
RAPPORT FINAL : COMPILATION D'UN NOYAU OPTIMISE POUR UNE UTILISATION SOUS VM

Elodie Kruczowy, ING2 TD3

LE NOYAU

DEFINITIONS ET EXPLICATIONS

Linux est un noyau (kernel) open-source. C'est le premier programme chargé en mémoire lors du démarrage de la machine sur le système d'exploitation. Il reconnaît le matériel physique installé sur la machine (carte-mère, cartes graphiques, processeurs, cartes son, disques durs, etc...) et en prend le contrôle pour lancer le système d'exploitation lui-même. Lorsque cette opération est terminée, le noyau reste actif et a pour rôle de faire communiquer entre eux les matériels et les programmes. Par exemple, si on lance un DVD dans VLC, c'est le noyau qui fait communiquer entre l'application et le lecteur. C'est un modèle à trois couches :

(les utilisateurs)		
shells et commandes compilateurs et interpréteurs bibliothèques système		
<i>interface entre appels système et noyau</i>		
gestion des signaux système d'E/S caractères pilotes de terminaux	système de fichiers système de permutation et E/S par blocs disques et lecteurs de bandes	Ordonnancement UC remplacement de page pagination à la demande mémoire virtuelle
<i>interface entre noyau et matériel</i>		
contrôleurs de terminaux terminaux	contrôleurs de périphériques disques et lecteurs de bande	contrôleurs mémoire mémoire physique

COMPILER LE NOYAU

Le noyau que l'on télécharge et installe en même temps qu'une distribution suffit le plus souvent à une utilisation normale et il est rare que la recompilation de son noyau soit nécessaire. Cependant reconfigurer son noyau pour le compiler et l'installer peut être utile dans deux cas de figure :

- Lorsqu'on dispose de matériel très spécifique non pris en charge par le noyau fourni par la distribution
- Lorsqu'on désire optimiser l'utilisation du matériel physique, soit en supprimant la gestion de matériel non présent sur la machine, soit précisant certaines options pour que le noyau soit paramétré

exactement pour le matériel dont on dispose (par exemple, si certains périphériques sont gérés par des modules génériques)

PARAMETRER LE NOYAU

LES MODULES

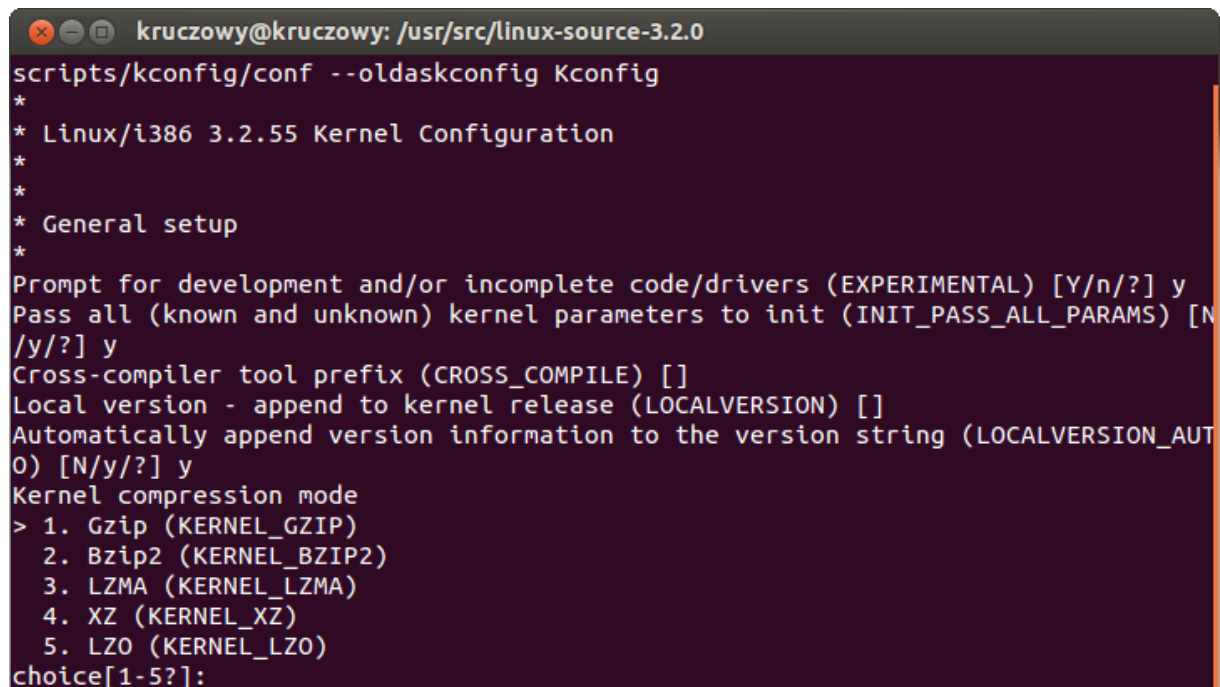
Lors de la configuration du noyau, de nombreuses options peuvent être ajoutées en tant que module. Une option compilée « en dur » dans le noyau est entièrement intégrée à celui-ci : elle est chargée en mémoire au démarrage du système et reste active en permanence lorsque le système est actif. Une option intégrée en tant que module n'est au contraire chargée en mémoire que quand elle est nécessaire au déroulement du système, de certains programmes ou pour l'utilisation d'un matériel. Cette méthode permet d'économiser de la mémoire mais n'est pas applicable à toutes les options : certaines options, comme par exemple celles liées à l'utilisation des disques durs, sont indispensables et donc considérées comme non modularisables.

LES DIFFERENTES INTERFACES

Il existe plusieurs interfaces pour paramétrer le noyau.

MODE TEXTE

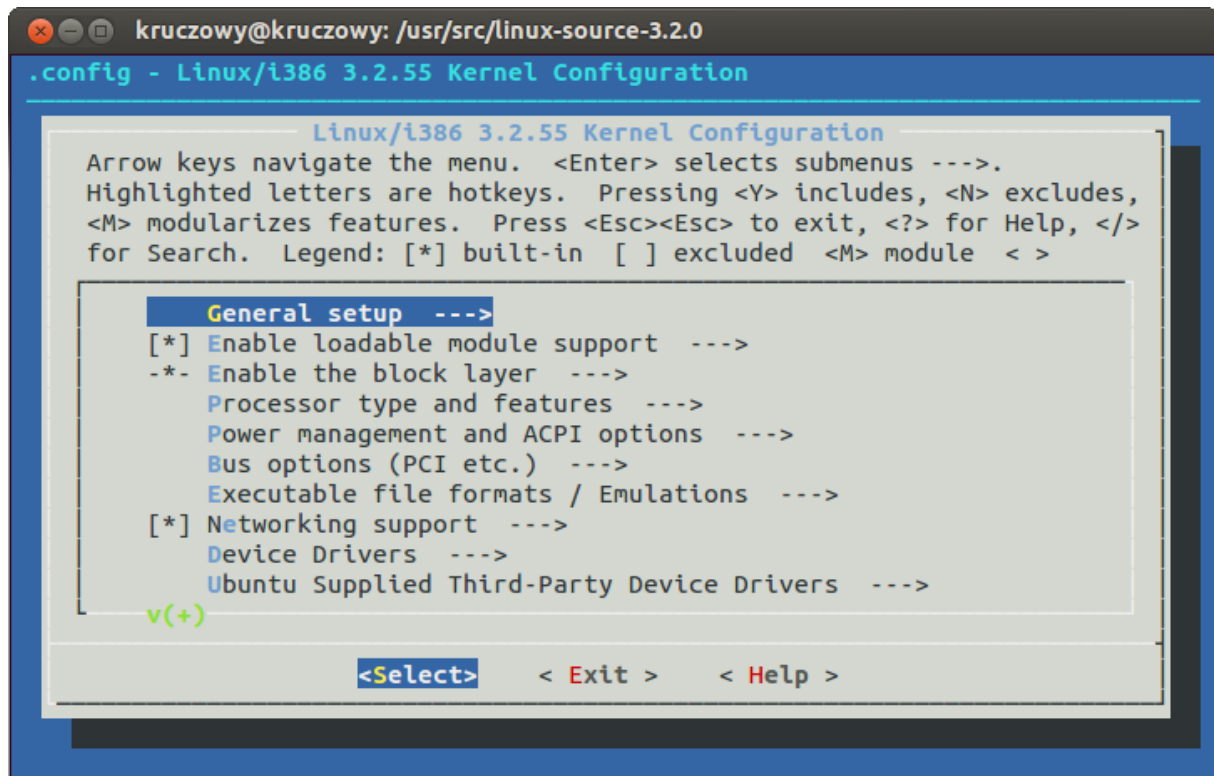
Le moyen le plus simple de configurer le noyau est de passer par les lignes de commande. Chaque option est proposée sous forme de question à laquelle on peut répondre par « n », « y », « m » ou « ? ». « n » signifie que le module ne sera pas intégré au noyau, « y » qu'il sera installé en dur, « m » qu'il sera intégré en tant que module, et « ? » permet d'obtenir de l'aide concernant l'option proposée. Cette méthode est très austère, peu rapide (on est obligé de voir défiler toutes les options une par une) et il est impossible de revenir sur un choix précédent (on doit alors recommencer toute l'opération).



```
kruczowy@kruczowy: /usr/src/linux-source-3.2.0
scripts/kconfig/conf --oldaskconfig Kconfig
*
* Linux/i386 3.2.55 Kernel Configuration
*
*
* General setup
*
Prompt for development and/or incomplete code/drivers (EXPERIMENTAL) [Y/n/?] y
Pass all (known and unknown) kernel parameters to init (INIT_PASS_ALL_PARAMS) [N/y/?] y
Cross-compiler tool prefix (CROSS_COMPILE) []
Local version - append to kernel release (LOCALVERSION) []
Automatically append version information to the version string (LOCALVERSION_AUTO) [N/y/?] y
Kernel compression mode
> 1. Gzip (KERNEL_GZIP)
  2. Bzip2 (KERNEL_BZIP2)
  3. LZMA (KERNEL_LZMA)
  4. XZ (KERNEL_XZ)
  5. LZO (KERNEL_LZO)
choice[1-5?]:
```

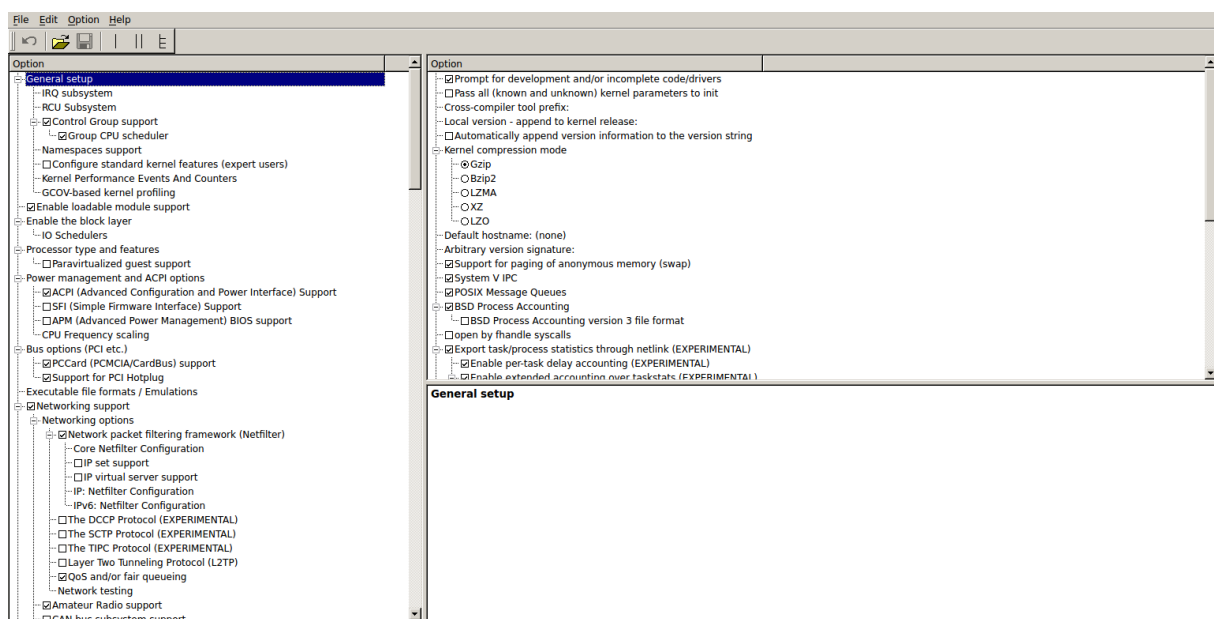
MENU EN MODE TEXTE

C'est un menu en couleur disponible avec la console. Contrairement au mode texte, il permet de naviguer avec le clavier dans un menu sous forme d'arborescence, les options sont regroupées par catégories et il est possible de modifier certaines options que l'on a choisies. Le principal avantage de cette interface est qu'elle peut être utilisée sans serveur graphique (X Window).



MENU EN MODE FENETRE

Ce mode est accessible en deux versions : l'une utilise la bibliothèque graphique Qt et l'autre Gtk. D'une manière générale, ces deux méthodes sont très semblables au menu en mode texte, si ce n'est que l'on navigue avec la souris et non le clavier.



LE PROJET

Pour la réalisation du projet personnel, j'ai voulu recompiler mon noyau pour l'adapter à l'utilisation quelque peu « minimaliste » que j'en ai sur une machine virtuelle Virtual Box et pour l'adapter à mon matériel. J'utilise une distribution Ubuntu version 12.04.

INSTALLATIONS PRELIMINAIRES

Il est nécessaire d'installer plusieurs paquets avant de configurer le noyau :

- `linux-source` : C'est un méta-paquetage (un regroupement de plusieurs paquets téléchargeables ensemble sous ce nom) qui contient toutes la dernière version des sources du noyau Linux
- `debconf-utils`, `dpkg-dev`, `debhelper`, `build-essential` : Ce sont des paquets qui permettent de manipuler, compiler, et mettre à jour des paquets de la distribution Debian (Ubuntu étant une distribution basée sur Debian)
- `kernel-package` : Ce paquet permet de créer un paquet Debian pour créer un noyau.
- `libncurses5-dev` : Bibliothèque graphique permettant d'ouvrir le menu en mode texte.
- `qt-sdk` : Bibliothèque Qt (mode fenêtré)
- `libgtk2.0-dev`, `libglib2.0-dev`, `libglade2-dev` : Bibliothèque GTK (mode fenêtré)

Puis on doit décompresser les sources Linux qui se trouvent dans `/usr/src` :

```
cd /usr/src
tar xjf1 linux-source-3.2.0.tar.bz2
```

Ensuite, on doit récupérer un fichier de configuration opérationnel du noyau ² (on peut également entièrement réécrire une configuration, mais ça peut être dangereux si on ne sait pas exactement de quelles options on a besoin pour que le noyau fonctionne normalement) :

```
cd /usr/src/linux-source-3.2.0
make3 defconfig
```

CHOISIR L'INTERFACE DE CONFIGURATION

Pour lancer la configuration, on doit se trouver dans le répertoire des sources (`linux-source-VERSION`) et on tape la commande qui correspond à l'interface que l'on a choisie :

- `make config` : mode texte
- `make menuconfig` : menu en mode texte
- `make xconfig` : mode fenêtré avec Qt
- `make gconfig` : mode fenêtré avec GTK

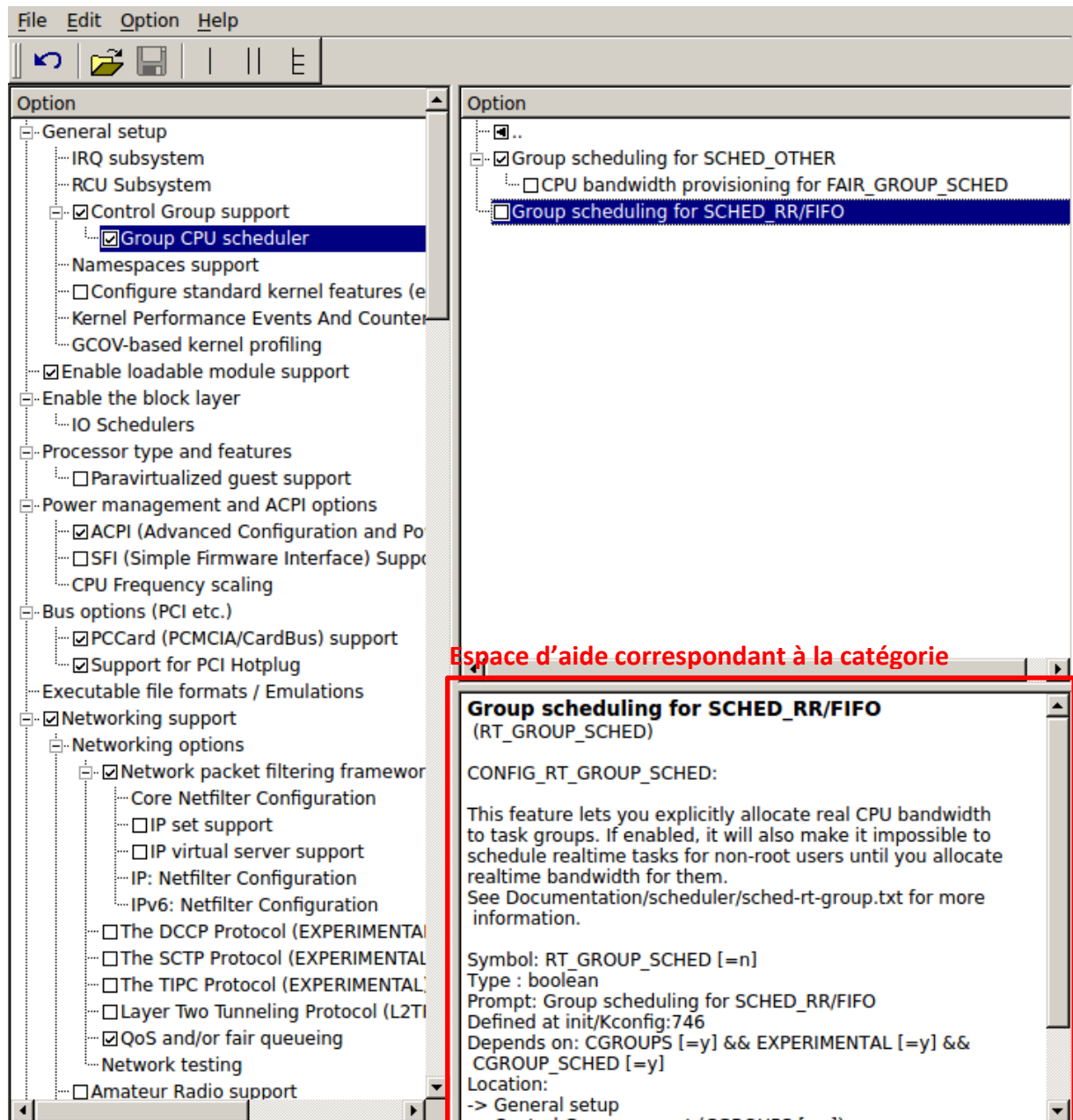
¹ Avec [`x` = extraire][`j` = compression Bzip2][`f` = fichier donné en paramètre]

² On peut également récupérer le fichier de configuration du noyau actuel déjà présent dans le répertoire `/boot`

³ On utilise ici un `makefile` présent dans le répertoire `linux-source-3.2.0`, comme chaque fois que nous ferons appel à la commande `make` dans la suite du projet

J'ai personnellement choisi d'utiliser l'interface en mode fenêtré Qt, car elle propose une aide très facile d'accès pour toutes les options à configurer.

```
cd /usr/src/linux-source-3.2.0
make xconfig
```



LES OPTIONS DE CONFIGURATION

LES DEPENDANCES

Les options à configurer sont regroupées sous des catégories et permettent de mieux visualiser quelles options sont interdépendantes. Par exemple, si on désactive `Enable loadable module support`, on n'a plus accès aux options qui sont sous cette catégorie et qui seront donc automatiquement désactivées. De même, on ne peut pas activer « en dur » une option qui dépend d'une autre activée « en module ».

MODIFICATION DE LA CONFIGURATION

Processor type and features – Dans la section Processor family, il faut renseigner le type de processeur de la machine : 386 ici.

Power management and ACPI options – La plupart des options ont été désactivées : c'est le système hôte qui gèrera les problèmes d'alimentation. Cela concerne notamment la mise en veille, la mise en veille prolongée, la gestion de la batterie, la gestion d'un possible switch entre batterie et alimentation secteur, le contrôle des ventilateurs associés, la gestion de certains appareils en mode économie d'énergie, etc...

Networking support – Cette section regroupe tous les supports réseaux, tels que l'adressage IP, l'utilisation de sockets ou de protocole tels de TCP/IP. Des options concernant des fonctionnalités soit gérées par le système hôte soit qui ne seront pas utiles si on considère l'utilisation de la VM, ont été désactivées :

- Amateur Radio support : permet d'écouter des radios amateurs
- Bluetooth subsystem support
- RF switch subsystem support : module de gestion de fréquences sur des cartes réseaux

Device drivers – **USB support** – **Activation de USB Serial Converter support** (convertisseur USB/série) en tant que module : j'aurais besoin de cette fonctionnalité pour un projet étudiant.

File systems – Etant donné que la machine tourne sous un OS Windows, le support des systèmes de fichiers NTFS (en lecture et en écriture) a été activé en tant que module. Cela permettra par exemple de travailler sur une clé USB ou un disque dur externe lisible par l'OS hôte.

Virtualization – Option désactivée : je ne virtualiserai aucun OS depuis cette machine virtuelle.

Enfin, on sauvegarde la configuration (simple clic sur le bouton Save).

COMPILER ET INSTALLER LE NOYAU

Pour la compilation et l'installation, j'ai choisi une méthode proposée sur le tutoriel « Compiler Linux »⁴ que j'ai suivie sans modification, à savoir :

- On compile le noyau

```
cd /usr/src/linux
make && make modules_install
```

- On copie le noyau et la configuration dans /boot

```
cp arch/i386/boot/bzImage /boot/kernel-3.2.0
cp .config /boot/config-3.2.0
cp System.map /boot/System.map.3.2.0
```

- On édite un fichier du grub, menu.lst, qui permettra de booter sur le noyau nouvellement créé

```
title                Ubuntu GNU/Linux 3.2.0
root                 (hd0,3)
kernel               /boot/kernel-3.2.0 root=/dev/hda4
```

⁴ Disponible dans la partie « Sources et Bibliographie »

PROBLEMES ET CONCLUSION

Au cours de ce projet personnel, j'ai rencontré plusieurs problèmes.

- Dans un premier temps, j'ai voulu configurer le noyau avec l'interface GTK, mais celle-ci s'est avérée non fonctionnelle sur ma machine, et je n'ai pas compris pourquoi.
- La famille de processeurs que j'ai choisie a été « par défaut ». N'étant pas familière du sujet et ayant eu du mal à comprendre la documentation que j'ai trouvée, j'ai donc choisi la famille 386 car elle correspondait à des vieilles versions de processeurs Intel (le modèle de mon processeur est : Intel® Core™ i5-2450M)
- Malgré toutes mes recherches, je n'ai pas réussi à installer le noyau que j'ai compilé. J'ai pourtant essayé de nombreuses méthodes, qui ont toutes mené soit à des erreurs de montage (notamment lorsque j'ai essayé de créer un paquet pour ensuite l'installer) ou alors n'ont absolument rien donné (je suis restée sur mon noyau d'origine : 3.11.0-20-generic)

Finalement, mon projet n'est pas abouti, et je n'ai malheureusement pas pu tester la configuration que j'avais préparée pour le noyau. Cela aurait été intéressant à mon avis, car j'ai choisi de désactiver beaucoup d'options, et j'avais beaucoup de doutes sur la fonctionnalité de celui-ci.

Pour autant, ce travail m'a poussé à faire de nombreuses recherches dans un domaine que je ne connaissais pas du tout : le hardware. De plus, parcourir les pages de documentation, les forums et les tutoriels sur Ubuntu et ses fonctionnalités m'a fait comprendre à quel point la communauté Linux était puissante, et que travailler sous une distribution Linux était à la portée de tous pour peu que l'on sache où et quoi chercher.

SOURCES ET BIBLIOGRAPHIE⁵

http://doc.ubuntu-fr.org/tutoriel/compiler_linux

http://doc.ubuntu-fr.org/tutoriel/comment_compiler_un_kernel_sur_mesure

<http://doc.ubuntu-fr.org/kernel>

<http://www.linux-france.org/article/kafkafr/node36.html>

<http://kistren.polytech.unice.fr/cours/sae/td1/02%20-%20TD.pdf>

<http://www.linux-france.org/article/mininet/noyau/noyau-3.html>

⁵ Seules les sources principales ont été indiquées : les ressources utilisées pour définir certains termes ou qui ne sont pas directement exploitées dans le document ne sont pas citées.