

# Rapport pour le tp2

Par Quentin Jeanmonod

## Introduction

Le but de ce deuxième TP est de compiler un nouveau kernel sur une machine linux.

## Déroulement

Tout d'abord, je commence par déterminer la version actuelle de mon noyau linux à l'aide de `uname`.

```
quentin@Quentin-laptop-debian:~$ uname -a
Linux Quentin-laptop-debian 3.2.0-4-686-pae #1 SMP Debian 3.2.68-1+deb7u3 i686 GNU/Linux
```

Après avoir téléchargé et extrait la version 4.3 du kernel (dernière version stable actuellement, 04 novembre 2015), je visite les dossiers créés par la commande `tar xpvf linux-4.3.tar.xz -C ./linux-4.3`.

J'utilise `make help` afin de déterminer comment créer le fichier `.config` pour la compilation.

```
Configuration targets:
config          - Update current config utilising a line-oriented program
nconfig         - Update current config utilising a ncurses menu based
                  program
menuconfig      - Update current config utilising a menu based program
xconfig         - Update current config utilising a Qt based front-end
gconfig         - Update current config utilising a GTK+ based front-end
oldconfig       - Update current config utilising a provided .config as base
localmodconfig  - Update current config disabling modules not loaded
localyesconfig  - Update current config converting local mods to core
silentoldconfig - Same as oldconfig, but quietly, additionally update deps
defconfig       - New config with default from ARCH supplied defconfig
savedefconfig   - Save current config as ./defconfig (minimal config)
allnoconfig     - New config where all options are answered with no
allyesconfig    - New config where all options are accepted with yes
allmodconfig    - New config selecting modules when possible
alldefconfig    - New config with all symbols set to default
randconfig      - New config with random answer to all options
listnewconfig   - List new options
olddefconfig    - Same as silentoldconfig but sets new symbols to their
                  default value
kvmconfig       - Enable additional options for kvm guest kernel support
xenconfig       - Enable additional options for xen dom0 and guest kernel support
tinyconfig      - Configure the tiniest possible kernel
```

L'option `olddefconfig` m'a l'air d'être la manière la plus simple de configurer en se basant sur la config actuelle, c'est donc cette option que j'utiliserai.

```

quentin@Quentin-laptop-debian:~/Documents/tp2/linux-4.3/linux-4.3$ make olddefconfig
HOSTCC scripts/basic/fixdep
HOSTCC scripts/kconfig/conf.o
SHIPPED scripts/kconfig/zconf.tab.c
SHIPPED scripts/kconfig/zconf.lex.c
SHIPPED scripts/kconfig/zconf.hash.c
HOSTCC scripts/kconfig/zconf.tab.o
HOSTLD scripts/kconfig/conf
scripts/kconfig/conf --olddefconfig Kconfig
#
# using defaults found in /boot/config-3.2.0-4-686-pae
#
/boot/config-3.2.0-4-686-pae:510:warning: symbol value 'm' invalid for ACPI_PCI_SLOT
/boot/config-3.2.0-4-686-pae:512:warning: symbol value 'm' invalid for ACPI_CONTAINER
/boot/config-3.2.0-4-686-pae:1626:warning: symbol value 'm' invalid for BMP085
/boot/config-3.2.0-4-686-pae:1816:warning: symbol value 'm' invalid for SCSI_DH
/boot/config-3.2.0-4-686-pae:3131:warning: symbol value 'm' invalid for POWER_SUPPLY
/boot/config-3.2.0-4-686-pae:3376:warning: symbol value 'm' invalid for MFD_WM8400
/boot/config-3.2.0-4-686-pae:4926:warning: symbol value 'm' invalid for COMEDI_MISC_DRIVER
S
/boot/config-3.2.0-4-686-pae:4933:warning: symbol value 'm' invalid for COMEDI_ISA_DRIVERS
#
# configuration written to .config

```

Après cette étape, j'utilise **make menuconfig** afin de pouvoir modifier manuellement certaines entrées, telles que le protocole IPv6 qui m'est inutile. Puisque j'ai eu différentes erreurs de symboles invalides après avoir utiliser **make olddefconfig**, j'ai commencé par vérifier ces symboles en les recherchant avec **/**.

```

Search Results
Symbol: ACPI_PCI_SLOT [=n]
Type : boolean
Prompt: PCI slot detection driver
Location:
-> Power management and ACPI options
(1) -> ACPI (Advanced Configuration and Power Interface) Support (ACPI [=y])
Defined at drivers/acpi/Kconfig:302
Depends on: ACPI [=y] && SYSFS [=y]

```

Le configurateur est plutôt malin puisqu'il leur a assigné automatiquement des valeurs (que j'imagine être les valeurs par défaut). Je ne les ai donc pas changé, n'ayant aucune idée de leurs utilités (et l'aide avec **?** ne m'a guère rendu service).

```

[*] TCP: MD5 Signature Option s
< > The IPv6 protocol ----
[ ] NetLabel subsystem support

```

```

Networking options --->
[ ] Amateur Radio support ----
<M> CAN bus subsystem support --

```

```

[*] Enable MPS table
[ ] Support for big SMP systems with more than 8 CPUs
[ ] Support for extended (non-PC) x86 platforms

```

J'ai enlevé le protocole IPv6, la radio amateur, l'option permettant d'avoir plus de 8 CPU. J'ai également enlevé tous les drivers de cartes et chipsets graphiques ne correspondant pas à mon laptop. J'ai également activé le logo au démarrage car je trouvais ça rigolo.

J'ai malheureusement perdu tous les screenshots suivants, mon linux n'ayant pas survécu à ce qui va suivre.

Après avoir fini la configuration, j'ai lancé la compilation avec **make -j8**, ce qui était une mauvaise idée car mon vieux laptop n'est plus ventilé correctement, il a donc surchauffé au milieu de cette opération. Après l'avoir relancé, j'ai effectué un **make clean**. J'ai ensuite posé mon laptop dans l'abri anti-atomique de la maison, où il fait approximativement 5°C et lancé **make -j2**. Cette fois la compilation s'est passée sans problème et j'ai pu l'installer avec **sudo make install** suivit de **sudo make modules\_install**.

Après avoir redémarré mon laptop, j'ai bien un nouveau kernel dans le GRUB, qui s'appelle 4.3.0. Lorsque je boot dessus, rien ne charge après le kernel. Je peux lancé une interface console avec **ctrl-alt-f1** mais la plupart des commandes ne marchent pas (par exemple ifconfig). Je pense qu'il me manque une étape où je configure le fichier de chargement de l'OS. Ce qui m'échappe est que mon ancien kernel (qui d'abord marchait niquel) a maintenant été contaminé par le même problème.

A noter qu'en mode "sans échec", j'ai accès à des commandes plus sympatiques que les commandes kernel de base, j'ai donc pu réinstallé les drivers graphiques de ma carte graphique manuellement, mais il est possible que c'est cette étape qui a cassé mon ancien kernel...

## En conclusion

D'après mon analyse, il est fort probable que mon kernel soit compilé et installé correctement. Malheureusement, il manque une étape cruciale à son utilisation, mais mes recherches sont restées infructueuses.