installation du nouveau noyau Linux

1. Quel est le paquet qui contient les outils de construction de paquet de noyau ?

Rechercher le mot clé kernel à l'aide du gestionnaire de paquets. Installer le paquet correspondant

La recherche dans les attributs du gestionnaire de paquets permet d'identifier le paquet kernel-package.

```

$ aptitude search kernel | grep package
p  debian-kernel-handbook - reference to Debian Linux kernel packages
p  kernel-package        - A utility for building Linux kernel relate
p  kernel-patch-grsecurity2 - transitional package for Debian Lenny
p  kernel-patch-scripts  - Scripts to help dealing with packaged kern

```

Suivant l'état antérieur de l'installation système, la liste des dépendances est plus ou moins importante lors de l'installation de kernel-package.

```

# aptitude install kernel-package
Les NOUVEAUX paquets suivants vont être installés :
autopoint{a} build-essential{a} dpkg-dev{a} fakeroot{a} g++{a} g++-4.6{a}
gettext{a} git{a} git-man{a} intltool-debian{a} kernel-package
libalgorithm-diff-perl{a} libalgorithm-diff-xs-perl{a}
libalgorithm-merge-perl{a} libcrococ3{a} libcurl3-gnutls{a} libdpkg-perl{a}
liberror-perl{a} libglib2.0-0{a} libglib2.0-data{a} libmail-sendmail-perl{a}
librtmp0{a} libstdc++6-4.6-dev{a} libsys-hostname-long-perl{a}
libtimedate-perl{a} libunistring0{a} po-debconf{a} rsync{a}
shared-mime-info{a}

```

2. Quelles sont les commandes de compilation du noyau ?

Rechercher les commandes dans le support *Manuel de référence Debian - Chapitre 9*²⁶ et donner la signification de chacune des commandes.

Pour faciliter les opérations de (dé)réinstallation du noyau, on se propose de construire un paquet Debian de noyau Linux. L'utilisation d'un paquet permet de s'assurer que tous les fichiers nécessaires ont bien été (copiés|supprimés) dans l'arborescence du système.

```

$ pwd

```

²⁵ http://www.inetdoc.net/travaux_pratiques/interco.kernel.qa/images/kernelcfgisdn3.png
²⁶ http://www.debian.org/doc/manuals/debian-reference/ch09.fr.html#_the_kernel

```
/usr/src/linux
$ export CONCURRENCY_LEVEL=`grep -c '^processor' /proc/cpuinfo`
$ make-kpkg clean
$ make-kpkg --rootcmd fakeroot --initrd --append-to-version -1st-try kernel_image
```

3. Quelles sont les étapes d'installation du noyau compilé ?

Quel outil faut-il utiliser pour gérer les paquets localement sur le système ?

Une fois le paquet de noyau construit, il ne reste plus qu'à procéder à l'installation de ce paquet local. Cette étape fait appel à l'outil de gestion de bas niveau des paquets Debian : **dpkg**. Cette opération nécessite les droits du super-utilisateur.

```
# pwd
/usr/src
# dpkg -i linux-image-3.0.0-1st-try_3.0.0-1st-try-10.00.Custom_amd64.deb
```

Après cette installation de paquet de noyau on peut valider la liste des paquets correspondant installés.

```
# dpkg -l linux-image* |grep ^ii
ii linux-image-2.6-amd64 3.0.0+39 Linux for 64-bit PCs (dummy package)
ii linux-image-3.0.0-1-amd64 3.0.0-3 Linux 3.0.0 for 64-bit PCs
ii linux-image-3.0.0-1st-try 3.0.0-1st-try-10.00.Custom Linux kernel binary image for version 3.0.0-1st-try
ii linux-image-amd64 3.0.0+39 Linux for 64-bit PCs (meta-package)
```

4. Que faut-il faire pour que le gestionnaire de démarrage propose le nouveau noyau compilé lors de l'initialisation du système ?

Identifier le gestionnaire d'amorce installé sur le système.

En fait, l'opération d'installation du paquet de noyau intègre l'ajout d'une nouvelle entrée dans le gestionnaire de démarrage. Aucune opération supplémentaire n'est donc nécessaire.

on peut tout de même valider la liste des noyaux disponibles au niveau du gestionnaire d'amorce. Dans le cas de grub avec la distribution Debian GNU/Linux, on obtient une liste du type suivant.

```
# update-grub
Generating grub.cfg ...
Found linux image: /boot/vmlinuz-3.0.0-1st-try
Found initrd image: /boot/initrd.img-3.0.0-1st-try
Found linux image: /boot/vmlinuz-3.0.0-1-amd64
Found initrd image: /boot/initrd.img-3.0.0-1-amd64
done
```

Une fois toutes ces étapes franchies, il ne reste plus qu'à relancer le système et vérifier que le noyau exécuté est bien celui qui a été recompilé à partir des sources.

6. Documents de référence

Debian Linux Kernel Handbook

*Debian Linux Kernel Handbook*²⁷ : guide sur les techniques de construction d'un paquet Debian de noyau Linux.

Manuel de référence Debian

*Manuel de référence Debian - Chapitre 9*²⁸ : La section 9.7 traite des opérations de configuration et de compilation d'un noyau Linux.

²⁷ <http://kernel-handbook.alioth.debian.org/>

²⁸ http://www.debian.org/doc/manuals/debian-reference/ch09.fr.html#_the_kernel