Mini-HOWTO Machines Multi-Ethernet

Don Becker, becker@cesdis.gsfc.nasa.gov 5 août 1995

Ce mini-Howto explique comment configurer Linux pour qu'il reconnaisse plusieurs cartes Ethernet sur une même machine.

Contents

1 Introduction

Dans le cas de la plupart des distributions classiques de Linux, il suffit d'ajouter la ligne suivante au début de votre fichier `/etc/lilo.conf` et de relancer `lilo` :

```
append = "ether=0,0,eth1"
```

C'est tout ce que vous avez à faire. Lors du prochain redémarrage de la machine, Linux devrait reconnaître la seconde carte.

2 Les différentes solutions

Par défaut, le noyau Linux ne recherche qu'une seule carte Ethernet, et ne va pas plus loin dès qu'il en a trouvé une. Il y a trois façons de contraindre le noyau à rechercher des cartes supplémentaires. Par ordre de simplicité (et de souplesse) croissante :

- fournir des paramètres au noyau lors du démarrage ;
- configurer le chargeur pour qu'il fournisse lui-même systématiquement ces paramètres au noyau ;
- modifier les tables de détection des cartes Ethernet du noyau dans le fichier `drivers/net/Space.c` (et recompiler le noyau après coup).

Dans la plupart des cas, c'est la deuxième solution qui convient le mieux, et correspond à ce que nous avons décrit en introduction. Les deux premières solutions reposent sur le passage de paramètres au noyaux et sont décrites dans la section suivante. La troisième solution est décrite ensuite.

3 Transmettre des paramètres au noyau

Le noyau Linux admet qu'on lui fournisse un certain nombre de paramètres lors de son lancement. Le plus souvent ces paramètres décrivent des aspects de la configuration qui ne peuvent être déterminés qu'au moment du démarrage. Pour les cartes réseaux, le paramètre est le suivant :

```
ether=IRQ,adresse-E/S,param1,param2,nom
```

Les valeurs numériques admises peuvent être exprimées en décimal, en octal (précédées par un '0') ou en hexadécimal (précédées par '0x'). Le premier argument qui n'est pas une valeur numérique est pris comme *nom* du périphérique (ici une carte réseau). Les paramètres vides (entre virgules) ont zéro comme valeur par défaut, et les paramètres manquants avant le nom ne sont pas modifiés.

`IRQ`

Ce paramètre indique l'IRQ (ligne d'interruption) à configurer (pour les cartes admettant un paramétrage logiciel de l'IRQ) ou à utiliser (pour celles où l'IRQ est configurée avec des cavaliers sur la carte). Une valeur nulle (0) indique de demander à la carte quelle IRQ utiliser (si elle le permet) ou d'utiliser l'autoIRQ si la carte ne le permet pas.

`adresse-E/S`

Ce paramètre indique l'adresse d'entrée/sortie à tester. Une valeur nulle (0) demande le test de toutes les adresses d'entrée/sortie raisonnables. Celles-ci sont déterminées d'après une carte des zones d'entrée/sortie habituelles pour les différents types de périphérique. Cette carte des zones est ignorée si une adresse d'entrée/sortie est spécifiée. Utilisé avec le paramètre `reserve=`*`base`*`,`*`taille`*`,...`[1] ceci permet d'empêcher l'auto-test d'une zone d'entrée/sortie par d'autres pilotes et d'éviter ainsi le dysfonctionnement d'un périphérique qui se trouverait pertubé par ces tests.

`param1,param2`

Au départ, ces paramètres permettaient d'indiquer l'adresse d'une zone de mémoire partagée pour les cartes qui utilisaient cette technique, comme la WD8013. Leur utilisation a ensuite été étendue à la transmission d'autres informations propres aux différents types de cartes.

`nom`

Ce paramètre indique le nom d'un périphérique prédéfini. Le noyau standard définit ainsi au moins `"eth0"`, `"eth1"`, `"eth2"` et `"eth3"`. D'autres noms peuvent être prédéfinis (pour PPP, SLIP, etc.) mais ils ont une sémantique différente (pour toute précision, se reporter aux FAQ et HOWTO correspondants).

Deux méthodes peuvent être utilisées pour fournir ces paramètres au noyau Linux lors de son lancement. La méthode habituelle est de les indiquer directement après le nom de l'image noyau à charger. L'exemple suivant permet de tester les quatre emplacements possibles :

```
linux ether=0,0,eth1 ether=0,0,eth2 ether=0,0,eth3
```

Pour éviter d'avoir à taper ceci à chaque démarrage, il est plus pratique de configurer votre chargeur.

[1] Se reporter à la documentation *Lilo*.

3.1 Configurer votre chargeur

Il est supposé dans ce qui suit que vous utilisez le chargeur Linux standard *Lilo*.

Il est bien évidemment pénible d'avoir à taper une série de paramètres lors de chaque démarrage, et de plus cela empêcherait tout redémarrage involontaire de s'effectuer correctement[2]. L'ajout d'une ligne `append` à votre fichier de configuration *Lilo* (`/etc/lilo.conf`) vous permet de fournir automatiquement ces paramètres au noyau (n'oubliez pas de relancer `lilo` pour mettre à jour votre configuration).

```
append = "ether=0,0,eth1 ether=0,0,eth2 ether=0,0,eth3"
```

Cet exemple est équivalent au précédent (test des quatre emplacements), en utilisant cette fois *Lilo* pour transmettre à chaque démarrage ces paramètres au noyau.

4 Modifier le noyau

Si vous pouvez configurer votre système sans toucher au code source du noyau, nous vous recommandons fortement de faire ainsi (cf. supra). Il est difficile de garder une trace d'une modification apportée au code source et cela complique grandement les mises à jour du noyau. Toutefois, cela s'impose dans les situations suivantes :

- lorsque vous avez besoin de plus de quatre cartes (seules `eth0` à `eth3` sont définies dans le source `drivers/net/Space.c`) ;
- vous devez resteindre les types de périphériques recherchés à un sous-ensemble précis de types de cartes quand, par exemple, le mécanisme de détection confond des types de cartes différents ;
- quand vous voulez utiliser un nom de périphérique différent de `eth`x.

5 Notes sur la détection de quelques cartes particulières

5.1 Cartes Lance/PCNET

Le pilote Lance a besoin de tampons DMA en mémoire basse, ce qui fait que la procédure de détection des cartes Lance est spécifique à ce type de cartes, et effectuée avant la détection des autres périphériques réseaux. L'avantage est que les cartes Lance multiples sont automatiquement détectées par cette procédure, l'inconvénient est que le pilote Lance ignore (pour le moment) les paramètres *Lilo* telle l'IRQ.

5.2 La 3C509 en mode ISA

La 3C509 présente la caractéristique unique de permettre une détection vraiment sûre par le bus ISA. C'est une caractéristique intéressante, mais malheureusement pour les situations qui nous intéressent ici, cela ne fait pas très bon ménage avec les autres mécanismes de détection.

Le problème le plus important est qu'il est difficile de savoir quelle carte sera reconnue en premier, l'ordre dépendant de l'adresse Ethernet des cartes. Cela signifie que la carte avec l'adresse la plus basse se verra affectée à `eth0`, et ainsi de suite. Si la carte correspondant à `eth0` est retirée, toutes les autres cartes voient leur nom de périphérique décalé d'une unité vers `eth0`.

[2]Bien que ce type de redémarrage ne se produise pas sous Linux `;-)` (*N.D.T.*).

Un problème lié est qu'il n'est pas possible de laisser une première carte inactive, ou une carte active à une adresse ou à une IRQ différentes de celles indiquées dans l'EEPROM, ou encore de configurer une carte à une adresse spécifique.

5.3 La 3C579 EISA et la 3C509 en mode EISA

Les noyaux de version antérieure à la 1.1.25 ne détecteront pas correctement les cartes multiples en mode EISA. Si plusieurs périphériques `eth`x sont indiqués, la *même* carte 3C509 sera détectée plusieurs fois. La solution est de spécifier l'adresse d'entrée/sortie directement. Les noyaux de version ultérieure détecteront correctement plusieurs cartes en mode EISA, et détecteront aussi des cartes en mode ISA supplémentaires, une fois toutes les adresses potentielles de cartes en mode EISA testées.

Don Becker, `becker@cesdis.gsfc.nasa.gov`