The Linux BootPrompt-HOWTO

Par Paul Gortmaker. v1.14, 1er Février 1998

Ce document est le BootPrompt-Howto, qui est un condensé de tous les paramètres de boot qui peuvent être transmis au noyau de **Linux** lors de la séquence de boot. Ceci inclut tous les paramètres concernant les périphériques. Une partie traitant de la façon dont le noyau trie les paramètres de démarrage ainsi qu'un tour d'horizon des logiciels les plus répandus pour démarrer le noyau de **Linux** sont aussi inclues. **Cette version française a été réalisée par** *Laurent RENAUD*(`lrenaud@hol.fr`).

Contents

1 Introduction

Le noyau a une capacité limitée pour accepter des informations au moment du démarrage sous la forme d’une ligne de commande, semblable à une liste d’arguments que vous pouvez passer à un programme. En général, ceci est utilisé pour donner au noyau des informations concernant les paramètres du matériel que le noyau n’est pas capable de déterminer tout seul, ou pour se substituer/écraser les valeurs que le noyau pourrait détecter.

Cependant, si vous avez juste copié une image du noyau directement sur une disquette, (c.a.d `cp zImage /dev/fd0`) alors vous n’avez aucune chance de pouvoir spécifier quelque argument que ce soit à ce noyau. C’est pourquoi beaucoup d’utilisateurs de **Linux** utilisent des logiciels comme *LILO* ou *loadlin* qui se chargent de transmettre ces arguments au noyau, et de le faire alors démarrer.

NOTE IMPORTANTE POUR LES UTILISATEURS DE MODULES : Les paramètres de démarrage en général, ne s’appliquent qu’aux pilotes de matériel qui sont compilés directement dans le noyau. Ils n’ont *aucun effet* sur les pilotes qui sont chargés en tant que modules. La plupart des distributions utilisent des modules. Si vous ne savez pas, regardez dans `man depmod` et `man modprobe` en suivant le contenu de `/etc/conf.modules`.

Cette version couvre les distributions du noyau jusqu’à la v2.0.33 incluse. Des informations qui font partie des noyaux en développement jusqu’à la version 2.1.84 sont aussi documentées.

Le BootPrompt-Howto est edité et mis à jour par :

Paul Gortmaker, `gpg109@rsphy1.anu.edu.au`

[Notez que les paramètres de démarrage qui sont spécifiques aux ports et périphériques non-i386 (ex : Atari/Amiga) ne sont actuellement pas documentés.]

1.1 Responsabilité et Copyright

Ce document *n’est pas* l’évangile ! Bien que ce soit probablement la source d’information la plus à jour que vous puissiez trouver. Personne n’est responsable de ce qui peut arriver à votre matériel à part vous. Si votre matériel s’enflamme brusquement (ce qui est quasiment impossible !) je ne suis pas responsable. C’est à dire QUE L’AUTEUR N’EST PAS RESPONSABLE DES DOMMAGES QUI PEUVENT ETRE PRODUITS PAR DES ACTIONS RESULTANT D’INFORMATIONS CONTENUES DANS CE DOCUMENT.

Si vous avez l’intention d’incorporer ce document au sein d’une publication, merci de me contacter, et je ferai un effort pour m’assurer que vous avez les informations les plus à jour disponibles. Par le passé, des versions périmées de HOWTO ont été publiées, ce qui a attristé les developpeurs qui ont été harcelés de questions auxquelles ils avaient déjà répondu dans des versions plus récentes.

1.2 Documentation Associée

Les documentations les plus à jour seront toujours les sources du noyau. Pas si vite ! Ne soyez pas effrayés. Vous n'avez pas besoin de connaître la programmation pour lire les commentaires dans les fichiers source. Par exemple, si vous recherchez un argument qui peut être transmis au pilote AHA1542 SCSI, il vous suffit d'aller dans le répertoire `linux/drivers/scsi`, et de regarder dans le fichier `aha1542.c` et dans les cent premières lignes vous trouverez en anglais une description simple et complète des paramètres de démarrage que le pilote 1542 peut recevoir.

Une autre bonne chose seront les fichiers de documentation livrés avec le noyau lui-même. Il y en a aujourd'hui pas mal, et la plupart d'entre eux peuvent-être trouvés dans le répertoire `linux/Documentation` et tous ses sous répertoires. Le répertoire `linux` se trouve généralement dans `/usr/src/`. Parfois des fichiers `README.foo` peuvent se trouver dans le répertoire associé aux pilotes (c.a.d. `linux/drivers/XXX/`, où `XXX` sera `scsi`, `char`, ou `net`.

Si vous avez trouvé quels sont les paramètres que vous avez l'intention d'utiliser, et que vous voulez savoir comment transmettre ces informations au noyau, alors regardez la documentation qui correspond au logiciel que vous utilisez pour démarrer le noyau (par exemple : LILO ou loadlin). Un bref survol est fourni ci-dessous, mais il ne remplace pas la documentation fournie avec le logiciel de démarrage.

1.3 Le groupe de discussion Linux

Si vous avez des questions sur la transmission des paramètres au noyau, s'il vous plait, LISEZ D'ABORD ce document. Si ce document et les documents associés qui sont mentionnés ci-dessus ne répondent pas à votre (vos) question(s), alors vous pouvez essayer de la (les) poser dans le groupe de discussion **Linux** (fr.comp.os.linux pour la France). Bien sûr, il serait bon de lire les messages du groupe avant de poser aveuglément vos questions, il se peut que quelqu'un d'autre ait déjà posé la même question, ou peut-être est-ce une question fréquemment posée (FAQ). Un coup d'oeuil rapide à la FAQ linux avant de poster est une *bonne* idée. On pourra trouver les FAQ quelque part, dans un répertoire proche de celui où vous avez trouvé ce document.

Les questions générales concernant la configuration de votre système peuvent être directement posées dans le groupe comp.os.linux.setup. Nous vous demandons *s'il vous plaît* de respecter ces quelques recommandations, et de ne pas cross-poster vos demandes dans d'autres groupes.

1.4 Nouvelles Versions de ce Document

Les nouvelles versions (en anglais) de ce document peuvent être recupérées par FTP anonyme sur le site sunsite.unc.edu, dans le répertoire `/pub/Linux/docs/HOWTO/`. Notez que *SunSITE* est souvent surchargé, donc il vaudrait mieux aller chercher ce document sur un des sites ftp miroir de Linux.

Ces documents en langue française se trouvent sur le site ftp.lip6.fr dans de répertoire `/pub/linux/french/docs/HOWTO`.

Des mises à jour seront faites chaque fois que de nouvelles informations / pilotes seront disponibles. Si la copie que vous êtes en train de lire date de plus de quelques mois, il serait bon de vérifier qu'il n'en existe pas une version plus récente.

Ce document est produit en utilisant le système SGML spécialement concu pour le projet **Linux** Howto, et il existe différents formats de sortie disponibles : postscript, dvi, ascii, html, et bientôt TeXinfo.

Je vous recommande de visualiser ce document en HTML (via un logiciel de navigation WWW) ou dans le format PostScript/dvi. Tous deux contiennent les références croisées qui sont perdues dans les conversions en ASCII.

Si vous voulez obtenir la copie officielle de sunsite, voici l'URL.

BootPrompt-HOWTO `http://sunsite.unc.edu/mdw/HOWTO/BootPrompt-HOWTO.html`

2 Vue d'Ensemble des Paramètres de Démarrage

Cette partie donne un certain nombre d'exemples de logiciels qui peuvent être utilisés pour transmettre les paramètres de démarrage au noyau. Elle donne aussi une idée de la façon dont les paramètres sont traités, quelles sont les limitations des paramètres de démarrage, et la façon dont ils sont répartis vers chaque périphérique pour lesquels ils ont été conçus.

Il est *important* de noter que l'on *ne peut pas* utiliser d'espaces dans un paramètre de démarrage, mais seulement entre des paramètres différents. Une liste de valeurs correspondant à un seul paramètre doit utiliser des virgules comme séparateur entre les différentes valeurs, là aussi, sans aucun espace. Voir les exemples ci-dessous.

```
ether=9,0x300,0xd0000,0xd4000,eth0  root=/dev/hda1           *BON*
ether = 9, 0x300, 0xd0000, 0xd4000, eth0  root = /dev/hda1    *MAUVAIS*
```

2.1 LILO (LInux LOader)

Le programme LILO (LInux LOader) écrit par Werner Almesberger est le plus couramment utilisé. Il a la capacité de démarrer différents noyaux, et stocke les informations de configuration dans un fichier contenant exclusivement du texte. Beaucoup de distributions fournissent LILO comme "boot-loader" (chargeur de noyau) par défaut. LILO peut démarrer DOS, OS/2, **Linux**, FreeBSD, etc. sans aucun problème, et il est très souple.

Une configuration classique est d'avoir LILO qui arrête le démarrage et affiche `LILO:` peu de temps après que vous ayez allumé votre ordinateur. Il attendra alors quelques instants en vue d'une eventuelle saisie de l'utilisateur, faute de quoi il lancera le système d'exploitation par défaut. Les étiquettes couramment utilisées dans les fichiers de configuration de LILO sont `linux` , `backup` et `msdos`. Si vous désirez entrer un paramètre de démarrage, vous le taperez ici, après avoir entré l'étiquette du système que vous voulez que LILO lance, comme indiqué dans l'exemple ci-dessous.

```
LILO: linux root=/dev/hda1
```

LILO est fourni avec une documentation excellente, et pour les paramètres de démarrage dont nous parlons ici, la commande `append=` de LILO est d'une très grande importance lorsque l'on veut ajouter un paramètre de démarrage de façon permanente dans le fichier de configuration de LILO. Vous ajoutez tout simplement quelque chose comme `append = "foo=bar"` dans le fichier `/etc/lilo.conf`. On peut l'ajouter soit en haut du fichier de configuration, afin qu'il s'applique à toutes les sections, ou dans une section correspondant à un système particulier en le mettant dans une section `image=`. Voyez la documentation de LILO pour une description plus complète.

2.2 LoadLin

L'autre chargeur de noyau couramment utilisé est 'LoadLin' qui est un programme DOS qui est capable de lancer un noyau **Linux** à partir du prompt du dos (avec des paramètres de démarrage) en supposant que certaines ressources sont disponibles. Ceci est très bien pour les gens qui utilisent le DOS et qui veulent basculer sur **Linux** à partir du DOS.

C'est aussi très pratique si vous possédez du matériel qui est dépendant du pilote fourni pour le DOS afin de mettre le matériel dans un état donné. Un exemple fréquent : les cartes son 'SoundBlaster Compatible' qui requièrent un pilote DOS pour positionner un ensemble de registres propriètaires pour mettre la carte dans un mode compatible SoundBlaster. Démarrez le DOS avec le pilote requis, et maintenant chargez **Linux** à partir du prompt du DOS avec `LOADLIN.EXE` en esquivant la remise à zéro de la carte qui intervient si on redémarre complètement la machine. De cette façon, la carte est laissée dans le mode compatible SB et par conséquent est utilisable sous **Linux**.

Il y a aussi d'autres programmes qui peuvent être utilisés pour démarrer **Linux**. Pour une liste complète, regardez sur votre miroir ftp **Linux** local, les programmes disponibles dans le répertoire `system/Linux-boot/`.

2.3 L'utilitaire "rdev"

Un certain nombre des paramètres de démarrage du noyau ont leurs valeurs par défaut stockées dans différents octets de l'image du noyau. Il existe un utilitaire baptisé `rdev` qui est installé sur la plupart des systèmes et qui sait où sont ces valeurs, et comment les changer. Il peut aussi modifier un certain nombre de choses qui ne possèdent pas de paramètre de démarrage équivalent, comme le mode vidéo utilisé par défaut.

L'utilitaire rdev est couramment associé à swapdev, ramsize, vidmode et rootflags. Les cinq paramètres que rdev peut modifier sont : le périphérique de démarrage, le périphérique de swap, les paramètres du disque RAM, le mode vidéo par défaut, et l'autorisation de lecture-seule/lecture-écriture sur le périphérique racine.

Des informations plus complètes sur `rdev` peuvent être obtenues en tapant `rdev -h` ou en lisant la page correspondante du manuel fourni (`man rdev`).

2.4 Comment le noyau gère t-il les paramètres ?

La plupart des paramètres de démarrage utilisent la syntaxe suivante :

```
nom[=valeur_1][,valeur_2]...[,valeur_11]
```

où 'nom' est un mot clé unique qui est utilisé pour reconnaître à quelle partie du noyau sont destinées les valeurs associées (si il y en a). Plusieurs paramètres de démarrage peuvent être transmis sous forme d'une liste d'éléments, comme celle situé ci-dessus, séparés par des espaces. Notez que la limite de 11 paramètres est réelle, c'est pourquoi le code ci-dessus ne comporte que 11 paramètres séparés par des virgules pour un mot clé. Toutefois, vous pouvez réutiliser le même mot clé avec 11 paramètres de plus dans des situations très complexes, en sachant que ceci est accepté par la fonction de configuration. Notez aussi que le noyau partage la liste en un maximum de 10 paramètres entiers, et une chaîne de caractères accompagnatrice, donc vous pouvez réellement fournir 11 entiers, dans la mesure ou vous assurez la conversion du 11ème paramètre, de chaîne en entier, dans le pilote lui même.

La plupart sont pris en charge par `linux/init/main.c`. Tout d'abord, le noyau cherche à voir si le paramètre fait partie des paramètres spéciaux comme 'root=', 'ro', 'rw', ou 'debug'. La signification de ces paramètres spéciaux est décrite plus loin dans ce document.

Il parcourt alors une liste de fonctions de configuration (contenues dans le tableau `bootsetups`) pour voir si la chaîne paramètre spécifiée (comme par exemple 'foo') a été associée à une fonction de configuration (`foo_setup()`) pour un périphérique particulier ou une partie du noyau. Si vous passez au noyau la ligne `foo=3,4,5,6,bar` alors, il cherchera dans le tableau `bootsetups` pour voir si 'foo' y figure. S'il y est, alors il pourra appeler la fonction de configuration associée à 'foo' (`foo_setup()`) et prendra en charge les paramètres 3, 4, 5 et 6 tels qu'ils sont donnés dans la ligne de commande adressée au noyau, et traitera aussi le paramètre de type chaîne `bar`.

2.5 Positionnement des Variables d'Environnement.

Quelque chose du type 'foo=bar', qui n'est pas accepté comme une fonction de configuration telle qu'elle est décrite ci-dessus, est interprétée comme une variable d'environnement à positionner. Un exemple (inutile ?) serait d'utiliser 'TERM=vt100' comme paramètre de démarrage.

2.6 Passer des paramètres au programme 'init'

Tous les paramètres restants qui ne sont pas pris par le noyau et qui ne sont pas considérés comme étant des variables d'environnement sont transmis au processus initial, qui est généralement le programme `init`. Le paramètre le plus couramment passé au processus `init` est le mot *single* qui demande à `init` de démarrer l'ordinateur en mode mono-utilisateur, et de ne pas lancer les "daemons" (démons) habituels. Regardez la page du manuel correspondant à la version de `init` installée sur votre système, afin de connaître les paramètres acceptés.

3 Paramètres Généraux non spécifiques à un Périphérique

Voici des paramètres qui ne sont pas liés à des périphériques particuliers. Ils sont simplement liés à un certain nombre de paramètres internes au noyau, comme la gestion mémoire, celle du disque RAM, celle du système de fichiers racine, etc.

3.1 Options du système de fichiers racine

Les options suivantes déterminent toutes la façon dont le noyau sélectionne et manipule le système de fichiers racine.

3.1.1 Le paramètre 'root='

Ce paramètre indique au noyau quel périphérique doit être utilisé comme "root filesystem" (racine du système de fichiers) pendant le démarrage. Par défaut, c'est le périphérique racine du système sur lequel le noyau a été construit. Par exemple, si le noyau en question a été construit sur un système qui utilise '/dev/hda1' comme partition racine, alors le périphérique racine par défaut sera '/dev/hda1'. Pour outrepasser cette valeur et sélectionner le second lecteur de disquette comme périphérique racine, il faut utiliser 'root=/dev/fd1'. Les périphériques racine valides sont un des périphériques suivants :

(1) /dev/hdaN à /dev/hddN, où N est la partition pour les disques 'a à d' compatibles ST-506.

(2) /dev/sdaN à /dev/sdeN, où N est la partition pour les disques 'a à e' compatibles SCSI.

(3) /dev/xdaN à /dev/xdbN, où N est la partition pour les disques 'a à b' compatibles XT.

(4) /dev/fdN, où N est le numéro du lecteur de disquette. La valeur N=0 correspond au disque DOS 'A:', et N=1 correspond à 'B:'.

(5) /dev/nfs, qui n'est pas vraiment un périphérique, mais plutôt un indicateur pour dire au noyau de rechercher le système de fichiers racine via le réseau.

La plus maladroite et la moins compatible des spécifications des périphériques disque ci-dessus, qui est le format nombre majeur/nombre mineur est aussi acceptée (par exemple /dev/sda3 a pour major 8, et pour minor 3, vous pouvez donc utiliser `root=0x803` comme alternative).

C'est un des paramètres de démarrage qui a sa valeur par défaut stockée dans l'image du noyau, et qui peut être aussi modifiée par l'utilitaire `rdev`.

3.1.2 Le paramètre 'ro'

Quand le noyau démarre, il a besoin du système de fichiers racine, pour énumérer les éléments de base de celui-ci. C'est le système de fichiers racine qui est monté au démarrage. Cependant, si le système de fichiers racine est monté avec un accès en écriture, vous ne pourrez pas contrôler de façon fiable l'intégrité du système de fichiers, car il peut y avoir des fichiers en cours d'écriture. L'option 'ro' indique au noyau de monter le système de fichiers racine en lecture seule, de façon que les programmes de contrôle de cohérence du système de fichiers (fsck) puissent être certain qu'il n'y a pas d'écritures en cours pendant la durée du test. Aucun programme ou processus ne peut écrire dans les fichiers situés sur le système de fichiers en question jusqu'à ce qu'il ait été 'remonté' avec un accès en lecture/écriture.

C'est un des paramètres de démarrage qui a sa valeur par défaut stockée dans l'image du noyau, et qui peut être aussi modifiée par l'utilitaire `rdev`.

3.1.3 Le paramètre 'rw'

Ceci est le contraire le plus parfait de ce qui précéde, c'est à dire que ce paramètre indique au noyau de monter le système de fichier racine en lecture/écriture. N'exécutez surtout pas un programme de type 'fsck' sur un système de fichiers monté en lecture/écriture.

La même valeur stockée dans le fichier image mentionné ci-dessus est aussi accessible via `rdev`

3.2 Options liées à la gestion des disques virtuels (disques RAM)

Les options suivantes correspondent à la façon dont le noyau gère le périphérique disque virtuel, qui est souvent utilisé comme zone d'amorçage durant la phase d'installation, ou pour des machines qui utilisent des pilotes modulaires qui doivent être installés pour accéder au système de fichiers racine.

3.2.1 Le paramètre 'ramdisk_start='

Pour permettre à une image du noyau de loger sur une disquette, conjointement avec une image compressée du disque virtuel, la commande 'ramdisk_start=<offset>' est ajoutée. Le noyau ne peut pas être inclus dans l'image compressée du système de fichiers du disque virtuel, car il doit être stocké à partir du bloc zéro de façon à ce que le BIOS puisse charger le secteur d'amorce (bootsector) et que le noyau puisse alors s'auto-lancer.

Note : Si vous utilisez une image du disque virtuel non compressée, alors le noyau peut faire partie de l'image du système de fichiers qui est chargé sur le disque virtuel, et la disquette peut-être lancée avec LILO, ou les deux peuvent être distincts comme c'est fait pour les images compressées.

Si vous utilisez deux disques boot/root (noyau sur le disque 1, image u disque virtuel sur le disque 2) alors, le disque virtuel démarrera au bloc zéro, et un déplacement (offset) de zéro sera utilisé. Etant donné que c'est la valeur par défaut, vous n'aurez pas besoin actuellement d'utiliser cette commande.

3.2.2 Le paramètre 'load_ramdisk='

Ce paramètre indique au noyau si il essaye de charger une image du disque virtuel ou pas. En spécifiant 'load_ramdisk=1' on indiquera au noyau de charger une disquette dans le disque virtuel. La valeur par défaut est zéro, ce qui signifie que le noyau n'essaiera pas de charger un disque virtuel.

Voyez le fichier `linux/Documentation/ramdisk.txt` pour une description complète des nouveaux paramètres de démarrage, et comment les utiliser. La façon dont ces paramètres peuvent être positionnés et stockés dans l'image du noyau via 'rdev' est aussi décrite.

3.2.3 Le paramètre 'prompt_ramdisk='

Ce paramètre indique au noyau si il doit ou non vous demander d'insérer la disquette contenant l'image du disque virtuel. Dans une configuration à une seule disquette, l'image du disque virtuel est sur la même disquette que le noyau qui vient juste de se charger/démarrer, et donc un message d'invite est inutile. Dans ce cas, on peut utiliser 'prompt_ramdisk=0'. Dans une configuration avec deux disquettes, vous devez avoir la possibilité de changer de disquette, et alors 'prompt_ramdisk=1' peut-être utilisé. Etant donné que c'est la valeur par défaut, on n'a pas vraiment besoin de l'indiquer.

Note Historique : Des gens sournois on l'habitude d'utiliser l'option de LILO 'vga=ask' pour stopper temporairement le démarrage et avoir ainsi une chance de pouvoir passer de la disquette boot à la disquette root.

Voyez le fichier `linux/Documentation/ramdisk.txt` pour une description complète des nouveaux paramètres de démarrage, et comment les utiliser. La façon dont ces paramètres peuvent être positionnés et stockés dans l'image du noyau via 'rdev' est aussi décrite.

3.2.4 Le paramètre 'ramdisk_size='

Bien que ce soit vrai que le disque virtuel augmente sa taille de façon dynamique, il existe une limite maximum afin qu'il n'utilise pas toute la mémoire vive (RAM) disponible et vous laisse dans une triste situation. Par défaut, la taille est de 4096 (c.a.d. 4MB) qui doit être suffisant pour la plupart des besoins. Vous pouvez écraser cette taille par défaut pour une plus grande ou une plus petite avec ce paramètre de démarrage.

Voyez le fichier `linux/Documentation/ramdisk.txt` pour une description complète des nouveaux paramètres de démarrage, et comment les utiliser. La façon dont ces paramètres peuvent être positionnés et stockés dans l'image du noyau via 'rdev' est aussi décrite.

3.2.5 Le paramètre 'ramdisk=' (obsolete)

NOTE : Ce paramètre est obsolète, et ne doit pas être utilisé exepté sur les noyaux v1.3.47 et ceux plus anciens. Les commandes que l'on peut utiliser pour les disques virtuels sont documentées ci-dessous.

Ceci indique la taille en Kilo-Octets du disque virtuel (RAM disk) que vous pouvez éventuellement utiliser. Par exemple, si vous souhaitez avoir un système de fichiers racine sur une disquette 1.44 Mo chargé sur le disque virtuel, vous devrez utiliser :

```
ramdisk=1440
```

C'est un des paramètres de démarrage qui a sa valeur par défaut stockée dans l'image du noyau, et qui peut être aussi modifié par l'utilitaire `rdev`.

3.2.6 Le paramètre 'noinitrd' (disque RAM initial)

La version v2.x du noyau et les versions plus récentes possédent la caractéristique de pouvoir avoir le système de fichiers racine initialement sur un disque virtuel, et le noyau exécute `linuxrc` sur cette image mémoire. Cette caractéristique est généralement utilisée pour permettre de charger des modules nécessaires au montage du système de fichiers racine réél (par exemple : charger les modules du pilote SCSI stockés dans l'image du disque virtuel, et alors monter le système de fichiers racine réél sur un disque SCSI).

Le paramètre 'noinitrd' actuel détermine ce qui arrive aux données initrd après que le noyau ait démarré. Lorsqu'il est indiqué, au lieu de se convertir en disque virtuel, il est accessible via `/dev/initrd`, et peut-être lu juste avant que la RAM soit libérée pour le système. Pour de plus amples détails sur l'utilisation du disque RAM initial, consultez `linux/Documentation/initrd.txt`. De plus, les versions les plus récentes `LILO` et `LOADLIN` doivent contenir des informations complémentaires très intéressantes.

3.3 Paramètres de Démarrage relatifs à la Gestion de la Mémoire.

Les paramètres suivants modifient la façon dont linux détecte ou gère la mémoire physique et virtuelle de votre système.

3.3.1 Le paramètre 'mem='

Ce paramètre vise deux objectifs : L'objectif principal est d'indiquer la quantité de mémoire installée (ou une valeur plus petite si vous désirez limiter le quantité de mémoire disponible pour linux). Le second ojectif (très utilisé) est de spécifier `mem=nopentium` qui indique au noyau de linux de ne pas utiliser les caractéristiques de la table de performance de pages de 4 MO (4MB page table performance).

L'appel initial au BIOS défini dans la spécification des PC, et qui renvoie la taille de la mémoire installée, a été conçu pour être capable de donner des tailles mémoire jusqu'à 64 Mo (Hé oui, encore une manque de prévoyance, tout comme les disques de 1024 cylindres...Pfffff). Linux utilise cet appel au BIOS au démarrage pour déterminer quelle est la quantité de mémoire installée. Si vous avez plus de 64 Mo de mémoire vive installée, vous pouvez utiliser ce paramètre de démarrage pour indiquer à Linux quelle est la quantité de mémoire dont vous disposez. Voici une citation de Linus sur l'utilisation du paramètre `'mem='`.

“Le noyau acceptera tous les paramètres 'mem=xx' que vous lui donnerez, et s'il s'aperçoit que vous lui avez menti, il plantera lamentablement tôt ou tard. Le paramètre indique la plus haute zone adressable, donc 'mem=0x1000000' signifie que vous avez 16 Mo de mémoire, par exemple. Pour une machine ayant 96 Mo de mémoire, le paramètre serait 'mem=0x6000000'.”

NOTE NOTE NOTE: certaines machines peuvent utiliser le sommet de la mémoire pour le cache du BIOS ou quelque chose d'autre, c'est pourquoi il se peut que vous n'ayez pas vraiment la totalité de ces 96 Mo comme mémoire adressable. Le contraire est aussi exact : certaines puces feront un plan de la mémoire physique couverte par la zone BIOS dans la zone située juste au dessus du sommet de la mémoire, donc le sommet de la mémoire peut être actuellement 96Mo + 384ko par exemple. Si vous indiquez à **Linux** qu'il a plus de mémoire qu'il doit en avoir actuellement, des choses plutôt désagréables vous arriveront : peut-être pas tout de suite, mais un jour sûrement.”

Notez que cet argument n'a pas besoin d'être en hexadécimal, et que les suffixes 'k' et 'M' (en majuscule ou minuscule, peu importe) peuvent être utilisés pour indiquer respectivement kilo-octets et Méga-octets (le 'k' multiplie par 10 votre valeur et le 'M' la multiplie par 20). La mise en garde exposée ci-dessus reste vraie en cela qu'une machine avec 96 Mo peut fonctionner avec `mem=97920k` mais échouer avec soit `mem=98304k` ou `mem=96M`.

3.3.2 Le paramètre 'swap='

Il permet à l'utilisateur de régler certains des paramètres de la mémoire virtuelle qui sont liés aux fichiers d'échange (swap) sur disque. Il accepte les huit paramètres suivants :

```
MAX_PAGE_AGE
PAGE_ADVANCE
PAGE_DECLINE
PAGE_INITIAL_AGE
AGE_CLUSTER_FRACT
AGE_CLUSTER_MIN
```

```
PAGEOUT_WEIGHT
BUFFEROUT_WEIGHT
```

Les utilisateurs avertis pourront jeter un coup d'oeuil au fichier `linux/mm/swap.c` et sur les données du répertoire `/proc/sys/vm`.

3.3.3 Le paramètre 'buff='

Comme le paramètre 'swap=', il permet à l'utilisateur de régler certains des paramètres relatifs à la gestion des tampons mémoire. Il accepte les six paramètres suivant :

```
MAX_BUFF_AGE
BUFF_ADVANCE
BUFF_DECLINE
BUFF_INITIAL_AGE
BUFFEROUT_WEIGHT
BUFFERMEM_GRACE
```

Les utilisateurs avertis pourront jeter un coup d'oeuil au fichier `linux/mm/swap.c` et sur les données du répertoire `/proc/sys/vm`.

3.4 Paramètres de démarrage pour les systèmes de fichiers racine NFS

Linux supporte des systèmes comme les stations de travail sans disques à condition que leur système de fichiers racine soit de type NFS (Network FileSystem ou Système de Fichiers Réseau). Ces paramètres sont utilisés pour indiquer à la station exempte de disque sur quelle machine elle doit aller chercher son système. Notez aussi que le paramètre `root=/dev/nfs` est requis. Des informations détaillées sur l'utilisation d'un système de fichiers racine NFS sont contenues dans `linux/Documentation/nfsroot.txt`. Je vous conseille de lire ce fichier, car ce qui suit est juste un résumé rapide extrait directement de ce document.

3.4.1 Le paramètre 'nfsroot='

Ce paramètre indique au noyau quelle machine, quel répertoire et quelles options NFS sont utilisées pour son système de fichiers racine. La structure du paramètre est la suivante :

```
nfsroot=[<server-ip>:]<root-dir>[,<nfs-options>]
```

Si le paramètre nfsroot n'est pas donné sur la ligne de commande, on utilisera par défaut '/tftpboot/%'. Les autres options sont les suivantes :

<server-ip> - Indique l'adresse IP du serveur NFS. Si ce champ n'est pas indiqué, l'adresse par défaut déterminée par la variable nfsaddrs (voir ci-dessous) est utilisée. Une des utilisations de ce paramètre est par exemple l'utilisation de serveurs différents pour RARP et NFS. Généralement vous pouvez le laisser à blanc.

<root-dir> - Nom du répertoire sur le serveur à monter en tant que racine. Si il y a un caractère '%' dans la chaîne, le caractère sera remplacé par la représentation ASCII de l'adresse IP du client.

<nfs-options> - Options NFS standard. Toutes les options sont séparées par des virgules. Si le champ option n'est pas indiqué, les valeurs suivantes sont utilisées par défaut :

```
port          = tel que donné par le démon portmap du serveur
rsize         = 1024
wsize         = 1024
timeo         = 7
retrans       = 3
acregmin      = 3
acregmax      = 60
acdirmin      = 30
acdirmax      = 60
flags         = hard, nointr, noposix, cto, ac
```

3.4.2 Le paramètre 'nfsaddrs='

Ce paramètre de démarrage positionne les différentes adresses qui sont nécessaires à la communication sur le réseau. Si ce paramètre n'est pas indiqué, le noyau essaie d'utiliser RARP et/ou BOOTP pour calculer ces paramètres. La structure est la suivante :

```
nfsaddrs=<my-ip>:<serv-ip>:<gw-ip>:<netmask>:<name>:<dev>:<auto>
```

<my-ip> - Adresse IP du client. Si elle est vide, cette adresse sera déterminée par RARP ou BOOTP. Le protocole utilisé dépend de ce qui a été activé pendant la configuration du noyau et sur le paramètre <auto>. Si ce paramètre n'est pas vide, ni RARP, ni BOOTP ne seront utilisés.

<serv-ip> - Adresse IP du serveur NFS. Si RARP est utilisé pour déterminer l'adresse du client et que ce paramètre N'EST PAS vide, seules les réponses du serveur spécifié seront acceptées. Pour utiliser différents serveurs NFS et RARP, indiquez votre serveur RARP ici (ou laissez le à blanc), et indiquez votre serveur NFS dans le paramètre nfsroot (voir ci-dessus). Si cette entrée est à blanc, l'adresse utilisée est celle du serveur qui répond à la requête RARP ou BOOTP.

<gw-ip> - Adresse IP d'une passerelle (gateway) si le serveur est sur un sous-réseau différent. Si cette entrée est vide, aucune passerelle n'est utilisée et le serveur est supposé être sur le réseau local, à moins qu'une valeur n'ait été reçue par BOOTP.

<netmask> - Masque de réseau pour les interfaces de réseau local. Si ce paramètre est vide, le masque de réseau est déduit de l'adresse IP du client, à moins qu'une valeur n'ait été reçue par BOOTP.

<name> - Nom du client. Si il est vide, l'adresse IP du client est utilisée en notation ASCII, sauf si une valeur a été reçue par BOOTP.

<dev> - Nom du périphérique réseau à utiliser. Si le paramètre est vide, tous les périphériques sont utilisés pour les requêtes RARP, et le premier trouvé pour BOOTP. Pour NFS, le périphérique utilisé est celui pour lequel on a reçu une réponse à RARP ou BOOTP. Si vous n'avez qu'un périphérique, vous pouvez sans aucun risque le laisser à blanc.

<auto> - Méthode à utiliser pour l'autoconfiguration. Si 'rarp' ou 'bootp' sont indiqués, le protocole spécifié est utilisé. Si la valeur est 'both' ou vide, les deux protocoles seront utilisés à condition qu'ils aient été activés durant la configuration du noyau. Utiliser 'none' signifie pas d'autoconfiguration; Dans ce cas, vous devez indiquer toutes les valeurs nécessaires dans les champs précédents.

Le paramètre <auto> peut apparaître seul comme valeur du paramètre nfsaddrs (sans tous les caractères ':' avant). Dans ce cas, l'autoconfiguration est utilisée. Toutefois, la valeur 'none' n'est pas disponible dans ce cas.

3.5 D'autres paramètres de démarrage divers

Ces différents paramètres de démarrage permettent à l'utilisateur de gérer certains paramètres internes du noyau.

3.5.1 Le paramètre 'debug'

Le noyau envoie des messages importants (et moins importants) à l'opérateur via la fonction `printk()`. Si le message est considéré comme important, la fonction `printk()` envoie une copie sur la console active, mais le transmet aussi à la fonction `klogd()` qui l'archive sur le disque. La raison pour laquelle le message est envoyé à la console et archivé sur disque, est simple : dans certaines circonstances malheureuses (par exemple une défaillance du disque) le message ne serait pas écrit sur le disque et serait perdu.

Le seuil à partir duquel un message est considéré comme important, ou ne l'est pas, est déterminé par la variable `console_loglevel`. Par défaut, l'affichage sur la console est déclenché pour tout ce qui depasse le `DEBUG` (niveau 7). Ces niveaux sont définis dans le fichier include `kernel.h`. Le fait de spécifier comme paramètre de démarrage `debug` forcera le niveau de suivi à 10, de façon que *tous* les messages du noyau apparaissent sur la console.

Le niveau de suivi de la console peut aussi être positionné pendant l'utilisation via une option du programme `klogd()`. Consultez la page du manuel correspondant à la version installée sur votre système, pour voir comment utiliser ce programme.

3.5.2 Le paramètre 'init='

Par défaut, le noyau lance le programme 'init' au démarrage, qui prend alors soin de configurer l'ordinateur pour les utilisateurs en lançant les programmes getty, les scripts 'rc' et tout le reste. Le noyau recherche d'abord `/sbin/init`, ensuite `/etc/init` (secondaire), et en dernier recours, il essaiera d'utiliser `/bin/sh` (éventuellement `/etc/rc`). Si par exemple, votre programme init est corrompu et donc stoppé vous serez en mesure de démarrer, en utilisant le paramètre de démarrage `init=/bin/sh` qui vous positionnera directement dans un shell au démarrage, vous permettant de remplacer les programmes corrompus.

3.5.3 Le Paramètre 'no387'

Certains coprocesseurs i387 ont des bogues qui apparaissent lorsqu'ils sont utilisés en mode protégé 32 bits. Par exemple, certaines puces ULSI-387 récentes, provoquent un blocage irréversible lorsqu'elles font des calculs un virgule flottante, apparemment dû à un bug dans les instructions FRSAV/FRRESTOR. L'utilisation du paramètre de démarrage 'no387' fait ignorer à **Linux** le coprocesseur mathématique s'il y en a un. Bien sûr, votre noyau doit alors obligatoirement être compilé avec l'option d'émulation du coprocesseur ! Cela peut aussi être intéressant si vous possédez une de ces *très* vielles machines 386 qui peuvent utiliser une FPU 80287, alors que **Linux** ne peut pas.

3.5.4 Le Paramètre 'no-hlt'

La famille des processeurs i386 (et les suivantes) ont une instruction 'htl' qui indique au processeur que rien ne va se produire jusqu'à ce qu'un périphérique externe (clavier, modem, disque, etc.) demande au processeur d'accomplir une tâche. Ceci permet au processeur de se mettre dans un mode 'low-power' (économie d'énergie) dans lequel il reste à l'état de zombi jusqu'à ce qu'un périphérique externe le réveille (généralement via une interruption). Certaines puces i486DX-100 récentes ont un problème avec l'instruction 'htl' qui est le suivant : elles ne peuvent pas retourner en mode opérationnel de façon fiable après que cette instruction ait été utilisée. L'utilisation de l'instruction 'no-hlt' indique à **Linux** de simplement exécuter une boucle infinie quand il n'y a rien d'autre à faire, et de *ne pas* arrêter votre processeur quand il n'y a aucune activitée. Ceci permet aux personnes qui utilisent ces puces défectueuses d'utiliser **Linux**, bien qu'ils doivent être informés du fait que le remplacement dans le cadre de la garantie est possible.

3.5.5 Le paramètre 'no-scroll'

L'utilisation de ce paramètre au démarrage désactive le défilement d'écran (scrolling) qui rend difficile l'emploi de terminaux Braille.

3.5.6 Le paramètre 'panic='

Dans le cas très désagréable d'une alerte du noyau (kernel panic), c'est à dire une erreur interne qui a été détectée par le noyau, et pour laquelle il a décidé qu'elle était suffisamment grave pour râler bruyamment et tout arrêter ; le comportement par défaut est d'en rester là jusqu'à ce que quelqu'un se penche sur le problème, visualise le message sur l'écran et redémarre la machine. Cependant, si une machine fonctionne sans surveillance dans un local isolé il peut-être souhaitable qu'il redémarre de lui-même afin que la machine revienne en ligne. Par exemple, l'utilisation de `'panic=30'` au démarrage forcera le noyau à essayer de redémarrer 30 secondes après que l'alerte du noyau se soit produite. Une valeur à zéro donne le comportement par défaut, qui est d'attendre éternellement. Notez que cette valeur d'attente peut aussi être lu et positionnée via l'interface sysctl `/proc/sys/kernel/panic`.

3.5.7 Le paramètre 'profile='

Les développeurs du noyau peuvent activer une option qui leur permet de suivre comment et ou le noyau consomme ses cycles CPU, dans le but d'augmenter ses capacités et ses performances. Cette option vous permet de positionner cet indicateur de suivi au moment du démarrage. Généralement il est positionné à deux. Vous pouvez aussi compiler votre noyau avec l'option de suivi par défaut. Dans tous les cas, il vous faudra un outil comme `readprofile.c` afin d'utiliser les données fournies par `/proc/profile`.

3.5.8 Le paramètre 'reboot='

Cette option contrôle le type de redémarrage que Linux fera lorsque vous ferez une remise à zéro de votre ordinateur (généralement via `/sbin/init` en faisant un Ctrl-Alt-Suppr). Le comportement par défaut des derniers noyaux v2.0 est de faire un redémarrage 'à froid' (c.a.d. remise à zéro complète, le BIOS comtrôle la mémoire, etc.) au lieu d'un redémarrage 'à chaud' (c.a.d pas de remise à zéro totale, pas de contrôle de la mémoire). Il a été modifié pour prendre la valeur froid par défaut depuis que cela semble fonctionner sur des matériels bon marché ou endommagés qui ne voulaient pas redémarrer lorsqu'un redémarrage à chaud était requis. Pour retrouver l'ancien comportement (c.a.d redémarrage à chaud) utilisez `reboot=w` en fait n'importe quel mot commançant par `w` fonctionnera.

Pourquoi cela pourrait-il vous ennuyer ? Certains disques incluant de la mémoire cache peuvent détecter un redémarrage à chaud, et écrire les données du cache sur le disque. Lors d'un redémarrage à froid, la carte peut-être remise à zéro, et les données stockées dans la mémoire cache seront perdues. D'autres ont signalé que des systèmes prenaient beaucoup de temps pour vérifier la mémoire, et/ou des BIOS SCSI qui étaient très long à s'initialiser lors d'un démarrage à froid, et c'est par conséquent une excellente raison pour utiliser le redémarrage à chaud.

3.5.9 Le paramètre 'reserve='

Ceci est utilisé pour *protéger* les zones des ports d'I/O des programmes de test. La syntaxe de la commande est la suivante :

```
reserve=iobase,extent[,iobase,extent]...
```

Sur certaines machines, il peut-être nécessaire d'empêcher les pilotes de périphériques de contrôler les périphériques à une certaine adresse (auto-test). Ceci peut-être nécessaire pour du matériel mal conçu qui peut provoquer un *bloquage* au démarrage (comme par exemple certaines cartes réseaux ethernet), du matériel mal reconnu, du matériel dont l'état a été modifié par un test récent, ou encore si vous ne voulez pas que le noyau initialise certains matériels.

Le paramètre de démarrage `reserve` s'attaque à ce problème en spécifiant une zone d'un port d'entrée/sortie qui n'a pas besoin d'être testée. Cette zone est "réservée" (verrouillée) dans la table d'enregistrement des ports du noyau comme si un périphérique avait déjà été trouvé dans cette zone (avec le nom `reserved`). Notons que ce mécanisme n'est pas nécessaire sur la plupart des machines. Il est indispensable d'utiliser ce paramètre uniquement en cas de problème ou dans certains cas particuliers.

Les ports d'entrée/sortie dans la zone spécifiée sont protégés contre les contrôles de périphériques qui font un `check_region()` au lieu de tester aveuglément une région d'entrée/sortie. Ceci a été introduit pour être utilisé lorsqu'un pilote plante, avec la NE2000 par exemple, ou identifie de façon incorrecte un autre périphérique comme étant le sien. Un pilote de périphérique correct ne doit pas tester une zone réservée, à moins qu'un autre paramètre de démarrage lui demande explicitement de le faire. Ceci implique que le paramètre `reserve` doit être le plus souvent utilisé avec un autre paramètre de démarrage. Par conséquent si vous spécifiez une région `reserve` pour préserver un périphérique particulier, vous devrez en général aussi spécifier de façon explicite un test pour ce périphérique. La plupart des pilotes ignorent la table d'enregistrement des ports si on leur donne une adresse spécifique.

Par exemple, la ligne de démarrage

```
reserve=0x300,32  blah=0x300
```

laisse tous les pilotes de périphériques, excepté le pilote pour 'blah', tester `0x300-0x31f`.

Comme d'habitude avec les paramètres de démarrage, il existe une limite à 11 paramètres, c'est pourquoi vous ne pouvez indiquer que 5 zones protégées par mot clé `reserve`. Plusieurs ordres `reserve` peuvent être utilisés si vous avez une requête vraiment très complexe.

3.5.10 Le paramètre 'vga='

Notez que ce n'est pas vraiment un paramètre de démarrage. C'est une option interprétée par LILO et non pas par le kernel, contrairement à tous les autres arguments. Pourtant, son utilisation est devenue si commune qu'une mention lui est réservée ici. Il peut aussi être positionné grâce à `rdev -v` ou par equivalence avec `vidmode` sur le fichier vmlinuz. Cela permet au programme de configuration d'utiliser le BIOS vidéo pour changer le mode d'écran par défaut, avant le démarrage du noyau de Linux. Les modes courants sont 80x50, 132x44, etc. Le meilleur moyen d'utiliser cette option est de demarrer avec `vga=ask`, qui vous demandera à l'aide d'une liste des différents modes que vous pourrez utiliser avec votre carte vidéo, avant de démarrer le noyau. Une fois que vous avez le nombre que vous voulez utiliser, provenant de la liste ci-dessus, vous pouvez, plus tard, le placer à la place de 'ask'. Pour plus d'informations, veuillez, s'il vous plait, regarder le fichier `linux`Documentation/svga.txt/ qui existe depuis les dernières versions du noyau. Notez que les noyaux récents (version 2.1 et supérieures) ont leur programme de configuration qui permettent de changer le mode vidéo, sous la forme d'une option, listée comme un *Support de sélection de mode vidéo* (*Video mode selection support*), donc vous devez sélectionner cette option si vous voulez cette caractéristique.

4 Paramètres de démarrage pour les Périphériques SCSI

Cette section contient une description des paramètres de démarrage qui sont utilisés pour passer des informations concernant les adaptateurs hôtes et les périphériques SCSI.

4.1 Paramètres pour les pilotes de niveau intermédiaire

Les pilotes de niveau intermédiaire prennent en charge des choses comme le disques, les CD-Roms et les bandes sans s'attacher aux spécificitées de chaque périphériques.

4.2 Nombre maximum de LUN contrôlés ('max_scsi_luns=')

Chaque périphérique SCSI peut avoir un nombre de 'sous-périphériques' qui le composent. L'exemple le plus courant est représenté par les nouveaux CD-ROM SCSI qui utilisent plus d'un disque à la fois grâce à un chargeur de CD. Chaque CD est adressable comme un 'Logical Unit Number' (LUN = Numéro d'Unité Logique) de ce périphérique multiple. Mais la plupart des périphériques comme les disques durs, les lecteurs de bandes et autres, sont des périphériques simples et on leur attribue le LUN zéro.

Le problème survient avec les périphériques à un seul LUN qui ont un mauvais microprogramme. Certains périphériques SCSI mal conçus (anciens et malheureurement nouveaux aussi) ne supportent pas d'être testés pour des LUN différents de zéro. Ils répondent en se bloquant, et peuvent aussi verrouiller tout le bus SCSI en même temps.

Les nouveaux noyaux ont une option de configuration qui vous permet d'indiquer le nombre maximum de LUN à tester. Par défaut, ils ne testent que le LUN zéro, pour éviter le problème décrit ci-dessus.

Pour spécifier le nombre de LUN à tester au moment du démarrage, il suffit d'entrer le paramètre de démarrage 'max_scsi_luns=n', où n est un nombre compris entre un et huit. Pour éviter les problèmes décrits précédemment, on peut utiliser n=1 pour éviter de perturber les périphériques défectueux.

4.3 Paramètres pour les Lecteurs de Bandes SCSI ('st=')

Certaines configurations de démarrage pour les lecteurs de bande SCSI peuvent être obtenues en utilisant ce qui suit :

```
st=buf_size[,write_threshold[,max_bufs]]
```

Les deux premiers nombres sont donnés en kilo-octets. La valeur par défaut du `buf_size` est 32 ko, et la taille maximum qui peut être donnée est la valeur ridicule de 16384 ko. La zone `write_threshold` est la valeur à laquelle le tampon est envoyé vers la bande, avec une valeur par défaut de 30ko. Le nombre maximum de tampons varie en fonction du nombre de lecteurs détectés, et a une valeur par défaut égale à deux. Voici un exemple d'utilisationnbsp;:

```
st=32,30,2
```

Des indications plus précises peuvent être trouvées dans le fichier `README.st` qui est dans le répertoire `scsi` de l'arborescence des sources du noyau.

4.4 Paramètres pour les adaptateurs SCSI

Notations utilisées dans cette section :

`iobase` Le premier port d'Entrée/Sortie que le serveur SCSI occupe. Ceux-ci sont donnés en notation hexadécimale, et sont généralement situés dans la fourchette `0x200` à `0x3ff`.

`irq` L'interruption matérielle pour laquelle la carte a été configurée. Les valeurs autorisées dépendront de la carte en question, mais seront généralement 5, 7, 9, 10, 11, 12, et 15. Les autres valeurs étant généralement utilisées pour les périphériques courants comme les disques durs IDE, les lecteurs de disquettes, les ports série, etc.

`dma` Le canal DMA (Direct Memory Access - Accès Direct à la Mémoire) Généralement appliqué aux cartes de pilotage du bus. Les cartes PCI et VLB pilotent directement le bus, et ne nécessitent pas de canal DMA ISA.

`scsi-id` L'identifiant que la carte-serveur utilise pour s'identifier elle-même sur le bus SCSI. Un certain nombre de cartes serveur vous permettront de modifier cette valeur, alors que d'autres ont cette valeur stockée de façon définitive sur la carte. La valeur par défaut la plus courante est sept, mais les cartes Seagate et Future Domain TMC-950 par exemple utilisent la valeur six.

`parity` Détermine si la carte serveur SCSI doit demander aux périphériques connectés de fournir une valeur de parité avec tous les échanges d'informations. La valeur 1 indique que la détection de parité est activée, et la valeur 0 désactive le contrôle de parité. Encore une fois, toutes les cartes ne supportent pas la sélection du contrôle de parité par les paramètres de démarrage.

4.4.1 Adaptec aha151x, aha152x, aic6260, aic6360, SB16-SCSI ('aha152x=')

Les valeurs aha font référence à des cartes et les valeurs aic font référence aux puces SCSI actuelles de ce type de cartes, y compris la Soundblaster-16 SCSI.

Le code de test de ces serveurs SCSI recherche s'il existe un BIOS installé, et s'il n'est pas présent, le test ne trouvera pas votre carte. Vous aurez alors à utiliser le paramètre de démarrage avec la syntaxe suivante :

```
aha152x=iobase[,irq[,scsi-id[,reconnect[,parity]]]]
```

Notez que si le pilote a été compilé avec l'option de recherche d'erreur activée, une sixième valeur peut être spécifiée pour fixer le niveau de recherche d'erreur.

Tous les paramètres sont décrits au début de cette section, et la valeur `reconnect` permet au périphérique de se déconnecter/reconnecter si une valeur différente de zéro est utilisée. Voici un exemple d'utilisation :

```
aha152x=0x340,11,7,1
```

Notez que les paramètres doivent être donnés dans l'ordre, ce qui signifie que si vous désirez spécifier une configuration de parité, vous devrez alors indiquer les valeurs de iobase, irq, scsi-id et reconnect aussi.

4.4.2 Adaptec aha154x ('aha1542=')

Ce sont les gammes de cartes aha154x. Les différentes cartes aha1542 ont un contrôleur de disquette i82077 en interne, tandis que les cartes de la série aha1540 n'en ont pas. Ce sont des cartes à "busmastering", (contrôle de bus) et elles ont des paramètres qui permettent d'indiquer le niveau "d'équité" qui est utilisé pour partager le bus avec les autres périphériques. Le paramètre de démarrage ressemble à ce qui suit.

```
aha1542=iobase[,buson,busoff[,dmaspeed]]
```

Les valeurs couramment utilisées pour `iobase` sont les suivantes : `0x130, 0x134, 0x230, 0x234, 0x330, 0x334`. Des clones de cartes peuvent autoriser d'autres valeurs.

Les valeurs `buson, busoff` indiquent le nombre de microsecondes pendant lesquelles la carte est prioritaire sur le bus ISA. Les valeurs par défaut sont 11 $\mu sprioritaire, et4\mu snonprioritaire, defaonqued'autrescartes(commeunecarteEthernetISALANCE)aientunechanced'$

La valeur `dmaspeed` fait référence à la vitesse (en Mo/s) à laquelle s'effectue le transfert DMA (Direct Memory Access, Mémoire à Accès Direct). La valeur par défaut est 5 Mo/s. Les nouvelles versions de ces cartes vous permettent de sélectionner cette valeur de façon logicielle alors que les anciennes cartes utilisait des cavaliers. Vous pouvez utiliser des valeurs allant jusqu'à 10 Mo/s en supposant que votre carte mère soit capable de les supporter. Expérimentez prudemment si vous utilisez des valeurs supérieures à 5 Mo/s.

4.4.3 Adaptec aha274x, aha284x, aic7xxx ('aic7xxx=')

Ces cartes peuvent recevoir un paramètre selon la syntaxe suivante :

```
aic7xxx=extended,no_reset
```

La valeur de `extended`, si elle est différente de zéro, indique que la traduction étendue pour les disques de grande capacité est activée. La valeur `no_reset`, si elle est différente de zéro, indique au pilote de ne pas réinitialiser le bus SCSI lorsqu'il configure la carte-serveur au démarrage.

4.4.4 Adaptateurs SCSI AdvanSys ('advansys=')

Le pilote AdvanSys peut accepter jusqu'à quatre adresses I/O qui seront testées pour une carte SCSI AdvanSys. Notez que ces valeurs (si elles sont utilisées) n'auront en aucun cas d'effet sur les tests EISA ou PCI. Elles sont seulement utilisées pour tester les cartes ISA et VLB. De plus, si le pilote a été compilé avec l'option de débogage activée, le niveau de détail des informations renvoyées par le débogage peut être indiqué en ajoutant un paramètre `0xdeb[0-f]`. Le `0-f` permet de faire afficher les 16 niveaux de messages de débogage.

4.4.5 Adaptateur Always IN2000 ('in2000=')

Contrairement aux autres paramètres de démarrage, le pilote IN2000 utilise des préfixes de type chaîne ASCII pour la plupart de ses paramètres entiers; Voici la liste des paramètres acceptés :

ioport:addr

- Où addr est l'adresse IO d'une carte (généralement sans mémoire morte 'ROM').

noreset

- Pas de paramètres optionnels. Evite la remise à zéro du bus SCSI au moment du démarrage.

nosync:x

- x est un masque d'octets (bitmask) ou les 7 premiers bits correspondent aux 7 périphériques SCSI possibles (bit 0 pour le périphérique #0, etc). Positionnez un bit pour PREVENIR une négociation de synchronisation sur ce périphérique. Par défaut sync est DESACTIVE sur tous les périphériques.

period:ns

- ns est la durée minimum en nanosecondes d'une période de transfert de données en SCSI. La valeur par défaut est 500; les valeurs doivent être comprises entre 250 et 1000.

disconnect:x

- x = 0 pour ne jamais autoriser les déconnexions, 2 pour toujours les autoriser. x = 1 fait des déconnexions 'selon le besoin', ce qui est la valeur par défaut et généralement le meilleur choix.

debug:x - Si 'DEBUGGING_ON' est positionné, x est un masque d'octets qui provoque différents types de sorties de débogage pour imprimer (voyez le DB_xxx définis dans in2000.h).

proc:x - Si 'PROC_INTERFACE' est défini, x est un masque d'octets qui indique comment fontionne l'interface /proc et ce qu'elle fait (voir la définition de PR_xxx dans in2000.h

Quelques exemples d'utilisation sont listés ci-dessous :

```
in2000=ioport:0x220,noreset
in2000=period:250,disconnect:2,nosync:0x03
in2000=debug:0x1e
in2000=proc:3
```

4.4.6 Matériel basé sur un AMD AM53C974 ('AM53C974=')

Contrairement aux autres pilotes, celui-ci n'utilise pas de paramètres de démarrage pour indiquer les E/S, les IRQ ou les DMA (depuis que le AM53C974 est un périphérique PCI, il n'a pas besoin de la faire). En revanche, les paramètres sont utilisés pour communiquer les modes de transfert et les vitesses qui doivent être utilisés entre le serveur (host) et le périphérique cible. Utilisons un exemple pour y voir plus clair :

```
AM53C974=7,2,8,15
```

Ceci peut être interprété de la manière suivante :

'Pour communiquer entre le contrôleur d'identifiant SCSI-ID 7 et le périphérique d'identifiant SCSI-ID 2, un taux de transfert de 8 MHz en mode synchrone, avec un décalage maximum de 15 octets doit être négocié.' De plus amples détails peuvent être trouvés dans le fichier `linux/drivers/scsi/README.AM53C974`

4.4.7 Les serveurs SCSI BusLogic avec les noyaux v1.2 ('buslogic=')

Dans les anciens noyaux, les pilotes buslogic n'acceptent qu'un seul paramètre, qui est l'adresse d'entrée/sortie. Elle doit correspondre à l'une des valeurs suivantes :

`0x130, 0x134, 0x230, 0x234, 0x330, 0x334`.

4.4.8 Les serveurs SCSI BusLogic aves les noyaux v2.x ('BusLogic=')

Avec les noyaux v2.x, le pilote BusLogic accepte de nombreux paramètres (notez la casse ci dessus ; B et L majuscule !!!). La description détaillée qui suit est extraite directement du pilote de Leonard N. Zubkoff inclus dans le noyau v2.0 .

Pour le pilote BusLogic, une ligne de commande destinée au noyau comprend l'identifiant du pilote "BusLogic=" éventuellement suivi par une série d'entiers séparés par des virgules, et accessoirement par une suite de chaines aussi séparées par des virgules. Chaque ligne de commande s'applique à un adaptateur BusLogic. Des lignes de commande multiples peuvent être utilisées sur des systèmes utilisant plusieurs cartes BusLogic.

Le premier entier indiqué est l'adresse d'Entrée/Sortie (I/O Address) à laquelle l'adaptateur est situé. Si il n'est pas spécifié, il est positionné à zéro, ce qui indique d'appliquer cette ligne de commande au premier adaptateur BusLogic trouvé lors de la séquence de détection. Si une adresse I/O est fournie sur la ligne de commande, la séquence de détection est ignorée.

Le second entier fourni est la profondeur de la 'Tagged Queue' à utiliser pour les périphériques cibles qui utilisent le 'Tagged Queuing'. La profondeur de cette file correspond au nombre de commandes SCSI qui peuvent être envoyées simultanément pour être éxécutées. Si rien n'est indiqué, la valeur par défaut est zéro, et indique d'utiliser une valeur déterminée automatiquement en fonction du 'Total Queue Depth' de l'adaptateur, ainsi que du nombre, du type, de la vitesse des périphériques cible détectés. Pour les adaptateurs qui requièrent des 'ISA Bounce Buffers', le 'Tagged Queue Depth' est automatiquement positionné à 'BusLogic_TaggedQueueDepth_BB' pour éviter une préallocation excessive de mémoire 'DMA Bounce Buffer'. Les périphériques cibles qui ne supportent pas le 'Tagged Queuing' utilisent une 'Queue Depth' ayant pour valeur 'BusLogic_UntaggedQueueDepth'.

Le troisième entier est le 'Bus Settle Time' (temps de stabilisation du bus) en secondes. C'est le temps à attendre entre une remise à zéro physique de l'adaptateur, qui initialise une remise à zéro du bus SCSI, et le moment où l'on peut passer une commande SCSI. Si rien n'est indiqué, il est à zéro par défaut, ce qui indique d'utiliser la valeur BusLogic_DefaultBusSettleTime.

Le quatrième entier correspond aux options locales. Si rien n'est indiqué, la valeur par défaut est 0. Notez que ces options locales sont uniquement utilisées sur un adaptateur hôte spécifique.

Le cinquième entier correspond aux options globales. Si rien n'est indiqué, le valeur par défaut est 0. Notez que les options globales sont appliquées à tous les adaptateurs hôtes.

Les chaînes d'options sont utilisées pour contrôler le 'Tagged Queuing', le recouvrement d'erreur, et le test de l'adaptateur hôte.

Les indications pour le 'Tagged Queuing' commencent par "TQ:" et permettent d'indiquer précisemment où le 'Tagged Queuing' est autorisé sur les périphériques cibles qui le supportent. Les spécifications suivantes sont disponibles :

TQ:Default

- Le 'Tagged Queuing' sera permis, basé sur la version de micro-code de l'adaptateur hôte BusLogic et conditionné par la valeur de 'Tagged Queue Depth' qui doit permettre la mise en file d'attente de multiples commandes.

TQ:Enable

- Le 'Tagged Queuing' est activé pour tous les périphériques de cet adaptateur hôte, outrepassant toutes les limitations qui seraient imposées par la version de micro-code de cet adaptateur.

TQ:Disable

- Le 'Tagged Queuing' sera désactivé pour tous les périphériques reliés à cet adaptateur hôte.

TQ:<Per-Target-Spec>

- Le 'Tagged Queuing' sera contrôlé individuellement pour chaque périphérique cible. <Per-Target-Spec> est une séquence de caractères "Y", "N", et "X". "Y" active le 'Tagged Queuing', "N" désactive le 'Tagged Queuing', et "X" correspond à la valeur par défaut basée sur la version du micro-code. Le premier caractère correspond au périphérique cible 0, le second au périphérique cible 1, et ainsi de suite ; Si la séquence de caractères "Y", "N", et "X" ne suffit pas pour tous les périphériques cibles, les caractères non-indiqués prendront la valeur "X".

Notez que la demande explicite de 'Tagged Queuing' peut conduire à des problèmes. Cette capacité est fournie principalement pour permettre de désactiver le 'Tagged Queuing' sur des périphériques qui ne l'utilisent pas correctement.

Les indications de la Stratégie de Recouvrement d'Erreurs commencent par "ER:" et permettent d'indiquer l'action de recouvrement d'erreur à effectuer quand la 'ResetCommand' est appellée en raison d'un incident sur une commande SCSI, de façon à finir correctement. Les options suivantes sont disponibles :

ER:Default

- Le Recouvrement d'Erreur choisira entre la remise à zéro physique (Hard Reset) et la remise à zéro du bus des périphériques (Bus Device Reset) selon les recommandations du sous système SCSI.

ER:HardReset

- Le Recouvrement d'Erreur demandera une remise à zéro physique de l'adaptateur hôte, ce qui provoquera aussi une remise à zéro du bus SCSI.

ER:BusDeviceReset

- Le recouvrement d'Erreur enverra un message 'Bus Device Reset' (remise à zéro du bus) individuellement au périphérique provoquant l'erreur. Si le Recouvrement d'Erreur est à nouveau appelé pour ce périphérique, et qu'aucune commande SCSI de ce périphérique n'a été éxecutée avec succès depuis le dernier message 'Bus Device Reset' a été envoyé, alors une remise à zéro physique est provoquée.

ER:None

- Le Recouvrement d'Erreur sera supprimé. Cette option peut seulement être sélectionnée si un 'SCSI Bus Reset' ou un 'Bus Device Reset' provoque un plantage du périphérique cible de façon totale et irrécupérable.

ER:<Per-Target-Spec>

- Le Recouvrement d'Erreur sera contrôlé individuellement pour chaque périphérique. <Per-Target-Spec> est une séquence de caractères "D", "H", "B", et "N". "D" correspond à 'Default', "H" à 'Hard Reset', "B" à 'Bus Device Reset', et "N" à 'None'. Le premier caractère correspond au périphérique 0 , le second au périphérique 1, et ainsi de suite. Si la séquence de caractères "D", "H", "B", et "N" ne suffit pas pour tous les périphériques possibles, les carractères manquants correspondront à "D".

Les spécifications de test de l'adaptateur hôte sont les suivantes :

NoProbe - Aucun test d'aucune sorte ne doit être fait, et par conséquent, aucun adaptateur hôte BusLogic ne sera détecté.

NoProbeISA - Aucun test des adresses I/O standard ISA ne sera fait, et par conséquent, seuls les adaptateurs hôtes PCI seront détectés.

NoSortPCI - Les adaptateurs hôtes PCI seront énumérés dans l'ordre fourni par le BIOS PCI, ignorant tous les paramètres de l'option "Utilisation du # des bus et périphériques pour la séquence d'analyse du bus PCI" de l'AutoSCSI.

4.4.9 Les cartes SCSI EATA ('eata=')

Depuis la déjà ancienne version v2.0 du noyau, les pilotes EATA acceptent un paramètre de démarrage permettant d'indiquer les adresses d'entrée/sortie qui doivent être testées. Il est de la forme :

```
eata=iobase1[,iobase2][,iobase3]...[,iobaseN]
```

Le pilote testera les adresses dans l'ordre où elles sont fournies.

4.4.10 Future Domain TMC-8xx, TMC-950 ('tmc8xx=')

Le code de test pour ces hôtes SCSI recherche un BIOS installé, et s'il n'en détecte aucun, le test ne trouvera pas votre carte. Ou si la signature de votre BIOS n'est pas reconnue, elle ne sera pas trouvée non plus. Dans ce cas, vous aurez à utiliser un paramètre de démarrage de la forme :

```
tmc8xx=mem_base,irq
```

La valeur `mem_base` est l'adresse dans le plan mémoire de la région d'entrée/sortie utilisée par la carte. C'est généralement une des valeurs suivantes :

`0xc8000, 0xca000, 0xcc000, 0xce000, 0xdc000, 0xde000`.

4.4.11 Future Domain TMC-16xx, TMC-3260, AHA-2920 ('fdomain=')

Le pilote détecte ces cartes selon une liste connue de signatures de BIOS ROM. Pour obtenir une liste complète des révisions connues de BIOS, voyez le fichier `linux/drivers/scsi/fdomain.c` qui contient beaucoup d'informations en début de fichier. Si votre BIOS n'est pas connu du pilote, vous pourrez utiliser un forçage de la façon suivante :

```
fdomain=iobase,irq[,scsi_id]
```

4.4.12 Le lecteur ZIP IOMEGA / Port Parallèle ('ppa=')

Ce pilote est pour l'adaptateur SCSI de l'IOMEGA Port Parallèle qui est intégré dans le lecteur IOMEGA ZIP. Il peut aussi fonctionner avec le périphérique d'origine IOMEGA PPA3. Le paramètre de démarrage pour ce pilote a la structure suivante :

```
ppa=iobase,speed_high,speed_low,nybble
```

où tous les paramètres sont facultatifs, sauf 'iobase'. Si vous souhaitez modifier un des trois éléments, il serait bon de lire au préalable le document `linux/drivers/scsi/README.ppa` afin d'obtenir des détails sur ces paramètres.

4.4.13 Contrôleurs utilisant un NCR5380 ('ncr5380=')

Selon votre carte, le 5380 peut-être soit 'i/o mapped' ou 'memory mapped' (répertorié en entrée/sortie ou répertorié en mémoire). Une adresse en dessous de 0x400 indique souvent l'i/o mapping, cependant, les matériels PCI et EISA utilisent des adresses d'entrée/sortie au dessus de 0x3ff. Dans tous les cas, vous indiquez l'adresse, la valeur de l'IRQ, et la valeur du canal DMA. Un exemple pour une carte 'i/o mapped' serait : `ncr5380=0x350,5,3`. Si la carte n'utilise pas les interruptions, une valeur d'IRQ 255 (`0xff`) désactivera les interruptions. Une IRQ à 254 indiquera d'activer l'autotest. Des détails supplémentaires sont fournis dans le document `linux/drivers/scsi/README.g_NCR5380`.

4.4.14 Contrôleurs utilisant un NCR53c400 ('ncr53c400=')

Le support du 53c400 est fait avec le même pilote que le support du 5380 mentionné ci-dessus. Le paramètre de démarrage est identique au précédent, sauf qu'aucun canal DMA n'est utilisé par le 53c400.

4.4.15 Contrôleurs utilisant un NCR53c406a ('ncr53c406a=')

Ce pilote utilise un paramètre de démarrage de la forme suivante :

```
ncr53c406a=PORTBASE,IRQ,FASTPIO
```

où les paramètres IRQ et FASTPIO sont optionnels. Une valeur d'interruption à zéro désactive l'utilisation des interruptions. L'utilisation d'une valeur à 1 pour FASTPIO active l'utilisation des instructions `insl` et `outsl` au lieu des instructions mono-octet `inb` et `outb`. Le pilote peut aussi utiliser le DMA comme une option utilisée lors de la compilation (compile-time option).

4.4.16 Pro Audio Spectrum ('pas16=')

La PAS16 utilise une puce NCR5380 SCSI, et les nouveaux modèles peuvent être configurés de façon logicielle. La syntaxe du paramètre est la suivante :

```
pas16=iobase,irq
```

La seule différence est que vous pouvez spécifier une valeur d'IRQ égale à 255, qui indique au pilote de travailler sans utiliser les interruptions, malheureusement au détriment des performances. La valeur de `iobase` est généralement `0x388`.

4.5 Seagate ST-0x ('st0x=')

Le code du programme de test de cet hôte SCSI recherche un BIOS installé, et s'il n'y en a aucun de présent, le test ne trouvera pas votre carte. Ou si la signature de votre BIOS n'est pas reconnue elle ne sera pas trouvée non plus. Dans ce cas, vous aurez à utiliser le paramètre suivant :

```
st0x=mem_base,irq
```

La valeur de `mem_base` est l'adresse dans le plan mémoire de la région d'entrée/sortie utilisée par la carte. En général, il s'agit d'une des valeurs suivantes : `0xc8000, 0xca000, 0xcc000, 0xce000, 0xdc000, 0xde000`.

4.6 Trantor T128 ('t128=')

Cette carte est aussi conçue autour de la puce NCR5380, et accepte les options suivantes :

```
t128=mem_base,irq
```

Les valeurs autorisées pour `mem_base` sont les suivantes : `0xcc000, 0xc8000, 0xdc000, 0xd8000`.

4.6.1 Cartes SCSI Ultrastor ('u14-34f=')

Notez que pour cette carte tout se présente sous la forme de deux pilotes indépendants, nommés `CONFIG_SCSI_U14_34F` qui utilise `u14-34f.c` et `CONFIG_SCSI_ULTRASTOR` qui utilise `ultrastor.c`. C'est le u14-34f qui (jusqu'au dernier noyau v2.0) accepte un paramètre de démarrage de la forme :

```
u14-34f=iobase1[,iobase2][,iobase3]...[,iobaseN]
```

Le pilote autotestera les adresses dans l'ordre dans lequel elles apparaissent.

4.6.2 Cartes Western Digital WD7000 ('wd7000=')

Le test du pilote pour le wd7000 cherche une chaine connue de BIOS ROM et connait quelques réglages standards de configuration. Si il ne retrouve pas les valeurs correctes pour votre carte, ou que vous avez une version de BIOS non reconnue, vous pouvez utiliser le pramètre suivant :

```
        wd7000=irq,dma,iobase
```

4.7 Cartes n'acceptant pas les paramètres de démarrage

Pour l'instant, les cartes SCSI suivantes n'utilisent aucun des paramètres de démarrage. Dans certains cas, vous pouvez "bricoler" les valeurs en éditant directement le pilote lui-même, si cela est nécessaire bien sûr.

```
        Adaptec aha1740 (autotest EISA),
        NCR53c7xx, 8xx (PCI, toutes les deux)
        Qlogic Fast (0x230, 0x330)
        Qlogic ISP (PCI)
```

5 Disque Durs

Cette section fait la liste de tous les paramètres de démarrage associés aux lecteurs de disques standards MFM/RLL, ST-506, XT, et IDE. Notez que les deux pilotes IDE et ST-506 HD acceptent l'option 'hd='.

5.1 Paramètres des lecteurs de Disques/CD-ROM IDE

Les pilotes IDE acceptent un certain nombre de paramètres, qui vont de la définition des caractéristiques du disque, à la correction des erreurs produites par les nouvelles puces ou celles qui sont défectueuses. Ce qui suit est un résumé des paramètres de démarrage possibles. Pour plus de détails, il faut *absolument* consulter le fichier `ide.txt` dans le répertoire `linux/Documentation`, duquel ce résumé est extrait.

```
 "hdx="  est reconnu pour toutes les valeurs de "x", de "a" to "h", comme "hdc".
 "idex=" est reconnu pour toutes les valeurs de "x" de "0" à "3", comme "ide1".

 "hdx=noprobe"          : le lecteur est peut-être présent, mais ne pas le tester
 "hdx=none"             : le lecteur n'est PAS présent, ignorer le cmos et
                          ne pas tester.
 "hdx=nowerr"           : ignorer le bit WRERR_STAT sur ce lecteur
 "hdx=cdrom"            : le lecteur est présent, et c'est un cdrom
 "hdx=cyl,head,sect"    : le lecteur est présent, avec la description indiquée
 "hdx=autotune"         : le pilote essaiera de régler la vitesse de l'interface
                          pour atteindre le plus rapide des modes PIO supportés,
                          si possible pour ce lecteur seulement.
                          Ce n'est pas supporté par tous les types de puces,
                          et peut de temps en temps poser des problèmes avec
                          les disques IDE anciens ou originaux.

 "idex=noprobe"         : ne pas tenter d'accéder ou utiliser cette interface
```

```
"idex=base"            : tester l'interface à l'adresse indiquée,
                         où "base" est généralement 0x1f0 ou 0x170
                         et "ctl" est considéré comme étant "base"+0x206
"idex=base,ctl"        : indiquer les deux, base et ctl
"idex=base,ctl,irq"    : indiquer les valeurs de base, ctl, et irq
"idex=autotune"        : le pilote tentera de régler la vitesse de l'interface
                         pour atteindre le plus rapide des modes PIO supportés,
                         pour tous les lecteurs de cette interface.
                         Ce n'est pas supporté par tous les types de puces,
                         et peut de temps en temps poser des problèmes avec
                         les disques IDE anciens ou originaux.

"idex=noautotune"      : le pilote n'essaiera PAS de régler la vitesse
                         de l'interface. Ceci est la valeur par défaut pour
                         le plupart des puces, excepté le cmd640.
"idex=serialize"       : ne pas empièter sur les opérations sur idex et ide(x^1)
```

Les suivants sont valides SEULEMENT pour ide0, et les valeurs par défaut pour base, ctl et ports ne doivent pas être modifiés.

```
"ide0=dtc2278"         : teste/supporte l'interface DTC2278
"ide0=ht6560b"         : teste/supporte l'interface HT6560B
"ide0=cmd640_vlb"      : *REQUIS* pour les cartes VLB avec la puce CMD640
                         (pas pour PCI - automatiquement détecté)
"ide0=qd6580"          : teste/supporte l'interface qd6580
"ide0=ali14xx"         : teste/supporte les puces ali14xx (ALI M1439/M1445)
"ide0=umc8672"         : teste/supporte les puces umc8672
```

Tout le reste est rejeté par un message "BAD OPTION" (mauvaise option).

5.2 Options du pilote standard ST-506 ('hd=')

Le pilote standard de disque accepte les mêmes paramètres que le pilote IDE. Notez cependant qu'il ne requiert que 3 valeurs (C/H/S) - Ni plus ni moins, et il vous ignorera -. De plus, il accepte uniquement le paramètre 'hd=', c'est à dire que 'hda=', 'hdb=' et tout le reste ne sont pas autorisés ici. Le format est le suivant :

```
hd=cyls,heads,sects
```

Si deux disques sont installés, la ligne ci-dessus est répétée avec les caractéristiques techniques du second disque.

5.3 Options du pilote de disque XT ('xd=')

Si vous êtes malchanceux au point d'utiliser une de ces vieilles cartes 8 bits qui transfère les données à la vitesse fulgurante de 125 ko/s, c'est ici qu'est le scoop. Le code de test pour ces cartes recherche un BIOS installé et s'il n'en trouve pas, le test ne détectera pas votre carte. Ou encore, si la signature de votre BIOS n'est pas reconnue, le test ne trouvera pas votre carte non plus. Dans n'importe lequel de ces cas, vous devrez utiliser le paramètre suivant :

```
xd=type,irq,iobase,dma_chan
```

La valeur de `type` indique qui est le constructeur de la carte et peut prendre les valeurs suivantes : 0=generic; 1=DTC; 2,3,4=Western Digital, 5,6,7=Seagate; 8=OMTI. La seule différence entre les différents types pour un même constructeur est la chaîne BIOS utilisée pour la détection, et qui n'est pas utilisée si le type est spécifié.

La fonction `xd_setup()` ne contrôle pas les valeurs, et supporte que vous saisissiez les 4 valeurs. Ne soyez pas déçu. Voici un exemple d'utilisation pour un contrôleur WD1002 avec un BIOS inactivé/supprimé, utilisant les paramètres 'par défaut' du controleur XT :

```
xd=2,5,0x320,3
```

6 CD-ROMs (Non-SCSI/ATAPI/IDE)

Cette section fait l'inventaire de tous les paramètres de démarrage possibles pour les lecteurs de CD-ROM. Ceci n'inclut pas les CD-ROMs SCSI ou IDE/ATAPI. Consultez les sections appropriées pour ces types de CD-ROMs.

Notez que la plupart de ces CD-ROM ont des fichiers de documentation que vous *devriez* lire, et ils sont tous dans le répertoire : `linux/Documentation/cdrom`.

6.1 L'interface Aztech ('aztcd=')

La syntaxe pour ce type de carte est :

```
aztcd=iobase[,magic_number]
```

Si vous positionnez le `magic_number` (nombre magique) à `0x79` alors le pilote essaiera puis laissera tomber dans le cas d'une microprogrammation inconnue. Toutes les autres valeurs seront ignorées.

6.2 L'interface Sony CDU-31A et CDU-33A ('cdu31a=')

On rencontre cette interface CD-ROM sur certaines cartes son Pro Audio Spectrum, ainsi que sur les autres cartes d'interface fournies par Sony. La syntaxe est la suivante :

```
cdu31a=iobase,[irq[,is_pas_card]]
```

Le fait de spécifier une valeur d'IRQ égale à zéro indique au pilote que les interruptions logicielles ne sont pas supportées (comme sur certaines cartes PAS). Si votre carte supporte les interruptions, vous devrez les utiliser car elles abaissent la consommation de CPU par le pilote.

Le 'is_pas_card' peut-être saisi sous la forme suivante 'PAS' si vous utilisez une carte Pro Audio Spectrum, mais on peut aussi ne pas l'indiquer.

6.3 L'interface Sony CDU-535 ('sonycd535=')

La syntaxe pour cette interface de CD-ROM est :

```
sonycd535=iobase[,irq]
```

La valeur zéro peut-être utilisée comme 'bouche-trou' pour l'I/O base si l'on désire spécifier une valeur d'IRQ.

6.4 L'interface GoldStar ('gscd=')

La syntaxe pour cette interface de CD-ROM est :

```
gscd=iobase
```

6.5 L'interface standard Mitsumi ('mcd=')

La syntaxe pour cette interface de CD-ROM est :

```
mcd=iobase,[irq[,wait_value]]
```

La valeur `wait_value` est utilisée comme une valeur interne de dépassement de temps pour les gens qui ont des problèmes avec leur disques, et peut, ou non, être implémentée en fonctions d'une instruction `DEFINE` lors de la compilation.

6.6 L'interface ISP16 ('isp16=')

la syntaxe pour cette interface de CD-ROM est :

```
isp16=[port[,irq[,dma]]][[,]drive_type]
```

Utiliser une valeur à 0 pour `irq` ou `dma` signifie qu'ils ne sont pas utilisés. Les valeurs possibles pour `drive_type` sont `noisp16`, `Sanyo`, `Panasonic`, `Sony`, et `Mitsumi`. L'utilisation de `noisp16` désactive les lecteurs totalement.

6.7 L'interface Mitsumi XA/MultiSession ('mcdx=')

Pour l'instant, ce pilote 'expérimental' possède une fonction de configuration mais aucun paramètre n'est encore implémenté (version 1.3.15). Le matériel est le même que ci-dessus, mais le pilote possède de nouvelles fonctionnalités.

6.8 L'interface Optics Storage ('optcd=')

La syntaxe pour ce type de carte est :

```
optcd=iobase
```

6.9 L'interface Phillips CM206 ('cm206=')

La syntaxe pour ce type de carte est :

```
cm206=[iobase][,irq]
```

La valeur de l'IRQ est comprise entre 3 et 11,et les adresses des ports d'entrée/sortie sont comprises entre `0x300` et `0x370`, vous pouvez donc spécifier un ou deux nombres, dans n'importe quel ordre. Il accepte aussi 'cm206=auto' pour activer l'autotest.

6.10 L'interface Sanyo ('sjcd=')

La syntaxe pour ce type de carte est :

```
sjcd=iobase[,irq[,dma_channel]]
```

6.11 L'interface SoundBlaster Pro ('sbpcd=')

La syntaxe de ce type de carte est :

```
sbpcd=iobase,type
```

Où `type` prend une des valeurs suivantes (Attention : le respect des majuscules et des minuscules est important) : 'SoundBlaster', 'LaserMate', ou 'SPEA'. L'adresse d'entrée/sortie de base est celle de l'interface de CD-ROM, et *non* celle de la partie son de la carte.

7 Autres Périphériques Matériels

Tous les autres périphériques qui ne peuvent être classés dans une des catégories ci-dessus sont entassés ici.

7.1 Périphériques Ethernet ('ether=')

Différents pilotes utilisent différents paramètres, mais ils partagent tous au moins une IRQ, une adresse d'entrée/sortie, et un nom. Dans sa forme la plus générique, cela ressemble à ça :

```
ether=irq,iobase[,param_1[,param_2,...param_8]]],name
```

Le premier argument non-numérique est pris comme nom. La valeur `param_n` (si elle est applicable) a généralement des significations différentes pour chaque carte/pilote. Les valeurs courantes de `param_n` sont utilisées pour indiquer des choses comme l'adresse de la mémoire partagée, la sélection d'interface, le canal DMA et ainsi de suite.

L'utilisation la plus courante de ce paramètre est de forcer le test d'une seconde carte ethernet, alors que par défaut on en teste une seule. Ceci peut être accompli avec un simple ordre :

```
ether=0,0,eth1
```

Notez que la valeur zéro pour l'IRQ et l'I/O base dans l'exemple ci-dessus indiquent au pilote de faire un autotest.

NOTE IMPORTANTE POUR LES UTILISATEURS DE MODULES : ce qui est indiqué ci-dessus *ne forcera pas* un autotest pour une seconde si vous utilisez les pilotes de périphériques en tant que modules chargeables au moment de l'exécution (au lieu de les avoir compilés dans le noyau). La plupart des distributions de Linux utilisent un noyau central dépouillé combiné avec une large sélection de pilotes modulaires. Le paramètre `ether=` s'applique seulement aux pilotes compilés directement dans le noyau.

Le Ethernet-HowTo décrit de façon exhaustive l'utilisation de plusieurs cartes simultanément, ainsi que la façon dont est utilisée la valeur `param_n` en fonction des spécificités de chaque carte/pilote. Les lecteurs concernés pourront faire référence à la section de ce document correspondant à leur carte pour une information plus précise.

Ethernet-HowTo `http://sunsite.unc.edu/mdw/HOWTO/Ethernet-HOWTO.html`

7.2 Le pilote du Lecteur de Disquettes ('floppy=')

Il existe de nombreuses options pour le pilote du lecteur de disquette, et qui sont listées dans le fichier `README.fd` dans le répertoire `linux/drivers/block`. Cette information est extraite directement du fichier.

floppy=mask,allowed_drive_mask

Positionne le "bitmask" (masque binaire) des lecteurs autorisés à la valeur `mask`. Par défaut, seules les unités 0 et 1 de chaque contrôleur de lecteur de disquette sont autorisées. Ceci est fait car certains matériels non-standards (cartes mères ASUS PCI) mettent la pagaille dans le clavier lorsque l'on accède aux unités 2 ou 3. Cette option est un peu obsolète en raison de l'option cmos.

floppy=all_drives

Positionne le "bitmask" (masque binaire) des disques autorisés à tous les disques. Utilisez ceci si vous avez plus de deux lecteurs de disquette connectés à un contrôleur de lecteur de disquettes.

floppy=asus_pci

Positionne le "bitmask" uniquement aux unités autorisées 0 et 1. (Par défaut)

floppy=daring

Indique au pilote du lecteur de disquette que vous avez un contrôleur de lecteur de disquette qui se conduit bien. Ceci permet des opérations plus efficaces et plus discrètes, mais peut échouer sur certains contrôleurs. Ceci peut accélérer certaines opérations.

floppy=0,daring

Indique au pilote du lecteur de disquette que votre contrôleur doit être utilisé avec précaution.

floppy=one_fdc

Indique au pilote de lecteur de disquette que vous n'avez qu'un contrôleur de lecteur de disquette (Par défaut).

floppy=two_fdc *ou* floppy=address,two_fdc

Indique au pilote de lecteur de disquette que vous avez deux contrôleurs de lecteurs de disquette. Le second contrôleur est supposé être à l'adresse indiquée. Si l'adresse n'est pas donnée on suppose qu'elle est égale à 0x370.

floppy=thinkpad

Indique au pilote de lecteur de disquette que vous avez un Thinkpad. Les Thinkpads utilisent une convention inversée pour la "disk change line" (ligne de changement de disque).

floppy=0,thinkpad

Indique au pilote de lecteur de disquette que vous ne possédez pas un Thinkpad.

floppy=drive,type,cmos

Positionne le type cmos du `drive` à `type`. De plus, ce lecteur est autorisé dans le "bitmask" (masque binaire). C'est pratique si vous avez plus de deux lecteurs de disquette (seuls deux peuvent être décrits dans la cmos physique), ou si votre BIOS utilise un type de CMOS non-standard. Si l'on positionne le CMOS à 0 pour les deux premiers disques (par défaut) le pilote de lecteur de disquette ira lire la cmos physique.

floppy=unexpected_interrupts

Imprime un message d'alerte lorsqu'une interruption inattendue est reçue (comportement par défaut).

floppy=no_unexpected_interrupts *or* floppy=L40SX

Ne pas imprimer de message lorsqu'une interruption inattendue est reçue. Ceci est nécessaire sur un IBM

L40SX portable dans certains modes vidéo (il semble qu'il y ait une interaction entre la vidéo et les disquettes). Les interruptions inattendues affectent seulement les performances, et peuvent être ignorées sans crainte).

7.3 Le pilote de sons ('sound=')

Le pilote de sons peut aussi recevoir des paramètres de démarrage qui écraseront les valeurs compilées dans le programme. Ceci n'est pas recommandé, et de plus c'est complexe. Ceci est décrit (était décrit ?) dans le fichier `Readme.Linux`, dans le répertoire `linux/drivers/sound`. Il accepte de recevoir un paramètre de la forme :

```
sound=device1[,device2[,device3...[,device11]]]
```

Où chaque valeur de `deviceN` est de la forme `0xTaaaId`, et les octets sont utilisés de la façon suivante :

T - type de périphérique : 1=FM, 2=SB, 3=PAS, 4=GUS, 5=MPU401, 6=SB16, 7=SB16-MPU401

aaa - adresse d'entrée/sortie en hexadécimal.

I - ligne d'interruption en hexadécimal (i.e 10=a, 11=b, ...).

d - canal DMA.

Comme vous pouvez le voir, ceci reste assez malpropre et vous ferez mieux de compiler vos propres valeurs comme c'est recommandé. Si l'on utilise un paramètre de démarrage 'sound=0' on désactive entièrement le pilote de sons.

7.4 Le pilote de souris sur bus "Bus Mouse" ('bmouse=')

Le pilote des souris sur bus accepte un seul paramètre, qui est la valeur de l'IRQ matérielle à utiliser.

7.5 Le pilote MS Bus Mouse ('msmouse=')

Le pilote MS mouse accepte un seul paramètre, qui correspond à l'IRQ à utiliser.

7.6 Le pilote d'imprimantes ('lp=')

Depuis le noyau 1.3.75, vous pouvez indiquer au pilote d'imprimante quels sont les ports qu'il doit utiliser et ceux qu'il *ne doit pas* utiliser. Vous devriez l'utiliser si vous ne voulez pas que le pilote demande tous les ports parallèles disponibles, alors que d'autres pilotes (c.a.d. PLIP, PPA) peuvent aussi les utiliser.

Le format du paramètre est des paires i/o, IRQ. Par exemple, `lp=0x3bc,0,0x378,7` utilisera le port d'adresse 0x3bc en mode IRQ-less (élection), et utilisera l'IRQ 7 pour le port d'adresse 0x378. Le port 0x278 (si il y en a un) ne sera pas testé, jusqu'à ce que l'autotest soit utilisé en l'absence d'un paramètre 'lp=' argument. Pour désactiver totalement le pilote d'impression, on peut utiliser `lp=0`.

7.7 Le pilote ICN ISDN ('icn=')

Le pilote ISDN nécessite un paramètre de démarrage de la forme suivante :

```
icn=iobase,membase,icn_id1,icn_id2
```

où `iobase` est l'adresse du port d'entrée/sortie de la carte, `membase` est l'adresse de base de la mémoire partagée de la carte, et les deux `icn_id` sont des chaines d'identification ASCII uniques.

7.8 Le pilote PCBIT ISDN ('pcbit=')

Ce paramètre de démarrage utilise des paires de valeurs de la forme :

```
pcbit=membase1,irq1[,membase2,irq2]
```

où `membaseN` est l'adresse de base de la mémoire partagée de la Nième carte, et `irqN` est l'interruption de la Nième carte. La valeur par défaut est IRQ 5 et l'adresse de base `0xD0000`.

7.9 Le pilote Teles ISDN ('teles=')

Le pilote ISDN nécessite un paramètre de démarrage de la forme suivantenbsp;:

```
teles=iobase,irq,membase,protocol,teles_id
```

où `iobase` est l'adresse du port e/s de la carte, `membase` est l'adresse de base de la mémoire partagée, `irq` est le canal d'interruption utilisé par la carte, et `teles_id` est l'identifiant ASCII unique.

7.10 Le pilote DigiBoard ('digi=')

Le pilote DigiBoard accepte une chaine de six identifiants ou entiers séparés par des virgules. Les 6 valeurs dans l'ordre sont :

```
Active/Désactive la carte
Type de la carte : PC/Xi(0), PC/Xe(1), PC/Xeve(2), PC/Xem(3)
Active/Désactive la mise en ordre alternative des broches
Nombre de ports sur cette carte
Port E/S sur lequel la carte est configurée  (en HEXA si on
utilise des chaines d'identification)
Adresse de base de la fenêtre mémoire (en HEXA si on utilise les
chaines d'identification)
```

Un exemple de paramètre de démarrage correct (dans ses deux formes) est :

```
digi=E,PC/Xi,D,16,200,D0000
digi=1,0,0,16,512,851968
```

Notez que le pilote prend les valeurs par défaut de `0x200` pour l'i/o et pour la mémoire partagée `0xD0000` en l'absence de paramètre de démarrage `digi=`. Il n'y a pas d'autotest effectué. Plus de détails peuvent être trouvés dans le fichier `linux/Documentation/digiboard.txt`.

7.11 le pilote RISCom/8 Multiport Serial ('riscom8=')

Jusqu'à quatre cartes peuvent être supportées en fournissant une valeur d'E/S unique pour chaque carte installée. Les autres détails pourront-être trouvés dans le fichier `linux/Documentation/riscom8.txt`.

7.12 Le modem Série/Parallèle Radio Baycom (‘baycom=’)

Le format du parmètre de démarrage pour ces périphériques est de la forme :

```
baycom=modem,io,irq,options[,modem,io,irq,options]
```

Utiliser modem=1 signifie que vous avez le périphérique ser12, modem=2 signifie que vous avez le périphérique par96. Utiliser options=0 signifie l’utilisation du DCD matériel, et options=1 signifie l’utilisation du DCD logiciel. L’`io` et l’`irq` sont l’adresse I/O de base du port, et la valeur de l’interruption. Il y a plus de détails dans le fichier `README.baycom` qui est généralement dans le répertoire `/linux/drivers/char/`.

8 Conclusion

Si vous avez trouvé des fautes de frappe manifestes, ou des informations périmées dans ce document, faites le moi savoir. Il est facile de laisser passer quelque chose.

Merci,

Paul Gortmaker, `Paul.Gortmaker@anu.edu.au`

Merci de faire parvenir vos remarques sur la traduction de ce document à Laurent Renaud, `lrenaud@hol.fr`

(`http://wwwperso.hol.fr/~lrenaud`)