Mini-howto Modules

rhw@bigfoot.com
date ?

Vendredi 14 janvier 1998. traduit par Frédéric Gacquer gacquer@neuronnexion.fr

Contents

1 But de ce document

Les documents existants sur linux et les modules n'ont pas réussi à me fournir une explication satisfaisante comme par exmple la façon d'installer linux avec succès avec les modules configurés et opérationnels. La procédure expliquée dans ce document a été utilisée avec succès plusieurs fois, aussi bien sur mon propre système que sur Internet pour donner des indications à quelqu'un essayant de faire fontionner un élément qui requiert un driver fourni seulement sous forme de module.

Mon prope système a été installé à partir d'une distribution linux Redhat 4.1, et ce fut lors de cette installation que j'ai développé cette procédure. Je l'ai installée depuis avec succès sur plusieurs systèmes avec la distribution slackware, et sur un système avec une distribution Debian. La procédure qui permet de configurer correctement les modules sous linux pour ces trois distributions est dans ce document.

2 AVERTISSEMENT

J'ai récemment utilisé la même procédure avec la RedHat 4.2, mais avec des résultats contradictoires sur des systèmes apparement identiques. Je n'ai PAS encore déterminé quel est le problème, et ne peut donc PAS garantir aujourd'hui que cela va ou non fonctionner sur votre système.

3 Prérequis

Avant d'appliquer les étapes de ce document, le lecteur DOIT avoir installé linux pour que l'on puisse se connecter en tant que root, car la majorité des étapes impliquées ne peuvent être entreprises que par le dit utilisateur.

Le noyau existant peut être compilé soit pour utiliser les modules soit pour ne pas les utiliser. Il peut même afficher des messages d'erreurs pendant le démarrage du système signalent les modules configurés qui ne sont pas disponible pour l'instant, pourvu que la condition ci-dessus soit remplie.

L'emplacement du source du noyau en cours est supposée se trouver à la racine `/usr/src/linux` qui est supposé être le répertoire courant, au travers de ce document, de n'importe quelle commande qui en est issue.

4 Accélérer la compilation

Si votre machine a 16 méga ou plus de RAM, on peut accélérer utilement les choses en permettant de compiler plusieurs modules en parallèle. Cela augmentera la charge cpu de la machine pendant la recompilation du noyau mais réduira le temps pendant lequel la compilation s'effectuera.

Avant d'utiliser cette méthode, vous devez vérifier la quantité de RAM présente dans votre machine, car si vous l'avez estimée trop grande, la compilation va en fait être moins rapide. L'expérience a montrée que la valeur optimum dépend de la quantité de RAM dans votre système selon la formule suivante, au moins pour les systèmes qui ont jusqu'à 32 Mo de RAM, même si elle est applicable aux systèmes avec une plus grande quantité de RAM:

```
N= [RAM en Mo] / 8 + 1
```

Pour ceux qui n'aime pas les maths, les valeurs pour les quantités courante de RAM sont :

La valeur à utiliser avec

- 16 Mo 3
- 24 Mo 4
- 32 Mo 5
- 40 Mo 6
- 48 Mo 7
- 56 Mo 8
- 64 Mo 9
- 80 Mo 11
- 96 Mo 13
- 112 Mo 15
- 128 Mo 17

Quand vous avez choisi la valeur correcte, éditer le fichier `/usr/src/linux/Makefile` et trouver la ligne :

```
MAKE=make
```

La remplacer par:

```
MAKE=make -j N
```

où N est le nombre calculé précédemment.

5 Recompiler le noyau avec les modules

Le noyau peut être reconfiguré pour utiliser les modules pour tout sauf le système de fichier monté root (dans la plupart des cas, c'est le système de fichier ext2).

Cependant, il y a certains éléments qui semble difficile à installer proprement en tant que modules, aussi je propose de compiler dans le noyau:

- Les pilotes des cartes ethernet
- Les pilotes des CDROMs SCSI.

D'autre part, il y a certaines combinaisons de pilotes qui ne fonctionne QUE comme module, particulièrement la combinaison d'un ou plusieurs des groupes suivants :

- L'imprimante parallèle
- Un pilote de port parallèle, comme le lecteur Zip de IOMEGA ou le lecteur Jazz, ou le CDROM de chez BackPack, et
- Le démon PLIP

Vous aurez à décider de ce qui sera compilé dans le noyau, ou comme module, en prenant en compte les points cités ci-dessus. Les vrais choix seront fait durant la compilation, au moment de la séquence d'instructions suivante :

```
cd /usr/src/linux
make menuconfig
make dep clean modules modules_install zImage
```

Après cela, les dépendances de modules doivent être référencées. C'est accompli par les commandes suivantes:

```
depmod -a
```

Maintenant le nouveau noyau a besoin d'être inséré dans la séquence d'initialisation du boot. Je vais supposer que le lecteur utilise LILO dans ce but, puisque c'est le seul gestionnaire de boot que j'ai utilisé.

Je recommande que le nouveau noyau ne soit pas inséré comme étant le noyau linux par défaut car s'il plante, il sera extrêmement difficile de récupérer le setup linux sans refaire une complète réinstallation, ce qui n'est pas recommandé. Pour cette raison, j'ai les entrées suivantes dans mon fichier `/etc/lilo.conf` :

```
image=/usr/src/linux/arch/i386/boot/zImage
   label=new
   alias=n
   read-only
   vga=ask
   optional
```

Cette entrée indique qu'il y a une OPTION de boot (ignorée si l'image en question n'existe pas) qui charge le fichier `/boot/newlinux` s'il est sélectionné, et permet de sélectionner le mode vidéo dans lequel il sera exécuté.

Supposant l'existence de l'entrée ci-dessus dans `/etc/lilo.conf` le noyau modifié est sauvegardé au bon endroit à la fin de la compilation, et il peut être installé via les commandes suivantes:

```
lilo
```

Ayant fait cela, le lecteur a besoin de suivre les étapes suivantes qui dépendent de la distribution choisie :

- Configurer Debian ou la Redhat pour les modules
- Configurer la Slackware pour les modules
- Configurer les autres distributions pour les modules

5.1 Configurer Debian ou la Redhat pour les modules

Avant la prise en compte des opérations décrites ici, celles listées dans "Recompiler le noyau" pour les modules sont supposées réalisées.

La distribution Debian et la Redhat ont des procédures de boot identiques, et ont aussi des procédures identiques pour configurer les modules inclus.

1. vous étant connecté root, utiliser votre éditeur de texte favori pour créer un nouveau fichier appelé `/etc/rc.d/init.d/modules.init` avec le contenu suivant :

```
# Initialisation des modules
#
# Démarrer démon d'auto-chargement des modules
/sbin/kerneld

# Monter toutes les partitions courantes auto-montables non montées
/sbin/mount -a
```

2. Ayant créé le fichier ci-dessus, exécuter les étapes suivantes alors que vous êtes connecté root:

```
cd /etc/rc.d
chmod 755 init.d/*
cd rc3.d
ln -s ../init.d/modules.init 05modules.init
```

Le système peut être relancé, les modules sont alors implémentés.

5.2 Configurer la Slackware pour les modules

Avant la prise en compte des opérations décrites ici, celles listées dans "Recompiler le noyau" pour les modules sont supposées réalisées.

1- Le fichier `/etc/rc.d/rc.M` doit être éditer comme suit :

- Aux alentours de la ligne 18, il y a une section du style :

```
# Screen blanks after 15 minutes idle time.
/bin/setterm -blank 15
```

Juste après, insérer le paragraphe suivant, avec les traditionnelles lignes blanches au dessus et en dessous:

```
# Charger le chargeur automatique de module noyau
/sbin/kerneld
```

- A peut près 12 lignes plus loin il y a ce qui suit :

```
# if there is no /etc/HOSTNAME, fall back on this default:
Immediately prior to this, insert the following paragraph, again
with the usual blank lines either side of it:

# Mount remaining unmounted auto-mount drives.
/sbin/mount -a
```

Quand ces changements ont été fait, sauver le fichier. Aucun changement supplémentaire n'est requis pour la Slackware.

5.3 Configurer les autres distributions pour les modules

Avant la prise en compte des opérations décrites ici, celles listées dans "Recompiler le noyau" pour les modules sont supposées réalisées.

La procédure précise pour les autres distributions n'a pas été déterminée, mais il s'agit probablement d'une de celle citées plus haut. Pour déterminer laquelle, afficher le contenu du répertoire `/etc/rc.d`, comme suit:

```
cd /etc/rc.d
ls -l *.d rc.*
```

A partir du résultat, vous pouvez choisir une des trois options suivantes:

- Si cette liste inclu un répertoire nommé `init.d` et quelques répertoires dont les noms correspondent à `rc?.d` (où le point d'interrogation remplace un caractère unique), et n'inclu PAS un fichier nommé `rc.M`, cette distribution peut être configurée pour les modules en suivant la procédure pour les distributions Debian et redhat.
- Si cette liste n'inclu pas un répertoires appelé `init.d` mais inclu un fichier appelé `rc.M` alors la distribution peut être configurée pour les modules en suivant la procédure pour la distribution slackware.
- Si à cette liste ne correspond aucun des critères ci-dessus, la distribution à un script de boot qui n'est pas présenté dans cet Howto. Dans ce cas, vous êtes invités à contacter l'auteur de ce document pour conseils.

6 Copyright et Loi

L'auteur peut être contacté par email à *rhw@bigfoot.com*