# PA−RISC/Linux Boot HOWTO


## Deb Richardson

The Puffin Group

`<deb@thepuffingroup.com>`

## Thomas Marteau

ESIEE

`<marteaut@tuxfamily.org>`



**Historique des versions**

| | | |
|---|---|---|
| Version 0.9 | 15 janvier 2002 | Revu par : tm |
| Cette version vous apporte quelques conseils utiles pour obtenir votre propre noyau sur les systèmes hppa. | | |
| Version 0.8 | 17 octobre 2001 | Revu par : tm |
| Cette version prend en compte le changement de nom des sites officiels FTP et CVS et modifie la licence. | | |
| Version 0.7 | 13 octobre 2001 | Revu par : tm |
| Cette version s'adapte au progrès de PA/Linux. | | |
| Version 0.6 draft | 26 septembre 2001 | Revu par : tm |
| Cette version contient des modifications mineures et complète la section "Utiliser bootp/tftpd". | | |
| Version 0.5 draft | 3 août 2001 | Revu par : tm |
| Cette version est une mise à jour majeure du papier de Deb. | | |
| Version 0.3 draft | 24 novembre 1999 | Revu par : dlr |
| Voici la première version publiée du manuel. | | |



Ce document explique les procédures pour obtenir un noyau issu du développementde PA−RISC/Linux fonctionnant sur votre système PA−RISC. Il vous montre aussi les fonctionnalités de PALO, le chargeur de noyau pour PA/Linux. Vous y trouverez de nombreuses informations pour compiler un noyau depuis le code source disponible sur http://cvs.parisc−linux.org/. Remarquez que cette version est basée sur le travail de Deb Richardson. Elle actualise certaines informations rendues obsolètes par les avancées du portage. Cependant, je dois dire que ce manuel conserve des textes de cette version et j'ai dévoilé quelques parties commentées à l'époque de Deb.

Si vous cherchez des informations liées au matériel HP mais pas directement liées au PA−RISC, veuillez lire HP−HOWTO de Bruno Cornec.

Pour plus d'informations sur le projet de portage de PA−RISC/Linux, veuillez visiter http://www.parisc−linux.org/. Ce site officiel détaille le développement et les améliorations du noyau. Pour l'espace utilisateur, veuillez vous référer aux pages Debian sur le portage.


---

***Table des matières***

## 1. Copyright and Licensing



---

## 2. Matériel supporté

En produisant la version 0.9.3 de PA/Linux, le support matériel a connu une grande amélioration tant en qualité qu'en quantité. C'est pourquoi même si votre modèle n'est pas listée, il faut l'essayer et rapporter le résultat à la liste de diffusion. Les machines PA−RISC suivantes peuvent fonctionner comme n'importe quelle machines d'une architecture différente. Cette liste peut changer à tout moment. La meilleure manière d'obtenir une liste à jour est d'aller voir http://pateam.esiee.fr/list.html. À cet endroit, vous saurez si votre matériel a été rapporté comme fonctionnel et comment. Par exemple, il est stipulé si vous pouvez utiliser PA/Linux *via* la carte graphique ou bien *via* une console Série. Vous trouverez tout ce dont vous avez besoin là−bas comme les mises à jour de votre microprogramme aussi connu sous le terme de *firmware*.

- Tous les modèles 712. Vous pouvez utiliser X via le périphérique de mémoire vidéo, *framebuffer* ;
- Tous les modèles 715 incluant la série Strider ;
- Les modèles 710, 720, 730, 750 devraient fonctionner avec la dernière ISO. Elle comprend des modifications, surtout pour les disques durs ;
- Certains modèles 725, 735, 755 fonctionnent avec les dernières versions du source. Mais, vu qu'il n'y a pas beaucoup de résultats fournis par les utilisateurs de telles machines, il est difficile d'en dire plus ;
- A180C et similaires ;
- A500 et similaires ;
- Les modèles BXXX comme B132, B180. Ces machines peuvent être utilisées avec une mémoire vidéo *via Standard Text Interface* ;
- Les modèles CXXX comme C110, C160 ;
- Les modèles CXXXX fonctionnent avec les périphériques d'entrées/sorties liés au premier IOMMU ;

- La class D fonctionne normalement. Cependant, beaucoup de variantes de ce modèle existent. Par conséquent, il est difficile de généraliser ;
- La classe J est équivalent aux modèles CXXXX avec les multi−processeurs en plus. Donc, on peut dire la même chose que pour les CXXXX ;
- L1000 et L2000.

Le matériel suivant devraient fonctionner dans le futur:

- Le modèle L3000 ;
- La classe N ;
- La classe SuperDome.

Cette liste indique le matériel qui ne sera pas opérationnel:

- Les très vieux modèles comme les 705 ;
- Les classes E,F,G,H et I. Il semble que Christoph Plattner a réussi à faire fonctionner son E55 ;
- Les classes T et V.

# 3. Préparer le démarrage

Comme tout autre machine, les systèmes à base de processeur PA−RISC ont plusieurs étapes à effectuer afin de fonctionner. Tout d'abord, vous devez paramétrer correctement votre matériel. Ensuite, vous devez démarrer une première fois pour lancer le processus d'installtion. Une fois le noyau est mis en place, vous pouvez redémarrer pour installer le reste de votre système de fichiers. À ce point, votre système est installé et opérationnel. Ce document est séparé en plusieurs parties correspondant à chacune de ces étapes.

Cette première section introduit quelques paramètres liés au matériel HP configurables *via* HP **BOOT_ADMIN**.

## 3.1. Consoles

Pour démarrer votre système PA−RISC avec un noyau PA/Linux, vous devez d'abord avoir une console sur ce système. Elle peut être graphique. Un moniteur devra être connecté à la machine. Si vous optez pour la console série, il vous faut l'équipement adéquat comme une seconde machine qui récupère par un cable croisé ce qu'émet le port série.

Pour la console graphique, vous devez être sûr que Linux supporte la carte graphique de votre système. De plus, il existe deux façons d'obtenir la console graphique. Si vous pensez faire un rapport de bogue pour ce genre de souci, vous devez différencier les deux types. Tout d'abord, la console STI est le mode texte classique vidéo comme pour votre PC par exemple. Le nom est dû au fait que chaque machine HP possède une interface standard pour l'affichage de texte appelée *Standard Text Interface* qui fournit les fonctions de base pour accéder à la mémoire vidéo. Le second mode graphique de la console est celui de la console *framebuffer*. En fait, lors du démarrage, vous verrez un pengouin apparaître en haut à gauche de votre écran. Ceci est la méthode la plus simple pour différencier les deux modes graphiques. La console série est la meilleure méthode si vous désirez poster un rapport de bogue complet. Elle vous permet même d'avoir la sortie alors que vous êtes dans **BOOT_ADMIN**. La plupart de serveurs sont utilisables qu'avec la console série.

Si vous avez un moniteur pour votre machine PA−RISC, la méthode la plus conviviale est le mode graphique. Si vous rencontrez des soucis, passez en console série.

### 3.1.1. Utiliser une console graphique

Utiliser une console graphique est juste une autre manière de dire «Utiliser l'écran relié à la machine». Si vous possédez un moniteur pour votre machine PA−RISC, vous devriez être capable d'interagir via la console avec cet écran sans trop de modification et sans avoir à configurer une console série.

Naturellement, si vous pouvez utiliser une console graphique, c'est le meilleur choix à faire. Cependant, vérifiez que votre matériel le supporte.

### 3.1.2. Utiliser une console série

Vous devriez utiliser une console série si vous n'avez pas d'écran disponible ou que votre machine ne peut pas accéder à l'un des modes graphiques. Aussi, le noyau ne gère pas certains types de matériel pour l'instant. Si vous devez mettre en place une console série, suivez les indications suivantes.

#### 3.1.2.1. Cable série

Pour connecter votre machine PA−RISC à votre PC afin de pouvoir démarrer le système en utilisant une console, vous avez besoin d'un cable null−modem 9 pin vers 9 pin femelle. Un tel cable est vendu dans tout bon magasin d'équipement informatique. Bien que vous puissiez connecter l'autre extrémité du cable à un terminal, il est conseillé de le brancher sur une machine qui exécute **minicom** pour avoir tous les messages et pouvoir les copier/coller dans un fichier pour compléter un rapport de bogue parfait.

#### 3.1.2.2. Configurer minicom pour Linux

Pour communiquer avec votre machine PA−RISC par ce procédé, vous devez mettre en place le mode console série et configurer un programme qui vous permettra d'utiliser le port série comme un terminal. Nous recommandons **minicom**, qui est disponible dans la plupart des distributions Linux par défaut. Si **minicom** n'est pas présent sur votre système,vous trouverez la plus récente version du paquet sur n'importe quel grand sites de logiciels liés à Linux.

La plupart de la configuration de **minicom** est dépendant des systèmes utilisés. Cependant, soyez sûrs que:

a. le taux de transfert doit être de 9600
b. la parité doit être 8−N−1

Ne vous inquiétez car les valeurs par défaut correspondent au matériel PA−RISC. Si vous avez un PC, vous aurez probablement besoin de changer le taux de transfert mais il semble que les machines PA−RISC apprécient cette configuration.

#### 3.1.2.3. Changer de console

Avant de pouvoir utiliser la connexion série entre la machine Linux et celle PA−RISC, vous devez changer le mode de la console au niveau du système PA−RISC.

Le mode de la console par défaut est `graphic_1`. Pour changer cela, suivez les instructions ci−dessous:

1. Allumez votre machine PA−RISC ;
2. Lors du démarrage, le message suivant devrait apparaître:

   ```
   Searching for Potential Boot Devices.
   To terminate search, press and hold the ESCAPE key.
   ```

   Quand ce message apparaît, appuyez et maintenez enfoncée la touche **Echap** jusqu'à l'apparition du menu Options.

3. Par défaut, vous obtenez la console **BOOT_ADMIN**. Sur certaines 715, le menu Otpions aura cet aspect:

```
b)     Boot from specified device
s)     Search for bootable devices
a)     Enter Boot Administration mode
x)     Exit and continue boot sequence
?)     Help

Select from menu:
```

Choississez "a) Enter Boot Administration mode". Vous atteindrez le prompt "`BOOT_ADMIN>`". Tout le reste est à faire en mode **BOOT_ADMIN**. Maintenant, tout le monde est dans la console **BOOT_ADMIN** !

#### 3.1.2.3.1. Vérifier le mode console actuel

Tapez: **path console** pour voir quel mode console est utilisé par la machine en ce moment.

Si vous êtes en mode graphique, vous lirez "`Console path = graphic_1`".

Si vous êtes en mode série, vous lirez "`Console path = rs232_a.9600.8.none`" ou quelque chose de similaire.

Remarquez que certains modèles, vous pouvez trouver de petites différences mais l'idée générale reste la même. Si vous désirez plus de descriptions, écrivez−moi un courriel avec le type de machine vous avez et la sortie écran que vous avez.

#### 3.1.2.3.2. Changer pour le mode série

Pour obtenir le mode console série, tapez la commande suivante dans le prompt **BOOT_ADMIN**:

**path console rs232_a.9600.8.none**

Pour vérifier si la modification a été prise en compte, tapez **path console**. Vous devriez lire `Console path = rs232_a.9600.8.none`, indiquant que le système est configuré pour démarrer en mode console série. Par défaut, **reset** redémarrera le système avec les modifications.

#### 3.1.2.3.3. Comment obtenir la console série de démarrage sur 712 ?

Malheureusement, cale n'est pas possbile. Bien que les 712 sont configurées pour le développement en interne chez HP pour utiliser la console série, cette option n'est pas possible sur le terrain. Vous n'avez que la console graphique mais si vous désirez réellement avoir la console série, regardez les archives de la liste de diffusion de PA/Linux pour plus d'informations. Mais, vraiment pourquoi utiliser une belle 712 avec une console série alors que vous pouvez avoir X !

#### 3.1.2.3.4. Changer pour le mode graphique

C'est l'opération opposée de celle ci−dessus. En vérifiant votre chemin de console, vous devez lire`Console path = rs232_a.9600.8.none`. Maintenant, vous devez mettre en place le mode graphique *via* **BOOT_ADMIN**.

**path console graphic_1**

Vous devriez avoir à l'écran une image après la commande **reset**. Si l'écran n'a pas l'air de fonctionner, essayez d'appuyer sur la touche **Tab** pour changer la résolution de votre écran. Appuyant de nouveau sur cette touche, le moniteur basculera vers la résolution supérieure. Il faudra peut−être répéter cette opération plusieurs fois pour être totalement satisfait. Ce point est aussi vrai si vous changez de moniteur.

## 3.2. Preparer à démarrer depuis le réseau

Cette méthode pour démarrer une machine était plus utilisée dans les phases initiales de développement mais elle reste dans certains la seule manière possible. Maintenant, le démarrage avec le support réseau n'est plus nécessaire sauf dans certains cas spécifiques. C'est le pourquoi de cette section, d'ailleurs.

### 3.2.1. rbootd ou bootp ?

Tous les modèles « récents » peuvent utiliser **bootp**, incluant 715/100, 715/120, mais aussi toutes les 712.

### 3.2.2. Utiliser rbootd

#### 3.2.2.1. Installer rbootd

Si votre machine plus ancienne nécessite **rbootd** pour démarrer *via* le réseau, suivez les instructions suivantes pour faire et installer un noyau PA−RISC/Linux et démarrer votre système ainsi.

Pour les plus anciennes machines, incluant les 715 Scorpio 715, **rbootd** est nécessaire. Pour l'obtenir **rbootd**:

- pour toute distribution autre que Debian, vous devez récupérer cette archive *via* http://pateam.esiee.fr/download.html
- pour Debian, il existe un paquet sur http://www.debian.org/Packages/stable/net/rbootd.html

#### 3.2.2.2. Configurer rbootd

Par exemple, pour démarrer une 715, il faut que rbootd soit installé sur un système Linux (c'est le « serveur de démarrage ») où vous entreposerez votre image du noyau PA−RISC/Linux que vous désirez utiliser pour démarrer votre machine PA−RISC.

Une fois installé, il faut configurer votre service **rbootd** pour qu'il soit opérationnel avec votre système PA−RISC:

1. Dans `/etc/rbootd.conf`, ajoutez la ligne suivante:

   **ethernet addr *`bootfile`***

   Remplacez *`bootfile`* avec le nom de votre image noyau PA−RISC/Linux, généralement `"lifimage"`.
2. Maintenant, il nous faut l'adresse ethernet de votre machine PA−RISC en tapant **lan_addr** dans le prompt **BOOT_ADMIN** sur votre machine PA−RISC.

   Vous aurez en retour une série de chiffre similaire à `080009−7004b6`. Notez ce chiffre à la place *`ethernet addr`*.
3. Dans `/etc/rbootd.conf` sur votre serveur de démarrage, l'adresse MAC doit être formatée avec « : » pour séparer chaque octet, après avoir enlever le tiret qui les unissait. Par exemple:

   ```
   080009−7004b6
   ```

   devient

   ```
   08:00:09:70:04:b6
   ```

Ajoutez cette adresse ainsi formatée dans votre `/etc/rbootd.conf` sur votre serveur. Le fichier au final doit ressembler à ça:

```
# ethernet addr         boot file(s)        comments

08:00:09:87:e4:8f       lifimage_715        # PA/Linux kernel for 715/33
08:00:09:70:04:b6       lifimage_720        # PA/Linux kernel for 720
```

Cet exemple de `rbootd.conf` contient les adresses MAC et les images noyau pour deux différentes machines.

Une fois le fichier de configuration mis à jour, relancez le service **rbootd**.

Par défaut, **rbootd** va chercher les fichiers images dans `/var/lib/rbootd/`. Si vous utilisez l'archive pour les autres distributions, ce répertoire est `/export/hp/rbootd/`. Ainsi, vous devrez entreposez vos images noyau amorçables dedans. Si vous avez une bonne raison de ne pas aimer ce chemin, il ne vous reste plus qu'une solution, recompiler le source.

Le plus facile est bien sûr de mettre l'image dans le répertoire par défaut !

## 3.2.3. Utiliser bootp/tftp

### 3.2.3.1. Mettre en place bootp/tftp

Pour les utilisateurs Debian, vous avez juste à installer les paquets correspondants *via* ces commandes en tant que **root**:

```
apt−get install bootp
apt−get install tftpd
```

Le paquet **bootp** peut être remplacé par **dhcp**. Mais, ce manuel ne développera pas plus ce point. Pour votre information, après avoir installé le paquet **dhcp**, vous avez une section dans `/etc/dhcpd.conf` sur BOOTP et vous pouvez toujours essayer **man dhcp**.

Si vous avez besoin des paquets rpm, le meilleur conseil est d'aller visiter http://rpmfind.net. Il semble qu'il est nécessaire de créer un utilisateur **nobody** appartenant au groupe **nogroup** pour les utilisateurs de Red Had. Les fichiers contenus dans `/tftpboot/` doivent lui appartenir.

### 3.2.3.2. Configurer bootp/tftp

Utilisez les instructions suivantes pour vous servir de **bootp** sur votre serveur de démarrage:

Cette section vise les utilisateurs de Debian. Pour les autres distributions, la logique reste la même mais il existe des différences comme les répertoires par défaut. Vu que Debian maintient un paquet **bootp** à jour, il en sera question dans ce qui suit. Pour la configuration de **dhcp**, il n'est pas prévu de l'ajouter.

1. Configurez `/etc/inetd.conf` sur le serveur en ajoutant ces lignes:

   ```
   tftp            dgram   udp     wait    nobody  /usr/sbin/tcpd        /usr/sbin/in.tftpd /tftpboot
   bootps          dgram   udp     wait    root    /usr/sbin/bootpd      bootpd −i −t 120
   ```

   Ici, vous indiquez que `/boot` est le répertoire à utiliser pour le service **tftpd**. Ce choix est libre. Selon **man tftpd**, le répertoire par défaut est `/tftpboot`.

   Une fois sauvegardé, redémarrez **inetd** avec: **/etc/rc.d/init.d/inetd restart**.

2. Selon **man bootptab**, créez un fichier `/etc/bootptab` contenant:

```
[hostname]:hd=/tftpboot/Image:\
                              :rp=/usr/src/parisc/:\
                              :ht=ethernet:\
                              :ha=[mac address]:\
                              :ip=[ip address]:\
                              :bf=[boot filename]:\
                              :sm=255.255.255.0:\
                              :to=7200:
```

Remplissez le champ `[hostname]`, `[mac address]` et `[ip address]` avec les informations appropriées:

- [hostname] est le nom de la machine PA−RISC.
- [mac address] est l'adresse MAC de la même machine, que vous obtenez toujours en tapant **lan_address** dans le prompt `BOOT_ADMIN>`.
- [ip address] est l'adresse IP de la machine.
- [boot file name] est le nom du fichier contenant l'image du noyau amorçable.

Ce qui vous donne:

```
vodka:hd=/tftpboot:\
                              :rp=/usr/src/parisc/:\
                              :ht=ethernet:\
                              :ha=080069088717:\
                              :ip=140.244.9.208:\
                              :bf=lifimage:\
                              :sm=255.255.255.0:\
                              :to=7200:
```

### 3.2.4. Démarrer votre système PA−RISC/Linux *via* le réseau et avec une console série

Pour conclure cette méthode de démarrage plutôt réservée au développement, cette section décrit comment amorcer votre machine HP si vous utlisez un serveur de démarrage. Encore une fois, il est à noter que cette technique est de moins en moins utilisée. Pour une utilisation plus « classique », reportez vous directement à la partie concernant **PALO**.

Nous voici. C'est une liste de trucs et astuces au cas où vous désirez démarrer *via* le réseau. Vous avez fait tout ce qui a été décrit précédemment, votre machine de développement est lié à votre machine PA−RISC et vous avez une image amorçable, vous voilà fin prêt pour vous lancer ! Si tout est bien à sa place, ces quelques lignes devraient vous permettre de voir votre machine PA−RISC s'amorcer sur GNU/Linux.

1. Vérifiez la connexion entre les deux machines qu'elle soit réseau et série. Cela paraît peut−être naturel mais on sait jamais, vérifiez !
2. Démarrez **minicom** sur votre machine de développement.
3. Démarrez votre machine PA−RISC.
4. Regardez votre console de minicom. Quand le message suivant apparaît à l'écran, appuyez et maintenez enfoncé la touche **Echap** pour interrompre le démarrage de votre machine:

```
Searching for Potential Boot Devices.
To terminate search, press and hold the ESCAPE key.
```

5. Choisissez "`a) Enter Boot Administration mode`" dans le menu. Vous entrez dans le prompt **BOOT_ADMIN**.
6. Tapez cette commande: **boot lan**.
7. Regardez comment votre machine PA−RISC devient un système PA/Linux.Ta dah !

# 4. Construire son propre noyau PA−RISC/Linux

Pour compiler un noyau Linux, vous devez avoir un compilateur C et le source de Linux. Le premier élément n'est pas trivial à trouver car tout dépend de comment vous comptez compiler le source. Le second est plus simple car pour l'instant la version la plus récente se trouve sur le site CVS officiel. Tout d'abord, nous parlerons du compilateur **GCC**. Ensuite, la configuration de votre futur noyau sera abordé. Le dernier paragraphe couvrira l'installation du noyau résultant de la compilation.

## 4.1. Le compilateur GCC

Vous pouvez compiler votre kernel avec votre machine HP. Mais, sur celles qui commencent à dater, vous préférerez sûrement utiliser une machine plus puissante pour la compilation. Regardons ces deux choix:

### 4.1.1. Compilation native

Vu que Debian est la seule distribution qui gère l'architecture PA−RISC, si vous voulez utiliser les pouvoirs de la « *Super Cow* », vous devez avoir quelques savoirs de base sur la gestion des paquets.

#### 4.1.1.1. apt−get et ses amis

**apt−get** est un utilitaire en ligne de commande simple qui gère votre système de paquets Debian. Tout d'abord, Gustavo Noronha Silva a écrit le APT HOWTO que je vous conseille si le sujet vous intéresse. Ici, nous voulons juste compiler un noyau. Vu que le portage hppa n'est pas encore public, vous devez être relativement prudent avec les mirroirs que vous choisissez dans votre `/etc/apt/sources.list`. Par exemple, en Allemagne, vous pouvez utilisez ces paramètres:

```
# non−US packages
deb http://www.ftp.uni−erlangen.de/pub/debian/ unstable/non−US main non−free contrib
# Binary packages
deb http://gluck.debian.org/debian unstable main contrib non−free
# source packages
deb−src http://gluck.debian.org/debian unstable main contrib non−free
```

#### 4.1.1.2. mettre à jour votre gcc

Si vous utilisez votre propre machine HP, vous avez besoin seulement du fameux compilateur **GCC**. Le conseil est de conserver une différence nulle entre les versions des développeurs et la vôtre.

```
apt−get update
apt−get upgrade
```

Si ce n'est pas une simple mise−à−jour, les dépendances du paquet kernel−source sont les suivantes:

- binutils
- fileutils
- gcc
- libc−dev
- make

Une fois à jour, il faut configurer votre noyau.

#### 4.1.2. Compilation croissée

Pour compiler ainsi votre kernel, tout dépend de l'architecture de la machine qui est utilisée pour la compilation. Pour les PCs, vous pouvez téléchargez un cross compilateur prêt à l'emploi sur le serveur FTP de PA/Linux. Pour les autres architectures ou si vous voulez compiler votre propre chaîne de compilation, lisez le HOWTO de Carlos O'Donell.

### 4.2. Configuration du kernel

L'une des meilleures manières de gagner de la performance est d'avoir un boyau bien configuré. Pour la plateforme PA−RISC, **make oldconfig** est une sorte de configuration par défaut. Si vous voulez faire votre propre noyau, la première étape est de connaître votre matériel. Pour ce faire, allez voir la description de votre machine sur la base de données officielle du matériel ou le HP partsurfer.

Quand vous savez ce qu'il y a dans votre boîte et aussi l'utilisation de votre futur PA/Linux, lancez simplement **make menuconfig** ou une autre commande de configuration du noyau. Voilà une liste de menus que vous allez parcourir pour vérifier si votre configuration correspond à votre matériel:

- *Processor type* indique le modèle de votre processeur
- *General options* vous informe sur ce qui est actuellement activer dans votre kernel
- *Network device support* contient les informations pour votre carte réseau
- *Character devices* montre vos entrées/sorties.
- *Console drivers* est directement relié à votre choix de console
- *Sound* active votre carte son Harmony

Comme vous le voyez, les menus propres au matériel HP ne sont pas nombreux mais il existe de nombreuses dépendances entre eux. Maintenant, il faut configurer le noyau par rapport à la future utilisation de la machine. Voici une liste de menus que vous pourrez visiter pour configurer les services que vous désirez:

- *General setup* détient les formats binaires exploitables
- *Parallel port support* vous permet d'activer ou non votre port parallèle
- *Block devices* valide l'utilisation de disque mémoire aussi connu sous le terme *ramdisk* et du périphérique virtuel *loopback*. Vous en aurez sûrement pas besoin
- *File Systems/Network File Systems* est l'endroit pour désactiver l'utilisation de NFS

Une fois terminé, sauvegardez votre configuration. Le tout est écrit dans le fichier .config. Vous pouvez en copier une version car **make distclean** le détruit. À cette étape, vous n'avez plus qu'à laisser faire **make dep vmlinux** et si tout va bien, voilà votre nouveau kernel.

### 4.3. Installation du noyau

Si vous avez réalisé la compilation en natif, vous sauvegardez le dernier noyau opérationnel avec une extension du style « .bk » et remplacez ce noyau par celui récemment compilé avec le même nom. Redémarrez et essayez−le. Si cela ne marche pas, vous pourrez toujours démarrer sur la sauvegarde en utilisant **PALO** pour renommer le nom du noyau que vous désirez utiliser (cf le prochain chapître pour plus d'informations).

Si vous amorcez votre machine *via* le réseau, vous devez configurer **PALO** comme l'explique la prochaine section et lancer **make palo**.

# 5. Démarrer votre machine PA−RISC/Linux *via* PALO, le chargeur du noyau

## 5.1. Qu'est ce que PALO ?

**PALO** est deux programmes, un chargeur de noyau, qui est exécuté par le microcode HP depuis la mémoire, et un gestionnaire de démarrage, qui initialise et met à jour les périphériques amorçables tels que les disques durs. L'exécutable du chargeur de noyau PALO est conservé dans un fichier iplboot. "IPL" signifie dans le jargon HP pour *Initial Program Loader* pour programme de chargement initial. Le gestionnaire est appelé **PALO**, tout simplement comme pour x86 où le gestionnaire de démarrage lilo est surnommé **LILO**, cependant il est notable que PALO n'est pas à relancer à chaque fois que vous compilez un nouveau noyau, à la différence de **LILO**.

## 5.2. Que fait PALO ?

L'idée principale est de permettre de démarrer un noyau en lui passant les paramètres dont il a besoin. La pratique est un peu plus compliqué que ça. En fait, PALO peut aussi transformer un fichier vmlinux classique en une image amorçable incluant une RAMDISK ou bien une redirection NFSROOT. Ce fichier est baptisé lifimage. Cependant, il peut aussi rendre démarrable un disque dur en lui spécifiant le type de console que l'on désire utiliser ainsi que le chemin vers la partion racine. Nous allons voir chacune de ces utilisations précisément. Le point important est que le fichier lifimage contient un noyau Linux amorçable et vmlinux est le noyau lui−même et il a besoin de **PALO** pour être amoçable.

## 5.3. Comment fabriquer une lifimage avec RAMDISK ?

Le premier point est d'expliquer quand et pourquoi le faire. À une époque antérieure du projet PA/Linux, les lifimages étaient très utiles. En fait, vous les utilisez pour votre serveur de démarrage pour amorcer votre machine HP *via* l'instruction boot lan. L'avantage de la ramdisk est de décompresser son propre système de fichiers et d'être complètement indépendant. Le principal inconvénient est le fait que vous devez construire votre propre ramdisk si vous avez des contraintes de mémoire ou que des fichiers de configuration doivent être personnalisés. Maintenant, voyons comment faire cette lifimage. Tout d'abord, il faut le dernier source de PA/Linux. À partir de là, il vous faut un bon cross−compilateur et les répertoires linux et palo. Tout ce dont vous avez besoin est sur http://www.parisc−linux.org/. Vous faites votre **make menuconfig**. Ensuite, lancez **make palo**. L'idée est que dans le répertoire palo réside le `Makefile` de **PALO** dans lequel vous devez sélectionner quelle ramdisk vous désirez. Initialement, vous lisez:

```
# RAMDISK = ${PA}/ramdisk.bin
```

Ainsi, si vous désirez utiliser l'option **RAMDISK** contenue dans un fichier appelé par exemple ramdisk.bin et placé à la racine du source linux, vous n'avez qu'à décommenter cette ligne:

```
RAMDISK = ${PA}/ramdisk.bin
```

Après avoir configuré le `Makefile`, allez dans le répertoire linux et lancez le fameux **make palo**. Le résultat, un fichier lifimage, vous attends dans le répertoire **PALO**.

## 5.4. Comment fabriquer une lifimage avec NFSROOT ?

Cette méthode est largement utilisée car le noyau et le système de fichiers sont visibles depuis votre serveur. Il est très simple de tester un kernel. Vous avez juste à générer un nouveau noyau et le placer dans le bon répertoire. Au redémarrage, la machine HP ira utiliser le nouveau kernel *via* instruction **boot lan**. Pour obtenir l'option **NFSROOT**, c'est encore plus facile que que celle **RAMDISK**. Editez le `Makefile` de **PALO** et spécifiez l'adresse IP du serveur tftpboot. En fait, si votre serveur a comme IP 10.10.10.2, alors le fichier par défaut est correct car on peut y lire:

```
NFSROOT = 10.10.10.2
```

Pour une autre adresse, ce champ doit contenir les bons paramètres. Après avoir configuré le `Makefile`, vous pouvez aller dans le répertoire linux et lancer **make palo**. Le résultat, un fichier lifimage, vous attends dans le répertoire **PALO**.

## 5.5. Comment rendre amorçable une partition ?

Cette partie correspond aux fonctionnalités de **PALO** similaires à celles de **LILO**. **PALO** est principalement un programme qui permet à une machine HP de démarrer *via* un kernel sur son propre disque dur. Cette section va expliquer comment le faire proprement. Quand vous installez le paquet **PALO**, Paul Bame, l'auteur et le responsable, a mis une copie du `/etc/palo.conf` par défaut dans `/usr/doc/share/doc/palo/palo.conf`. Si vous désirez comprendre comment **PALO** fonctionne, vous devez juste lire ce fichier ! Cet échantillon vient de `palo.conf`. C'est la valeur par défaut. Nous vous conseillons de ne pas trop vous écarter du schéma utilisé. Cependant, le paramètre **recoverykernel** est le chemin du noyau à choisir lorsque vous désirez démarrer en mode sans échec ! Le suivant, **bootloader**, est le chemin de l'utilitaire qui est produit par **PALO** quand vous exécutez **make iplboot**. **init−partitioned** informe le système sur le périphérique que vous désirez rendre amorçable. L'effet est immédiat. Cela signifie que **PALO** va écrire sur les premiers octets de la partition. Un bon conseil est de vérifier cette information avec **fdisk**, pour être sûr de mettre à jour le bon disque. Finalement, le dernier paramètre est **commandline** ! Le premier chiffre indique quelle partition du disque **PALO** doit monter pour obtenir le noyau ! Logiquement, la chaîne suivante est le chemin absolu jusqu'au kernel. **HOME** and **TERM** sont des variables d'environnement qui sont passées à **init**. Elles ne sont pas obligatoires mais elles peuvent être utiles. le paramètre **root** dit au noyau quelle partition monter comme racine du système de fichiers en lecture seule au démarrage. Cale peut être piégeux quand vous avez deux disques attachés. Vous pouvez ajouter des paramètres intéressants comme **console** où vous spécifiez la sortie de votre console. Vous devez savoir que **console=ttyS0** est pour la console série et **console=tty0** pour celle graphique. Dans la dernière version de **PALO**, ce chemin est placé automatiquement et correctement. Sinon, rapportez votre problème à la liste de diffusion.

```
# The following arguments are set up for booting from /dev/sda3, specifically
# mounting partition 3 as root, and using /boot/vmlinux as both the
# recovery kernel, and the default dynamically-booted kernel.
# --recoverykernel=/boot/vmlinux
# --bootloader=/boot/iplboot
# --init-partitioned=/dev/sda
# --commandline=3/boot/vmlinux HOME=/ TERM=linux root=/dev/sda3
```

## 5.6. Comment se servir de PALO au démarrage ?

### 5.6.1. La théorie

Après avoir installé votre **PALO**, vous pourrez peut−être avoir envie de modifier les paramètres passés au noyau. Tout d'abord, vous devez savoir comment interagir avec **PALO** au démarrage. Pour certains modèles plus anciens, vous devez ajouter « ipl » à votre commande. dans la console de **BOOT_ADMIN**: **BOOT_ADMIN> boot pri ipl** Pour les machines plus récentes, le système vous demandera si vous désirez interagir avec IPL. Vous n'avez qu'à répondre par un « y ». Ensuite, vous voilà dans **PALO** avec une liste de tous les paramètres et un chiffre leur correspondant. Entrez le chiffre du paramètre que vous voulez changer. Appuyez sur Entrée et modifiez la valeur et validez. Le système va afficher la nouvelle liste de paramètres. Cette modification n'est pas permanente ! Pour l'inscrire, il faut que vous exécutiez **/sbin/palo** et il écrira sur le disque tous les paramètres contenus dans le fichier par défaut à savoir `/etc/palo.conf`. Si vous désirez ajouter un autre paramètre, choississez un paramètre et ajoutez le vôtre en les séparant par un espace. Pour en enlever un, il suffit de le choisir et d'effacer l'entrée complètement. Vous verrez que la liste en s'affichant de nouveau compte un paramètre en moins.

Pour plus d'informations à propos de **PALO**, allez lire le fichier lisezmoi de PALO. Cette section est largement inspirée de ce fichier écrit par Paul Bame et ma page actuelle sur **PALO** que vous trouvez à http://pateam.esiee.fr/palo.html.

### 5.6.2. Un exemple

Cet exemple est le travail de *Michael Damaschke*. Alors, c'est parti pour l'histoire de l'heureux utilisateur de PA/Linux démarrant un noyau aussi appelée *Je ne sais pas comment je peux configurer ma machine pour qu'elle démarre avec le noyau que je veux ?*.

Après avoir allumer votre machine et votre moniteur, un message apparaît à l'écran vous informant que le démarrage va se poursuivre dans quelque instants sauf si vous pressez sur la touche **Echap** pour interrompre le processus de démarrage automatique. À ce moment, maintenez enfoncée la touche **Echap**.

Selon votre modèle, vous devez appuyer pendant un certain temps la touche. Aussi, votre moniteur peut être trop long pour pouvoir afficher ce message à l'écran. Donc, si vous voyez les diodes de votre clavier clignotées, c'est le signal pour maintenir enfoncée la touche **Echap** key. Si vous avez encore des soucis, référez−vous à la section *Consoles*.

Maintenant, il existe des variantes pour atteindre le prompt de **BOOT_ADMIN**. Si vous avez une ancienne machine, vous obtenez un autre message vous informant que le microcode de la machine recherche tous les périphériques amorçables. Vous pouvez interrompre cette recherche en appuyant encore sur la touche **Echap**. Là encore, appuyez sur **Echap**.

Là, un menu s'affiche. Choississez le choix avec la touche **a** et validez. Maintenant, vous voilà dans le prompt **BOOT_ADMIN**. D'abord, annulez l'autodémarrage en tapant ceci dans le **BOOT_ADMIN**:

```
BOOT_ADMIN> autoboot off
```

et puis valider avec **Entrée**.

Après ça, vous devez dire au système sur quel périphérique SCSI vous voulez démarrer. Ce disque doit contenir une partition de type *f0* où le chargeur **PALO** réside.

Pour cet exemple, l'ancien noyau sera `vmlinux−2.4.9−32` et le nouveau `vmlinux−2.4.17−pa3`. Le format d'un périphérique SCSI est `SCSI.X.0` où *X* est l'ID du disque SCSI vous désirez atteindre. Par exemple:

```
BOOT_ADMIN> boot SCSI.5.0
```

Après cette commande **boot** et le SCSI−ID, vous devez ajouter **IPL** si vous avez une machine des séries 7xx pour dire que vous voulez interagir avec **IPL**. Si vous avez une machine plusrécente, le système vous demandera ce que vous voulez faire:

```
Interact with IPL (Y or N)?>
```

Maintenant, vous pouvez configurer les paramètres de démarrage de **PALO**. Par exemple:

```
BOOT_ADMIN> boot scsi.5.0 ipl
```

et puis validez avec **Entrée**.

Voilà le nouveau Menu où vous configurez dans la ligne 0 (par défaut) le numéro de la partition, le chemin et le nom du noyau. Voici un exemple:

```
HARD Booted.
palo ipl 0.92 root@spqr Mon Jun 25 23:03:13 CEST 2001
```

```
3/vmlinux-2.4.9-32 3585851 bytes @ 0x6d8f800
Current command line:
3/vmlinux-2.4.9-32 root=/dev/sda3 console=ttyS0 TERM=vt102
 0: 3/vmlinux-2.4.9-32
 1: root=/dev/sda3
 2: console=ttyS0
 3: TERM=vt102

Edit which field?
(or 'b' to boot with this command line)? 0
```

Vous demandez à **PALO** de démarrer le fichier `vmlinux−2.4.9−32` contenant le noyau sur la troisième partition du disque SCSI.5.0. Mais cette fois−ci, vous désirez un autre kernel. Alors, vous validez avec **Entrée** et vous modifiez la chaîne de caractère pour qu'elle corresponde avec votre souhait, ici `vmlinux−2.4.17−pa3`. Vous concluez la modification *via* **Entrée**. Puis, il redemande si vous désirez effectuer des changements, mettez simplement « b » pour lancer le nouveau kernel ! Encore un fois, confirmez avec **Entrée**, évitez de changer d'autres paramètres si vous en connaissez pas le sens ! Maintenant, **PALO** n'a plus de secret pour vous :−)

# 6. Que faut−il savoir de BOOT_ADMIN ?

**BOOT_ADMIN** est une console primaire où vous pouvez exécuter des commandes précises. Ici, vous devriez trouver tout ce dont vous avez besoin. Sur tout système HP à base de processeur PA−RISC, vous trouverez **BOOT_ADMIN**, l'affichage peut varier mais l'idée reste la même. C'est pourquoi cette liste n'est pas complète mais contient ce qu'il faut. Un autre point important est que chaque commande peut être invoquée par un raccourci. Il est indiqué par les lettres en majuscule. Il n'est pas standard d'un modèle à un autre donc j'utiliserais le nom complet à chaque fois.

## 6.1. Les commandes dans main

Ces commandes sont celles de base.

- **boot** doit être suivi d'un argument qui indique le chemin que vous désirez. Le chemin est la définition du périphérique par exemple `FWSCSI.6.0` ou `PRI` si vous avez initialisé la variable correctement.
- **path** affiche ou inscrit les chemins de référence. Vous n'avez qu'à taper: `path pri fwscsi.6.0`. Mais, vous pouvez aussi mettre les chemins de console (graphique/série) et aussi pour le clavier (ps2/hil).
- **search** est très utile. Elle vérifie automatiquement et affiche tous les chemins amorçables possibles. Dans les versions récentes, les résultats sont liés à un raccourci. Elle détecte même si on peut faire un **boot lan**.
- **display** affiche le menu actuel.
- **help** vous donne une idée générale de ce que font les commandes. Par défaut, vous pouvez avoir toutes les commandes en tapant `help main`.
- Sur tout système, vous avez une commande **reset**. Elle redémarre la machine avec les derniers paramètres fixés.

## 6.2. Les commandes dans configuration

Ces commandes sont disponibles dans le menu **configuration**. Ainsi pour les utiliser, vous devez tout d'abord entrer dans ce menu avec la commande **configuration**.

- **auto** vous montre si la machine démarre automatiquement par défaut ou si elle fait une recherche. Vous pouvez modifier ce comportement avec les mots−clés `ON` and `OFF`.
- **bootinfo** liste tous les paramètres de démarrage du systéme.
- **default** rétablit la configuration de base.

- **monitor** règle l'affichage en tapant `mo <path> <type>` qui indique le chemin de votre console et ensuite le type de résolution. Si vous ne le connaissez pas, avec `mo list`, vous saurez tout.

---

## 6.3. Les commandes dans information

Ici, vous accédez à toutes les informations du système. Allez dans le menu en demandant **information**.

- **all** affiche tout.
- **fwrversion** vous donne la version de votre microcode. Vérifiez si votre microcode est à jour grâce à ce fichier.
- **lanaddress** affiche l'adresse MAC.

---

## 6.4. Les commandes dans service

Ce menu est réservé aux gouroux PA−RISC. Vous ne trouverez rien d'utile pour un utilisateur final. Si vous pensez réellement le contraire, vous pouvez toujours m'écrire pour me faire connaître vos arguments.

---

# 7. Contributeurs de ce manuel

Les personnes suivantes ont contribué ou relu la version de Deb de ce manuel d'une manière ou d'une autre.

- David Alexander deVries`<adevries@thepuffingroup.com>`
- Philip Imperial Schwan `<pschwan@thepuffingroup.com>`

Pour la version de Thomas.

- Michael Damasche `<sps01@uni−koeln.de>` Thanks for your example
- Helge Deller `<deller@gmx.de>`
- Grant Grundler `<grundler@puffin.external.hp.com>`
- Richard Hirst `<rhirst@puffin.external.hp.com>`