

Vine Linux 3.1 ppc 版について

MATSUBAYASHI 'Shaolin' Kohji*

平成 16 年 11 月 25 日

目次

1	はじめに	3
2	ディレクトリ構成	3
3	インストールの準備	4
3.1	機種についての情報収集 (1)	4
3.2	機種についての情報収集 (2)	5
3.3	キーボードとマウス	6
3.4	パーティションについて	7
3.5	パーティションを分割する	9
4	Vine Linux/ppc をインストールする	12
4.1	インストーラ起動の準備 (OldWorld 機のみ)	12
4.2	インストーラを起動する (OldWorld 機の場合)	13
4.3	インストーラを起動する (NewWorld 機の場合)	13
4.4	キーボードの設定	14
4.5	マウスの設定	16
4.6	X の設定 (Xautoconfig4)	17
4.7	X インストーラ起動	18
4.8	ネットワークの設定	19
4.9	パーティションの設定	20
4.10	パーティションの選択	23
4.11	パッケージの選択	24
4.12	root のパスワードの設定	27
4.13	インストール終了....	28
5	ブートローダの設定 (OldWorld 機のみ)	30
5.1	BootX のインストールと設定	30

*mailto:shaolin@vinelinux.org

6	Vine/ppc の設定	32
6.1	X サーバを設定する	32
6.2	マウスのエミュレーションを設定する	33
6.3	MacOS とデータをやりとりする	34
6.3.1	Linux から直接マウントする	34
6.3.2	HFS ユーティリティを使う	34
6.4	Mac-On-Linux を使ってみる	35
6.4.1	初期設定ファイルの編集	35
6.4.2	MOL を起動する	36
6.4.3	サウンドを有効にする	36
7	最後に	39
A	PowerMac 用ブートローダについて	40
A.1	調べておくこと	40
A.2	BootX について	43
A.2.1	Mac OS ボタン	43
A.2.2	Linux ボタン	43
A.2.3	Kernel ポップアップメニュー	44
A.2.4	Root device エントリ	44
A.2.5	Options... ボタン	44
A.2.6	Save to prefs ボタン	45
A.2.7	No video driver チェックボックス	45
A.2.8	More kernel arguments エントリ	45
A.3	miBoot について	46
A.3.1	miBoot 設定の準備	47
A.3.2	起動時の引数の設定	47
A.3.3	カーネルと ramdisk の指定	48
A.4	yaboot / ybin / xybin について	50
A.4.1	起動パーティションの必要性	50
A.4.2	ybin を使った yaboot のインストール	51
A.4.3	yaboot.conf の編集	52
A.4.4	ybin の実行	55
A.4.5	xybin について	56
A.4.6	Option ブートで切替える	56
B	カーネル引数の詳細	58
B.1	ランレベルに関する引数	59
B.2	画面に関する引数	60
B.2.1	フレームバッファ名の指定	61
B.2.2	画面解像度の指定	62
B.2.3	画面の色数の指定	62
B.2.4	VRAM 搭載量	62
B.2.5	複数のビデオカードを搭載した機種の場合	63

1 はじめに

この文書では、Vine Linux 3.1 ppc 版 (以下 Vine/ppc) に固有の情報を扱います。

Vine/ppc が動作対象とするのは、CPU に 32bit PowerPC を使い、Open Firmware が搭載された Apple 社 (および過去の一時期は互換機メーカー) から発売された PowerMac アーキテクチャです。具体的な動作情報については以下のページや web 上で公開している最新確認情報¹ で詳しく触れています。

Vine/ppc の構成コンポーネントは i386 版とごく一部を除いて全く同じですので、インストールが終了すれば i386 版と全く同じように利用することができます。アーキテクチャ、ハードウェアの相異に起因する設定の違いや、ppc 版固有のインストーラ、設定ツール、ハードウェア互換情報などを扱います。

2 ディレクトリ構成

Vine/ppc の CD-ROM (Vine Linux 3.1/ppc) は、以下のディレクトリから構成されます。

/

Manual/	MacOS 側から見る為のドキュメント類 (HTML, PDF)
MacOS Utilities/	MacOS 側で使用するツール類 (pdisk, MacRPM, MacGzip など)
System Folder/	NewWorld 機で CD ブートするためのダミーシステムフォルダ
SRPMS-ppc/	ppc 版固有のソースコード (src.rpm 形式)
Vine/base/	apt で使うデータ (MacOS 側からは見えません)
live/	インストーラで使うファイル (MacOS 側からは見えません)
maps/	インストーラで使うファイル
software/	RPM パッケージ

¹<http://vinelinux.org/ppc/compatibility/>

3 インストールの準備

このセクションでは、実際に Vine Linux を PCI PowerMac 機にインストールする為に調べておかなければならない各種情報、そしてインストールを始める前の準備作業について説明します。

3.1 機種についての情報収集 (1)

Vine/ppc が動作対象とする PowerMac には大きく二種類に分けられます。インストールしようとしている機種がどちらに属するかによって、インストール前、インストール後の手順が大きく異なります。

OldWorld MacOS が動作するための ROM がオンボードで搭載されている機種で、Open Firmware のバージョンは 3.0「未満」です。デスクトップ機では PowerMac G3 DT/MT/All-In-One (ページ G3)、ノート機では PowerBook G3 (WallStreet, キーボードが黒いもの) までがこれにあたります。

NewWorld MacOS が動作するための ROM はオンボードに搭載されておらず、システムフォルダ内の「Mac OS ROM」ファイルを使う機種です。Open Firmware のバージョンは 3.0「以上」となります。デスクトップ機では PowerMac G3 (Blue & White) や iMac 以降、ノート機では PowerBook G3 (Lombard, キーボードが半透明のもの) 以降がこれにあたります。

なお、初代 PowerMac (6100/7100/8100 など NuBus 搭載マシン) やその互換機、PowerBook Duo 2300c, PowerBook 5300, PowerBook 1400 など初期の PowerBook, Performa 5200, 6200, 6300 などは、Vine/ppc ではサポートされていません。これらの機種で Linux を動作させるには MkLinux² あるいは Nubus-pmac カーネル³ を使う必要があります (Vine/ppc には収録されていませんし、動作確認も行っていない)。

最も新しい 64bit PowerPC 搭載 Macintosh (PowerMac G5, iMac G5 など) についても、動作確認を一切行っておりませんので、Vine/ppc ではサポートされていません。

²<http://www.mklinux.org/>

³<http://nubus-pmac.sourceforge.net>

3.2 機種についての情報収集 (2)

続いてマシンに装着しているカードや周辺機器について調べておきます。最新の一部機種を除き、純正の構成のものではほぼ問題なく動作する様ですが、PCI スロットに拡張カードを追加している場合等は特に調べる必要があります。

PowerMac でよく増設/交換され利用されているものを含め、以下のものの型番や搭載チップなどを調べるとトラブルを防ぐことがより容易になります。

- ビデオカード
- モニタの垂直 / 水平周波数
- ネットワークカード
- SCSI カード
- ATA33/66/100/133 カード
- USB カード
- IEEE1394 (FireWire) カード
- その他ハードウェア

MacOS で利用可能だからといって、必ずしも Linux で利用可能とは限りませんので注意して下さい。このマニュアルを執筆している段階で分かっている問題点のうち主なものをいくつか記しておきます。

- PowerMac 9600/300 に標準搭載されているビデオカードである IMS TwinTurbo (8MB VRAM) は X の動作に難があることが確認されています。他の 9600 に搭載されている TwinTurbo (4MB VRAM) は、15/16bpp での動作はほぼ問題ない様です。
- 最近の PowerMac や iMac G4 に搭載されているビデオカード/チップのうち nVidia 社製のもの (GeForce2/3/4 など) では X の動作に問題がでる場合があります。
- 比較的最近の機種に搭載されている内蔵モデムの多くは、現状では Linux からは利用できません。
- Vine-2.6 のカーネルまでは動作していた AEC-6280M などの AEC 社製 PCI IDE カードの動作が不安定になっています。現在調査中です。
- その他サードパーティから販売された PowerMac 用各種拡張カードでも、Linux での利用が困難なものがあります。

対応していない周辺機器や増設カードの中には、それらをマシンから外しておかないと Linux がうまく動かないものも一部あるようですので注意して下さい。

特に Linux カーネルや X の対応状況は、開発が進むにつれて変わっています。より詳しい情報は「web 上で公開している最新確認情報⁴」を御覧下さい。

⁴<http://vinelinux.org/ppc/compatibility/>

3.3 キーボードとマウス

キーボードとマウスについては、ADB (Apple Desktop Bus) 接続、USB 接続のもの共に、大抵のものがそのまま使えます。一部の Mac 互換機には PS/2 コネクタが装備されている機種がありますが、これらも使用可能です (内部的には ADB として扱われているようです)。

キーボードはできれば拡張タイプ (ファンクションキーがついているもの) をお使いになるのをお勧めします。仮想ターミナルの切替えを行う際にファンクションキーが必要になるからです。

マウスは可能ならば 2 ボタン以上のものを用意して下さい。X では 3 ボタンマウスが前提となったプログラムが多いためです。最悪 1 ボタンマウスでも、キー操作による中ボタン/右ボタンのエミュレーションは可能ですが、操作性は悪くなります。

キーボードによるマウスボタンのエミュレーションについては「マウスのエミュレーションを設定する (Section 6.2 参照)」を御覧下さい。

また、一部の 3rd パーティ製 ADB 2 ボタンマウスは、Linux 上でうまく使えない場合があるので注意して下さい。

3.4 パーティションについて

i386 版同様，Vine/ppc をインストールするには，ハードディスクに Linux 用パーティションを確保する必要があります．全ての PowerMac や PowerBook は MacOS や MacOS X が一つのパーティションにプリインストールされて販売されていますので，まずパーティションを切り直して Linux 用の領域を確保しなければなりません．

ただしデスクトップ機に内蔵 SCSI/IDE ハードディスクや外付 SCSI ハードディスクを増設し，それを Linux 専用で利用する場合はこの限りではありません．

既に MacOS がインストールされているハードディスクのパーティションを分割する最も簡単な方法は，購入時付属のシステムインストール CD やシステムリストア CD を利用するものです．

1. ハードディスクの内容を別ディスクにバックアップしておく
2. システムリストア CD から起動する
3. CD に入っている「ドライブ設定」を使ってパーティションを分割する
4. MacOS 用に確保したパーティションに，MacOS をインストールあるいはリストアする
5. バックアップしておいたものから MacOS / MacOS X の環境を戻します

これらの作業は，可能であればマシン購入直後にやっておくとよいでしょう．

パーティションは OldWorld 機の場合最低 3 つ、NewWorld 機の場合最低 4 つ必要となります。

1. MacOS / MacOS X 用パーティション
2. Linux ルート用パーティション
3. Linux swap 用パーティション
4. bootstrap パーティション (NewWorld 機のみ)

Linux ルート用パーティション は Linux のシステムがインストールされるメインのパーティションです (より多くのパーティションに分割することももちろん可能ですがこのセクションで扱う内容ではないので割愛します)。Vine/ppc の場合、最低でも 1.3GB、通常は 2GB 以上は確保しておくのが無難です。より小さいサイズのパーティションにも工夫すればインストールすることは可能ですが、お勧めしません。

Linux swap 用パーティション は、システム搭載メモリ容量と同量を目安に確保して下さい。搭載メモリが少ないマシンの場合は、メモリ容量の 2 倍程度のサイズの swap パーティションを確保しておくといでしょう。

bootstrap パーティション は NewWorld 機でのみ必要な小さなパーティションです。通常は 5MB ~ 50MB 程度の小さなサイズで充分です。このパーティションにブートローダ (yaboot) がインストールされ、起動時に OS を選択したり、Linux 起動時には様々なオプションを与えたり出来るようになります。ブートローダはインストーラ内から簡単にインストールすることが出来ます。またインストール後にブートローダの設定を変更することも容易です。

なお、MacOS と Linux でデータのやりとりをしたい場合には、別途 Apple 標準 (HFS) 形式のパーティションも用意する必要があります。Linux から Apple 拡張 (HFS+) 形式のパーティションにアクセスするのは現時点ではまだ不安定だからです。



図 1: 「ドライブ設定 (英語版では Drive Setup)」

3.5 パーティションを分割する

以下、内蔵ハードディスクが 4GB で、

1. Mac OS に 2GB
2. Mac OS / Linux のデータやりとり用に 200MB
3. Linux ルートパーティションに 1.7GB
4. Linux swap パーティションに 100MB

を割り当てる場合を例に説明します。

現在起動中のディスクのパーティションを変更することは出来ませんので、一旦 Macintosh に付属の Mac OS CD、またはパーティションを変更しようとしているハードディスク「以外」のハードディスクから Mac OS を起動する必要があります。その後、「ドライブ設定」(図 1) というツールを使ってパーティション割り当て作業を行います。

パーティショニングを行う際、Mac OS や Mac OS X 用のパーティション以外は「Apple 標準」にしておいて下さい。今の例では、2 番目のパーティション (データやりとり用)、3 番目のパーティション (Linux swap)、4 番目のパーティション (Linux ルート) を「Apple 標準」(HFS) にしています (図 4)。Linux ルートパーティションと Linux swap パーティションは後にインストーラ内で HFS から Apple.UNIX.SVR2 に変更します。

フォーマットが済んだら、Mac OS / OS X 用に確保したパーティション (今の例では 1 つめのパーティション) に Mac OS / OS X を再インストールするなり、バックアップしていたデータを戻すなりして下さい。買ったばかりの Macintosh の場合は、システムリストア用 CD を使って、今用意したばかりの Mac OS 用パーティションにリストアするのが簡単でしょう。

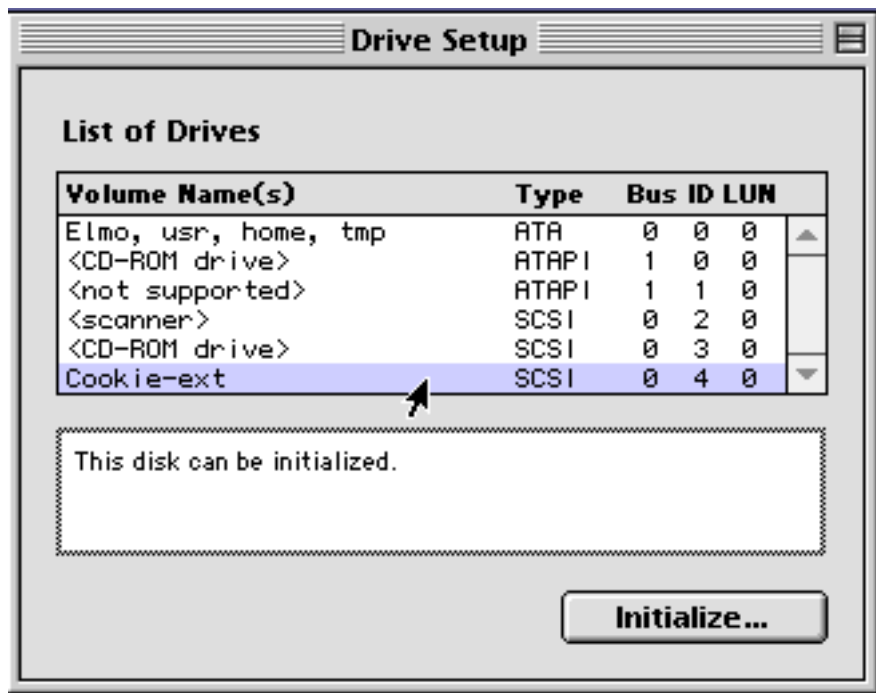


図 2: フォーマットするディスクを選択

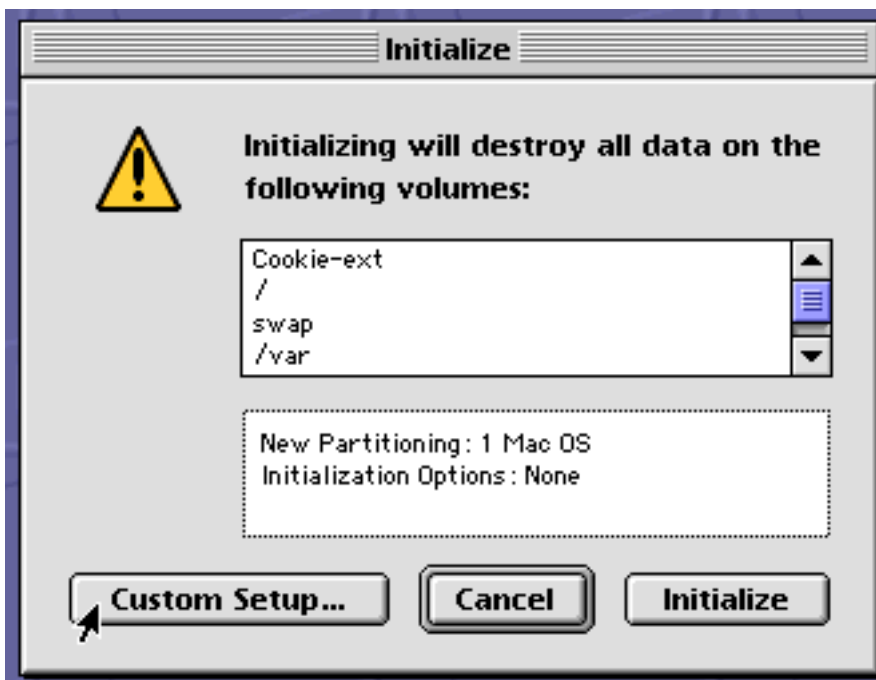


図 3: 「カスタム設定...」を選択

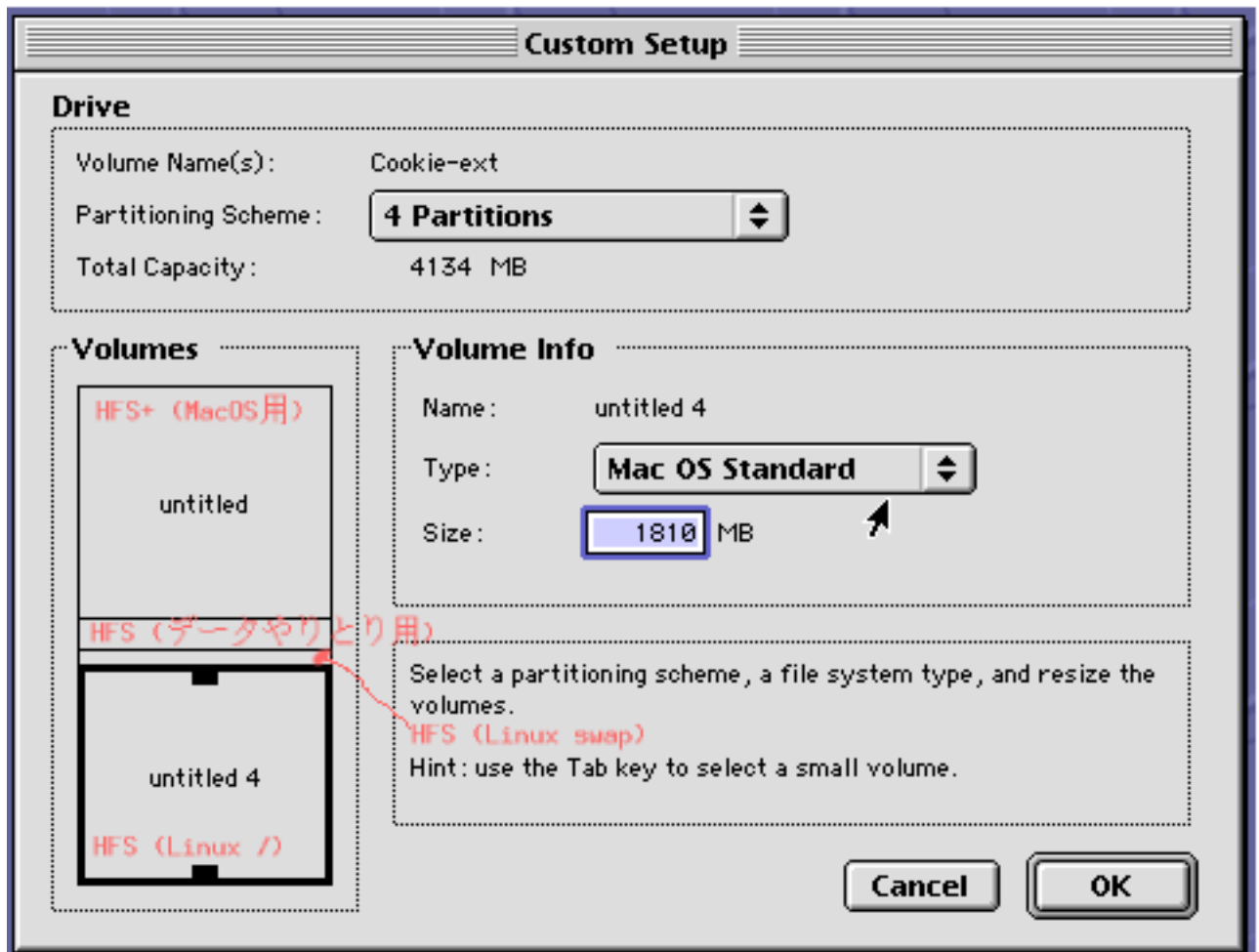


図 4: ドライブ設定でのパーティション設定

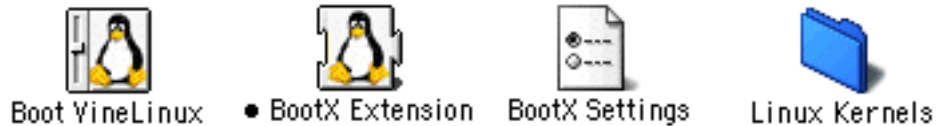


図 5: BootX 起動に必要なファイル一式

4 Vine Linux/ppc をインストールする

このセクションでは、CD-ROM から Vine Linux/ppc をインストールする手順について説明しています。ハードディスクからインストールする事も可能ですが、その方法については本マニュアルでは触れていません。

4.1 インストーラ起動の準備 (OldWorld 機のみ)

Vine/ppc のインストール CD は、ライセンスの関係上 OldWorld 機では CD からブートさせることが出来ません。そのため MacOS 側にブートローダをインストールし、それを介してインストーラを立ち上げる必要があります。

CD-ROM 中にあるファイル (図 5) をそれぞれハードディスクの MacOS パーティションにインストールして下さい。

- Boot VineLinux 「コントロールパネル」フォルダへ
- BootX Extension 「機能拡張」フォルダへ
- BootX Settings 「初期設定」フォルダへ
- Linux Kernels フォルダ システムフォルダ直下へ
- ramdisk.tar.gz システムフォルダ直下へ



図 6: OldWorld 機からインストーラを起動する際の画面 (BootX)

4.2 インストーラを起動する (OldWorld 機の場合)

Vine の CD-ROM をドライブに入れておいてシステムを再起動すると、MacOS の起動途中で BootX のパネルが現れますので (図 6) 「Linux」ボタンを押します。するとインストーラが起動します。

これでうまく起動しない場合は、起動カーネルを変更したり、「More kernel arguments」に適切な引数を追加する必要があります。

4.3 インストーラを起動する (NewWorld 機の場合)

Vine Linux の CD-ROM を 内蔵 CD-ROM ドライブ / DVD-ROM ドライブに入れておきます。そして Mac を再起動させ、起動音がなったらキーボードの「C」キーを押し続けて下さい。これにより CD-ROM から起動させることができます。

黒い背景に白色の文字が現れますので (図 7)、install と入力しリターンキーを押して下さい。

これでうまく起動しない場合は、install の後にスペースをあげ、更に適切な引数を与える必要があります。

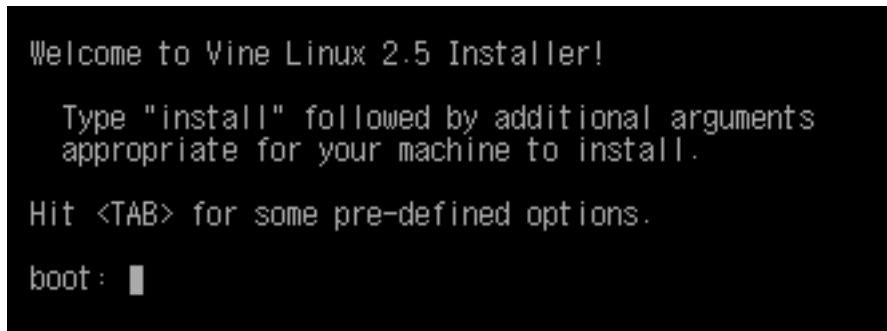


図 7: NewWorld 機で CD ブートした直後の画面 (yaboot)

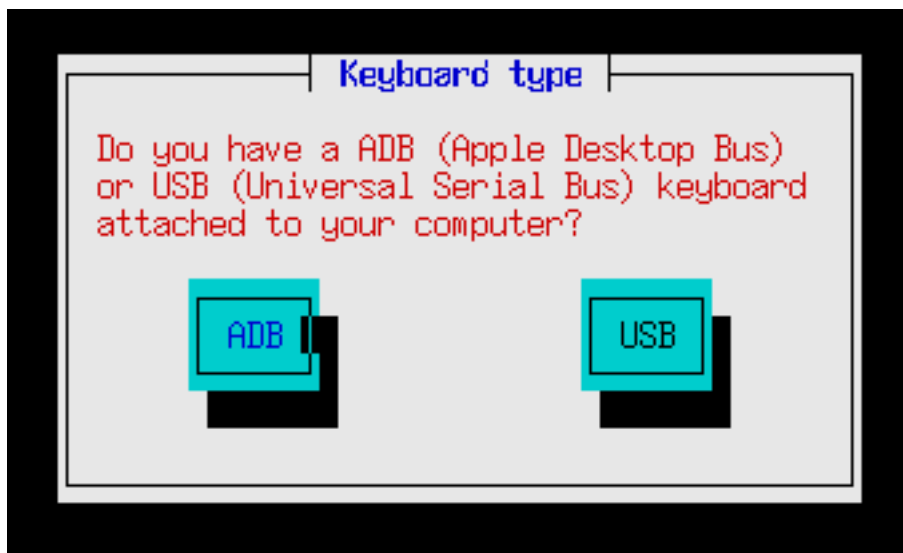


図 8: 「キーボード設定画面 (1)」

4.4 キーボードの設定

まず最初にキーボードの設定画面が現れます (図 8)。キーボードが ADB が PS/2 接続の場合は「ADB」を，USB 接続の場合は「USB」を選択して下さい。

項目間の移動はタブキーで行い，ボタンを選択するにはリターンキーを押して下さい。

「ADB」を選択すると，次に ADB キーボード用キーマップ選択画面 (図 9) が現れます。カーソルキーを使って該当するキー配列を選んで下さい。

日本語 JIS 配列キーボードの場合は「mac-jp-jis」を，ASCII 配列拡張キーボード (ファンクションキーあり) の場合は「mac-us-ext」を，ASCII 配列標準キーボード (ファンクションキーなし) の場合は「mac-us-std」を，それぞれ選んで次に進んで下さい。

「USB」を選択すると，次に USB キーボード用キーマップ選択画面 (図 10) が現れます。日本語 JIS 配列キーボードの場合は「usb-jp-jis」を，ASCII 配列キーボードの場合は「usb-us」を，それぞれ選んで次に進んで下さい。

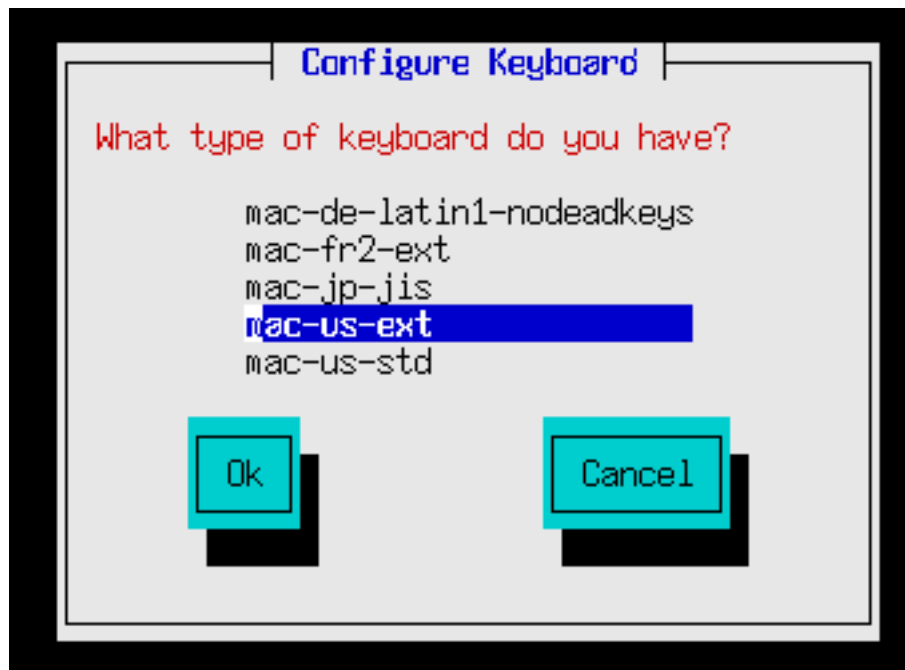


图 9: 「キー配列選択画面 (ADB)」

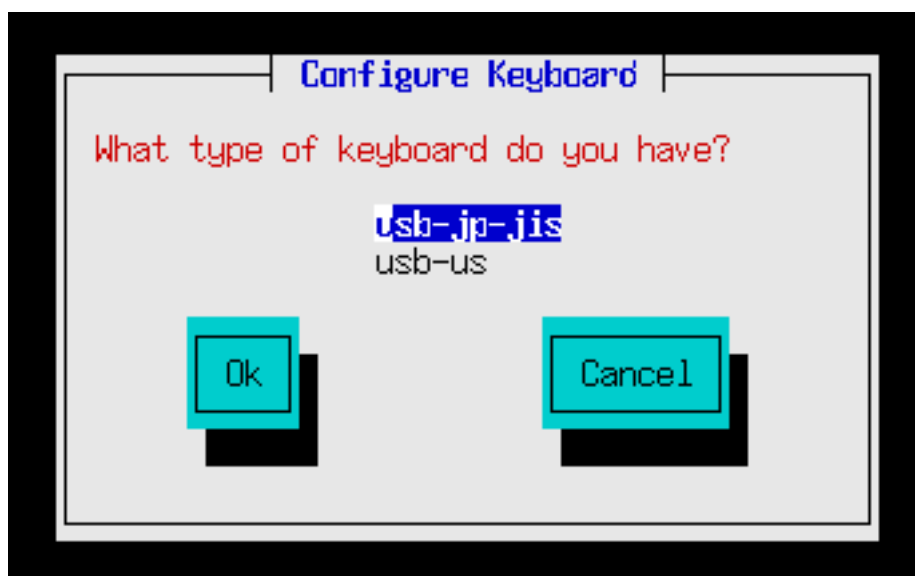


图 10: 「キー配列選択画面 (USB)」

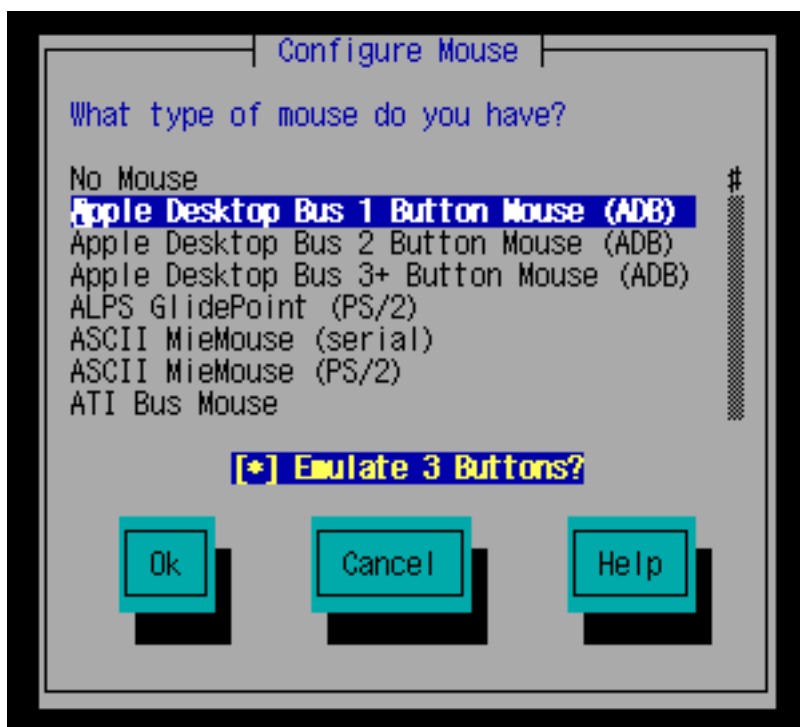


図 11: 「マウスの設定画面」

4.5 マウスの設定

続いてマウスの設定画面が現れます (図 11)。

カーソルキーを使って、使用しているマウスを選んで下さい。

ADB マウスや PS/2 マウスの場合は、マウスボタンの数にあわせて「Apple Desktop 1 Button Mouse (ADB)」～「Apple Desktop 3+ Button Mouse (ADB)」から選んで下さい。

USB マウスの場合は、同様にマウスボタンの数にあわせて「Universal Serial Bus 1 Button Mouse (USB)」～「Universal Serial Bus 3+ Button Mouse (USB)」から選んで下さい。

「Emulate 3 Buttons?」は、2 ボタンマウスで左ボタン/右ボタンを同時に押した場合に中ボタンと認識させるかどうかを選択します。チェックの ON/OFF はスペースキーを押すことで行います。

設定が終わったら「OK」ボタンを押して次に進みます。

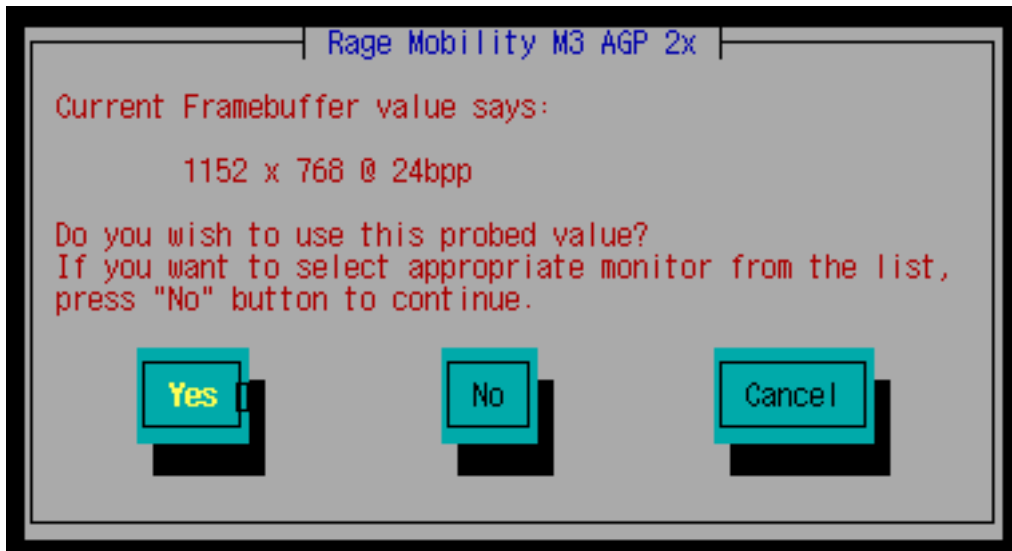


図 12: 「X の設定画面」

4.6 X の設定 (Xautoconfig4)

続いて X の設定を行います (図 12)。最初に自動検出された値が表示されますが、設定を変更したい場合はモニタの種類、解像度、色数、使用するドライバを手動で指定することも出来ます。



図 13: 「X インストーラの起動画面」

4.7 X インストーラ起動

ここまで終ると X が立ち上がり、インストーラが起動します (図 13)。

ここで「続ける」ボタンを押して下さい。続いて、インストールに関する簡単な手順の説明が現れます。

図 14: 「ネットワーク設定画面」

4.8 ネットワークの設定

続いて、ネットワーク設定の画面が現れます (図 14)。

固定 IP アドレスを割り当てる場合は「IP アドレス」「ゲートウェイ」「ネットマスク」「ホスト名」「ネームサーバ」の欄を正しく設定して「保存する」ボタンを押して下さい。

DHCP による接続を行う場合は「DHCP 設定」ボタンを押して下さい。その後、ホスト名を正しく設定して「保存する」ボタンを押して下さい。

ネットワークによる常時接続をしない場合は、何も設定せずに「閉じる」ボタンを押して次に進んで下さい。

なお、ここでネットワーク設定をしなかった場合、間違えた内容で設定してしまった場合でも、インストール後に改めて設定しなおすことが可能です。

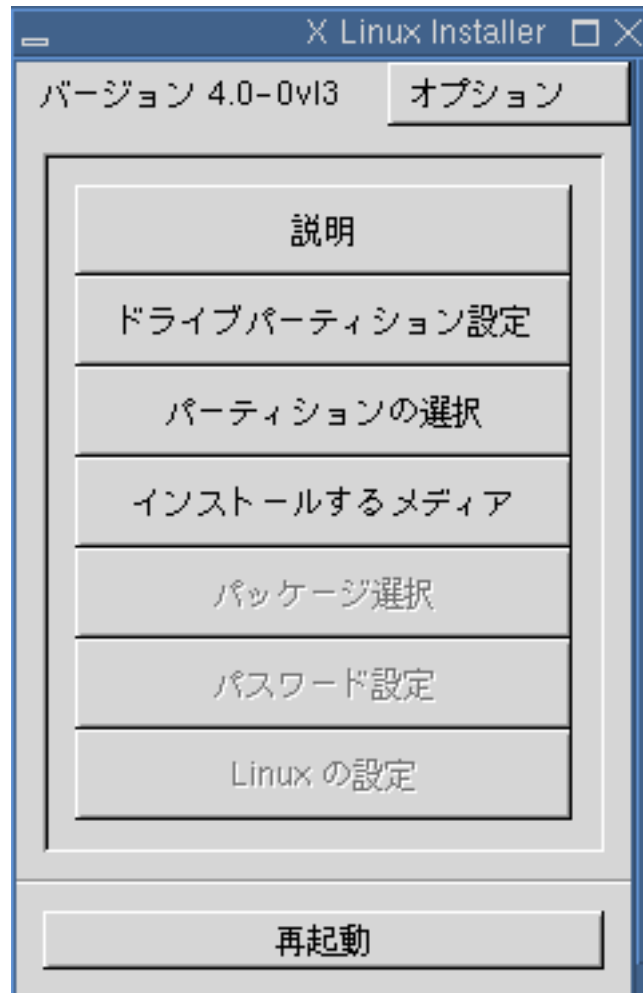


図 15: 「X インストーラのメニュー」

4.9 パーティションの設定

続いて、ハードディスクのパーティションを設定します。ただし、既に Linux がインストールされていて、パーティション設定はそのまま使う場合や、アップグレードインストールする場合はこのセクションは飛ばして下さい。

X インストーラのメニュー (図 15) から「ドライブパーティション設定」を選んで下さい。すると、perldisk というパーティション設定ツールが起動します (図 16)。

ここで、「パーティションを分割する (Section 3.5 参照)」で Linux 用に確保したパーティションを設定します。

確保したパーティションを Linux 用にするには

- パーティションタイプ名を「Apple_UNIX_SVR2」に変更する
- 「名称」を Linux で認識可能な名前に変更する

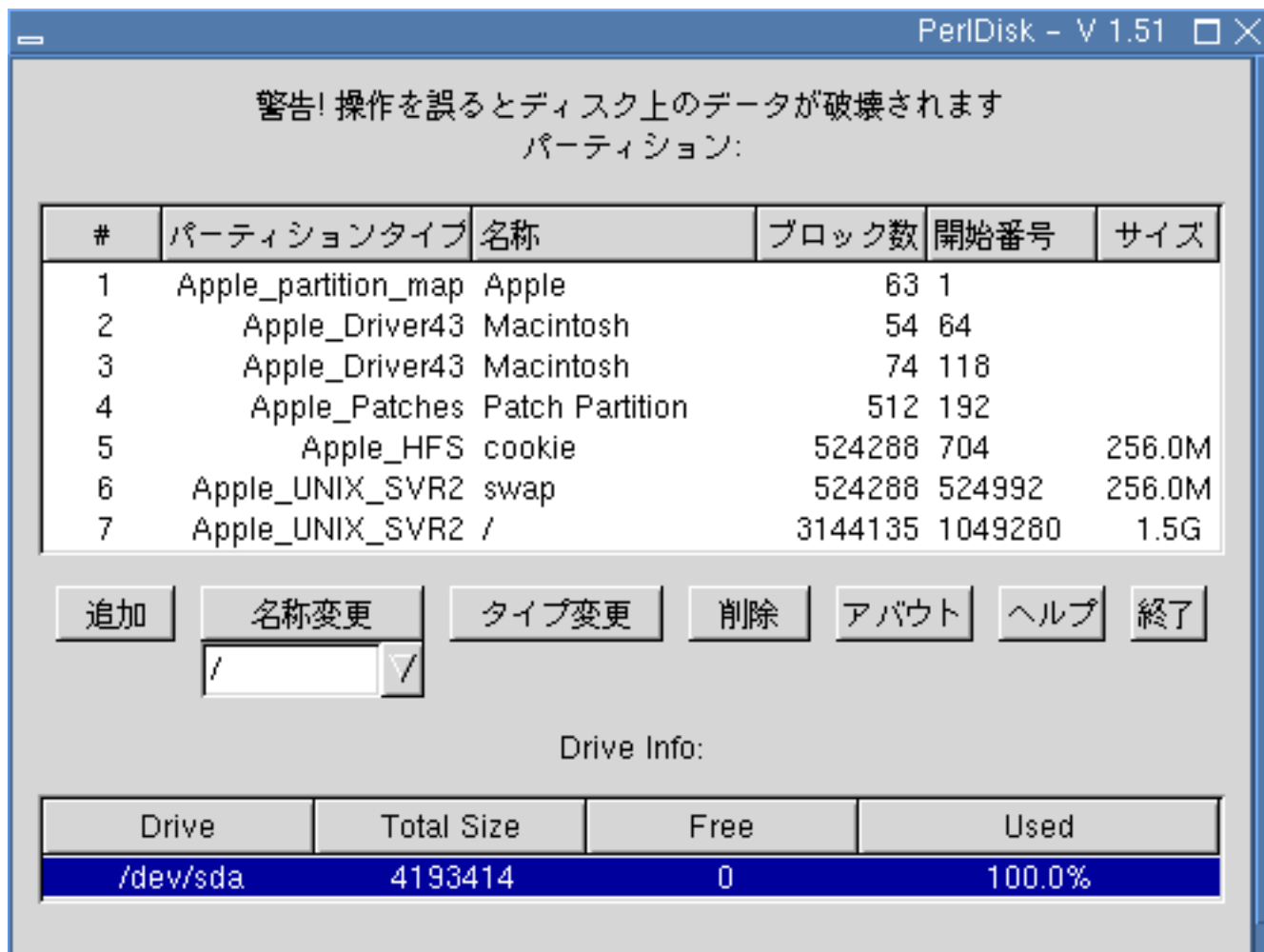


図 16: 「ドライブパーティション設定 (perldisk)」

の2つを行う必要があります。先の「パーティションを分割する (Section 3.5 参照)」ですと、

- swap 用に確保したパーティション (100MB) の
 - パーティションタイプ名を「Apple_UNIX_SVR2」に変更
 - 名称を「swap」に変更
- / (root) 用に確保したパーティション (1.7GB) の
 - パーティションタイプ名を「Apple_UNIX_SVR2」に変更
 - 名称を「/」に変更

という作業を行う必要があります。

NewWorld ROM 機ではこれらに加えて **bootstrap パーティション** として 5MB ~ 50MB 程の HFS (Apple 標準) パーティションを用意したはずですので、

- パーティションタイプを「Apple_Bootstrap」に
- 名称を「magicboot」に

それぞれ変更しておいて下さい。

パーティションタイプ名を変更するには、変更したいパーティションを選択しておき、「タイプ変更」ボタンを押して下さい。押す度に「Apple_UNIX_SVR2」「Apple_Bootstrap」「Apple_HFS」と変更されます。

パーティションの名称を変更するには、変更したいパーティションを選択しておき、「名称変更」ボタンの下にあるテキスト欄に名称を入力、最後に「名称変更」ボタンを押して下さい。

設定が終了したら「終了」ボタンを押して下さい。ここで設定を間違えると最悪ハードディスク上のデータが失われることとなりますので、間違いのないことを確認して「保存して終了」を選びます。もし間違っていると思った場合は「保存せずに終了」を選んで下さい。

「保存して終了」を選んだ場合は、再起動を促すウィンドウが現れます。パーティションタイプ名やパーティションの名称を変更しただけの時は再起動せずに次に進んで構いません。ただし、perldisk 上でパーティションサイズの変更や新規パーティションの作成を行った場合は、再起動して下さい。サイズが変更された新しいパーティションマップは、再起動しないと認識されないことがあります。

この perldisk を使って、パーティションの削除、パーティションの新規作成などを行うこともできますが、Apple_partition_map, Apple_Driver43, Apple_Driver_ATA, Apple_Driver_IOKit, Apple_FWDriver, Apple_Patches といったタイプの (ドライバ) パーティションは**絶対に削除しないで下さい**。

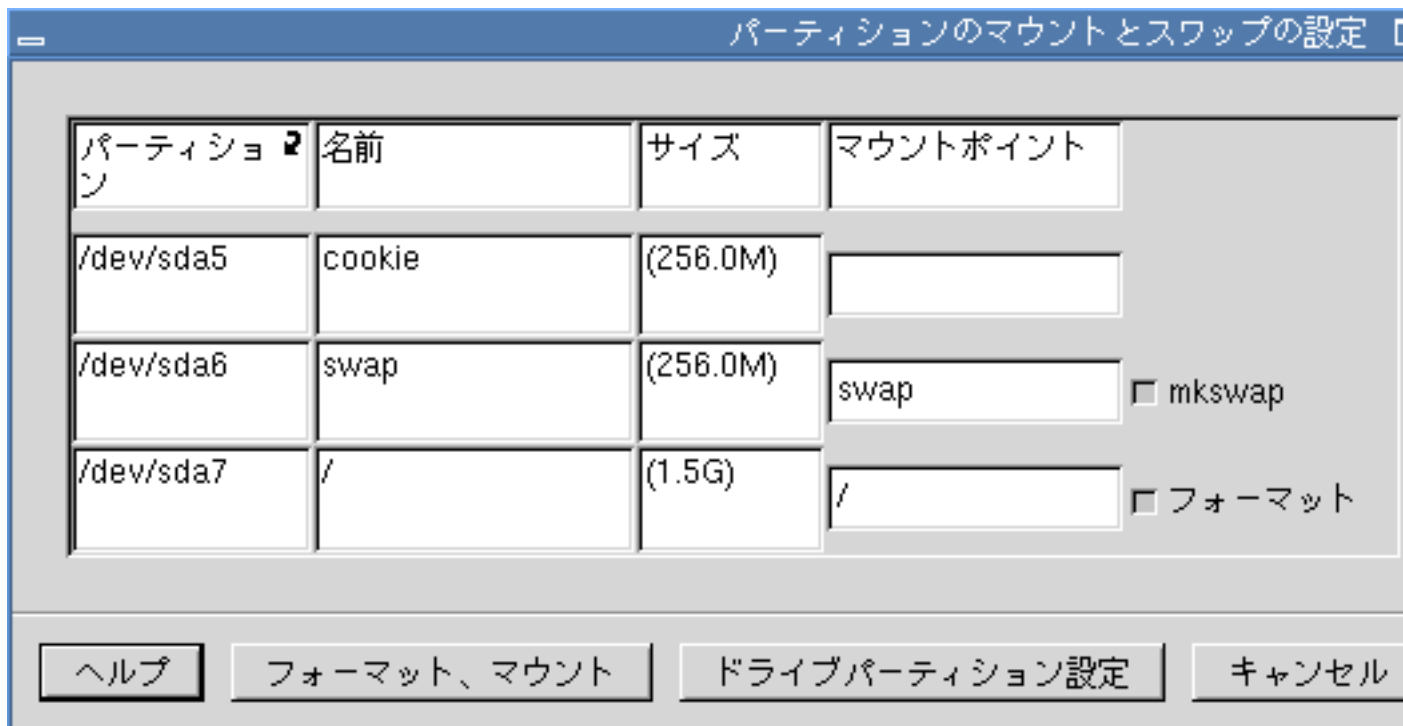


図 17: 「パーティションのマウントとスワップの設定」

4.10 パーティションの選択

続いて、X インストーラのメニュー (図 15) から「パーティションの選択」を選んで下さい。すると、どのパーティションをどこにマウントするかを設定するウィンドウが現れます (図 17)。

スワップ用のパーティションの場合は、マウントポイントを `swap` に、その他の Linux パーティションの場合は、マウントポイントを `/` から始まる絶対パスで指定して下さい。

「mkswap」や「フォーマット」チェックボタンは、そのパーティションを初期化するかどうかを選択するものです。新規インストールの場合は、必ずチェックを入れて下さい。逆にアップグレードインストールの場合は、チェックを外して下さい。

NewWorld 機の場合は、**bootstrap パーティション** として用意したパーティションのマウントポイントとして「magicboot」を指定して下さい (実際には Linux からはマウントされません)。

設定が済んだら「フォーマット、マウント」ボタンを押して下さい。フォーマットが行われた後、次の「パッケージ選択」画面が現れます。

4.11 パッケージの選択

最後に、パッケージ選択ウィンドウが現れます (図 18)。

ここで、インストールするパッケージを選択します。

デフォルトでは「(default)」とついているパッケージ集が選択されているはずですが、それぞれのパッケージ群について簡単に説明しておきます。

基本パッケージ (必須) 必ずインストールして下さい

X ウィンドウシステム (必須) 必ずインストールして下さい

PowerBook 専用 PowerBook にインストールする場合には選択して下さい

GNOME (推奨) GNOME をインストールしたい場合には選択して下さい

印刷関係 (推奨) CUPS 関連のパッケージをインストールする場合には選択して下さい

開発環境 コンパイルやパッケージのリビルド等を行う場合に選択して下さい。ただし、Vine-3.0 から、ごく基本的なパッケージしか含まれていません。Vine-2.6 と同等の開発パッケージ群をインストールするにはインストール終了後に apt 等を使って下さい。

ドキュメント (推奨) 各種パッケージのドキュメント群をインストールしたい場合には選択して下さい

Emacs (推奨) Emacs をインストールしたい場合には選択して下さい

基本サーバ (推奨) ftp / telnet / openssh 等の基本的なサーバをインストールする場合には選択して下さい

インターネットサーバ 常時 IP 接続されたマシン上で www / dhcp / nfs / netnews / netatalk などのサーバを稼働させる場合に選択して下さい

TeX 関係 TeX 環境をインストールする場合には選択して下さい

PPP 関係 ダイヤルアップ環境でネットワークに接続する場合には選択して下さい

その他 上のパッケージ群に含まれない残りのパッケージがリストアップされています。必要に応じて選択して下さい。

Linux に関する知識があって、サブパッケージを選択する意味が分かっている場合を除いて

- 基本パッケージ (default)
- X ウィンドウシステム (default)
- GNOME (default)

の 3 つだけは必ず選択しておいて下さい。

パッケージの選択が済んだら、「インストール」ボタンを押して下さい。選択したパッケージの数や、マシンの速度/CD-ROM のアクセス速度にもよりますが、インストールが終了するまでに 10 分～30 分程かかります。

アップグレードの場合には、もともとインストールされているパッケージを調べ、何をアップグレードし、何を削除するかを選ぶのに余分な時間がかかりますのでご注意ください。

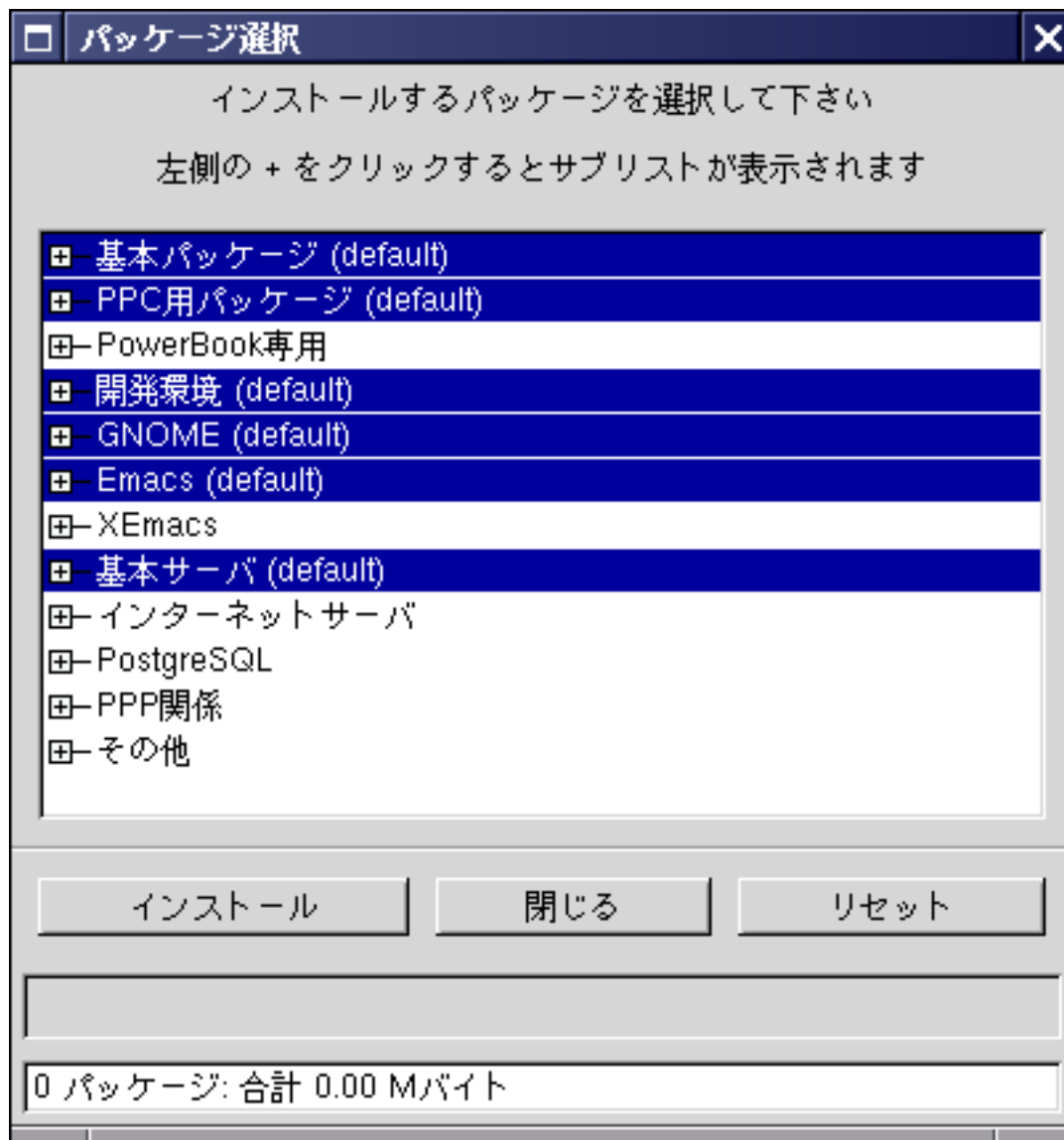


図 18: 「パッケージ選択ウィンドウ」

また、2.6 と 3.1 では収録されているパッケージが大幅に異なるため、アップグレード時には既にインストールされている**多くのパッケージが削除される場合があります**。アップグレード後に、apt-get 等を使って、それらのパッケージの 3.1 対応版 (恐らく main, devel, plus といったカテゴリに移動されているはず) をインストールしなおして下さい (削除されたパッケージに関するログは/tmp/Remv.lst を参照して下さい)。

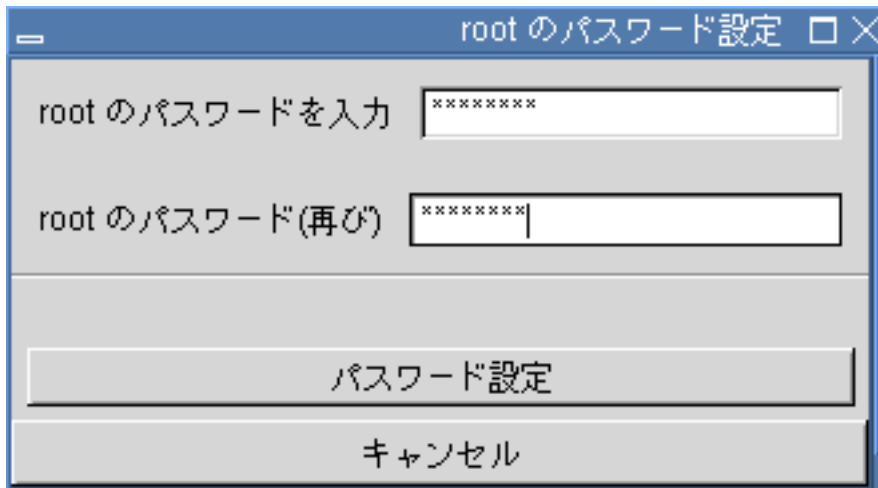


図 19: 「root パスワード設定ウィンドウ」

4.12 root のパスワードの設定

インストールが終了したら、root のパスワードを設定するウィンドウが現れます (図 19)。

ここで、root のパスワードを 2 回入力して「パスワード設定」ボタンを押して下さい。2 回入力したパスワードが一致していない場合は次に進みませんので、再び入力しなおして下さい。

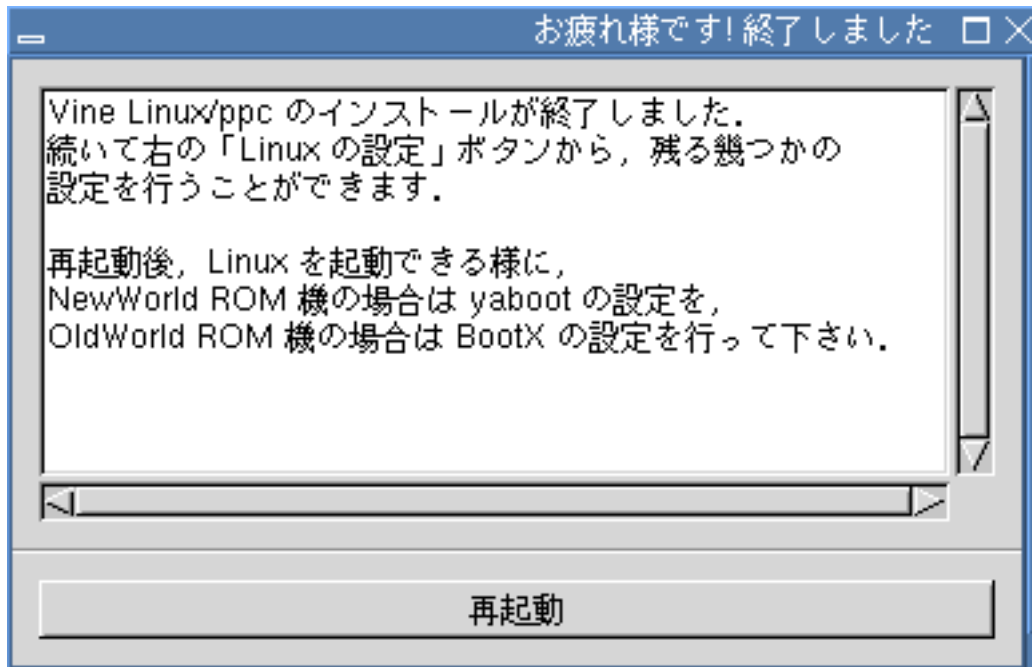


図 20: 「インストール終了」

4.13 インストール終了...

パスワードの設定が終ると、インストールが終了した旨表示されます (図 20)。

NewWorld ROM 機の場合は、**bootstrap パーティション** にブートローダをインストールする必要があります。X インストーラのメニューから「起動パーティションの設定」を選び xybin を起動して設定を行って下さい (図 21)。

終わったら、「再起動」ボタンを押して、インストーラを終了します。OldWorld ROM 機の場合は、このあと Linux を起動させる設定を行うために、MacOS を起動して下さい。



図 21: xybin の設定ウィンドウ

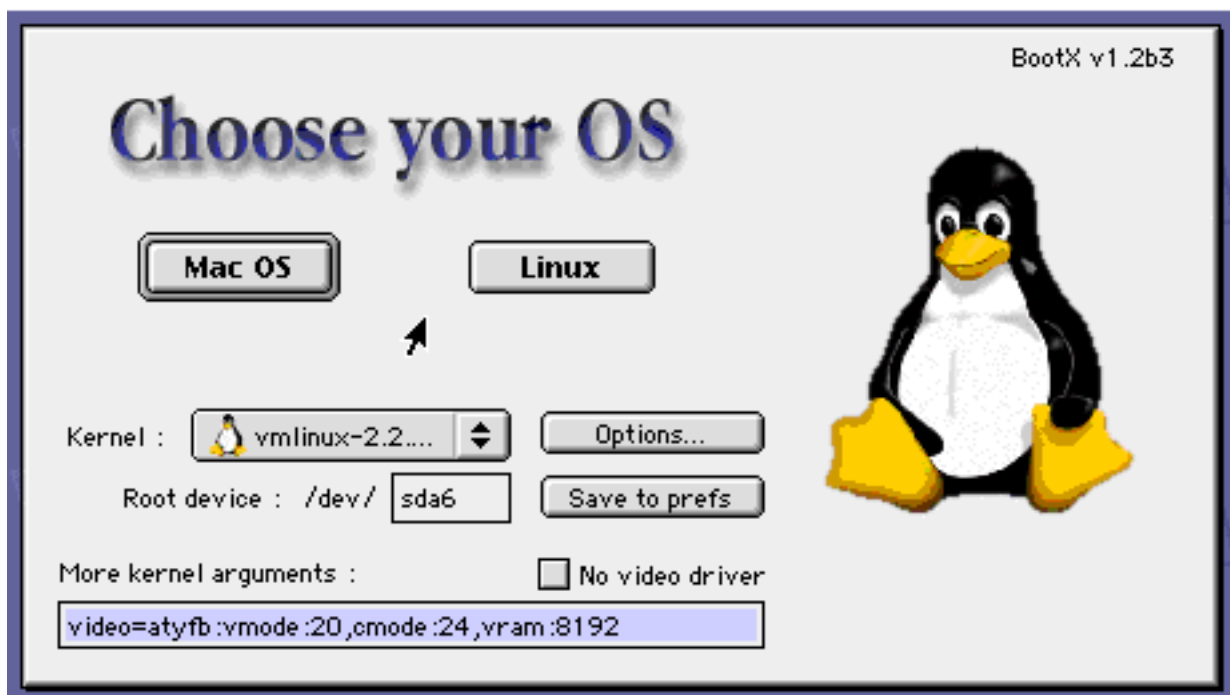


図 22: BootX パネル

5 ブートローダの設定 (OldWorld 機のみ)

NewWorld 機の場合は、インストーラ内からブートローダ (yaboot) をインストールしましたが、OldWorld 機は MacOS を介した Linux の起動が一番簡単なのでここではその方法を紹介しします。

より詳しい設定の仕方、設定項目の意味については別ドキュメント「BootX / miBoot / yaboot について」を参照して下さい。

5.1 BootX のインストールと設定

BootX は OldWorld 機用の Linux ブートローダで、MacOS 用のコントロールパネル/機能拡張の形態をとっています。MacOS らしい方法で設定を行うことが出来ます。

Vine/ppc のインストールが終了したら、一旦 MacOS を起動し、コントロールパネルから Boot VineLinux を開いて設定を行います (図 22)。

- 「Options...」ボタンを押して、「Use specified RAM Disk」のチェックを外します
- 「Kernel:」ポップアップメニューが「Vine26.Default」になっていることを確認します (SMP 機の場合は Vine26.SMP を選ぶと良いでしょう)
- 「Root Device」欄に、Linux のルート (/) パーティションを指定します
先程の例 (Section 4.10 参照) (図 17) の例ですと、/ パーティションは /dev/sda7 となります
- 「No Video Driver」チェックを外します

- 「More Kernel Arguments」欄に、必要であれば適切な記述を追加します。通常は空で構いません。
- タブキーを押して、「Mac OS」「Linux」のどちらをデフォルトにするかを選択します
- 最後に「Save to prefs」ボタンを押して今行った設定内容を保存します

慣れて来たら、お使いの Mac に応じて、より適切なカーネル引数を指定しましょう。

「No Video Driver」チェックボックスの有無、「More Kernel Arguments」欄の詳しい設定方法については別ドキュメント「BootX / miBoot / yaboot について」や web 上で公開している最新確認情報⁵を参照して下さい。

ここまで出来たら Mac OS を再起動して下さい。BootX 関係のファイルが正しくインストールされていれば、アイコンパレードが始まる直前に BootX の選択画面が現れます。ここで「Linux」ボタンを押すと、Linux が起動します。

⁵<http://vinelinux.org/ppc/compatibility/>

6 Vine/ppc の設定

このセクションでは、Vine/ppc に固有の設定について説明しています。i386 版 と共通の項目に関しては i386 版 のドキュメント⁶ を参照して下さい。

6.1 X サーバを設定する

Vine-3.1/ppc では、他アーキテクチャと同様 XOrg サーバを使用します。通常インストール時の設定がそのまま引き継がれますのでインストール直後から X を利用することが出来ますが、なんらかの理由で X の設定を変更する場合やデフォルトで利用するグラフィックカードを変更した場合には設定を変更する必要があるでしょう。

XOrg サーバを使うには、`/etc/X11/X` が `../../usr/X11R6/bin/Xorg` へのシンボリックリンクになっている必要があります。もしなっていない場合は

```
(cd /etc/X11; ln -sf ../../usr/X11R6/bin/Xorg X)
```

とコマンドを入力して下さい。

Vine/ppc での XOrg サーバの設定は、`Xautoconfig4` を使います。起動したフレームバッファの設定から自動的に `/etc/X11/xorg.conf` ファイルを生成してくれます。

解像度や色数を手動で変更したい場合には `Xautoconfig4 -i` という風に `-i` オプションをつけて実行すると、画面解像度、色数、使用ドライバなどを対話的に変更することが出来ますのでこちらをお勧めします。

⁶file:/usr/doc/HTML/documentations.html

6.2 マウスのエミュレーションを設定する

Macintosh の標準マウスは、ADB / USB 用共に 1 ボタンマウスです。ところが Linux で使用されている X ウィンドウシステム用のウィンドウマネージャやアプリケーションでは 3 ボタンマウスを前提としたものがかなりの数にのぼります。そのため、Macintosh の標準マウスでは、非常に不便な思いをしつつ X を使用する羽目になります。

ADB 接続用の多ボタンマウスは今ではほとんど流通していませんが、USB 接続用のものは Mac で使用可能なものが何種類か販売されており、それらを購入することも解決策になります。

多ボタンマウスが用意できなかった場合、キーボードの特定のキーを押してマウスの中ボタン右ボタンをエミュレートすることが可能です。

マウスボタンのキーボードによるエミュレーションを設定するには `/proc/sys/dev/mac_hid/` 以下のファイルに値を書き込むことによって行います。この設定は全 X 上のみならずコンソールマウス (gpm) においても有効です。

この新しい設定方法を有効にするには

```
# echo "1" > /proc/sys/dev/mac_hid/mouse_button_emulation
```

と行います。これでキーボードによるマウスボタンのエミュレーションが有効になります。どのキーでエミュレーションを行うかを設定するには

```
# echo "99" > /proc/sys/dev/mac_hid/mouse_button2_keycode
```

```
# echo "70" > /proc/sys/dev/mac_hid/mouse_button3_keycode
```

と値を書き込むことにより行います。

この例では F13 (SysRq) と F14 (Scroll Lock) を設定していますが、この値は `/usr/src/linux/include/linux/input.h` で定義されている値であることに注意して下さい。

Vine/ppc では `/etc/sysctl.conf` を介してシステム起動時に値を設定していますので、この設定ファイル内のエントリを変更しておくといでしょう。Vine/ppc デフォルトでは以下の様に設定されています。

```
# Controls mouse-key emulation feature
# (most pmacs have one-button mouse by default)
dev.mac_hid.mouse_button_emulation = 1

# Set the middle/right button emulation keycode
# Below we set F11 and F12 for middle and right mouse buttons respectively
# (note that you have to use values as seen in <linux/input.h>)
dev.mac_hid.mouse_button2_keycode = 87
dev.mac_hid.mouse_button3_keycode = 88
```

6.3 MacOS とデータをやりとりする

Mac OS 側とデータのやり取りをする場合は、パーティションについて (Section 3.4 参照) のところで触れた、データやり取り用に確保した HFS パーティションを介して行います。

現時点では HFS+ (HFS 拡張) パーティションを Linux 側から安定利用するにはまだ問題があります。可能な限り HFS (HFS 標準) パーティションを用意して下さい。

6.3.1 Linux から直接マウントする

一番簡単なのは、Linux 上から HFS パーティションを直接マウントしてしまう方法です。例えば、Linux 側から MacOS 側にデータをコピーする場合は、以下の様な手順を踏みます。データやり取り用の HFS パーティションが `/dev/sda7` だとすると、

```
# mkdir /mnt/mac (マウントポイントの作成)
# mount -t hfs /dev/sda7 /mnt/mac
# cp [コピーしたいファイル] /mnt/mac/.
# umount /mnt/mac
```

の様にします。

逆に MacOS 側から Linux 側にデータをコピーする場合も同様に、MacOS 上でデータやり取り用のパーティションにコピーしておき、Linux 側から上と同様にマウントし、コピーします。

ただし、Linux 上から直接 HFS パーティションの操作 (書き込み/削除) を行うと、最悪パーティションが壊れる可能性もあることが報告されていますので、あまり大量のファイルを Linux 上から HFS に書き込みしない方がよいでしょう。

6.3.2 HFS ユーティリティを使う

データのやり取りを行うもう 1 つの方法は、`hfsutils` パッケージに含まれているプログラムを使用するものです。このパッケージに含まれるプログラムを使うことで、HFS パーティション上の様々な操作 (コピー、リネーム、属性変更、フォーマット等) を行うことが可能です。

ここでは簡単な使用例を説明するに留めますが、詳しくは `hfsutils(1)` の `man` ページを参照して下さい。

```
[root@elmo root]# hmount /dev/hda5
(/dev/hda5 の HFS パーティションを仮想マウント)
[root@elmo root]# hls -l
f ???/????          0   2985330 May 30 14:18 vmlinux-2.2.14-19v13
[root@elmo root]# hcopy -m /tmp/SomeFile.bin :
(/tmp/SomeFile.bin を HFS カレントディレクトリに MacBinaryII でコピー)
[root@elmo root]# hls -l
f ???/????          0   2985330 May 30 14:18 vmlinux-2.2.14-19v13
F APPL/????         51200   102400 Jun 10 22:10 Some Application
[root@elmo root]# humount /dev/hda5
(/dev/hda5 の HFS パーティションをアンマウント)
```

6.4 Mac-On-Linux を使ってみる

MOL⁷ (Mac-On-Linux) は、Linux 上で動作するネイティブ Mac エミュレータです。描画速度はあまり速くありませんが、Mac OS に切替えることなく Linux 内から Mac OS を利用することができます。MacOS のエミュレータではないので、MOL 内で Linux を動作させることも可能です。

Vine-2.6/ppc 以降に収録した MOL では、Mac OS X も動作させられる様になっています。

6.4.1 初期設定ファイルの編集

Vine-3.1/ppc 収録の MOL 設定ファイルは、`/etc/mol/` 以下の複数のファイルに分散しています。コメントを参考に編集して下さい。

`/etc/mol/molrc.input` マウス、キーボードに関する設定ファイルです。

`/etc/mol/molrc.net` ネットワーク周りの設定ファイルです。

ネットワーク周りの実装と設定方法は、以前のバージョンとは大きく変更されています。

詳しくは `/usr/doc/mol-0.9.65/mol/doc/Networking` を参照して下さい。

`/etc/mol/molrc.video` MOL 起動直後の解像度/色数を設定したり、フルスクリーンモードでの動作を設定するためのファイルです。

`/etc/mol/molrc.macos` Mac OS 起動用の設定ファイルです。割り当てるメモリのサイズ、サウンドドライバ、使用する Mac OS 用パーティションなどを指定します。

`/etc/mol/molrc.osx` Mac OS X 起動用の設定ファイルです。割り当てるメモリのサイズ、使用する Mac OS X 用パーティションなどを指定します。

その他、数多くの設定項目がありますので、詳しくは設定ファイルのコメントや付属の `man` ページを御覧下さい。

⁷<http://www.maconlinux.org/>

6.4.2 MOL を起動する

初めて MOL を実行する際には、その前に `molvconfig` というコマンドを実行する必要があります。

```
# molvconfig
```

画面に関する質問に Y や N で答えてゆくと、テストパターンが表示され、終了します。

ここまで出来たら、MOL を起動してみましょう。

```
# startmol
```

設定が正しく行われていれば、X 上の 1 ウィンドウ、もしくは 1 コンソールで Mac OS が起動します。

X 上の 1 ウィンドウとしてではなく、1 コンソール上でフルスクリーンで MOL を動作させるには、`/etc/mol/` 以下の設定ファイルを修正する必要があります。MOL バージョン 0.9.53 以降では `molvconfig` というツールを使ってフルスクリーンモードの設定を行うことができます。

6.4.3 サウンドを有効にする

Vine-3.1/ppc に収録したバージョンの MOL では、Mac OS X 上でのサウンドはサポートされていません。Mac OS 上では以下の方法でサウンドを有効にできます。

MOL を使って Mac OS を起動すると、「MOL」というボリュームがマウントされているはずですので、この中に入っている「MOLAudio」という機能拡張をシステムフォルダ内の「機能拡張」フォルダにコピーして下さい。

以前に古いバージョンの MOL を使っていた場合は、MOLAudio の互換性がない場合がありますので、改めて MOLAudio をコピーし直して下さい。

その後、サウンド出力を「MOLAudio」に指定します。「サウンド」コントロールパネルを使って設定します。

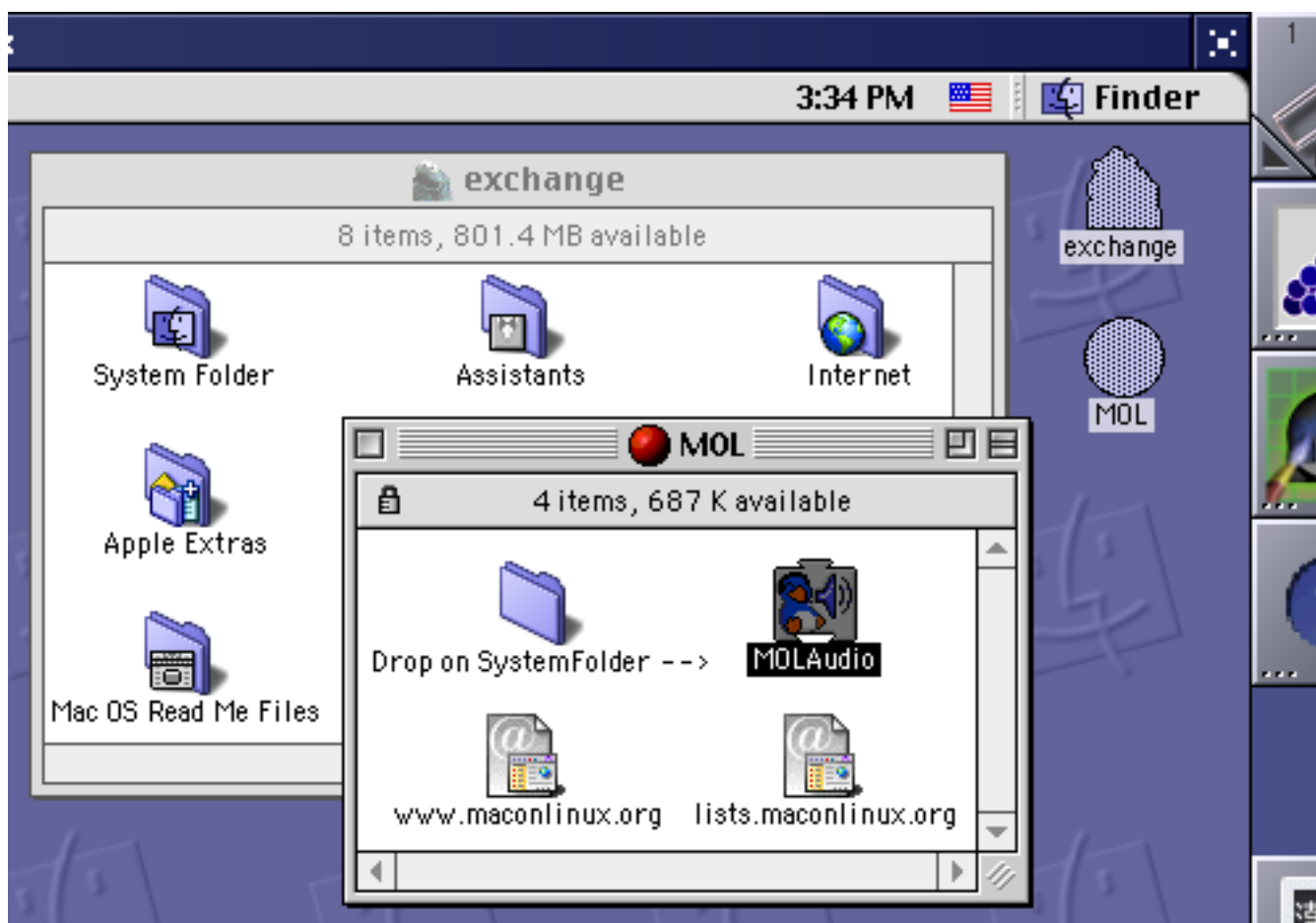


図 23: MOLAudio 機能拡張をシステムフォルダにコピー

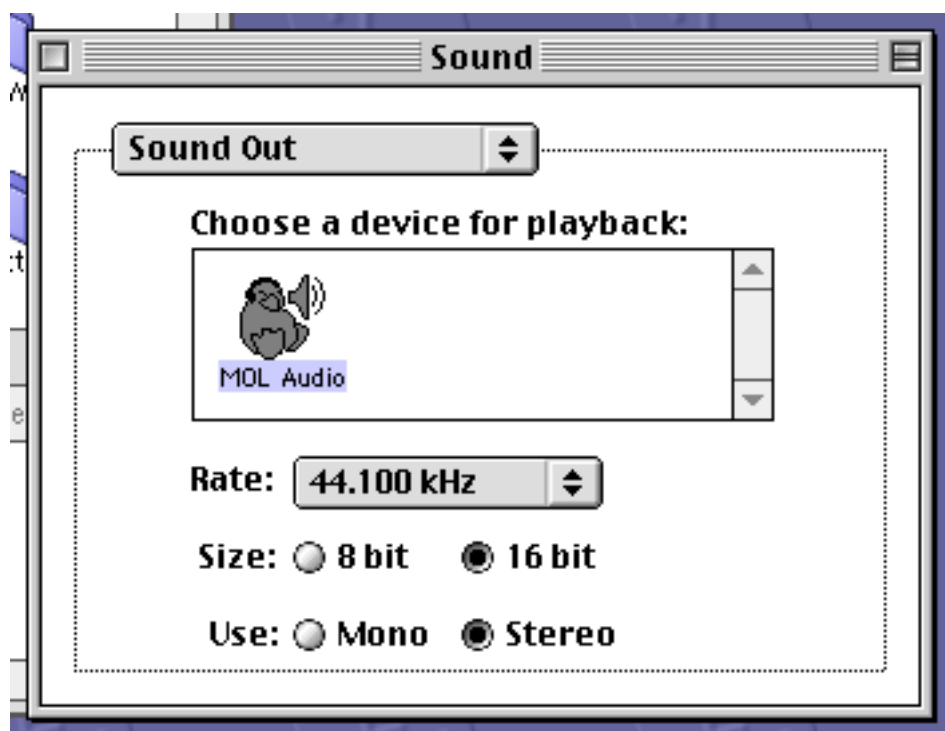


図 24: 「サウンド」コンパネで MOL Audio を選択

7 最後に

Vine Linux/ppc をまとめるにあたり，ここに書き切れない程多くの個人/団体（順不同）に多大な御指南/御協力を頂いたり，公開されているパッケージを参考にさせて頂いたりしました．この場を借りて関係各位に感謝の意を表させていただきます．

VineSeed ML の皆さん	-
VineSeed-ppc ML の皆さん	動作テスト，バグ出し
linuxppc-user ML の皆さん	-
linuxppc-jp ML の皆さん	-
小林 泰三 さん	kernel / pcmcia / pmud 関連
天野 和浩 さん	XFree86 関連
及川 隆一 さん	Xpmac 関連
花井 一光 さん	iMac DV / iMac (Summer 2000) 上の X サーバ関連
福井 薫 さん	gcc / glibc 関連
大江 貴志 さん	mol 関連
かねこ ただひろ さん	ybin の iMac Rev.B での動作確認等
もんでん あきと さん	604e SMP 機でのテスト

A PowerMac 用ブートローダについて

このセクションは、PowerMacintosh 上で Linux を起動するのに使われるツール (ブートローダ) のうち、Vine Linux に同梱している BootX、miBoot、yaboot (ybin/xybin も含む) という 3 つの紹介とその設定の仕方についての説明です。

これらのツールは全て、PPC 向け Linux で数多くの貢献をされている Benjamin Herrenschmidt⁸ さんによって作られたものです。BenH さんの web ページ⁹ でこれらツールの最新版などが公開されています。

A.1 調べておくこと

どの方法で起動するにせよ、必ず調べておかなければならないのは

Linux の root パーティションがどこにあるか

ということです。インストール時のパーティション設定を行う時に控えておくとよいでしょう。

X インストーラ上で「ドライブパーティション設定」や「パーティションの選択」を行った際に図 25 や図 26 のようになっていたとします。

この例の場合、

- MacOS のシステムフォルダがあるパーティション は hda8
- Linux の root パーティション は hda10

になります。

⁸<mailto:benh@kernel.crashing.org>

⁹<http://penguinppc.org/%7Ebenh/>

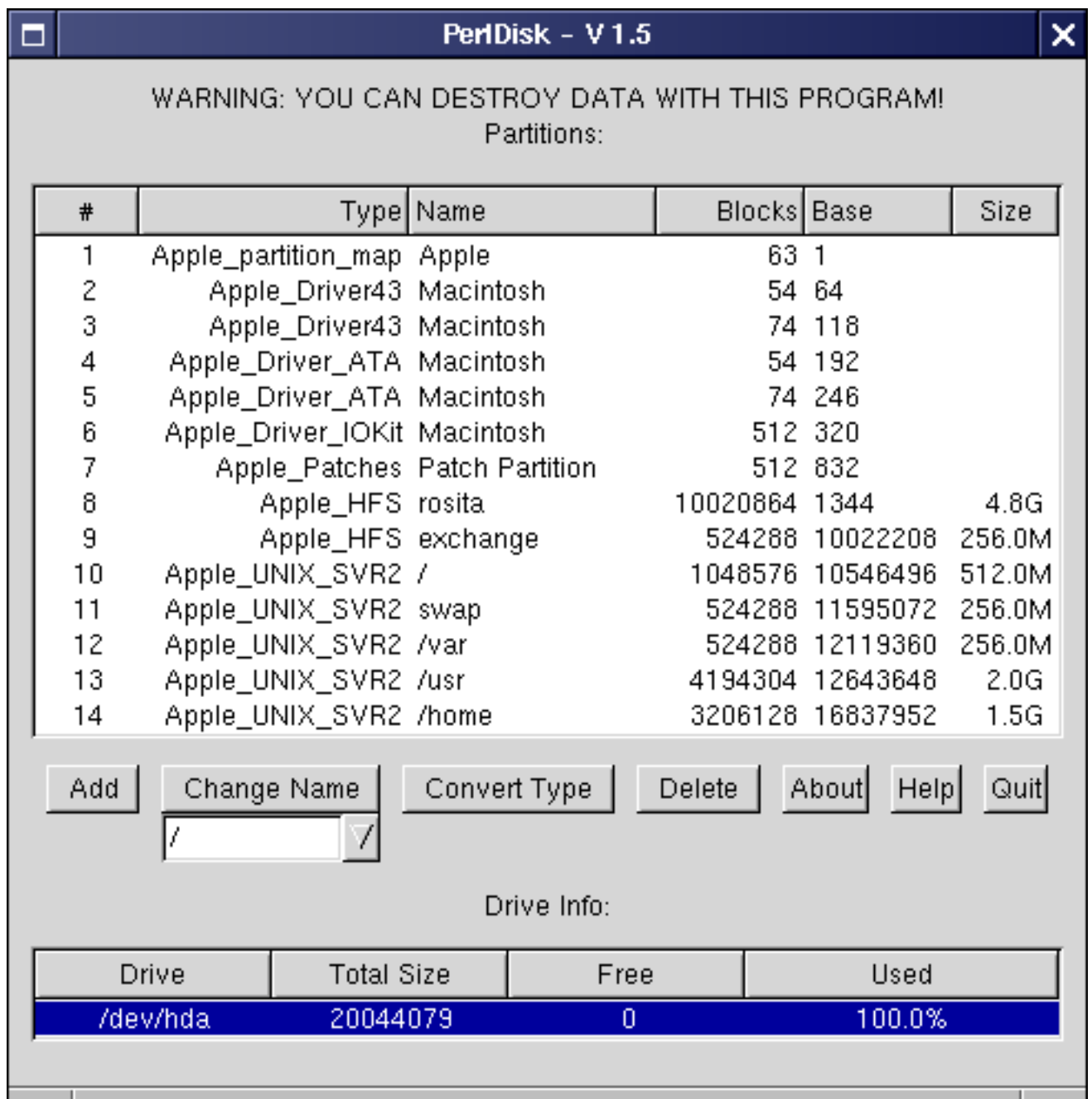


図 25: 「ドライブパーティション設定」画面 (perldisk)



図 26: 「パーティションの選択」画面 (xinstaller)



図 27: BootX パネル

A.2 BootX について

BootX は、OldWorld ROM 機用の Linux カーネルローダで、MacOS 用コントロールパネル/機能拡張の形態をとり、MacOS の起動途中で MacOS と Linux を切替えるパネルを表示するものです。起動する際のカーネルを選んだり、起動時のパラメータなどを与えることができます。

OldWorld 機で Linux と MacOS を共存させたい場合に使われます。Vine-2.6/ppc ではバージョン 1.2.2 が収録されています。

BootX および BootX Extension が正しくインストールされていれば、MacOS の起動時に MacOS と Linux の切替えパネル (図 27) が現れます。以下、各部分の説明を行います。

Vineppe では BootX のファイル名は **Boot VineLinux** となっていますが、全く同じものです。

A.2.1 Mac OS ボタン

Mac OS ボタンを押すと、Linux を起動することなくそのまま MacOS の起動を続けます。

A.2.2 Linux ボタン

Linux ボタンを押すと、

- Kernel ポップアップメニュー (A.2.3) で選んだカーネルを使い、
- Root device (A.2.4) エントリ、

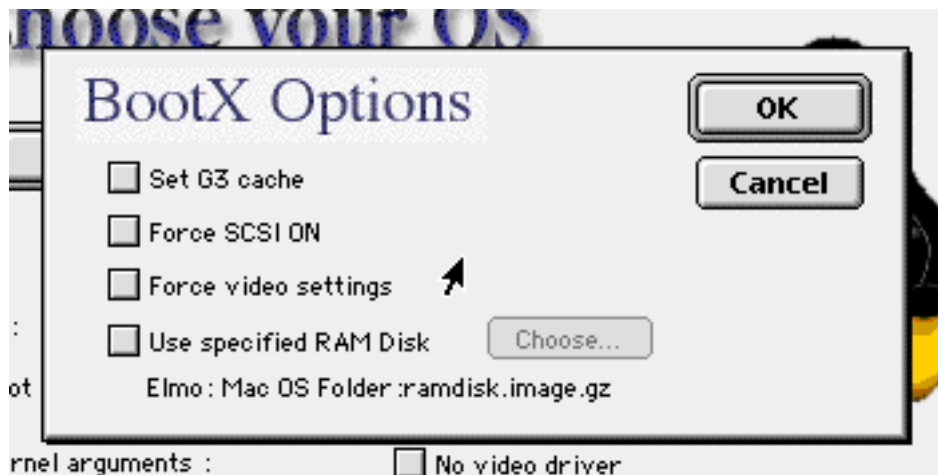


図 28: BootX のオプションパネル

- Options... (A.2.5) での設定 ,
- No Video driver (A.2.7) チェックの有無 ,
- More kernel arguments (A.2.8) の記述

に従って Linux を起動します .

A.2.3 Kernel ポップアップメニュー

ここには , システムフォルダ内にある「Linux Kernels」フォルダの内容が表示されます . このポップアップで , 起動するカーネルを選択して下さい .

通常はデフォルトの「Vine26_Default」を選択して下さい . CPU を 2 個以上積んでいる機種の場合は「Vine26_SMP」を利用することも出来ます . 古い機種の場合は「Vine26_Kernel22」で起動する方が安定する場合があります .

自分でコンパイルしたカーネルで起動する場合には , linux 上でビルドしたのち , Mac OS 上に持ってきて「Linux Kernels」フォルダに入れて下さい .

A.2.4 Root device エントリ

ここに「調べておくこと (Section A.1 参照)」で事前に調べておいた , Linux の root ディレクトリが含まれるパーティションを指定します . このエントリを書き間違った場合は Linux は正しく起動できませんので注意して下さい .

なお , 以下で述べる「Use specified RAM Disk」がオンになっている場合は「Root device」の代わりに「Ramdisk size」エントリが表示されます .

A.2.5 Options... ボタン

このボタンを押すと , 各種オプション設定 (図 28) が行えます .

- Set G3 cache

G3 アップグレードカードを使用しているマシンで G3 Level2 キャッシュを有効にするにはここをチェックします。

- Force SCSI ON

一部の機種で、MacOS 9.0 のドライブ設定でフォーマットされた SCSI ディスクがうまく使えない場合、ここをチェックすると直る場合があるとのことです。

- Force video settings

No video driver (A.2.7) を使うと MacOS の解像度/色数を引き継いで Linux を起動することができますが、一部の機種/ディスプレイの組合せでは、BootX のパネルが表示されるタイミングでは 640x480 になっている場合があります。

このオプションをチェックしておく、Linux を起動する直前に MacOS で本来設定されている解像度/色数に変更してくれます。ただし、このチェックをしても必ずしも正しく動作するとは限らないようです。

- Use specified RAM Disk

インストール時などの特殊な場合、起動する際に読み込む RAM ディスクを使う際にはここをチェックします。普段 Linux を使う際にはチェックする必要はありません。

A.2.6 Save to prefs ボタン

このボタンを押すと、BootX パネル上で設定した内容が全て保存されます。Mac OS ボタン (A.2.1) と Linux ボタン (A.2.2) のどちらをデフォルトにするかをタブキーを押して変更し、その後 Save to prefs ボタンを押すとよいでしょう。

A.2.7 No video driver チェックボックス

このチェックボックスは、Mac OS での解像度/色数をそのまま引き継ぎ、Open Firmware のフレームバッファを使って Linux を起動する場合に使用します。ただし、これをオンにしていると画面描画のアクセラレーションが有効になりませんし、X が正しく動作しなくなることもあります。

お使いの Mac に対応するフレームバッファ名と最大解像度が分かるならば、ここのチェックはオフにして、次に述べる「More kernel arguments」欄に詳細を記述する方がよいでしょう。

A.2.8 More kernel arguments エントリ

このエントリには、カーネルに渡す引数を記述します。使用するフレームバッファ名、解像度、色数、起動時のランレベルなどを指定するのによく使われます。ただし、通常は特に何も指定しなくても構いません。

詳しい記述方法については「カーネル引数の詳細 (Section B 参照)」を御覧ください。

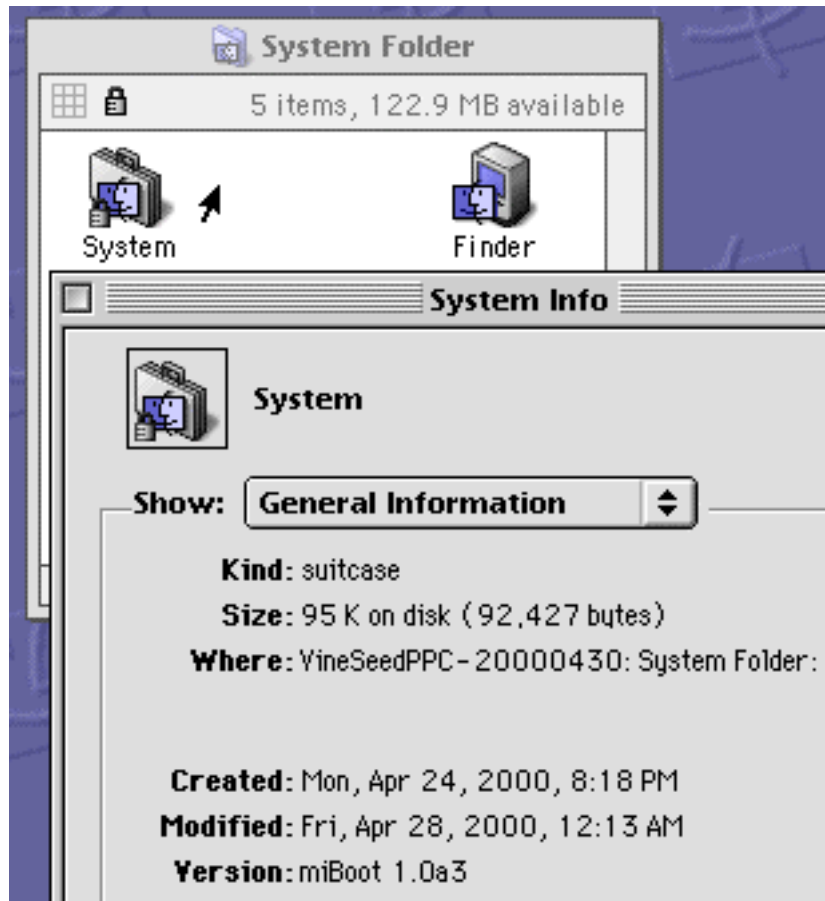


図 29: miBoot

A.3 miBoot について

miBoot (図 29) も OldWorld 機用のブートローダで、MacOS 上で見る限りはシステムファイルの様に見えますが、その実態は BootX と同等の Linux カーネルローダで、各種設定はリソースを編集することで行います。

OldWorld 機で MacOS を完全に排除して Linux を使う場合に使われます。Vine-2.6/ppc ではバージョン 1.0a3 が収録されています。

NewWorld 機の Macintosh は起動直後、ハードディスクから「Finder」と「System」が含まれるシステムフォルダを探しだし、その System ファイルを使って起動を始めます。その原理を利用して、偽の「System」ファイル (miBoot) とダミーの「Finder」を置いておくことで、Linux を起動させるというのが miBoot の仕組みです。

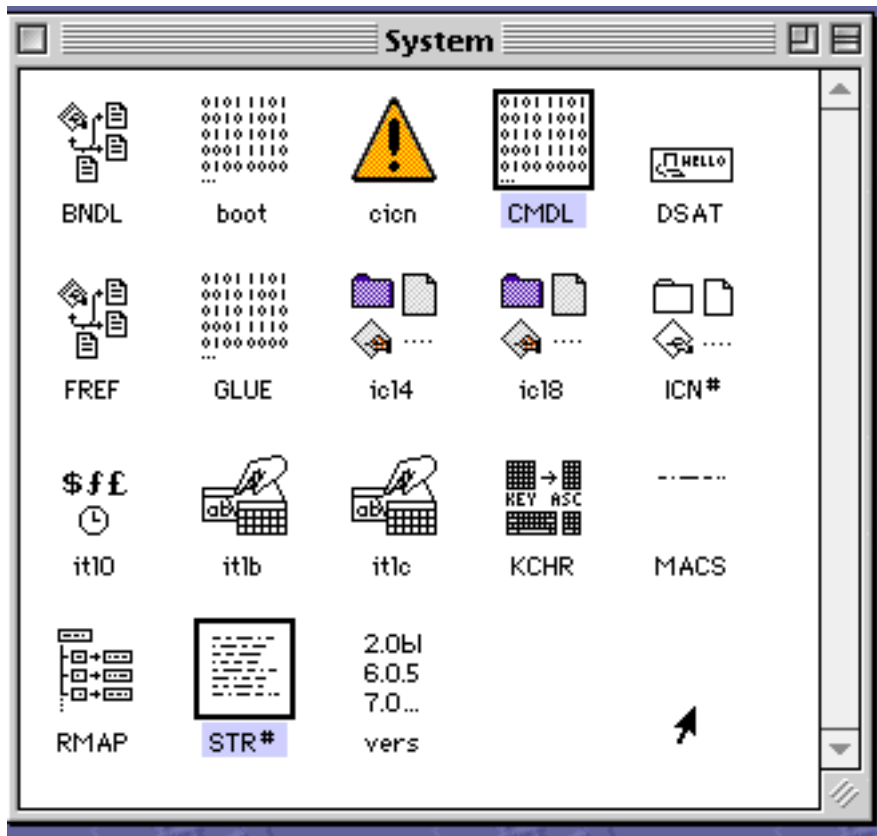


図 30: miBoot を ResEdit で開いたところ

A.3.1 miBoot 設定の準備

miBoot の設定には ResEdit が必要です。手元にない場合は

http://www.apple.co.jp/ftp-info/reference/resedit_2.1.3.html

などから取得して下さい。また、ResEdit の操作を間違えるとそのファイルを最悪壊してしまいますので、よく分からない場合は止めておいた方が無難かと思われます。

System ファイル (miBoot) を ResEdit で開きます (図 30)。この中の CMDL リソースと STR# リソースを編集する必要があります。

A.3.2 起動時の引数の設定

CMDL リソース (図 31) では、起動時に Linux カーネルに渡す引数を指定します。

CMDL ID=128 リソースには通常の起動時の引数を記述します。CMDL ID=129 リソースにはシフトキーを押したまま起動した際の引数を記述します。これによりシフトキーを押しているか押していないかでカーネルを切替えたりフレームバッファの指定を変更したりできます。

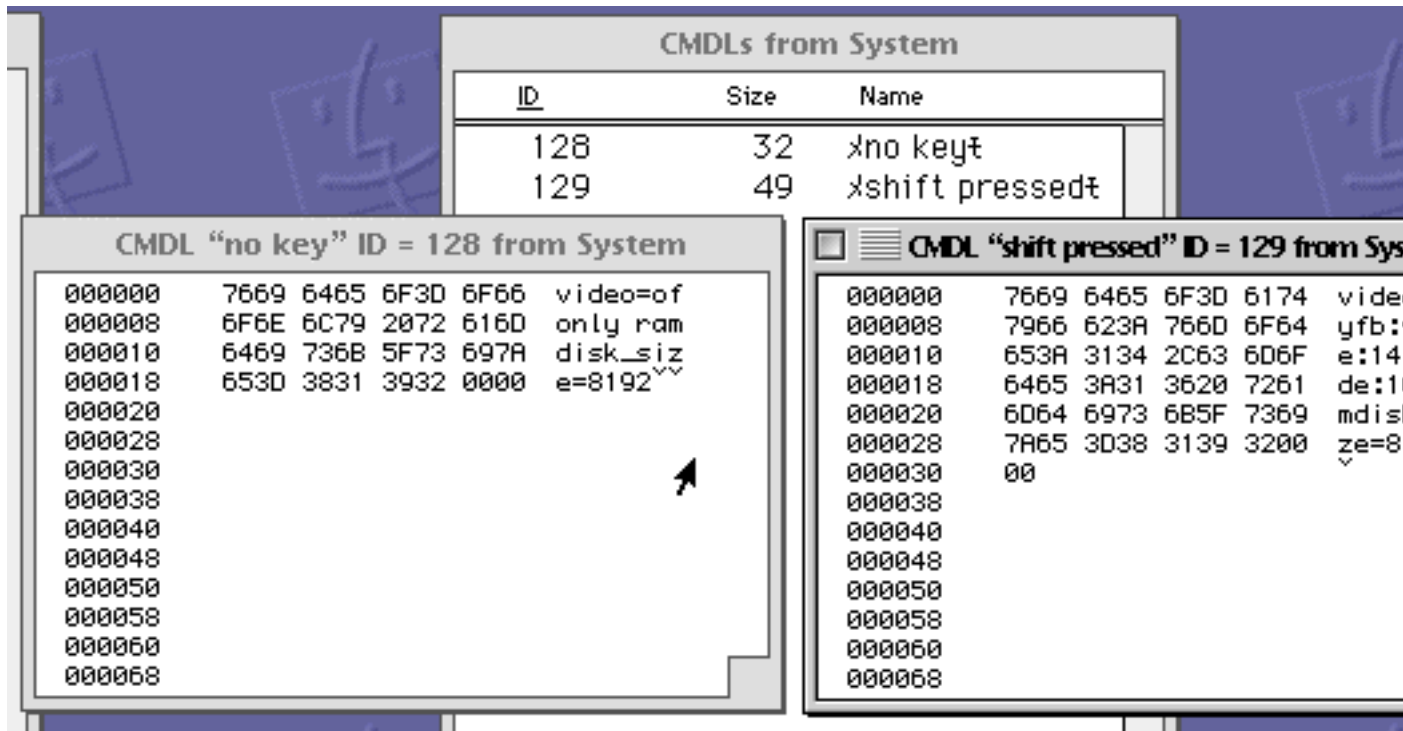


図 31: miBoot の CMDL リソース (例)

A.3.3 カーネルと ramdisk の指定

STR# リソース (図 32) では、起動時に使用するカーネルのファイル名と、ramdisk のファイル名を記述します。

STR# ID=128 リソースにはカーネルのファイル名を記述します。

STR# ID=129 リソースには ramdisk のファイル名を記述します。ここは通常は空欄にしておいて下さい。

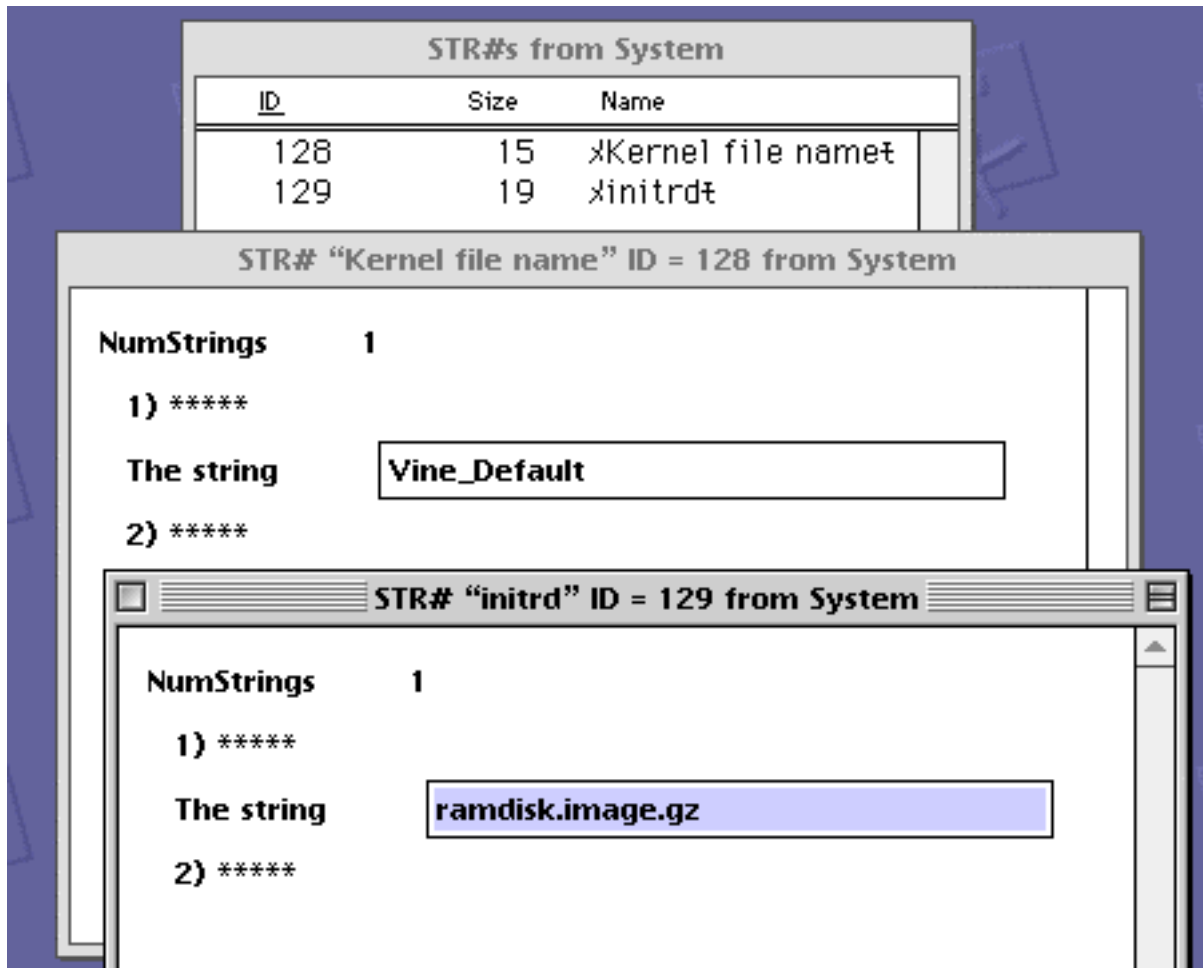


図 32: miBoot の STR# リソース (例)

A.4 yaboot / ybin / xybin について

yaboot は iMac, iBook, G&W G3, G4, Lombard 以降の PowerBook などいわゆる NewWorld 機用の Linux ブートローダです。Vine-2.6/ppc では yaboot バージョン 1.3.6 が収録されています。

yaboot では、Intel 版で使われるブートローダ lilo と同様の設定ファイルを編集することで簡単に設定が行えます。小さな起動用パーティションを用意しておけば、ybin / xybin を併用することで、更に起動設定を楽に行えます。

以下、インストール時に起動用パーティション (magicboot) を用意している、という前提で説明します。

A.4.1 起動パーティションの必要性

Vine/ppc では、NewWorld 機の場合はインストール時に小さな起動パーティションを確保することを推奨しています。このパーティション上に、OS 切替え用の Open Firmware スクリプトと yaboot をインストールするのが ybin と xybin です。

電源投入

[1st ステージブートローダ]

起動パーティション上の Open Firmware スクリプトが実行される
ここで Linux, Mac OS, Mac OS X 切替えを行う

(Mac OS 起動)

(Mac OS X 起動)

[2nd ステージブートローダ]

起動パーティション上の yaboot 起動
ここで、どの kernel で起動するかを切替える

(Linux 起動)

```

Partition map (with 512 byte blocks) on '/dev/hda'
#:          type name          length  base    ( size )
1: Apple_partition_map Apple          63 @ 1
2:  Apple_Driver_ATA*Macintosh          54 @ 64
3:  Apple_Driver_ATA*Macintosh          74 @ 118
4:  Apple_FWDriver Macintosh          200 @ 192
5: Apple_Driver_IOKit Macintosh          512 @ 392
6:  Apple_Patches Patch Partition          512 @ 904
7:  Apple_HFS "Macintosh HD" 2048000 @ 1416 (1000.0M)
8:  Apple_Boot MOSX_OF3_Booter  16384 @ 2049416 ( 8.0M)
9:  Apple_Loader SecondaryLoader  1024 @ 2065800
10: Apple_UFS Mac_OS_X 12270592 @ 2066824 ( 5.9G)
11: Apple_UNIX_SVR2 swap          262144 @ 14337416 (128.0M)
12: Apple_UNIX_SVR2 /          2097152 @ 14599560 ( 1.0G)
13: Apple_UNIX_SVR2 /usr          8388608 @ 16696712 ( 4.0G)
14: Apple_UNIX_SVR2 /home        15052800 @ 25085320 ( 7.2G)
15: Apple_HFS "Vine Linux Boot" 50830 @ 40138120 ( 24.8M)

```

図 33: パーティションマップの例

A.4.2 ybin を使った yaboot のインストール

yaboot を起動パーティションにインストールする為のツール **ybin** を使うと起動設定がより簡単に、かつ Linux の中から行うことができます。ybin はコマンドラインベースのシェルスクリプトスクリプトです。これらを使うと、yaboot.conf の自動設定はもちろん Open Firmware の環境変数の設定も自動で行うことができます。

以下の説明では、ハードディスクのパーティションが次の様になっている場合を例にします。

この例では

- /dev/hda7 - MacOS のパーティション
- /dev/hda8 - (MacOS X PB の起動パーティション)
- /dev/hda9 - (MacOS X PB の起動パーティション)
- /dev/hda10 - MacOS X PB のパーティション (UFS)
- /dev/hda12 - Linux の root パーティション
- /dev/hda15 - 起動パーティション (magicboot)

となります。

A.4.3 yaboot.conf の編集

まず yaboot.conf ファイルを、お使いのマシンにあわせて編集します。以下、各項目を簡単に説明します。

boot=(起動パーティション)

yaboot と起動スクリプトをインストールするパーティションを指定します。

図 33 の例では、/dev/hda15 です。

defaultos=(デフォルト起動する OS 名)

起動 OS 選択画面で、何も入力しなかった場合にデフォルトで起動する OS を指定します。

linux, macos, macosx のいずれかを指定します。

delay=(秒単位)

上の defaultos を起動するまでの、入力待ち時間を秒単位で指定します。

ここまでが、起動 OS 選択スクリプト用の設定です。以下、Linux ブートローダ yaboot の設定になります。

init-message="(文字列)"

yaboot 起動時に表示されるメッセージを指定します。改行を表すには \n を使います。

timeout=(1/10 秒単位)

yaboot 起動後、何も入力されなかった時に、次の default が実行されるまでの待ち時間を 1/10 秒単位で指定します。つまり 300 と指定すれば待ち時間が 30 秒になるということです。

default=(設定名)

yaboot プロンプトで、単純にリターンを押した場合に選択されるデフォルト設定のラベル名を指定します。以下の label を使って与えられる設定名です。

image=(読み込むカーネルのありか)

読み込むカーネルファイルの名前とそのパーティション名を指定します。書式は [デバイスへのパス名]:[パーティション番号],[カーネル名] となります。読み込むカーネルファイルが Mac OS のシステムフォルダ内にある場合には [カーネル] の先頭に \\\ をつけます。逆に、ハードディスクのトップディレクトリに置いてある場合はつける必要はありません。

linux 用 ext2/ext3 パーティション上のファイルを直接指定することができます。先程の 図 33 の場合だと、次の様な指定が可能です。

```
image = hd:12,/boot/vmlinuz-2.4.19-0v18
```

label=(ラベル名)

この設定に対応するラベル名を指定します。

root=(Linux の root パーティション)

Linux の root パーティションを指定します。

先程の 図 33 の例の場合は root = /dev/hda12 となります。

append=“ (カーネルに与える引数)”

BootX の More kernel arguments (Section A.2.8 参照) に記述するのと同様に、カーネルに渡す引数を指定します。

引数の最初にはスペースを入れておく方が良いでしょう。そうでないと、カーネルにうまく引数が渡らない場合があります。

novideo

BootX の No video driver (Section A.2.7 参照) のチェックを与えるのと同様の効果を得たい場合、この行を追加します。

注意する必要があるのは、BootX の場合はパネルが表示されている状態の解像度/色数 (つまり MacOS の設定) を Linux に引き継ぐのに対し、yaboot 上で novideo を与えた場合は Open Firmware の解像度/色数を引き継ぐということです。一般的に Open Firmware コンソールの色数は 8bpp です。コンソールの解像度/色数をもっと高精度のものにしたい場合は、この novideo は使わず、上の append = "... " で指定する必要があります。

特に append と novideo に関しては機種によって設定をシビアに行う必要がありますので、「web 上で公開している最新確認情報¹⁰」の該当マシンの解説を熟読されることをお勧めします。

¹⁰<http://vinelinux.org/ppc/compatibility/>

`macos=(Mac OS の起動パーティション)`

Mac OS の起動パーティションを指定します。
先程の 図 33 の例では `/dev/hda7` となります。

`macosx=(Mac OS X の起動パーティション)`

Mac OS X の起動パーティションを指定します。

先程の 図 33 の例では、Mac OS とは別に UFS パーティションを切り、そこに OS X をインストールしていますので、その起動パーティション (Apple_Boot) である `/dev/hda8` を指定します。

最近の Mac は、単一の HFS+ パーティションに Mac OS と Mac OS X の両方が出荷時にインストールされています。この場合には、`macosx=` という行の直後に `brokenosx` と書いて下さい。

ただし、同一パーティションにインストールされた Mac OS と Mac OS X の切替えを `ybin` スクリプトから行うには、Vine Linux インストール前に「起動ディスク」コントロールパネルを使って Mac OS 9 を起動 OS に指定しておく必要があります。

A.4.4 ybin の実行

yaboot.conf の設定内容をよく確認したあと, ybin を実行します. 実行内容を確認するために -v オプションをつけるとよいでしょう.

```
[root@bart /root]# ybin -v
ybin: Finding OpenFirmware device path to '/dev/hda15'...
ybin: Finding OpenFirmware device path to '/dev/hda7'...
ybin: Finding OpenFirmware device path to '/dev/hda8'...
ybin: Installing first stage bootstrap /usr/lib/yaboot/ofboot onto /dev/hda15...
ybin: Installing primary bootstrap /usr/lib/yaboot/yaboot onto /dev/hda15...
ybin: Installing /etc/yaboot.conf onto /dev/hda15...
ybin: Setting attributes on ofboot...
ybin: Setting attributes on yaboot...
ybin: Setting attributes on yaboot.conf...
ybin: Blessing /dev/hda15 with Holy Penguin Pee...
ybin: Updating OpenFirmware boot-device variable in nvram...
[root@bart /root]#
```

これで, /dev/hda15 の起動パーティション (magicboot) に OS 切替え用スクリプトと yaboot が適切にインストールされます.

再起動すると, ブートストラップメニューが表示されます. M を押すと MacOS を起動, L を押すと Linux を起動, C を押すと CD-ROM から起動, という風に切替えることができます. 10 秒待つと上の「デフォルト起動 OS」で指定した OS が起動します.

A.4.5 xybin について

ybin の GUI ラッパーとして **xybin** を用意しています (図 34) . これを使うことにより , yaboot.conf を直接編集することなく対話的にブートローダの設定を行うことができます .

xybin は , X インストーラの「起動パーティションの設定」ボタンを押すことで起動することが出来ます . また , インストール後には , 例えば GNOME デスクトップ環境では「プログラム」 「システム」 「yaboot インストーラ」から起動できます .

A.4.6 Option ブートで切替える

PowerMac G4 (AGP Graphics) , iMac DV , PowerBook G3 (Pismo) , iBook などより新しい機種では , 起動時にキーボードの Option キーを押しておけば , 接続されている全デバイスをスキャンして起動可能なパーティションをアイコン表示してくれる機能が備わっています .

Linux インストール後に Mac OS や Mac OS X を起動し , 間違えて「起動ディスク」コントロールパネル設定してしまうと , Linux の起動パーティションから起動しなくなってしまいます .

この際は , 電源投入直後に Option キーを押し続け , ペンギンアイコンのついたパーティションから起動します . その後 , Linux を起動し , ybin を再度実行すれば元に戻すことができます .



図 34: xybin の設定ウィンドウ

B カーネル引数の詳細

このセクションでは、BootX (Section A.2 参照) の More kernel arguments (Section A.2.8 参照) , yaboot (Section A.4 参照) の append 行 (Section A.4.3 参照) , miBoot (Section A.3 参照) の CMDL リソース (Section A.3.2 参照) で指定する、カーネルに渡す引数のうちよく使うであろう代表的なものについて説明しています。

カーネルに渡す引数を適切に指定することにより、

- 起動時のランレベル
- 使用するフレームバッファ
- 画面解像度/色数

などを設定することができます。

B.1 ランレベルに関する引数

Linux 起動時に、どのランレベルに入るかを指定できます。代表的なものには、以下の様な指定があります。

ランレベルの指定	
single	シングルユーザモードに入ります
1	シングルユーザモードに入ります
3	マルチユーザモード (X なし) に入ります
5	マルチユーザモード (X あり) に入ります
0	シャットダウンします
6	再起動します

B.2 画面に関する引数

BootX で「No Video Drivers」チェックボタンをオフにしている場合や yaboot で「novideo」行を追加していない場合に、使用するフレームバッファ名や画面解像度/色数を指定することができます。

各機種毎に、具体的にどのような設定が望ましいか、どのような設定だと正しく動作しないことが確認されているかは、「web 上で公開している最新確認情報¹¹」を御覧ください。

通常は Linux カーネル起動時に自動検出されますので、特に指定する必要はありませんが、ビデオカードを複数装着しているマシンの場合や、引数をきちんと与えないと画面表示がされないビデオカードの場合は、適切な設定を行う必要があります。

画面に関する引数の書式は以下の通りです。

`video=[フレームバッファ名]:vmode:[解像度],cmode:[色数],vram:[VRAM 搭載量]`

(例: `video=aty128fb:vmode:20,cmode:24`)

radeon や riva フレームバッファの場合は書式が若干異なります。

`video=[フレームバッファ名]:[横ピクセル数]x[縦ピクセル数]-[色数]@[周波数]`

(例: `video=radeon:1024x768-8@60`)

実際にはこれらの他に、フレームバッファドライバごとに `rclk`, `mclk`, `pll`, `flatpanel`, `dfp` 等数多くの引数を与えることが可能です。詳しくはカーネルソースと付属ドキュメントを御覧ください :-)。

¹¹<http://vinelinux.org/ppc/compatibility/>

B.2.1 フレームバッファ名の指定

PowerMac で使用可能なフレームバッファ名には以下の様なものがあります．ここに記したマシン名は，全てオンボードのビデオ端子，あるいはシステム純正付属のビデオカードについての情報です．御自分でビデオカードを増設されてそちらをメインで使っているいる場合は，そのビデオカードに対応するフレームバッファ名を指定して下さい．

フレームバッファ名と対応機種	
ofonly	No video driver チェックボックス ON (BootX), novideo (yaboot) と同じ動作するフレームバッファドライバがない場合は，これを選びます 8bpp でしか動作しません
controlfb	PowerMac 7300/7500/7600/8500/8600 オンボードビデオ
platinumfb	PowerMac 7200/8200, PowerCenter オンボードビデオ
valkyriefb	PowerMac 5400/6400 オンボードビデオ
chipsfb	PowerBook 2400/3400/初代 PowerBook G3 (3500)
matroxfb	Matrox Millenium シリーズ / Mistique
imsttfb	PowerMac 9600 IMS TwinTurbo カード
atyfb	PowerMac 9500 PowerMac G3 (Beige) DT/MT/All-In-One iMac Rev.A ~ Rev.D PowerBook G3 1999 (Lombard) iBook (オリジナル) その他 ATI Rage C / Pro などのチップを搭載したビデオカード
aty128fb	PowerMac G3 (Blue & White) / G4 iMac DV (Slot-Loading) 以降 iBook (FireWire) 以降 PowerBook G3 2000 (Pismo) PowerBook G4 (Titanium I) その他 ATI Rage128 系列のチップを搭載したビデオカード
radeon	Radeon PCI/AGP カード iBook (16M VRAM) PowerBook G4 (Titanium II/III) PowerBook G4 (Aluminium 15 インチ) その他 ATI Radeon 系列のチップを搭載したビデオカード
riva	nVidia GeForce 系列のカード iMac G4 (15 インチ LCD モデル) PowerBook G4 (Aluminium 12/17 インチ) その他 nVidia GeForce 系列のチップを搭載したビデオカード

B.2.2 画面解像度の指定

これらの引数は、radeon ドライバ、riva ドライバ以外の古い Macintosh 向けフレームバッファで有効です。

vmode で指定する解像度は数字で指定します。解像度は同じでも、ディスプレイによっては特定の周波数しか対応しない場合、これらを使って明示的に指定する必要があります。

VMODE に指定可能な数字とその意味			
1	512x384 60Hz (Interlaced-NTSC)	11	800x600 72Hz
2	512x384 60Hz	12	800x600 75Hz
3	640x480 50Hz (Interlaced-PAL)	13	832x624 75Hz
4	640x480 60Hz (Interlaced-NTSC)	14	1024x768 60Hz
5	640x480 60Hz	15	1024x768 72Hz
6	640x480 67Hz	16	1024x768 75Hz
7	640x870 75Hz (Portrait)	17	1024x768 75Hz
8	768x576 50Hz (Interlaced-PAL)	18	1152x870 75Hz
9	800x600 56Hz	19	1280x960 75Hz
10	800x600 60Hz	20	1280x1024 75Hz
		21	1152x768 75Hz (PowerBook G4 用)
		22	1600x1024 60Hz (Cinema Display 用)

B.2.3 画面の色数の指定

cmode で指定する色数は数字で指定します。

CMODE に指定可能な数字とその意味	
8	8bpp カラー
15	15bpp カラー
16	16bpp カラー
24	24bpp カラー
32	24 と同じ

B.2.4 VRAM 搭載量

この項目は、一般的には指定する必要はありません。web 上で公開している最新確認情報¹² に注記した ATI XClaim 3D の例の様に、ビデオカードに搭載されている VRAM 容量を誤認されたために動作が不安定になった場合にのみ指定するとよいでしょう。

¹²<http://vinelinux.org/ppc/compatibility/>

B.2.5 複数のビデオカードを搭載した機種の場合

オンボードビデオを搭載した機種に、PCI 接続のビデオカードを増設している場合など、ビデオ出力が複数搭載されている環境では、どのビデオコントローラを利用するかを明示的に指定した方が無難です。

複数のビデオコントローラがある環境でも、1つしかディスプレイを接続していない場合は、Linux カーネル起動時に、アクティブなコントローラを自動認識してくれますが、特に複数のモニタを接続している場合には、どのビデオカードを優先するかを指定する必要があります。

例えば、オンボードで control コントローラ (対応フレームバッファドライバは controlfb) を搭載した PowerMac 7600/200 の PCI スロットに Rage Radeon カード (対応フレームバッファドライバは radeon) と Rage 128 カード (対応フレームバッファドライバは aty128fb) を増設し、全てにモニタを接続している環境で、radeon 側を優先させたい (Linux kernel 起動直後の起動コンソールにする) 場合は、以下の様にカーネル引数を指定します。

```
video=radeon video=aty128fb video=controlfb video=map:001122
```

これにより、起動コンソール (仮想コンソール 1 番) と仮想コンソール 2 番が radeon に、仮想コンソール 3 番と 4 番が aty128fb に、仮想コンソール 5 番と 6 番が controlfb に割り当てられます。