Compilation du noyau Linux sous Red Hat 8.0

Ce document rappelle les 5 phases permettant de personnaliser le noyau 2.4.xx sous Linux Red Hat 8.0. Les 5 phases permettant de recompiler le noyau et d'aboutir à un nouveau noyau personnalisé et parfaitement bootable sont les suivantes :

- Le téléchargement des fichiers sources
- La préparation de l'arborescence source
- La configuration des options du noyau
- La compilation du noyau et les modules
- L'installation du nouveau noyau

I – Téléchargement des sources et de leurs dépendances

Pour disposer des fichiers sources du noyau il suffit d'installer le RPM **kernel-source** avec toutes ses dépendances, soit à partir des CD-ROM dès l'installation de Linux (noyau 2.4.18), soit en le téléchargeant sur Internet (noyau 2.4.20 actuellement), soit en utilisant une sauvegarde des RPM téléchargés antérieurement.

Pour télécharger sur Internet via RHN les sources du noyau avec toutes leurs dépendances et les enregistrer dans le répertoire **/var/spool/up2date** :

```
up2date –d –f kernel-source
```

Pour installer les sources du noyau avec toutes leurs dépendances à partir du répertoire **/var/spool/up2date** :

```
up2date –f kernel-source
```

Pour installer les sources du noyau avec toutes leurs dépendances à partir d'un répertoire particulier (le répertoire **/home/rpm** dans l'exemple suivant) :

```
up2date –f kernel-source –k /home/rpm
```

Après l'installation du package **kernel-source**, les sources du noyau se trouvent dans le répertoire **/usr/src/linux-2.4/**

II – Préparation de l'arborescence source

Toutes les commandes **make** suivantes se tapent dans le répertoire **/usr/src/linux-2.4/** en tant que **root**.

Il est important de commencer la construction d'un noyau avec l'arborescence source en situation connue. C'est pourquoi il est conseillé de commencer par la commande suivante :

```
make mrproper
```

Celle-ci supprime tous les fichiers de configuration en même temps que les restes de toute construction précédente qui pourraient avoir été éparpillés autour de l'arborescence source.

III – Configuration des options du noyau

Les options de configuration du noyau seront enregistrées dans le fichier **/usr/src/linux-2.4/.config**

Pour créer ce fichier de configuration il y a 4 solutions au choix :

make config : crée le fichier de configuration **.config** en utilisant un programme en mode texte interactif. Les composants sont présentés et on répond par **Y** (yes), **N** (no), ou **M** (module).

make menuconfig : utilise une interface utilisateur en mode semi-graphique, qui utilise la bibliothèque **Ncurses**. Pour utiliser cette interface semi-graphique, il faut installer le RPM **ncurses-devel**.

make xconfig : utilise une interface utilisateur en mode graphique sous X Windows pour configurer le noyau.

make oldconfig : crée le fichier **.config** en utilisant les options par défaut du noyau : on obtiendra alors après compilation un noyau identique au noyau fourni lors de l'installation de Linux.

Remarque : l'option **oldconfig** est généralement utilisée pour tester la première compilation, afin d'être sûr que le fichier **.config** ne contient pas d'erreur, et que les options du noyau sont toutes compatibles entre elles. Pour personnaliser le noyau, on préfèrera l'option :

- **xconfig** si le serveur X est installé
- Sinon **menuconfig** si le RPM **ncurses-devel** est installé
- Sinon **config**

A l'issue de la configuration des options du noyau, et quelque soit le méthode utilisée parmi les 4 méthodes possibles [**config**, **menuconfig**, **xconfig**, ou **oldconfig**], on option un fichier **.config** dans le répertoire **/usr/src/linux-2.4**

Une fois que le fichier **.config** est créé, pour finir de préparer l'arborescence source il faut lancer les 2 commandes suivantes :

1 – make dep : recherche et gère les dépendances, c'est-à-dire que si un pilotes a besoin d'un autre pilotes pour fonctionner, make dep se charge d'inclure celui-ci automatiquement.

2 – make clean : détruit tous les fichiers objets et autres fichiers qui doivent être recompilés

IV – Compilation du noyau et des modules

Cette phase est de loin la plus longue. Elle se déroule en 2 étapes :

1 – compilation du noyau : **make bzImage** : crée un fichier noyau compressé nommé **bzImage** dans le répertoire **/usr/src/linux-2.4/arch/i386/boot**

2 – compilation des modules : **make modules**

V – Installation du nouveau noyau

Cette phase modifie automatiquement le fichier de configuration **/boot/grub/grub.conf** de Grub. Elle se déroule en 2 étapes :

1 – installation des modules : **make modules_install**

2 – installation du noyau : **make install**

Remarques diverses :

- Grâce à **make install**, aucun fichier n'est à copier manuellement dans le répertoire **/boot** : ni le fichier du noyau **bzImage**, ni le fichier **initrd** correspondant (qui est généré automatiquement), ni le fichier **system.map**

- Pour créer manuellement un fichier **initrd** correspondant à un noyau particulier, il faut utiliser la commande **mkinitrd** (en principe inutile si on utilise une installation automatique avec **make install**) en lui passant en paramètre le nom du fichier **initrd** à générer ainsi que la version exacte du noyau. Exemple :

```
mkinitrd toto.img 2.4.20-28.8
```

- Après la configuration du noyau, toutes les commandes make peuvent se concaténer en une seule ligne de commande :

```
make dep clean bzImage modules modules_install install
```

- Pour créer rapidement une disquette de secours bootable afin de démarrer la machine en cas de problème avec Grub ou sur le MBR, on peut utiliser la commande **mkbootdisk** de Red Hat.