

Installation de la Debian stable et unstable, sous linux 2.4 et 2.6, sur un ibook 2.2

David Decotigny

Revision : 1.40

Date: 2005/05/05 15:53:17

Résumé

Notes sur l'installation de GNU/linux Debian 3.0 R1 (aka Woody r1) à partir des CD sur un ibook 2.2 800MHz (acheté février 2003) avec carte video ATI Radeon 7500 (M7 LW), en dual-boot Mac OS X / debian, clavier français. Suivies de notes sur le passage en *unstable*, puis sur le passage au noyau linux 2.6.

Table des matières

1	Prérequis	4
2	Partitionnement	5
2.1	Informations préliminaires	5
2.2	Repartitionnement côté MacOS X	6
2.3	Booter linux	7
2.4	Partitionnement côté Linux	7
3	Installation du reste de la debian	10
4	Noyau	11
4.1	Linux 2.4	11
4.2	Linux 2.6	12
5	Ethernet	12
6	Serveur X	12
6.1	Version de base (<code>framebuffer</code>)	12
6.2	Version avec DRI	12
6.3	Sortie vers écran extérieur	13
6.4	Anti-aliasing	14

7	Clavier	14
7.1	Configuration du noyau	14
7.2	Config en console et sous X	15
7.3	Touches spéciales Apple pour le son et l'eject du CD	16
7.3.1	Debian stable : <code>ikeyd</code>	16
7.3.2	Debian testing/unstable : <code>pbuttonsd</code>	16
7.3.3	Raccourcis clavier sous X	16
7.3.4	<code>pbuttonsd</code>	16
7.4	Bug X debian unstable semaine du 20/02/2004	17
8	Souris	17
8.1	Configuration X/console	17
8.2	<code>gpm</code>	17
8.3	Émulation 3 boutons	17
8.4	Marre du mode "cliquer quand on rabaisse le doigt rapidement sur le <i>trackpad</i> "	18
8.5	Souris externe USB	18
9	Gestion de l'énergie	18
9.1	Noyau et applications de suivi	18
9.2	Outils de gestion de l'énergie	19
9.2.1	<code>pmud</code>	19
9.2.2	<code>pbuttonsd</code>	19
9.3	Améliorer l'autonomie	19
9.3.1	Fréquence du processeur	19
9.3.2	Limiter les accès en écriture au disque	21
10	Son	21
11	Performances disque dur	22
12	CD/Graveur	29
13	Video	29
14	Modem RTC interne	29
15	Modem ADSL de Free	30
16	Configuration d'une partition d'échange MacOS X / Linux	31
16.1	Sous Linux	31
16.2	Sous MacOS X	32
17	IEEE 802.11b (Airport/Wifi/Wavelan)	33
18	IEEE 1394 (firewire)	33

19	Mise à jour vers unstable	33
19.1	Changements par rapport à la stable	34
19.1.1	Touches spéciales	34
19.1.2	Mise en veille, gestion de l'énergie	34
20	Passage au noyau Linux 2.6	34
A	Remerciements	35
B	Support d'un memory stick	35
C	Pense-bête	35
D	Liens	37

Introduction

Voilà quelques notes que j'ai prises (et que je continue de prendre) durant l'installation. Je ne prétends pas donner **LA** solution, ces notes sont là surtout pour que je me souvienne comment j'ai fait, histoire de ne pas perdre trop de temps sur des détails qui m'ont déjà suffisamment énervé une première fois. Alors si ça peut servir à d'autres, tant mieux.

Je précise qu'une connaissance de debian est hautement recommandée (avoir les bons réflexes, savoir interpréter les messages d'erreur, savoir se débrouiller en console et sous X, avoir certaines bases sur la configuration d'une debian).

Je précise également que je n'ai jamais touché à un Mac de ma vie (je découvre, c'est chouette), que je ne suis abonné à aucune mailing-list linux/mac, ni à aucun newsgroup sur le sujet, que je suis énormément pollué par debian/PC, que les subtilités du clavier et de l'ADB me dépassent, que je ne suis pas 100% au point côté magie du partitionnement pour MacOS X et l'openfirmware, etc... donc il y'a sûrement des inexactitudes, des approximations, ou des complications.

En fin de fichier, il y a une série de liens que j'ai utilisés. On trouvera mes fichiers de config dans ce répertoire (linux et macosx) : <http://david.decotigny.free.fr/libre/ibook2-debian>

Sauf mention explicite, la suite de ce document se rapporte à la debian woody (3.0) sur un noyau linux 2.4. Toutes les indications sur le passage en unstable (principalement en fin de document) supposent qu'une installation de la woody a été préalablement réalisée. Et, de même, les indications sur le passage au noyau linux 2.6 (principalement en fin de document) supposent qu'on a effectué le passage en unstable.

Je reste bien évidemment ouvert a toutes les remarques, surtout constructives : mail david point decotigny at free point fr.

Nouveautés

28 Février 2004 : Passage au noyau 2.6 (cf section 20), bug clavier avec les certains XFree 4.3 de la unstable (semaine du 20 Février 2004, cf section 7.4), bidouille qui marche pour activation vers le VGA externe sans booter sous OS/X avant.

20 Mai 2003 : Passage en *unstable* (cf section 19).

1^{er} Mai 2003 : Modem ADSL USB de Free (Sagem Fast 800), Modem RTC interne, souris USB, memory stick USB, anti-aliasing sous X, fonctionnement à mi fréquence pour économiser la batterie.

Février 2003 : Console, clavier, X, CD, Son, souris, ...

Qu'est-ce qui (ne) marche (pas) ?

Au final, voilà l'état de la chose :

- noyaux 2.4 et 2.6, debian stable (woody) et unstable
- clavier français sous X (mapping MacOS) et console (mapping un peu différent), avec support des touches "spéciales" Apple (volume, eject CD)
- trackpad Ok sous X et avec gpm
- XFree86 avec support (DRI) ATI Radeon M7 LW (glxgears fait du 1200 fps en 16bpp environ, quelques centaines en 24bpp) et anti-aliasing des fontes
- PMU (équivalent APM) : la led blanche en façade bat quand on rabat l'écran (*ie* sommeil), et (surtout) l'ibook se réveille quand on le relève. Fonctionnement à mi-fréquence quand on est sur batterie (meilleure autonomie)
- ethernet interne Ok
- carte son interne Ok
- cdrom : lecture CD data/audio Ok. Gravure Ok
- modem interne Ok
- modem USB ADSL de Free (Sagem Fast 800) Ok
- firewire : pas pu tester, mais reconnu
- Airport (aka Wifi/802.11b) : je ne l'ai pas
- déport écran vers VGA Ok (option `CloneDisplay` de XFree86); vers S-Video ⇒ pas réussi à configurer correctement.
- partition d'échange (MSDOS) MacOS X / Linux montée au démarrage de MacOS X / Linux.
- Divers : souris USB, memory stick USB

1 Prérequis

- Un ibook 2. Le mien, acheté en Février 2003, est un ibook2.2, c'est marqué dans `/proc/cpuinfo` sous Linux. L'écran est un 14", et cela semble ne pas avoir d'importance pour l'installation.

- Les 2 CD d’installation de Mac OS X. J’ai aussi installé les 2 CD de “*restauration des applications*” Apple, utiles pour avoir les outils de développement (on n’en a pas besoin du tout pour la suite, mais c’est toujours intéressant à regarder) et aussi la suite bureautique Apple.
- Les 7 CD debian 3.0 r1, le premier étant le “Non US”. Je les ai fabriqués en utilisant jigdo depuis solaris (<http://www.debian.org/CD/jigdo-cd/>).
- Je conseille un autre ordinateur disposant d’une prise RJ45 (via câble croisé ou hub) pour le connecter à l’ibook durant les phases délicates de la configuration (clavier, X).

2 Partitionnement

Il faut réinstaller OS X fourni par défaut, puisqu’un repartitionnement du disque est évidemment nécessaire. Donc ne pas trop perdre de temps à configurer OS X, ni trop bosser dessus avant d’installer la debian, ça évite d’avoir à faire des sauvegardes.

2.1 Informations préliminaires

Il y a 3 choses à noter, dont certaines peuvent surprendre les habitués du monde PC-Intel :

- Il y a un ordre logique des partitions défini dans la table des partitions, qui peut être différent de l’ordre physique sur le disque. Avec l’utilitaire `mac-fdisk` sous linux, il y a moyen de changer l’ordre logique des partitions (commande ‘`r`’).
- OS X se contrefiche visiblement de l’ordre logique et physique dans lequel on met les partitions. Ce qui veut dire qu’on peut mettre la partition principale d’OS X à un endroit, et rajouter plus tard des partitions avant (et après) cet endroit.
- Il n’y a pas de notion de partition étendue/logique/primaire comme sous PC. Je ne connais pas la limite du nombre de partitions possibles, mais visiblement c’est pas mal (chez moi, ça va jusqu’à `hda20` sans problème).
- La phase de boot accorde de l’importance à l’ordre logique des partitions. S’il y a plusieurs partitions bootables, alors l’ibook bootera sur la première parmi ces partitions bootables, dans l’ordre logique de la table des partitions. On verra que c’est un paramètre à prendre en compte pour le partitionnement côté Linux.

Une fois tout ceci établi, les phases de partitionnement qui suivent ne devraient pas poser de problème. Le but de ces 2 manip est de créer une partition pour MacOS X (éventuellement d’autres pour MacOS 9 et Darwin), et de laisser de la place pour linux, sous la forme d’une partition libre, ou d’une partition bidon. Cet espace sera effacé plus tard lors de l’installation de Linux qui l’utilisera pour créer ses partitions de swap, la racine, `/usr`...

Il est conseillé (mais pas nécessaire) de mettre l’espace libre pour linux *avant* les partitions Mac (avec l’utilitaire MacOS décrit ci-dessous), car linux installe

un outil pour le multiboot (**yaboot**, équivalent a **Lilo** sur PC) dans une partition bootable spéciale, dite “*bootstrap*”, qui sera créée dans la partition laissée à linux : pour éviter d’avoir à modifier l’ordre logique des partitions plus tard, le mieux est de prévoir le coup le plus tôt possible, c’est-à-dire qu’il vaut mieux que cet espace soit *avant* les partitions MacOS, afin que le *bootstrap yaboot* soit *avant* (ordre logique) celles-ci, pour être booté en priorité (*cf* ci-dessus). Si on ne fait pas ça, vu qu’il y’a moyen de changer l’ordre logique des partitions avec l’outil de partitionnement sous linux, ce n’est pas grave (*cf* la suite).

Enfin, Linux n’est pas top au point sur les partitions HFS+ (*ie* HFS extended), et Mac ne sait pas lire les partitions linux (ext2/3). Donc pour partager des données entre Linux et Mac, il faut :

- soit installer MacOS X sur une partition UFS (*aka* Unix FS) lisible depuis linux (lecture seule semble-t’il) en sachant que ça posera des problèmes pour Mac OS 9 (... que j’ai pu lire à droite à gauche, mais je ne confirme pas) et aussi (on m’a dit) pour MacOS X (l’installation se bloque) ;
- soit on installera MacOS sur HFS+ (Ok avec OS X et OS 9) et on ajoutera une partition spéciale d’échange entre MacOS X et linux (par exemple au format MS-Dos, que linux et MacOS savent manipuler). C’est ce que j’ai fait (partition de 800M), voir section 16.

2.2 Repartitionnement côté MacOS X

Marche à suivre :

- booter sous OS X (disque dur)
- insérer le CD 1 d’install MacOS X
- une fenêtre apparaît, cliquer bouton “install”
- [la machine reboote toute seule sur le CD]
- Une fois rebooté sur le CD, dans le menu **install** en haut de l’écran, sélectionner **Disk Utility**
- une fenêtre avec une espèce d’arborescence des volumes (disques durs, CD, partitions) apparaît, sélectionner le disque dur (et pas une de ses partitions)
- sélectionner l’onglet **partition**
- on a moyen de diviser la grande partition qui existe par défaut en plusieurs autres plus petites, et éventuellement de supprimer des partitions qu’on vient de rajouter (boutons *divide/suppress* en bas).

On peut également modifier la taille des partitions directement en faisant glisser la glissière dessinée entre les partitions. À ce stade, les partitions qu’on crée ne seront validées (*ie* écrites sur disque) qu’au moment où on cliquera sur le bouton “**partition**”.

- avec cet utilitaire, l’objectif est de créer une partition pour MacOS X, plus petite que le disque, les autres pour Mac si besoin (MacOS 9, Darwin, ...), et de laisser de la place pour linux (sous la forme d’une partition bidon par exemple, ou d’une partition de type Free space). Puisqu’on ne peut modifier la taille d’une partition que si la glissière entre 2 partitions apparaît, on est obligé de découper la grosse partition qu’on avait à l’origine, en (au

moins) 2 plus petites (bouton **divide**). On doit au moins voir 2 partitions apparaître, sachant que l'une est pour linux et l'autre pour MacOS X. Si on veut installer MacOS 9 sur une partition séparée, on doit en faire apparaître une 3ème. Une 4ème pour Darwin si besoin, *etc...*

- il faut choisir un nom pour chaque partition. Par exemple “*MacOS X*” (et “*Debian*” si on a préféré définir une vraie partition pour debian plutot que laisser l'espace libre). C'est ce nom qui apparaîtra dans le bureau MacOS. La partition debian sera effacée plus tard, donc son nom est sans importance (... ne rien mettre dans cette partition bien sûr!).
- il faut choisir un type de partition pour chaque partition. MacOS X peut s'installer sur HFS Extended (HFS+), UFS (Unix FS, mais c'est déconseillé), et (à vérifier) HFS. MacOS 9 sur HFS+ et HFS seulement. La partition Linux, on lui met n'importe quel type (y compris Free Space) puisque de toutes facons elle sera supprimée lors de l'install linux, pour la diviser en partitions spécialement linux (on ne peut pas les créer avec l'utilitaire d'Apple).
- une fois la taille des partoches et leur type définis, on peut écrire ces changements sur le disque (bouton **partition**), et quitter l'utilitaire. Perso, à ce niveau j'ai créé 2 partoches : une HFS+ de 12G de nom “**Mac OS X**” pour MacOS X, et le reste en espace libre pour Linux. En sachant que dans l'espace pour Linux j'ai prévu de mettre la partoches d'échange MacOS X / Linux (voir section 16).

Ensuite, on continue l'installation de MacOS X, ce qui ne devrait pas poser de problème.

2.3 Booter linux

- Avoir partitionné le disque avec l'utilitaire de MacOS X décrit ci-dessus, avoir installé MacOS X, et avoir laissé de la place (Free Space ou partition speciale) pour linux
- être sous MacOS X
- insérer le CD Debian 1 (préférer le **non-US** si on a le choix)
- rebooter. Au moment du “**bong**” de boot, presser la touche ‘c’ du clavier, et la maintenir jusqu'à ce que le prompt de choix de la version de linux à installer apparaisse (un obscur “**Boot :**” je crois).
- taper :
`install24 video=ofonly`
(entrée)
- linux est en train de booter. La debian procède au choix du clavier, puis au partitionnement...

2.4 Partitionnement côté Linux

L'utilitaire `mac-fdisk` (mode texte) est lancé automatiquement par debian. Il ressemble beaucoup au `fdisk` du monde PC : commande ‘?’ pour avoir la liste des commandes possibles, commande ‘p’ pour afficher la liste des partitions, ‘w’

pour enregistrer les modifications sur le disque (à n'utiliser qu'à la toute fin du partitionnement), 'q' pour quitter l'utilitaire.

Attention, avec le clavier français chargé à ce niveau, `mac-fdisk` est très sensible aux fausses manip (genre taper 2 lettres là où il en attend une, taper une lettre là où il attend un chiffre, toucher aux signes spéciaux genre @, etc...). À ce niveau, si on fait une fausse manip et que le `mac-fdisk` se bloque, on peut switcher vers la console d'urgence avec la touche `commande` (aka celle avec une pomme) jusqu'au terminal où on nous dit "*press enter to activate this terminal*", ce qu'on s'empresse de faire pour jouer avec 'ps' et 'kill'. Pour revenir au terminal initial, on appuie sur `commande` (la liste des terminaux est circulaire). On peut aussi appuyer sur `option-fn-F2` (resp. `option-fn-F1`) pour récupérer directement le terminal d'urgence (resp. terminal initial) [la touche option est la touche marquée Alt].

Dans l'utilitaire `mac-fdisk`, marche à suivre :

- taper `p<entrée>`, ça affiche la liste des partitions. On s'aperçoit que Mac OS en a créé une bonne dizaine rien que pour lui. On note que les noms des types de partitions sont assez verbeux. Normalement, on devrait reconnaître la/les partitions MacOS. Si on a choisi de créer une partition spéciale debian, on devrait la reconnaître aussi, sinon on devrait reconnaître un espace libre avec le doux nom `Free Space`.
- si on avait créé une partition spéciale debian plutôt que de laisser en espace libre, on la supprime, sinon on ne fait rien. Chez moi, cette partition s'appelait `/dev/hda9` (et était de type HFS+ mais on s'en contrefiche), ce qui donne :
`d<entrée> # Pour delete`
`9<entrée> # Pour le numero de la partition`
- taper `p<entrée>`. Voilà, maintenant on a un endroit marqué 'Free Space' dans la liste des partitions. C'est là qu'on va rajouter les partitions debian. Chez moi, cet espace s'appelle `/dev/hda9`.
- La première partition à créer est la partition de *bootstrap* (qui fait une taille fixe standard non modifiable de 800ko), qui contiendra *yaboot* (ie équivalent Lilo) : commande `'b<entrée>'`. On nous demande à quelle adresse sur le disque elle doit commencer. `mac-fdisk` permet de se dispenser de taper l'adresse en termes de numéro de secteur, ce qui est pratique : on peut taper `9p<entrée>`, ce qui signifie "à l'adresse du premier secteur de `/dev/hda9`".
- Un petit `p<entrée>` à ce niveau devrait montrer la partition de *bootstrap* fraîchement créée (`/dev/hda9` chez moi), et une partition Free Space juste après (`/dev/hda10` chez moi). Les partitions qui étaient éventuellement après l'espace debian du début ont été décalées d'1 numéro. Ça ne pose pas de problème, car Mac OS X se fiche des numéros logiques. Par contre, si on a la partition principale MacOS X avant (ordre logique) la partition *bootstrap* qu'on vient de créer, il faut utiliser la commande `r<entrée>` pour échanger leurs 2 numéros logiques (plus simplement, on peut l'échanger avec la partition numéro 2, mais jamais avec la partition numéro 1, qui contient la table des partitions).

- 2ème partition à créer : le swap. C'est avec la commande `c<entrée>` que ça se passe. On nous demande une adresse de début, comme pour le *bootstrap* ⇒ on met ça en début de l'espace libre (qui s'appelle `/dev/hda10` chez moi maintenant), donc chez moi je tape `10p<entrée>`. On nous demande la taille, on répond par exemple "500m" pour une taille de 500 Mo. On nous demande ensuite un nom : taper obligatoirement "swap", et `mac-fdisk` ajustera le type comme il faut pour que la partition soit de type Linux Swap.
- Vérifier avec `p<entrée>` que la partition swap est de la bonne taille avec le type Linux Swap, et que le Free Space a été décalée d'un numéro de partition (*ie* elle s'appelle `/dev/hda11` chez moi).
- 3ème partition à créer : la racine linux. C'est toujours avec `c<entrée>` que ça se passe. Pareil : l'adresse de début c'est maintenant (chez moi) `11p<entrée>`, la taille comme on veut, le nom par exemple "root". Avec la commande `p<entrée>`, vérifier que le type de la partition créée est bien Linux native.
- on peut créer d'autres partitions, tous les noms sont permis même *root* (mais pas *swap*; ou alors il faut changer le type de la partition après coup). Même manip.
- Pour le reste : commande 'd' pour supprimer des partitions, 'c' pour en créer d'autres... on peut faire ça autant de fois qu'on veut, tant qu'on n'utilise pas la commande 'w'. Ne pas oublier la partition d'échange MacOS/Linux si on a choisi d'en mettre une ("prévoir" ne veut pas forcément dire "le faire maintenant"; ça peut rester Free Space pour l'instant. Voir section 16 ci-dessous).
- Une fois que `p<entrée>` affiche un résultat qui convient, taper `w<entrée>` pour écrire le partitionnement sur le disque.
- pour quitter, c'est `q<entrée>`.

Chez moi le *swap* fait 500M, la racine fait 250M (s'appelle "root"), `/var` aussi (s'appelle "var"), `/usr` fait 3G (s'appelle "usr"), `/tmp` fait 500M (s'appelle "tmp"), `/home` (contient `/usr/local` aussi) fait 10G (s'appelle "home"). La partition d'échange MacOS/Linux fait 800M, auquel cas `fdisk` donne :

```

/dev/hda
#                type name                length  base    ( size ) system
dump: name /dev/hda len 8
/dev/hda1      Apple_partition_map Apple           63 @ 1    ( 31.5k) Partition map
/dev/hda2      Apple_Driver43 Macintosh          56 @ 64    ( 28.0k) Driver 4.3
/dev/hda3      Apple_Driver43 Macintosh          56 @ 120   ( 28.0k) Driver 4.3
/dev/hda4      Apple_Driver_ATA Macintosh          56 @ 176   ( 28.0k) Unknown
/dev/hda5      Apple_Driver_ATA Macintosh          56 @ 232   ( 28.0k) Unknown
/dev/hda6      Apple_FWDriver Macintosh          512 @ 288   (256.0k) Unknown
/dev/hda7      Apple_Driver_IOKit Macintosh          512 @ 800   (256.0k) Unknown
/dev/hda8      Apple_Patches Patch Partition          512 @ 1312 (256.0k) Unknown
/dev/hda9      Apple_Bootstrap bootstrap          1600 @ 1824 (800.0k) NewWorld bootblock
/dev/hda10     Apple_UNIX_SVR2 swap          1024000 @ 3424 (500.0M) Linux swap
/dev/hda11     Apple_UNIX_SVR2 root           512000 @ 1027424 (250.0M) Linux native
/dev/hda12     Apple_UNIX_SVR2 usr           6291456 @ 1539424 ( 3.0G) Linux native
/dev/hda13     Apple_UNIX_SVR2 var           512000 @ 7830880 (250.0M) Linux native
/dev/hda14     Apple_UNIX_SVR2 tmp           512000 @ 8342880 (250.0M) Linux native

```

```

/dev/hda15      Apple_UNIX_SVR2 home          22528000 @ 8854880 ( 10.7G) Linux native
/dev/hda16      MSDOS Echange                 1638992 @ 31382880 (800.3M) Unknown
/dev/hda17      Apple_Boot MOSX_OF3_Booter   16384 @ 33021872 ( 8.0M) Unknown
/dev/hda18      Apple_Loader SecondaryLoader 1024 @ 33038256 (512.0k) Unknown
/dev/hda19      Apple_HFS Mac OS X          25565832 @ 33039280 ( 12.2G) HFS
/dev/hda20      Apple_Free                   8 @ 58605112 ( 4.0k) Free space

```

```

Block size=512, Number of Blocks=58605120
DeviceType=0x0, DeviceId=0x0
Drivers-
1: @ 64 for 23, type=0x1
2: @ 120 for 36, type=0xffff
3: @ 176 for 21, type=0x701
4: @ 232 for 34, type=0xf8ff

```

Pour fixer les idées, `df` donne :

Sys. de fich.	1K-blocs	Occupé	Disponible	Capacité	Monté sur
/dev/hda11	247919	206872	28247	88%	/
tmpfs	192804	0	192804	0%	/dev/shm
/dev/hda12	3096336	2676320	262732	92%	/usr
/dev/hda13	247919	181030	54089	77%	/var
/dev/hda14	247919	4319	230800	2%	/tmp
/dev/hda15	11087104	8502740	2021164	81%	/home
/dev/hda16	819264	46608	772656	6%	/mnt/échange
none	10240	644	9596	7%	/dev

3 Installation du reste de la debian

Prévoir un trombone pour éjecter les CD à la main (petit trou dans le lecteur de CD), car après le 1er reboot la commande `eject` n'existe plus.

Pas de différence notable avec les installations debian PC. Ne pas oublier d'initialiser chaque partition (`ext3` recommandé sur portable) avant de les monter, sinon `mount` explose et le reste aussi (mais ce n'est pas dramatique, il suffit de rebooter et reprendre l'installation à cet endroit). Le clavier français n'est pas terrible (il manque des touches, genre “#” et “|”, ce qui n'est pas bien pratique quand on veut traficoter des choses dans le terminal d'urgence).

Perso, j'ai installé la base en me limitant à `tasksel`, et je n'ai pas lesiné (vu la taille de `/usr`) : installé *dialup*, *laptop*, *dev C/C++*, *dev Tcl/Tk* (pour faire flan avec le `make xconfig` du noyau), `TEX/LATEX`, *French env*, *X*, et *desktops gnome/kde*. Puis j'ai supprimé les `isdn-utils`, `leafnode`, `xtel`, `irda-utils`, `wvdial`, `netenv`, `pcmcia` qui ne servent à rien.

`X` ne fonctionnera pas tout de suite, donc pas la peine de se fatiguer sur sa configuration, mais il vaut mieux l'installer quand même, même si on le remplace par une autre version plus tard. J'ai demandé à avoir `gdm` comme *session manager*.

Une fois l'installation réalisée, ne pas oublier de rajouter l'entrée `macosx` dans `/etc/yaboot.conf`, de vérifier celle pour linux, de rajouter `enablecdboot`

pour plus de confort (*cf* mon `/etc/yaboot.conf`), et de taper `ybin` pour mettre à jour la partition de *bootstrap*. Éventuellement rebooter pour tester tout ça. Après reboot, `gdm/X` essaieront de se lancer, planteront 1 fois, 2 fois, puis la `debian` s'en apercevra et demandera si on veut désactiver `gdm`, répondre *oui* pour continuer à configurer la suite en mode console.

Si on a besoin de caractères exotiques (genre `\`, `|`, `#`, `@`) pour les opérations qui suivent, accéder à l'`ibook` depuis une autre machine (`ssh`, `telnet`, ...), c'est plus confortable que de se casser la tête avec le clavier tout de suite. Il faut bien sûr demander à avoir configuré la carte réseau; perso, j'ai choisi de faire configurer par `dhcp` lors de l'install.

4 Noyau

4.1 Linux 2.4

Au moins pour que la mise en veille se passe correctement et pour faire marcher `X`, on a besoin de compiler son noyau. J'ai récupéré le noyau de Benjamin Herrenschmidt (<http://www.linuxppc.org/>) par `rsync`. Attention, pour que l'`ibook` se mette en veille et (surtout) se réveille quand on baisse/lève l'écran (avec `pmud`, *cf* section 9 plus loin), il faut au moins la version 2.4.20-ben5 (le `rsync` du 9 Février 2003 était Ok).

J'ai repris le `.config` hispalinux (<http://www.hispalinux.es/~data/ibook/configs/config-2.4.20-rc1-> tel quel avec juste 2 modifs *optionnelles* :

- virer l'option `CONFIG_MAC_ADBKEYCODES` (menu *Macintosh device drivers*), car elle ne facilite pas la config du clavier.
Si on oublie de le faire (j'ai volontairement oublié de le faire dans mon `.config`), il est toujours possible de rajouter :
`dev/mac_hid/keyboard_sends_linux_keycodes=1`
dans `/etc/sysctl.conf` (puis exécuter `/etc/init.d/procps.sh`).
- virer l'option `CONFIG_DRM_RADEON` parce que le module `DRM radeon` fourni avec le noyau ne supporte pas correctement la carte vidéo. On utilisera un `DRM` plus récent (*cf* section 6.2)
- activer `CONFIG_CPU_FREQ` et `CONFIG_CPU_FREQ_PMAC`
- activer tous les supports `ADB` et `HID` sauf `CONFIG_MAC_ADBKEYCODES`
- activer le support `ide-scsi`
- la carte réseau s'appelle `sungem`
- activer `apm_emu`
- activer le support son `DMA powermac (dmasound_pmac)`
- activer le `framebuffer radeon` (pas le `DRM`!)
- activer le support générique `DRM`, mais pas précisément la carte `radeon`
- activer le support `USB UHCI` et `OHCI` (1 seul des 2 est nécessaire mais je ne sais plus lequel), `CONFIG_USB_DEVICEFS`. Et `CONFIG_USB_STORAGE` si on envisage d'utiliser d'un `memory stack`
- compiler le support `MS DOS + VFAT` (en module si possible) si on envisage d'utiliser une partition d'échange `MacOS/Linux` (*cf* section 16).

J'ai mis pas mal de choses en modules là-dedans (mais pas le framebuffer, pour avoir le pingouin au reboot ;).

Une fois le noyau compilé (`make dep clean vmlinux modules modules_install`, ou ce qu'il faut à la sauce `make-kpkg`), copier `vmlinux` à l'emplacement `/boot/vmlinux-2.4.20-ben5` et `System.map` dans `/boot/System.map-2.4.20-ben5`. Modifier `/etc/yaboot.conf` pour ajouter ce nouveau noyau, en gardant l'ancien. Et lancer `ybin` pour mettre à jour la partition `bootstrap`. À partir de maintenant on peut rebooter. Si besoin, on peut rajouter `append="video=ofonly"` dans `/etc/yaboot.conf` (je ne l'ai pas mis) avant de faire `ybin`.

Si on a compilé pas mal de choses en modules, pour avoir `reseau/son/modem/XFree...` voir la suite.

4.2 Linux 2.6

Voir la section 20.

5 Ethernet

Ça marche bien de base. Si on l'a compilé en module, il faut modifier `/etc/motutils/aliases` (cf mes fichiers de config) :

```
alias eth0 sungem
```

Puis faire `update-modules`.

Pour configurer le reseau, on peut reprendre mon `/etc/network/interfaces`, et pour s'en servir il faudra alors faire :

```
ifup eth0 # Activation
ifdown eth0 # Desactivation
```

6 Serveur X

6.1 Version de base (framebuffer)

La version debian de base fonctionne bien avec le fichier de config `hispalinux`, mais seulement en utilisant le pilote `fbdev` (section `Driver` du fichier de config). Ça ne suffit pas pour utiliser la carte video un peu plus efficacement (par exemple pour bénéficier de l'accélération 3d).

6.2 Version avec DRI

Pour une version plus complète (accélération 2d et 3d), utiliser la version 4.2.99 de `XFree`, disponible en paquet Debian sur <http://people.debian.org/~daenzer/dri-trunk> (`apt-cdrom add` si on a gravé les paquets sur CD, ou `modif /etc/apt/sources.list :` ajouter par exemple `"deb http ://people.debian.org/~daenzer/dri-trunk/"`) :

```
apt-get update
apt-get install xlibmesa3-dri-trunk xserver-xfree86-dri-trunk
```

Linux 2.4 seulement (dri du 2.6 de base est Ok, cf section 20) : compiler le module DRM pour la Radeon :

```
apt-get install drm-trunk-module-src
cd n/importe/ou ; tar xzvf /usr/src/drm-trunk/le_fichier.tar.gz
cd modules/drm-trunk
make -f Makefile.linux TREE=chemin/vers/sources/linux/include radeon.o
mkdir /lib/modules/[version-noyau]/kernel/drivers/char/drm/
cp radeon.o /lib/modules/[version-noyau]/kernel/drivers/char/drm/
```

(chez moi, version-noyau, c'est 2.4.20-ben5)

Mettre a jour /etc/modutils/arch/powerpc.pmac en rajoutant :

```
pre-install radeon modprobe agpgart
```

Avant de faire `update-modules`.

Récupérer le fichier de config XF86 adapté (*cf* mes fichiers de config). Personnellement, je suis en 16bpp parce que sinon la 3D est très poussive (en 24bpp par exemple).

Éviter de lancer X à ce niveau. Car si on lance le nouveau X sur le nouveau noyau (`/etc/init.d/gdm restart` si on utilise `gdm`), d'abord il risque de râler parce qu'il ne trouve pas le clavier demandé, et si on a du bol et que la fenêtre de login `gdm` apparaît alors on est content. **Mais...** Le clavier est foireux. C'est là que la connexion depuis une autre machine est intéressante, parce qu'avec un beau X sous le nez inutilisable à cause du clavier, on ne peut plus revenir en console pour modifier le reste (*ie* le clavier). Bref, si on ne peut pas se connecter à l'ibook depuis une autre machine, éviter de lancer le serveur X maintenant.

6.3 Sortie vers écran extérieur

C'est de la bidouille mais ça marche pour une sortie vers un écran type VGA :

- pas besoin de booter sous OSX, pas besoin d'avoir branché le câble de déport VGA avant de booter. Si cependant on faisait l'un et/ou l'autre, sans doute qu'au niveau Linux ça marche d'office sans la bidouille qui suit.
- lancer XFree normalement (avec `UseFBDev True`), s'apercevoir que le CRT est dans les choux
- commenter `UseFBDev` du `XF86Config-4`, relancer X, s'apercevoir que le CRT est OK mais que le LCD n'est pas terrible (stries)
- revenir au `XF86Config-4` initial (*ie* décommenter `UseFBDev True`), relancer X, et constater que le LCD **et** le CRT sont Ok.

Je n'ai pas réussi à activer convenablement la sortie extérieure sur S-Video, ni en PAL ni en NTSC : le mode est tout à fait incorrect. Peut-être ça marche mieux avec XFree 4.3.

6.4 Anti-aliasing

Étant donnée la faible résolution de l'écran (1024x768), il vaut mieux utiliser l'anti-crênelage sous X, sinon les textes (en particulier sous mozilla) sont difficiles et désagréables à lire.

Installer le package `libgdkxft0` puis utiliser la session `gnome-AA` sous `gdm` si vous l'utilisez, ou sinon demander à charger `LD_PRELOAD=/usr/lib/libgdkxft.so` avant le reste du `Xsession`.

La config `gdkxft` de base est peu efficace, demander à être un peu plus violent : un fichier de configuration `$HOME/.gdkxft` comme celui qui suit fait très bien l'affaire :

```
!-*-fixed-*
```

7 Clavier

Un conseil, avoir un autre ordinateur connecté à l'iBook pour faire les essais, c'est toujours désagréable de se retrouver avec un clavier inutilisable et de devoir faire un reboot hard (bouton on/off pendant 5s). J'ai trouvé la config clavier particulièrement pénible, et visiblement la config clavier console influence la config clavier pour X...

7.1 Configuration du noyau

Première précaution à prendre, s'assurer que le driver clavier génère les *scan-codes* (ou les *keycodes*, j'y ai jamais rien compris) linux, plutôt que ceux dits "ADB" : si il y'a un "fichier" `/proc/sys/dev/mac_hid/keyboard_sends_linux_keycodes`, alors il faut s'assurer qu'il contient "1". Sinon faire :

```
echo 1 > /proc/sys/dev/mac_hid/keyboard_sends_linux_keycodes
```

Et pour que ce soit remis à 1 à chaque reboot, rajouter :

```
dev/mac_hid/keyboard_sends_linux_keycodes=1
```

Dans `/etc/sysctl.conf`. Attention, dans ce cas, il faudra que les scripts d'initialisation rechargent les scancodes après que ce fichier ait été traité (par `/etc/init.d/procfs.sh`); pour ça, rajouter le script `/etc/init.d/rekeymap.sh` suivant :

```
#!/bin/sh
/bin/loadkeys /etc/console/boottime.kmap.gz
```

Puis faire :

```
chmod 755 /etc/init.d/rekeymap.sh
update-rc.d start rekeymap.sh 60 S .
```

Autre solution (alternative à tout ce qui précède), compiler le noyau **sans** `CONFIG_MAC_ADBKEYCODES` (menu "Macintosh device drivers").

7.2 Config en console et sous X

Faire la manip suivante (inspirée de celle décrite par David Mentré dans http://www.linux-france.org/macintosh/clavier_rpm3.html) :

1. récupérer `ftp://ftp.linux-france.org/pub/macintosh/kbd-mac-fr.tar.gz`
2. clavier sous X
 - (a) recopier `fr_new` dans `/usr/X11R6/lib/X11/xkb/symbols/macintosh/`
 - (b) dans `/etc/X11/XF86Config-4`, les options pour la config du clavier sont :

```
Option "XkbRules"      "xfree86"
Option "XkbModel"     "macintosh"
Option "XkbLayout"    "fr_new"
Option "XkbOptions"   "ctrl:nocaps"
```

IMPORTANT : veiller à ce que `XkbDisable` ne soit pas activée!
 - (c) Le nouveau *layout* sera pris au prochain redémarrage ou après avoir tué le serveur X (`Ctrl-Option-Del` ou `/etc/init.d/gdm restart`).
3. clavier console
 - (a) recopier `mac-fr-ext_new.kmap.gz` dans `/usr/share/keymaps/mac/`
 - (b) faire `install-keymap /usr/share/keymaps/mac/mac-fr-ext_new.kmap.gz`
 - (c) le nouveau clavier est utilisé par la console.

Le clavier devrait se comporter convenablement. En console :

- le 'l' se tape avec `option-<`
- changer de console, c'est `commande-fn-F1..F7` (F7=serveur X)

Sous X :

- le 'l' c'est `shift-option-l`
- switcher vers une console, c'est... compliqué (gymnastique des doigts) : `commande-ctrl-fn-F1..F7` (ou `shift-ctrl-option-fn-F1..F7` si on a une config sans l'option `ctrl :nocaps`)
- tuer le serveur X : `commande-ctrl-del` (ou `shift-ctrl-option-del` pour une config sans `ctrl :nocaps`)

En console et sous X :

- pour changer la luminosité de l'écran, c'est avec `F1/F2`
- les touches de fonction, c'est avec `fn-F1..F12`
- les accolades c'est avec `option-(et)`, et les crochets avec `shift-option-(et)`.
- le '\ ' c'est `option-shift-/`
- le tilde, c'est `shift-option-n`

À partir de maintenant on devrait pouvoir relancer X sans surprise, et constater avec émerveillement que `glxgears` fait du 1200 fps en 16bpp (beaucoup moins en 24bpp), et que `tuxracer` marche à merveille (environ 45fps en 16bpp, un peu plus de 10 en 24bpp) !

7.3 Touches spéciales Apple pour le son et l'eject du CD

Par défaut, seules les touches spéciales de réglage de la luminosité sont accessibles (F1/F2), pas les touches de volume ni celle d'eject du CD. Il y'a plusieurs manières de s'en sortir pour en profiter. Pour ma part, je me contente de la première méthode ci-dessous (`ikeyd`), qui marche très bien.

7.3.1 Debian stable : `ikeyd`

C'est un petit *démon* dispo par défaut avec la debian, et qui scrute `/dev/input/event` à la recherche des touches frappées : ça court-circuite donc à la fois X et la console, et ça permet de gérer le son directement (*via* `/dev/mixer`), et le cd aussi (`/dev/cdrom` avec un `umount` préalable si besoin). Pour l'installer, un simple :

```
apt-get install ikeyd
```

suffit. C'est plus réactif que `pbuttonsd` (voir section 7.3.4), et ça ne fait juste que gérer les touches "spéciales" contrairement à `pbuttonsd` qui fait plein d'autres choses très bien supportées par ailleurs, comme remplacer `pmud`, gérer le rétro-éclairage...

7.3.2 Debian testing/unstable : `pbuttonsd`

En testing/unstable, préférer utiliser `pbuttonsd` qui s'occupe également de remplacer avantageusement `pmud` pour la gestion de la mise en veille. Voir la section 19.1.1.

7.3.3 Raccourcis clavier sous X

On peut aussi définir un raccourci du style `Ctrl-fn-F3/F4/F5/F12` dans son gestionnaire de fenêtres (`sawfish`) pour faire quasiment la même chose, au prix d'une combinaison de touches un peu plus compliquée : il suffit de définir un raccourci clavier de type `Run shell command` dans `sawfish` pour ces combinaisons, avec respectivement `/usr/local/bin/beep 0` (voir script `/usr/local/bin/beep` que je fournis), `/usr/local/bin/beep -10`, `/usr/local/bin/beep +10` et `eject` comme commandes associées.

7.3.4 `pbuttonsd`

Il y a une troisième solution, un peu usine à gaz celle-ci, car elle permet en plus de remplacer `pmud` (section 9) : le logiciel `pbuttonsd` (<http://www.cymes.de/members/joker/>) à récupérer, compiler et installer à la main.

Le fichier de config `pbuttonsd.conf` installé par défaut redéfinit en profondeur le mapping des touches de fonction pour qu'on n'ait pas à taper `fn` quand on veut accéder aux touches de fonction, et ce mapping reste sous MacOS X : en conséquence, avec ce fichier de configuration de base, il faut maintenant taper `fn` quand on veut accéder aux fonctionnalités luminosité/volume/eject, même sous MacOS X. Pour revenir au comportement "normal" (*ie* `fn-F1..F12`

pour accéder aux touches de fonction, F1/F2/F3/F4/F5/F12 direct pour accéder aux fonctionnalités de luminosité/volume/eject), modifier l'option `kbdmode` du `pbuttonsd.conf` pour préciser `fkeyslast`. Je n'ai pas réussi à faire fonctionner la touche F12 (eject du CD), même en repositionnant correctement son scancode à 108 (au lieu de 69 par défaut). Et au final je n'utilise pas `pbuttonsd` parce que je trouve que son temps de réaction à l'appui des touches de volume est plus lent qu'en utilisant `ikeyd` ou en définissant un raccourci à la mano dans le window manager (cf ci-dessus).

7.4 Bug X debian unstable semaine du 20/02/2004

Voir le bug debian #234113 : X cherche la map `fr_new` dans un sous-répertoire `pc` au lieu de tenir compte du fichier `XF86Config-4` qui indique d'utiliser un sous-répertoire `macintosh`. Bref, une fois X lancé, la touche `alt` n'est plus prise en compte, plus moyen de taper un `|`, `⌘` ..., et évidemment plus moyen de revenir en console... En attendant que ce soir réparé, une fois dans une `xterm`, faire `setxkbmap fr_new` et voilà.

8 Souris

8.1 Configuration X/console

Par défaut le *trackpad* marche très bien. J'ai juste rajouté les liens suivants :

```
cd /dev/input
ln -s mice mouse
ln -s mice psmouse
```

en mettant le `/etc/X11/XF86Config-4` à jour.

8.2 gpm

C'est pour faire du copier/coller avec la souris en console. Faire un petit :

```
apt-get install gpm
```

Ça marche nickel en console et sans perturber X, et sans avoir à se poser des questions avec le `gpm_data`.

8.3 Émulation 3 boutons

Pour l'émulation 3 boutons, j'ai repris la manip de David Mentré (ou presque), c'est-à-dire que j'ai rajouté :

```
dev/mac_hid/mouse_button_emulation=1
dev/mac_hid/mouse_button2_keycode=68
dev/mac_hid/mouse_button3_keycode=87
```

dans `/etc/sysctl.conf` (puis faire `/etc/init.d/procps.sh`). Ça veut dire que le bouton du milieu correspond à F10, et le bouton de droite à F11. Pour choisir d'autres touches pour les boutons supplémentaires, utiliser `showkey` en console.

8.4 Marre du mode “cliquer quand on rabaisse le doigt rapidement sur le *trackpad*”

Déjà qu'il n'y a qu'un bouton de souris sur un Mac, si c'est pour ne pas l'utiliser... Je n'aime pas que la bécane décrète pour moi que j'ai cliqué quand je n'ai fait que rabaisser un peu brutalement mon doigt sur le *trackpad*.

Pour modifier ce comportement (passage en mode dit `notap`), utiliser par exemple `/sbin/trackpad` du package `powerpc-utils` installé par debian par défaut. J'ai rajouté le script d'initialisation `/etc/init.d/trackpad` pour que le *trackpad* soit configuré en mode `notap` à chaque reboot :

```
#!/bin/sh -e

case "$1" in
start|restart)
/sbin/trackpad notap
;;
stop|reload|force-reload)
;;
esac
```

L'installer avec `update-rc.d trackpad start S 60 .`

Sinon, si on utilise `pbbuttosd` (voir section 7.3.4), il suffit de modifier l'option `tpmode` du `pbbuttosd.conf` pour y mettre `notap`.

8.5 Souris externe USB

Suffit de brancher, et hop, ça marche!

9 Gestion de l'énergie

C'est pour que le cœur (la led blanche en façade) de l'ibook batte quand on rabat l'écran, et (surtout) qu'elle s'éteigne et que tout revienne en ordre quand on le relève... Pour faire cette prouesse technologique, le service de gestion de l'énergie adapté s'appelle `PMU` sur l'ibook, et non pas `APM` qui est installé par debian par défaut.

9.1 Noyau et applications de suivi

Il faut un noyau récent ($\geq 2.4.20$ -ben5), et `pmud` ou `pbbuttosd` (*cf* section 9.2 ci-dessous) pour que ça marche au poil. Si ce n'est pas le cas, il faut récupérer une version récente, et se le compiler aux petits oignons.

Important : pour que ça marche *vraiment*, c'est-à-dire pour que l'ibook se réveille correctement quand on relève l'écran, il faut que le périphérique caractères `/dev/apm_bios` existe (il permet d'indiquer au pilote X de la Radeon que le portable s'apprête à s'endormir quand on rabaisse l'écran, histoire qu'il retrouve ses petits au réveil). Il est créé par `tasksel` par défaut quand on choisit l'ensemble d'applications `laptop` lors de l'installation de la debian. Mais, s'il n'existe pas, faire :

```
mknod /dev/apm\_bios c 10 134
chmod 770 /dev/apm\_bios
```

Après ces vérifications, quand on rabat l'écran, le portable se met en sommeil et la led blanche s'allume. Et quand on relève l'écran, la bécane se rallume, toute fraîche (que ce soit sous X ou en console) ! Si on a oublié de créer `/dev/apm_bios`, l'ibook ne se réveille tout simplement pas quand on relève l'écran...

Sous gnome, utiliser l'applet `battery_applet` qui marche très bien pour savoir où en est la batterie (et en plus ça signale quand elle passe dessous les 5%). Il y a aussi une applet spéciale `pmu`, mais je la trouve trop grosse et pas jolie.

9.2 Outils de gestion de l'énergie

9.2.1 pmud

Première solution, installer `pmud` fourni en standard avec la debian :

```
apt-get install pmud pmud-utils
```

en demandant de virer `apmd` installé par défaut par `tasksel`.

J'ai juste modifié un peu la config pour que l'ibook ne se mette pas en veille quand on rabat l'écran alors qu'il est sur le secteur, mais ce n'est pas fondamentalement nécessaire : ça se passe dans `/etc/default/power` :

```
PMUD_FLAGS="-a -K"
```

Puis faire `"/etc/init.d/pmud stop; /etc/init.d/pmud start"`.

9.2.2 pbbUTTONSD

Deuxième solution : récupérer, compiler et installer `pbbUTTONSD` (voir section 7.3.4), et lui demander de remplacer `pmud` (option de configuration `replace_pmud`). Personnellement, je préfère n'utiliser que `pmud` de base.

9.3 Améliorer l'autonomie

9.3.1 Fréquence du processeur

Pour que `pmud` fasse tourner le processeur 2 fois moins vite quand on est en mode nomade (*ie* sur batterie), il faut compiler son noyau avec les options `CONFIG_CPU_FREQ` et `CONFIG_CPU_FREQ_PMAC`, et avoir un fichier `/etc/power/pwrctl-local` qui ressemble (merci David Mentré) à :

```

#!/bin/bash
logger=/usr/bin/logger

case "$1" in
minimum)
$logger -p daemon.info -t pwrctl-local "minimum power $2"
    case "$2" in
    ac)
        echo -n "0%50%100%performance" > /proc/cpufreq
echo 400000 > /sys/devices/system/cpu/cpu0/cpufreq/scaling_min_freq
echo 800000 > /sys/devices/system/cpu/cpu0/cpufreq/scaling_max_freq
echo performance > /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
        ;;
    *)
        echo -n "0%50%100%powersave" > /proc/cpufreq
echo 400000 > /sys/devices/system/cpu/cpu0/cpufreq/scaling_min_freq
echo 800000 > /sys/devices/system/cpu/cpu0/cpufreq/scaling_max_freq
echo powersave > /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
        ;;
    esac
;;
medium)
$logger -p daemon.info -t pwrctl-local "medium power $2"
    case "$2" in
    ac)
        echo -n "0%50%100%performance" > /proc/cpufreq
echo 400000 > /sys/devices/system/cpu/cpu0/cpufreq/scaling_min_freq
echo 800000 > /sys/devices/system/cpu/cpu0/cpufreq/scaling_max_freq
echo performance > /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
        ;;
    *)
        echo -n "0%50%100%powersave" > /proc/cpufreq
echo 400000 > /sys/devices/system/cpu/cpu0/cpufreq/scaling_min_freq
echo 800000 > /sys/devices/system/cpu/cpu0/cpufreq/scaling_max_freq
echo powersave > /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
        ;;
    esac
;;
maximum)
$logger -p daemon.info -t pwrctl-local "maximum power $2"
    case "$2" in
    ac)
        echo -n "0%50%100%performance" > /proc/cpufreq
echo 400000 > /sys/devices/system/cpu/cpu0/cpufreq/scaling_min_freq
echo 800000 > /sys/devices/system/cpu/cpu0/cpufreq/scaling_max_freq
echo performance > /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
        ;;
    *)
        echo -n "0%50%100%powersave" > /proc/cpufreq
echo 400000 > /sys/devices/system/cpu/cpu0/cpufreq/scaling_min_freq
echo 800000 > /sys/devices/system/cpu/cpu0/cpufreq/scaling_max_freq
echo powersave > /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
        ;;
    esac
$logger -p daemon.info -t pwrctl-local "skipping pwrctl-main"
exit 1
;;

```

```

warning)
/usr/bin/wall "Low battery, system going down any minute now"
;;
lid-closed)
$logger -p daemon.info -t pwrctl-local "lid-closed $2"
;;
lid-opened)
$logger -p daemon.info -t pwrctl-local "lid-opened $2"
;;
sleep)
$logger -p daemon.info -t pwrctl-local "sleep $2"
# Some modules don't support sleep/resume
/sbin/rmmod -a
sleep 3
/sbin/rmmod -a
sync;sync;sync
fblevel off
;;
wakeup)
$logger -p daemon.info -t pwrctl-local "wakeup $2"
/sbin/rmmod -a
    case "$2" in
        ac)
            echo -n "0%50%100%performance" > /proc/cpufreq
echo 400000 > /sys/devices/system/cpu/cpu0/cpufreq/scaling_min_freq
echo 800000 > /sys/devices/system/cpu/cpu0/cpufreq/scaling_max_freq
echo performance > /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
            ;;
        *)
            echo -n "0%50%100%powersave" > /proc/cpufreq
echo 400000 > /sys/devices/system/cpu/cpu0/cpufreq/scaling_min_freq
echo 800000 > /sys/devices/system/cpu/cpu0/cpufreq/scaling_max_freq
echo powersave > /sys/devices/system/cpu/cpu0/cpufreq/scaling_governor
            ;;
    esac
fblevel on
;;
*)
$logger -p daemon.error -t pwrctl-local "invalid arg $1 $2"
exit 2
;;
esac

exit 0

```

9.3.2 Limiter les accès en écriture au disque

Pour limiter les mises à jour sur disque lors des accès en lecture, spécifier l'option `noatime` dans le `/etc/fstab`.

10 Son

Rien à signaler si le noyau a été compilé comme indiqué ci-dessus. Si on a compilé en modules, rajouter dans `/etc/modutils/aliases` :

```
alias char-major-14 soundcore
alias sound-slot-0 dmasound_pmac
alias sound-slot-1 dmasound_pmac
alias char-major-14-3 dmasound_pmac
alias /dev/dsp dmasound_pmac
alias sound-service-0-0 i2c-keywest
alias sound-service-0-3 i2c-keywest
alias sound-service-1-0 i2c-keywest
alias char-major-14-0 i2c-keywest
alias /dev/mixer i2c-keywest
```

Puis faire `update-modules`.

Pour modifier le volume, utiliser par exemple `aumix`. Attention, `gnome` par défaut utilise `ESD`, ce qui peut poser des problèmes si les logiciels veulent utiliser directement `OSS` : `killall esd` (et/ou intervention dans le panneau de contrôle `gnome`) peut résoudre le problème. Autre détail : on entend un craquement au moment où on relève l'écran quand l'`ibook` était en sommeil.

11 Performances disque dur

J'ai rajouté ce script `/etc/init.d/hdparm` :

```
#!/bin/sh

set -e

#In certian cases you may wish to run this script twice. Once at S07
#and once later in the boot process. If you do this call /etc/init.d/hdparm
#again from rcS.d with a name such as S27hdparm.second.
#
#See /usr/share/doc/hdparm/README.Debian for more details.

case "$0" in
    *hdparm)
        FIRST=yes
        ;;
    *)
        FIRST=no
        ;;
esac

MYNAME="$0"

report()
{
    echo "${MYNAME}: $*"
}
```

```

}

report_error()
{
    echo "${MYNAME}: Error: $" >&2
}

report_error_and_exit()
{
    report_error "$*. Exiting."
    exit 1
}

case $1 in
    start|restart|reload|force-reload)
        ;;
    stop)
        exit 0
        ;;
    *)
        echo "Usage: $MYNAME {stop|start|restart|reload|force-reload}" >&2
        exit 3
        ;;
esac

if [ "$FORCE_RUN" != 'yes' ]; then
    if grep -w -q "nohdparm" /proc/cmdline ; then
        report "*** Skipping setup of disc parameters. ***"
        exit 0
    fi

    raidstat=OK
    if [ -e /proc/mdstat ]; then
        if grep -iq resync /proc/mdstat; then
            raidstat=RESYNC
        fi
    elif [ -e /proc/rd/status ]; then
        raidstat='cat /proc/rd/status'
    fi

    if ! [ "$raidstat" = 'OK' ]; then
        report_error "*** RAID status not OK. Exiting. ***"
        exit 0
    fi
fi

```

```

echo -n "Setting parameters of disc:"

DISC=
DEFAULT=
OPTIONS=
DEF_QUIET=
OPT_QUIET=

# set_option() adds $1 to the $OPTIONS list if in a disk stanza
#           and adds $1 to the $DEFAULT list if not in a disk stanza
#
# the block beginning:
#   if test x${i%#??} != x${1%#??}; then
# checks to see if $1 is already in the list and
#   if so removes the first instance

set_option()
{
  if test -n "$DISC"; then
    NEW_OPT=
    for i in $OPTIONS; do
      if test x${i%#??} != x${1%#??}; then
        NEW_OPT="$NEW_OPT $i"
      else
        NEW_OPT=${NEW_OPT%-q}
      fi
    done
    OPTIONS="$NEW_OPT $OPT_QUIET $1"
  else
    NEW_DEF=
    for i in $DEFAULT; do
      if test x${i%#??} != x${1%#??}; then
        NEW_DEF="$NEW_DEF $i"
      else
        NEW_DEF=${NEW_DEF%-q}
      fi
    done
    DEFAULT="$NEW_DEF $DEF_QUIET $1"
  fi
}

eval_value()
{
  case $1 in
    off|0)
      set_option "$2"0
  esac
}

```

```

        ;;
    on|1)
        set_option "$2"1
        ;;
    *)
        return 1
        ;;
esac
}

ITEMS_ECHOED=no

# Get blocks as far as the drive's write cache.
/bin/sync

# Set options for a group of disks in /etc/default/hdparm
[ -e /etc/default/hdparm ] && . /etc/default/hdparm

if [ -n "$harddisks" -a -n "$hdparm_opts" ]; then
    for drive in $harddisks; do
        /sbin/hdparm -q -f $drive
        hdparm -q $hdparm_opts -q $drive
        if [ "$ITEMS_ECHOED" = yes ] ; then
            echo -n ", $drive"
        else
            echo -n " $drive"
            ITEMS_ECHOED=yes
        fi
    done
fi

egrep -v '^[[:space:]]*(#|$)' /etc/hdparm.conf |
{
    while read KEY SEP VALUE; do
        if [ "$NEXT_LINE" != 'go' ]; then
            case $SEP in
                '{')
                    case $KEY in
                        command_line)
                            NEXT_LINE=go
                            unset DISC
                            unset OPTIONS
                            unset OPT_QUIET
                            ;;
                        *)
                            DISC=$KEY

```

```

        OPTIONS=$DEFAULT
        OPT_QUIET=$DEF_QUIET
        WAS_RUN=0
        ;;
    esac
    ;;
=)
case $KEY in
    read_ahead_sect)
        set_option -a$VALUE
        ;;
    lookahead)
        eval_value $VALUE -A
        ;;
    bus)
        eval_value $VALUE -b
        ;;
    apm)
        set_option -B$VALUE
        ;;
    io32_support)
        set_option -c$VALUE
        ;;
    dma)
        eval_value $VALUE -d
        ;;
    defect_mana)
        eval_value $VALUE -D
        ;;
    cd_speed)
        set_option -E$VALUE
        ;;
    mult_sect_io)
        set_option -m$VALUE
        ;;
    prefetch_sect)
        set_option -P$VALUE
        ;;
    read_only)
        eval_value $VALUE -r
        ;;
    spindown_time)
        set_option -S$VALUE
        ;;
    interrupt_unmask)
        eval_value $VALUE -u

```

```

;;
write_cache)
    eval_value $VALUE -W
;;
transfer_mode)
    set_option -X$VALUE
;;
acoustic_management)
    set_option -M$VALUE
;;
keep_settings_over_reset)
    eval_value $VALUE -k
;;
keep_features_over_reset)
    eval_value $VALUE -K
;;
chipset_pio_mode)
    set_option -p$VALUE
;;
ROOTFS)
    ROOTFS=$VALUE
;;
*)
    report_error_and_exit "unknown option $KEY"
;;
esac
;;
"")
case $KEY in
    '}'')
        if [ -z "$DISC" ]; then
            if [ "$WAS_RUN" != '1' ]; then
                report_error_and_exit "No disk enabled. Exiting"
            fi
        fi
        if [ -n "$OPTIONS" ]; then
            if [ -b "$DISC" ]; then
                if [ -n "$ROOTFS" ]; then
                    if [ "$FIRST" = 'yes' -a "$DISC" != "$ROOTFS" ]; then
                        continue
                    fi
                    if [ "$FIRST" = 'no' -a "$DISC" = "$ROOTFS" ]; then
                        continue
                    fi
                fi
            fi
        fi
        # Flush the drive's internal write cache to the disk.

```

```

        /sbin/hdparm -q -f $DISC

        /sbin/hdparm $OPTIONS $DISC
        if [ "$ITEMS_ECHOED" = yes ] ; then
            echo -n ", $DISC"
        else
            echo -n " $DISC"
            ITEMS_ECHOED=yes
        fi
    fi
    ;;
quiet)
    if [ -n "$DISC" ]; then
        OPT_QUIET=-q
    else
        DEF_QUIET=-q
    fi
    ;;
standby)
    set_option -y
    ;;
sleep)
    set_option -Y
    ;;
disable_seagate)
    set_option -Z
    ;;
*)
    report_error_and_exit "unknown option $KEY"
    ;;
esac
;;
*)
    report_error_and_exit "unknown separator $SEP"
    ;;
esac
else
    $KEY $SEP $VALUE
    NEXT_LINE=no-go
    WAS_RUN=1
fi
done

if [ "$ITEMS_ECHOED" = yes ] ; then
    echo "."

```

```

else
    echo " (none)."
fi
}

```

Puis faire “`chmod 755 /etc/init.d/hdparm; update-rc.d start hdparm 60 S .`”.

12 CD/Graveur

Dans `/dev`, j’ai modifié/rajouté les liens `/dev/cdrom` et `/dev/dvd` vers `/dev/scd0` puisque j’utilise `ide-scsi` pour m’en servir comme graveur. Puisque j’ai demandé `ide-scsi` en module, il a fallu modifier `/etc/modutils/arch/powerpc.pmac` avec :

```

pre-install sg modprobe ide-scsi
pre-install sr_mod modprobe ide-scsi

```

Puis faire `update-modules`.

Pour éjecter les CD, c’est avec la commande `eject` que ça se passe (ou `cdeject`, package `cdtool` pas installé par défaut), et on peut utiliser les touches de fonction prévues à cet effet en définissant des raccourcis, ou en espérant que `pbuttonsd` fonctionne correctement (cf section 7.3.4).

Pour lire des CD audio, utiliser `xmms` avec le plugin *audio CD reader*. Pour lire des dvd : `xine`, `ogle`, ou `mplayer /dev/dvd`.

Le graveur marche sans problème avec le support `ide-scsi` (testé et approuvé).

13 Video

`xine` et `ogle` marchent (`apt-get install...`). `mplayer` et `videolan` compilent et marchent sans problème (pour `gnome`, ne pas oublier `apt-get install libesd0-dev` avant pour que `mplayer` détecte et compile le support ESD sous `gnome`). Par contre les perfs ne sont pas au rendez-vous : la lecture de DVD sous ces 4 logiciels est un peu saccadée. La lecture de DIVX;) par contre est tout à fait fluide avec `xine`.

14 Modem RTC interne

Après avoir installé un pilote récent (fin Avril 2003) du modem de Marc Boucher (<http://www.mbsi.ca/cnxtlindrv/>), le modem interne de l’ibook fonctionne correctement. Marche à suivre :

- récupérer le `hcfusbmodem-xxxx.tar.gz` (cf url ci-dessus)
- `tar zxvf hcfusbmodem-xxxx.tar.gz`
- `su #` eh oui, il fait des choses bizarres avec `/dev` et cie...

- `make install`
- `hcfusbconfig` # il pose des questions avant de compiler le module

Puis s'inspirer des configurations que free propose sur son site (<http://support.free.fr/docs/lin>) avant de faire `pon` ou `pon paris-ppp` suivant le nom que vous donnez au fichier créé dans `/etc/ppp/peers/`. Et voilà la connexion établie! Pour se déconnecter, faire `poff`. Un petit `modprobe ucfusbserial` peut être nécessaire dans certains cas si `pon` échoue.

15 Modem ADSL de Free

Depuis fin Avril 2003, le modem ADSL USB Sagem Fast 800 que Free fournit à ses abonnés est supporté par Linux sur l'ibook.

Installer une version récente du package `hotplug` (par exemple : <http://www.fs.tum.de/~bunk/de>)
Avoir les headers du noyau sur le système (c'est le cas si vous avez compilé votre noyau : dans ce cas, assurez-vous qu'ils soient accessibles sous `/usr/src/linux/include`, sinon installez le package `kernel-headers`).

Récupérer la version 1.0.4pre5 ou plus récente du driver `eagle-usb` sur <http://www.eagle-usb.fr.st>, ne pas brancher le modem tout de suite, décompresser le driver, puis faire `make install`, répondre aux 2 questions (login/mot de passe). Et hop : brancher le modem, attendre 10s, faire `startadsl` et ça marche. Par la suite, lorsqu'on change de noyau, éviter de refaire le `make install` (pour éviter de répondre aux questions login/passwd dont il connaît déjà la réponse) : se contenter d'un `cd driver ; make install`.

Pour le fun (ce n'est pas obligatoire), j'ai juste modifié `/usr/sbin/startadsl` pour que le `ifconfig` avec l'adresse 192.168.1.30 bidon par défaut devienne plus simplement : `/sbin/ifconfig $INTERFACE up`.

Pour information, voici ce que donne `/sbin/ifconfig`

```
adsl0      Lien encap:Ethernet HWaddr 00:60:4C:10:FD:B9
          UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
          RX packets:2712 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2585 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:100
          RX bytes:1151880 (1.0 MiB) TX bytes:229038 (223.6 KiB)

ppp0      Lien encap:Protocole Point-à-Point
          inet adr:82.64.69.69 P-t-P:192.168.254.254 Masque:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST MTU:1492 Metric:1
          RX packets:2508 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2374 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:3
          RX bytes:1100931 (1.0 MiB) TX bytes:175392 (171.2 KiB)
```

Et voici ce que dit `plog` :

```
May  1 22:10:12 baloo pppd[21431]: pppd 2.4.1 started by root, uid 0
```

```

May  1 22:10:12 baloo pppd[21431]: Using interface ppp0
May  1 22:10:12 baloo pppd[21431]: Connect: ppp0 <--> /dev/pts/2
May  1 22:10:36 baloo pppd[21431]: kernel does not support PPP filtering
May  1 22:10:36 baloo pppd[21431]: Cannot determine ethernet address for proxy ARP
May  1 22:10:36 baloo pppd[21431]: local  IP address 82.64.69.69
May  1 22:10:36 baloo pppd[21431]: remote IP address 192.168.254.254
May  1 22:10:36 baloo pppd[21431]: primary  DNS address 213.228.0.68
May  1 22:10:36 baloo pppd[21431]: secondary DNS address 212.27.32.177

```

Si on veut effectivement avoir le nom `adsl0` pour l'interface (plutôt que le `ethXXX` par défaut), rajouter `:options adiusbadsl if_name=adsl0` dans `/etc/modutils/adiusb` par exemple (puis faire `update-modules`).

Si on veut se connecter automatiquement au démarrage de la machine, prévoir un script `/etc/ppp/ppp_on_boot` du type :

```

#! /bin/sh
( sleep 15 ; /sbin/ifconfig ppp0 || /usr/sbin/startadsl ) &

```

16 Configuration d'une partition d'échange MacOS X / Linux

Par défaut, Linux ne sait que lire (un peu) et pas (trop) écrire les systèmes de fichiers UFS et HFS, et MacOS ne sait pas (encore) lire les partitions `ext2/ext3`. Une solution est de passer par une partition de type MSDOS (avec les noms longs, *aka* `vfat` sous Linux) pour les échanges de données entre MacOS et Linux.

16.1 Sous Linux

On doit avoir prévu un peu d'espace libre sur le disque, ou une partition dédiée pour la partition d'échange. Chez moi ca fait 800M et ca s'appelle `/dev/hda16`.

- booter sous linux, puis...
- lancer `fdisk` (qui est un lien vers `mac-fdisk`), s'assurer que la partition en question est du **type MS-DOS** : majuscules et tiret sont **OBLIGATOIRES** sinon MacOS refusera de la monter.
- Si ce n'est pas le cas, supprimer la partition en question (commande `d`), et en créer une à la place avec la commande `C<entrée>` (`C` majuscule!) qui fait la même chose que la commande `c` minuscule, mais sans deviner toute seule le type à affecter à la nouvelle partition. On peut mettre n'importe quel nom, seul le type `MS-DOS` est important.
- `mkdosfs /dev/hda[num.partition] : chez moi ça donne mkdosfs /dev/hda16`
- tester : `mount -t vfat /dev/hda[num.partition] /mnt`
- Si ça ne marche pas, c'est peut-être parce que le module noyau `vfat` n'est pas disponible, revoir la config noyau.
- Si ça marche bien, rajouter un point de montage dans `/etc/fstab` de type `vfat`.

16.2 Sous MacOS X

- (booter sous MacOS X, puis...)
- Normalement, la partochte est reconnue, mais non montée par défaut. On teste quand même d'abord : dans un terminal MacOS (Applications/Utilitaires/Terminal), faire :

```
mkdir /Volumes/Echange
mount_msdos /dev/disk0s[num_partition] /Volumes/Echange
(chez moi : /dev/disk0s16)
```

Normalement ça doit se passer sans problème, et on doit pouvoir lire et écrire dans /Volumes/Echange des fichiers quelconques (noms longs, espaces, plusieurs ".", ...).

Si il y a un problème de droit pour faire le `mount_msdos`, voir pense-bête plus bas (section C, *changer le mot de passe root*) et refaire la manip ci-dessus apres être passé root (commande `su`)

- Pour qu'aux prochains reboots l'icône de disque "Echange" apparaisse automatiquement sur le desktop au même titre que la partochte MacOS de base, faire (passer root) :

```
mkdir -p /Library/StartupItems/MountMSDOS
Puis créer /Library/StartupItems/MountMSDOS/StartupParameters.plist:
{
  Description      = "Mount MSDOS partitions";
  Provides         = ("Manual MSDOS partitions");
  Requires         = ("Disks");
  OrderPreference = "None";
  Messages =
  {
    start = "Mounting MSDOS partitions";
    stop  = "Unmounting MSDOS partitions";
  };
}
```

Ainsi que /Library/StartupItems/MountMSDOS/MountMSDOS :

```
#!/bin/sh

##
# Local filesystems
##

. /etc/rc.common

StartService ()
{
  ConsoleMessage "Mounting MSDOS partition"
  mount_msdos /dev/disk0s16 /Volumes/Echange
}
```

```

StopService ()
{
    return 0;
}

```

```

RestartService ()
{
    return 0;
}

```

```

RunService "$1"

```

Puis faire :

```

chown -R root /Library/StartupItems/MountMSDOS

```

```

chmod a+rx /Library/StartupItems/MountMSDOS/MountMSDOS

```

Pour plus de details, cf http://www.opendarwin.org/documentation/SystemStarter_HOWTO.php

- Et voilà! Au prochain reboot, la belle icône *Echange* apparait sur le desktop.

17 IEEE 802.11b (Airport/Wifi/Wavelan)

Je ne l'ai pas. Suivre les directives de http://mij.oltrelinux.com/ibook/step_step/index.html sur le sujet.

18 IEEE 1394 (firewire)

Pas testé.

19 Mise à jour vers unstable

A la date du 20 Mai 2003, ça marche bien. Juste le terminal gnome qui est difficile a lire a cause de l'antialiasing sur la fonte `monospace` (utiliser `rxvt` ou `xterm`). Evidemment, se procurer une bonne connexion reseau (genre ADSL), puis :

- Sauvegarder `/etc` (`tar xzvf ... /etc`)
- Mettre à jour `/etc/apt/source.list` (mettre simultanement les sources pour stable, testing et unstable, avec les mises a jour security pour stable et testing)
- Rajouter un fichier `/etc/apt/apt.conf.d/90perso` :

```
// APT::Default-Release "stable"; // Si on veut downgrader
APT::Cache-Limit 10000000; // Sinon erreur "E: Dynamic MMap ran out of room"
```
- `apt-get remove manpages-fr` (conflit avec `coreutils` et `debianutils`).
- `apt-get update`; `apt-get dist-upgrade`. Repondre aux questions.

- Petit conflit avec `gdk-imlib-dev` : faire le `remove` a la main en cours de route pour qu'il installe `gdk-imlib1-dev` a la place.
- `apt-get install xscreensaver` (a ete supprime?...)
- `apt-get install manpages-fr`.

19.1 Changements par rapport à la stable

19.1.1 Touches spéciales

Utiliser `pbuttonsd` plutôt que `ikeyd`. Le changement est que, pour accéder aux touches de fonctions en tant que touches de fonctions, il ne faut plus appuyer sur `fn` : on appuie directement sur la touche et ça marche. En conséquence, pour passer en console depuis X11, c'est comme sur x86 : `Ctrl-pomme-Fx`. Et inversement : pour accéder à ces touches en tant que touches spéciales Apples (volume, CD, etc.), il faut appuyer sur `fn`.

19.1.2 Mise en veille, gestion de l'énergie

Utiliser `pbuttonsd` (eh oui, encore lui) plutôt que `pmud`.

20 Passage au noyau Linux 2.6

Testé et approuvé avec un noyau 2.6.3 de base. Marche à suivre :

- (testé et approuvé en debian unstable, cf section 19)
- récupérer les sources du 2.6 standard, pas besoin du noyau `benh`
- `apt-get install init-module-tools` (j'ai eu à le faire 2 fois de suite)
- Le noyau 2.6 repose sur l'architecture ALSA au lieu de OSS pour le son : `apt-get install alsa-utils`, puis choisir le driver "powermac"
- config noyau : voir mon fichier de config. En gros : clavier `adb`, `drm` et `dri` pour la radeon (ne plus compiler le `dri` séparément comme dans la section 6.2), son `dmасound_pmac` et le driver `alsa snd-powermac` (car il y a un bug `alsa` : le module `snd-powermac` ne semble pas initialiser correctement le son, ie le mixer ne marche pas si on ne charge pas le module `dmасound_pmac` un peu avant, avant de le décharger car les 2 ne peuvent être utilisés en même temps), et les wrappers `snd-pcm-oss` et `snd-mixer-oss` d'`alsa`, et plus besoin de `ide-scsi`
- `make all modules_install` puis mettre le `vmlinuz` dans `/boot` et mettre à jour `/etc/yaboot.conf` avant de faire `ybin` comme d'habitude
- fichier `/etc/modules-2.6` :


```

apm_emu
radeon

```
- fichier `/etc/modprobe.d/custom` :


```

alias eth0 sungem
install radeon /sbin/modprobe agpgart && /sbin/modprobe uninorth-agp && /sbin/
# For some strange reason we need to load/unload dmасound_pmac
# before being able to manage the mixer of the soundcard with the

```

```
# snd-powermac alsa mixer (linux 2.6.3)
install snd-powermac /sbin/modprobe dmasound_pmac && /sbin/modprobe -r dmasound_pmac && /
- comme d'hab : faire update-modules
- rebooter
- si en 2.4 on utilisait ide-scsi, il n'y en a plus besoin en 2.6 (le sous-système
  IDE gère bien la gravure directement) : enlever les liens /dev/cdrom vers
  /dev/scd0, les remplacer par des liens vers /dev/hdb
```

A Remerciements

- David Mentré,
- Stephane Lentz,
- Christophe Lejeune.

B Support d'un memory stick

Ce n'est pas à proprement parler spécifique à l'ibook. Il suffit d'installer le support hotplug (j'ai récupéré la version plus récente <http://www.fs.tum.de/~bunk/debian/dists/woody/bunk> d'avoir un noyau avec CONFIG_USB_STORAGE, et de rajouter un fichier /etc/hotplug/usb/usb-storage du type :

```
#!/bin/sh

case $PRODUCT in
58f/9380/100|7b4/105/1)
cat > $REMOVER << EOF
umount /usb-storage
EOF
chmod 755 $REMOVER
mount /usb-storage
;;
*)
;;
esac
```

(pour l'id 58f/9380/100, soit remplacer par *, soit le récupérer en regardant /proc/bus/usb/devices). Et avoir une entrée dans /etc/fstab du type :

```
/dev/sda1 /mstick vfat noauto,rw,user,sync,gid=50,umask=002
```

C Pense-bête

- Extinction d'urgence : bouton marche/arret pendant au moins 5s

- Remise a zero de la config hard sauvegardee [aka NVRAM] (on perd la date) : éteindre la machine, puis `ctrl-option-shift-bouton marche/arret` (aucun beep ni voyant allumé pour témoigner de la prise en compte)
 - Changer le mot de passe root MacOS : booter sur le CD 1 d'installation de MacOS X, puis dans le menu `install` sélectionner `reset password`
 - Activer ssh, dhcp, ... sous MacOS : `system settings/partages`
 - Changer le gecos (shell, ...) des utilisateurs : utiliser `Applications/Utilitaires/Net Util` et patienter le temps qu'il charge tout, et modifier les gecos.
Ou a la main, depuis un terminal (passer root) :


```
# Pour voir :
niutil -read / /users/username
# Pour changer :
niutil -createprop / /users/username shell /bin/zsh
```
 - Activer le multiboot graphique MacOS (qui ne remplace pas `yaboot`, mais qui le précède uniquement!!!) :
 - sous MacOS X, dans un terminal (`Applications/Utilitaires/Terminal`), passer root (commande `su`) et faire :


```
# Pour voir :
nvram -p
# Pour changer :
nvram boot-command="multi-boot"
```
 - sous Debian, utiliser `nvsetenv` fourni avec le package `powerpc-utils` installé par défaut, passer root, et faire :


```
# Pour voir :
nvsetenv
# Pour changer :
nvsetenv boot-command multi-boot
```
- Si `yaboot` est installé, une icône avec un pingouin apparaît au boot, et si on la sélectionne, c'est la phase de boot `yaboot` qui est lancée. Chez moi, la souris reste à représenter une montre qui tourne 30s avant de devenir une flèche normale qu'on peut utiliser pour sélectionner l'OS a booter (il doit faire quelques requêtes `dhcp` [?]).
- Pour revenir au boot mac par défaut (=i `yaboot` direct), faire : `nvram boot-command="mac-boot"` dans un terminal MacOS X, ou `nvsetenv boot-command mac-boot` sous Linux.
- Monter une partoché NFS sous macOS X :

s'assurer qu'un serveur NFS tourne quelque part, et est accessible depuis le mac. Pour ca, dans un terminal MacOS X :

```
showmount -e adresse_serveur
```

Note : l'exportation depuis le serveur doit être faite avec l'option (`/etc/exports`) `insecure`, et si possible en n'utilisant pas `knfsd` de Linux (car impossible de traverser les points de montage intermédiaires) ⇒ `apt-get install nfs-user-server` sur le serveur si c'est une debian
 - Sous MacOS : aller dans le menu `Aller du Finder`, puis "se connecter au serveur", et entrer :


```
nfs ://adresse_serveur/repertoire_exporte
```

Pour un montage systématique au boot, il faut sûrement s'attaquer à la base NIS =*i* Applications/Utilitaires/Net Info.

- divx sous MacOS X : ça se passe sur divx.com, ça installe le driver divx pour quicktime, et un utilitaire *divx validator*. Si on n'a pas le son sur un divx, faire un *drag'n drop* sur ce *validator* qui remet le flux en forme (en fait : génération d'un petit fichier caché `._nom_original`) pour que quicktime lise le son correctement.
- divx sous linux : `mplayer` (manque de fluidité), `xine` (superbe même en plein écran, plus fluide que sous MacOS X!).

D Liens

- Linux sur laptop : <http://tuxmobil.org/mylaptops.html>, <http://www.linux-laptop.net/>
- Meta : <http://www.linux-france.org/macintosh/>
- Meta + adresses rsync/bk noyau pppc/benh : <http://penguinppc.org/dev/>
- Meta : <http://www.debian.org/ports/powerpc/>
- divers partitionnement (debian seule, dual boot macosx, triple boot macosx/macos9, ...). Mais quand on ne sait pas que l'ordre logique des partitions n'affecte pas MacOS X, tout ça paraît un peu bizarre : <http://people.debian.org/~branden/iobook>.
- Une référence pour X, le son et le modem : http://mij.oltrelinux.com/iobook/step_step/index.html
- doc hispalinux, assez complète dans la partie configuration du noyau et de X : <http://www.hispalinux.es/~data/iobook/>
Traduction française : <http://www-alt.pasteur.fr/~edeveaud/MacDebian/iobook/index.html>
Détails sur l'économie d'énergie : <http://bulmalug.net/body.phtml?nIdNoticia=1511>
- clavier (rpm mais aussi infos debian par David Mentré) : <http://www.linux-france.org/macintosh/clavier>
- carte video (paquets debian XFree86 4.2.99) : <http://people.debian.org/~daenzer/dri-trunk>
- son : <http://www.wrietman.com/iobook/index.html?main>
- modem + cdrw + X + souris + config kernel : <http://www.mayin.org/aragorn/computers/iobook.html>
- drivers modem linux : <http://www.mbsi.ca/cnxtlindrv/> et <http://www.mbsi.ca/cnxtlindrv/hcf/download>
- Homepage Benjamin Herrenschildt (aka *benh*) : <http://penguinppc.org/~benh/>
- divers doc montage disque sur macos : <http://www.netbsd.org/Ports/macppc/faq.html>
http://www.opendarwin.org/documentation/SystemStarter_HOWTO.php
 - s'inspirer de `/System/Library/StartupItems/Disks/Disks` sous MacOS X
- changer phase de boot (multi-boot graphique, mac-boot) : <http://osxfaq.com/tips/beatty/index.ws>
- misc MacOS (admin type unix) : <http://www.bombich.com/mactips>
- misc urgence : http://www.mansd.org/departments/technology/help/iobook_tips.htm
- Doc Apple (url vers la doc processeur, ...) : <http://developer.apple.com/techpubs/hardware/Developer>
- Doc processeur : http://www-3.ibm.com/chips/techlib/techlib.nsf/products/PowerPC_750CX_and_7