LINUX PPC 2000

LinuxPPC 2000 User's Guide

Last modified 15 February 2000

http://www.linuxppc.com/

Foreword

Congratulations on deciding to use LinuxPPC 2000! This is the latest version of the rapidly-developing and growing open source operating system for PowerPC computers. Many, many developers have engaged themselves in making the software in this distribution work well. We hope you are pleased with the results of their efforts!

This User Guide is intended to get you started using LinuxPPC 2000 as quickly and easily as possible. We will be offering updates to this guide online from time to time, as new information becomes available. Just keep an eye on the LinuxPPC website at http://www.linuxppc.com/ for more information and updates for your LinuxPPC distribution.

Once again, thanks for using LinuxPPC!

Credits

Information on this guide is based on excellent documentation for Linux-pmac installation by Paul Mackerras, Jeff Carr's original PowerPC Linux Install and User Guide, Nick Bastin's PBG3 FAQ, Craig Sadler's G3 FAQ, the FAQ-O-Matic, as well as other tidbits gathered around the web, various readmes and man files, mailing lists, newsgroups, and finally my own experiences. A big thanks to all the people who have e-mailed contributions and suggestions.

Guide design and layout, LinuxPPC logo and further contributions from John Grantham. For more information, visit http://surf.to/multimedia.

Additional contributions and LinuxPPC 2000-related updates from Jason Haas.

It has to be said…

Apple Computer, Mac OS, Power Macintosh, Power Computing, Umax, Motorola, PowerPC, BeOS, Red Hat, and all other trademarks and registered trademarks listed on this User's Guide are property of their respective owners.

Linux is a trademark of Linus Torvalds.

Contents

LinuxPPC 2000 User’s Guide

About LinuxPPC 2000

LinuxPPC is a version of the Linux kernel and operating system for the PowerPC processor. Linux is a free version of Unix that runs on a variety of computers, including Intel x86, DEC Alpha, SPARC, Motorola 680x0, MIPS, and ARM. The Linux kernel was written by Linus Torvalds, and much of the inital port to the PowerPC processor was performed by Gary Thomas, Paul Mackerras, and Cort Dougan. Subequently, hundreds of people have worked on Linux/PowerPC, and are too many to name here. We thank them all.

Linux is developed under the GNU General Public License (GPL), and its source code is freely available online and on the source CD-ROM that LinuxPPC sells with its LinuxPPC product.

LinuxPPC is a completely PowerPC-native operating system that can co-exist peacefully on the same hard drive as your original operating system, such as Mac OS, AIX, or BeOS. By placing LinuxPPC on separate partitions from your other operating system, it can function independently without interfering with any of your other data. You can also share files between the two systems by mounting the partitions within LinuxPPC. If you prefer, you can also install LinuxPPC on a separate hard drive or have it completely replace your original OS as your primary system.

A few features of LinuxPPC include:

- Stable and fast - protected memory, preemptive multitasking, advanced virtual memory
- SCSI and IDE drive support
- Runs on PCI-based PowerPC 601, 603/603e, 604/604e/604r, 750 (G3) and 7400 (G4) computers. NuBus-based PowerMacs like the 6100, 7100, and 8100 cannot use LinuxPPC
- Shared library support, glibc 2.1, Linux 2.2 kernel
- Floppy, sound, ethernet, and CD-ROM support
- Serial and USB input device support (can print to supported printers and use PPP through a modem).
- Netscape Communicator, MP3 players, and other popular internet/multimedia tools
- X Window window system, and several window managers AfterStep, WindowMaker, Enlightenment, and desktop environments such as KDE and GNOME
- Free, powerful web server (Apache), anonymous FTP, file server, multi-user support
- Java support (JDK 1.1.7, 1.2)
- Bootable CD-ROMs
- Easy dual boot (LinuxPPC/Mac OS capability)
- X-based installer or Red Hat style installer
- Graphical partitioning tool

In addition, people are working on many other projects including emulators of the Mac OS that run within Linux. These include a port of SheepShaver from BeOS and the Mac-on-Linux project. (Mac-on-Linux ships with LinuxPPC 2000.) The ApplixWare suite of office productivity tools (word processing, spreadsheet, database, graphics) is also available for LinuxPPC.

LinuxPPC runs on PCI-based PowerMacintosh computers and compatibles as well as PowerPC BeBox, PReP and CHRP machines. Machines that can run LinuxPPC include:

Apple Computer

iMac, iMac DV, G4, Blue & White G3s, G3 desktop and Powerbooks, 9600, 9500, 8600, 8500, 8200, 7600, 7500, 7300, 7200, 6500, 6400, 6360, 5500, 5400, 4400, 20th Anniversary Macintosh; Powerbook 2400, 3400, G3 (all), iBook

Be

BeBox

IBM

RS6000 (PowerPC-based), 830, 850, 40P, Nobis, INDI

Motorola

StarMax (and all StarMax clones from APS, PowerTools, Mactell), PowerStack, Series E, PowerStack II

Power Computing

PowerBase, PowerWave, PowerCenter, PowerCenter Pro, PowerTower, PowerTower Pro

Umax

C500, C600, J700, S900, Apus 2000 and 3000

What's new in LinuxPPC 2000

We're proud to release the latest version of the LinuxPPC operating system. LinuxPPC 2000 includes a raft of new features to make LinuxPPC even easier to use and more powerful. Here is an overview of what's new in LinuxPPC 2000:

- *Bootable CD-ROM.* Certain machines (listed in *Installation setup* and the *Appendices*) can boot directly from the LinuxPPC 2000 CD-ROM.
- *Mac OS support.* With the open source tool Mac-on-Linux, now included with LinuxPPC 2000, you can run Mac OS applications in a special environment under LinuxPPC 2000. Mac OS applications run native, not emulated.
- *New Perldisk partitioning utility.* We have created a new Perl-based utility to make partitioning your hard drive much easier than with using pdisk or other tools, while natively creating Linux partition types and file systems.
- *Added support for G4, Yosemite G3, and "trayless" iMacs.* Now you can run Linux on your candy-colored Macs!
- *New window manager – Sawmill with GNOME.* We have changed default window managers and included Sawmill with GNOME support. Sawmill is a new, fast and stable window manager that we think will impress you. You can still use Enlightenment, KDE, WindowMaker or others, which are included on the CD.

There are many more new features, but these are the most obvious. For a complete list of changes and updates, see the various changelogs for the software you're looking for.

Important information – before you install

We have added documentation for our new graphical partitioning utility and for setting up PowerMacs to boot directly into LinuxPPC. Please read these pages before installing LinuxPPC if you want to use these tools.

Booting into LinuxPPC 2000

It is now possible to set up a Mac to boot directly into Linux without first booting into the Mac OS. Please carefully read the documentation for this process as listed in the chapter “Booting directly into LinuxPPC 2000” before attempting to set up your computer to use it.

HFS and HFS Extended: Compatibility and Hard Disk Formatting

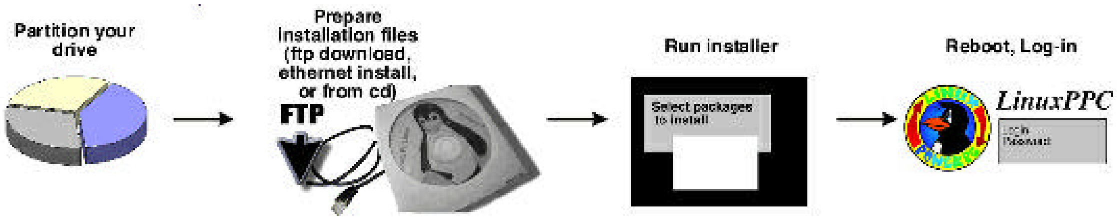
You do not need to reformat your entire hard disk to HFS (regular) to be able to use LinuxPPC. Please read the "HFS/HFS+ Compatibility Notes" in Appendix I for complete information.

Installation overview

This section is intended to give you a brief, at-a-glance look at installing LinuxPPC. It is focused on installing LinuxPPC 2000, but much of the information presented here can also be used when installing older versions of LinuxPPC, like LinuxPPC 1999 Q3.

Installation in a nutshell

This guide is focused on installation of LinuxPPC 2000. Resources that document installation not specific to the PowerPC are located at the Linux Documentation Project.


Partition your drive
Prepare Installation files (ftp download, ethernet install, or from cd)
FTP
Run installer
Select packages to install
Reboot, Log-in
LinuxPPC
Login
Password

LinuxPPC can be installed either using an X-based installer (which is called xinstaller) or a Red Hat Installer. The X-based installer is the recommended method of installing LinuxPPC 2000, while the Red Hat installer is the traditional method of installation used in previous releases.

The installer runs off of a Linux kernel. The kernel may be booted from a floppy, or for Mac users it can run off of a disk image on the hard drive when used with BootX. It reads the necessary installation files from one of four places:

- CD-ROM (available from https://order.linuxppc.com/)
- FTP via ethernet (with Ethernet adapters that the installer can recognize)
- NFS via ethernet (with Ethernet adapters that the installer can recognize)
- an existing local Mac OS HFS disk partition (HFS+ is not yet fully supported)

On PowerMacintoshes and clones, you have three different ways of booting:

- Directly from the CD-ROM.
- Via the BootX application or extension.
- Directly, via Open Firmware.

CD booting, BootX, and Open Firmware: A comparison

CD booting. The LinuxPPC 2000 CD-ROM is the first release that features the ability to boot a PowerMac directly from the LinuxPPC CD-ROM. CD booting is very simple: insert the LinuxPPC CD-ROM, restart the Macintosh, and hold down the C key immediately upon restarting. You can release the C key when you see a large penguin icon in the upper left hand corner of the screen.

When you boot the computer from CD, the computer will boot directly into LinuxPPC and load the LinuxPPC installer. You can use the installer to partition hard drives and install LinuxPPC. For details on partitioning, please see the chapter *Partitioning your hard drive*.

BootX is a Mac OS utility from Benjamin Herrenschmidt that can be used in the form of an extension or as an application. The BootX extension provides a simple graphical interface for you to choose between booting Mac OS or LinuxPPC at startup. It also allows the user to choose different boot-up partitions for LinuxPPC or to boot off of a ram disk (such as the one used for the installer). It provides the added capability, to use Mac OS initialized video if no video driver exists for your system (currently only ATI and ixMicro Twin Turbo video have built-in drivers). Usually, ATI video is built into Powermacs. The BootX application does the same thing as the extension, except it can be used to quit Mac OS and launch LinuxPPC from the Finder. Using BootX will be covered in more detail in the sections describing the installation and boot-up process.

Open Firmware is software built into all PowerMacintoshes since the 7200. If you are unable to use BootX, then you can use Open Firmware to boot into LinuxPPC. Open Firmware builds a device tree of the computer's hardware and allows control of which operating system to boot. In order to make Open Firmware boot LinuxPPC rather than Mac OS, you will have to modify Open Firmware's boot variables. Boot variables are stored in non-volatile ram, or nvram. These settings can be reset to their default values using the following method: at restart, hold down the Command-Option-P-R key combination until the computer resets itself twice (note that this will also clear your PRAM settings, such as settings in the memory control panel, volume, and video settings). Setting boot variables using the Boot Variables application will be covered in Appendices A and B.

Disk space requirements

For a default installation, you will need about 500 MB of drive space on its own partition(s) dedicated to LinuxPPC. These partitions may be on the same drive as another OS or on their own dedicated drive. If you want a bare minimum setup (no compilers, no X windows) you can get away with around 70 MB, but its usefulness will be limited. For LinuxPPC 2000, a minimum of 1.0 GB is recommended.

It is recommended that your drive be split into at least two partitions: a root partition and a swap partition. Root is where the Linux system files are kept, swap is used for virtual memory. The size of swap depends on the load you expect on your system. For most single-user systems, 50 MB should be fine. A general rule of thumb for multi-user systems, like servers, is that you should have twice as much swap as you do RAM: in other words, if you have 64 MB of RAM, you should have 128 MB of swap.

For LinuxPPC 2000, the recommended partitioning scheme is to split a 1 GB drive into 950 MB for root and 50 MB for swap. You can choose to create separate partitions for some of the larger directories that reside in the root directory, such as /usr, /home, and /var. Keeping these directories in separate partitions is a good idea to help protect your data in the rare case of file system corruption or other problems. If you have to reformat, you will only have to reformat that one partition. If you choose not to create separate partitions for these directories, all the files will be placed in directories in root.

If you will be partitioning a drive that has existing data on it (such as your startup disk), *back it up*. Editing the partition table of a drive usually results in the *loss of everything* in the partitions modified. There are some third party disk utilities that will allow you to resize a partition so that existing data is not erased, but use of these utilities will not be covered here.

If you plan on downloading the installation files onto a Mac OS drive, you'll need to partition your disk first. After partitioning, you can then reinstall Mac OS and download the installation files onto your Mac OS HFS partition (HFS+ is not yet supported). Alternatively, if you have two drives, you can retain the first drive for Mac OS and partition the second for LinuxPPC installation files. If you are installing from the CD-ROM, NFS, or FTP, you do not have to worry about copying the files onto a HFS partition.

More details on how to split your drive will be covered in the next chapter, *Partitioning the hard drive*.

Partitioning your hard drive

This is the first necessary step in preparing your PowerPC computer for running LinuxPPC 2000. Linux, like other variants of UNIX, uses partitions to control data allocation on the hard drive. This is to help prevent data loss through damage, possible instability in the software and so on.

Actually, Mac OS also uses partitions, though most people aren't aware of it. A typical Apple-formatted hard drive will have one large partition (usually HFS or sometimes HFS+ formatted) containing the user's data and the System Folder, but there are other hidden partitions as well – like the hard drive's driver software, for example. Linux simply makes more visible use of partitions than the Mac OS does.

Why partition?

LinuxPPC cannot be installed on HFS or HFS+ volumes. LinuxPPC uses a different type of disk partition called A/UX, also called Apple_UNIX_SVR2 partitions. A/UX was Apple Computer's first version of UNIX, designed for 680x0-based Macs. A/UX's disk partition type lives on as a part of LinuxPPC.

You should partition your drive as you see fit, but you will need a minimum of two partitions – root and swap – to install Linux. You can, if you want, have additional partitions, like /usr, /var or /home, which will contain those parts of the Linux system data and information. This protects the partitions' data from each other and can prevent data (like logfiles) from overfilling your computer.

Back up your files

Before you do any partitioning, remember to *backup all important files* on a separate hard drive, since partitioning a disk typically makes all old data on that disk inaccessible. If you have a brand new Apple machine that includes the "Apple Software Restore" application on the system CD-ROM, a backup won't be necessary if you haven't installed any additional files on your machine.

Partition types

Partitions come in several types: Apple Hierarchical File System (Apple_HFS, used by Mac OS), HFS+ (also type Apple_HFS, used by Mac OS 8.1 and above), and A/UX (Apple_UNIX_SVR2), among others. LinuxPPC requires A/UX partitions.

Partition sizes and setup: general notes

At a minimum, you will need two partitions – root and swap – but major directories within root can be separate partitions or part of the main root partition. (See "Where do I find stuff?" under *Getting started with LinuxPPC* for a basic view of the major directories in LinuxPPC.) If you are planning to use LinuxPPC as a multi-user server, you may also consider creating a separate partition for /home (where all user accounts are stored).

Swap must be a separate partition from root. The size of swap should be around 50 MB, or more if you expect your system to have a heavy load as may be the case if your system is going to be used for a server. 128 MB is the upper limit for individual swap partitions. If you require additional swap space, you can add separate swap partitions (note that all swap partitions should be named "swap").

LinuxPPC 2000 recommends at least 500 MB for a default installation, and 1 GB if you choose to install everything.

During partitioning, remember the order your partitions are in - Drive Setup uses the first few partitions of the drive for various drivers, patches, and partition maps, so your usable partitions might start numbering from partition 4, 5 or higher. Linux uses the following scheme for accessing partitions on an IDE drive: /dev/hdXY, where X is the device ID (a letter like "a," "b," "c," etc.) that the kernel assigns to the drive and Y is the number of the partition on that drive. For example, the first usable partition on an IDE drive might be /dev/hda7. For SCSI drives, the only difference is that partitions are accessed using the /dev/sdXY scheme. The internal SCSI drive is usually /dev/sdaY.

In the example here we will partition an IDE drive. Remember that you cannot partition the disk that you booted from - you will have to launch Drive Setup from the Mac OS CD, a Disk Tools floppy, or a separate drive. You can also download Drive Setup from Apple. Be aware that Drive Setup will format the partitions once they've been created, erasing all data, so backup your stuff. As mentioned earlier, if you have a brand new machine from Apple and haven't installed any new files, you can utilize the "Apple Software Restore" program found on the system CD-ROM to reinstall all the original files.

Partitioning utilities

In this section, we will take a look at two possible ways of partitioning your hard drive. The first method uses the tool “Drive Setup” that is included with Mac OS. The second uses the new Perldisk tool that is a new feature of LinuxPPC 2000.

There are other possibilities, too, like pdisk (an older text-based utility), Hard Disk Toolkit from FWB, and so on; however, we will concentrate on Drive Setup and Perldisk.

Drive Setup

The advantage to Drive Setup, for Mac users, is that it is familiar. However, older versions (before Mac OS 9) are not able to create A/UX partitions, meaning you’ll have to use either pdisk or Perldisk eventually. Another disadvantage is that you can’t resize existing partitions – you have to start from scratch each time.

For a beginner coming from Mac OS, it may be easiest to do the initial partitioning with Apple's Drive Setup program so that you get the sizes right and have the proper drivers for partitions used with Mac OS. Although older versions of Drive Setup cannot create A/UX partitions, you can create HFS partitions that can be easily converted to A/UX partitions with the LinuxPPC installer using the new Perldisk utilitiy, or the older tool pdisk (also referred to as fdisk). Users of other operating systems may go straight into using pdisk to partition the drive.

Note for Mac OS 9 users: Mac OS 9 includes a new version of Drive Setup that can create Linux partitions. If you have a copy of Mac OS 9, follow the same instructions below, but choose “LinuxPPC root”, “LinuxPPC swap”, etc. instead of “Unallocated” in Drive Setup and skip the steps with pdisk. Note that, in some cases, you may need to downgrade the driver later if you use the OS 9 version of Drive Setup. To do this, use an older version of Drive Setup and use it to install a new driver.

If you are using a version of Drive Setup older than Mac OS 9, you will have to use either pdisk or Perldisk to change the partitions to A/UX.

Drive Setup can normally be found in the Utilities folder. After launching Drive Setup, select the drive you want to partition, and click on the "Initialize..." button. Then click on the "Custom Setup..." button. This will bring up the custom partitioning window. (See examples on next page.)

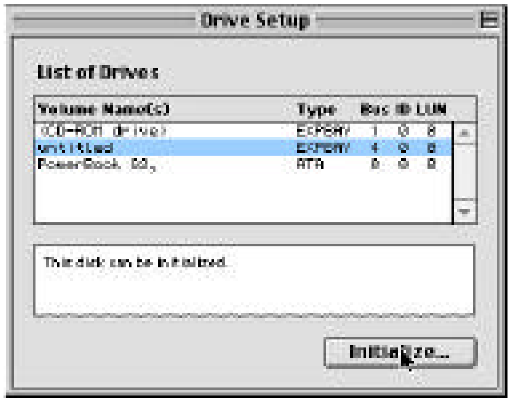
Mac OS 9 users: Remember that you can create A/UX (LinuxPPC) partitions using the version of Drive Setup provided with Mac OS 9. (See notes above.) You do not need to use pdisk or Perldisk to convert your partitions later.

The Drive Setup icon

Drive Setup

Left: Selecting the drive(s) to be parti - tioned.

Right: Select "Custom Setup..." to choose a special partitioning scheme.

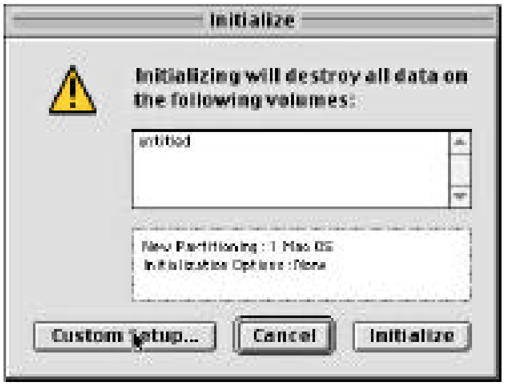


Drive Setup

List of Drives

Volume Name(s)	Type	Bus	ID	LUN
<CD-ROM drive>	EXPERY	1	0	0
untitled	EXPERY	4	0	0
PowerBook G3,	ATA	0	0	0

This disk can be initialized.



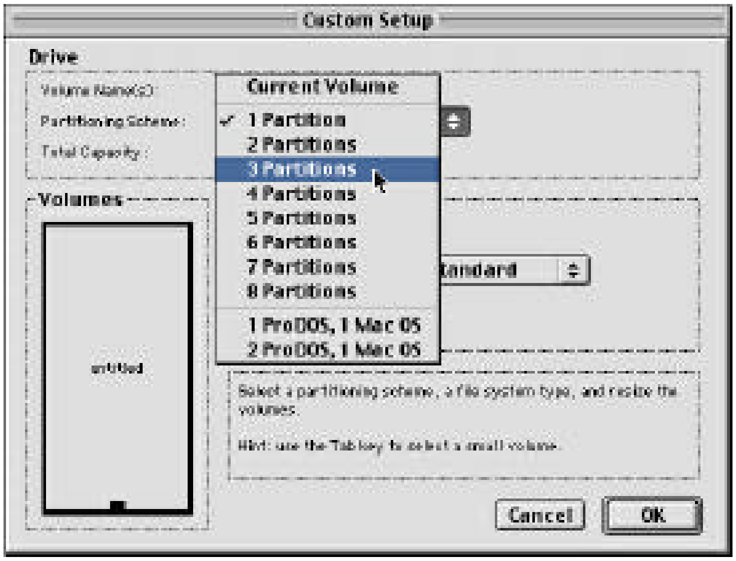
Initialize

Initializing will destroy all data on the following volumes:

untitled

New Partitioning : 1 Mac OS
Initialization Options : None

Choose the number of partitions. You will need at least two for LinuxPPC (one for root, one for swap), and one additional partition for HFS and the Mac OS (for PowerMac users).



Custom Setup

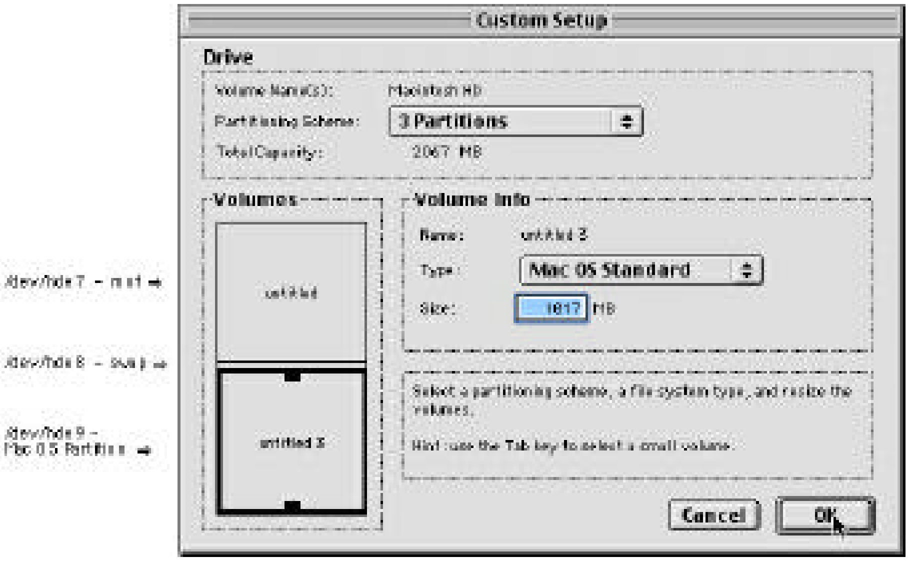
Drive

Volume Name(s):
Partitioning Scheme:
Total Capacity:

Select a partitioning scheme, a file system type, and resize the volumes.

Hint: use the Tab key to select a small volume.

Set the partition types. Mac OS 9 users can choose A/UX partitions for LinuxPPC here.



Custom Setup

Drive

Volume Name(s): Macintosh HD
Partitioning Scheme: 3 Partitions
Total Capacity: 2067 MB

Volume Info

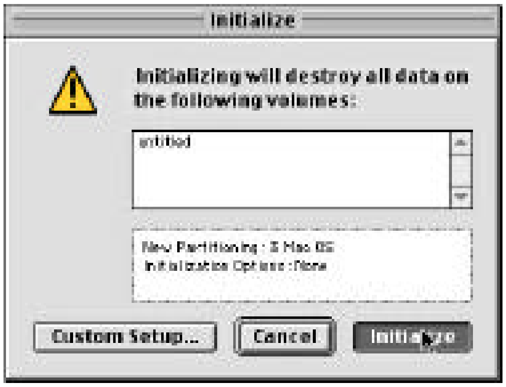
Name: untitled 3
Type: Mac OS Standard
Size: 1017 MB

Select a partitioning scheme, a file system type, and resize the volumes.

Hint: use the Tab key to select a small volume.

Partitioning the hard drive with Drive Setup

Here we selected the 3 partition scheme, and customized the size of each partition. You can utilize the tab key to select each partition and type in its size. The first shown partition is root, which is actually the seventh partition overall - Drive Setup keeps the partitions for the partition maps and drivers hidden. Thus, root is at /dev/hda7, swap is at /dev/hda8, and the partition used for Mac OS will be at /dev/hda9.

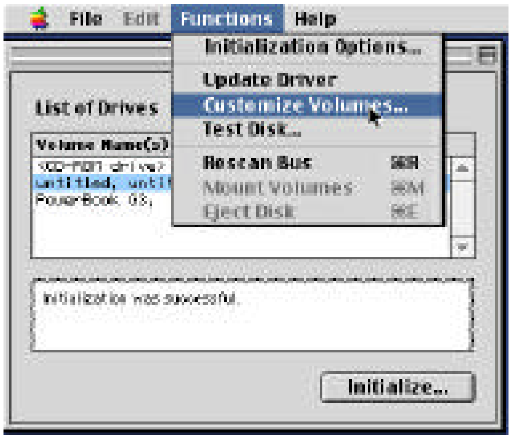


Initializing will destroy all data on the following volumes:

untitled

New Partitioning : 3 Mac OS
Initialization Options : None

Note: at this time, LinuxPPC does not recognize HFS+ drives. If you're creating partitions that will be converted to A/UX later, use HFS (Mac OS Standard). If you will be downloading the installation files to a hard drive partition, that partition must use standard HFS for the installer to find them. (Later you can have BootX and the kernel on an HFS+ partition, but you can't read or write on that partition from Linux.)

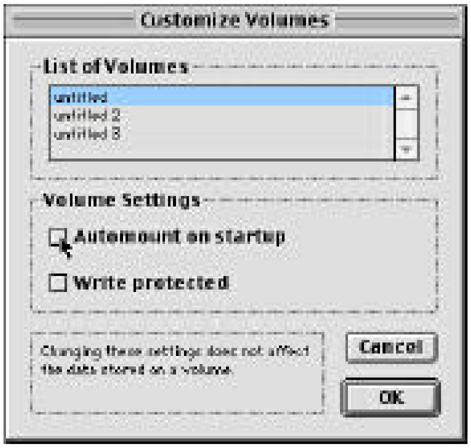


<CD-ROM drive>
untitled, unti
PowerBook G3,

Initialization was successful.

Don't forget to reserve space on your Mac partitions to download the installation files if you're installing from the hard drive (this is not necessary if you're installing from the LinuxPPC CD, NFS, or FTP). You will need around 500 MB of drive space for the installation files (and over 850 MB if you are installing everything).

When you're done, click on the "OK" button, then click the "Initialize" button.



untitled
untitled 2
untitled 3

□ Automount on startup

□ Write protected

Changing these settings does not affect the data stored on a volume.

After partitioning is complete, select the drive again from the list and go to the "Functions" menu and select "Customize Volumes...". This will bring up a window that allows you to select each individual partition. Select your Linux partitions and uncheck the "Automount on startup" option.

When you have disabled automount on each of your Linux partitions, click OK and quit Drive Setup. If you have problems using Drive Setup, please refer to Apple's documentation in the Help menu.

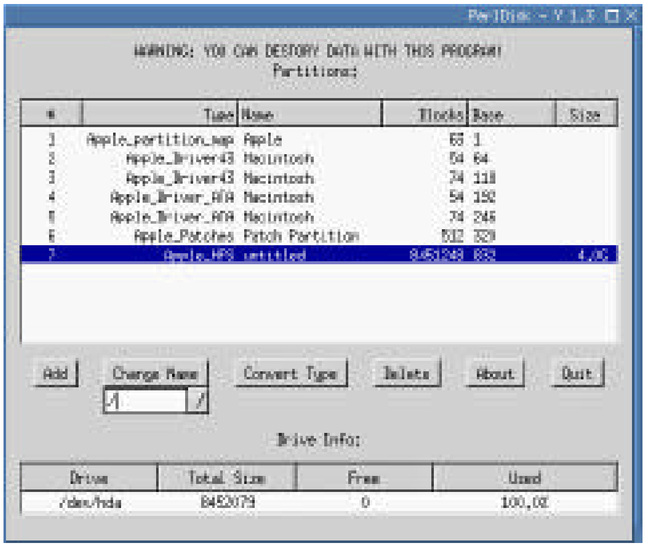
If you're planning on sharing the same drive with Mac OS, now is a good time to reinstall Mac OS on one of the partitions you're not using for LinuxPPC. You can also utilize the "Apple Software Restore" application that came on the CDs of recent PowerMacintoshes to restore Mac OS to its original state. In this example, I reinstalled Mac OS from backup onto the partition I created for Mac OS (partition #10). The Mac OS partition will also be used for the LinuxPPC installation files (again, this won't apply for those doing CD, NFS, or FTP installs).

Perldisk

Perldisk is a new disk partitioning utility that is included with LinuxPPC 2000. It was written by Brian Bresen of LinuxPPC Inc. Perldisk is an entirely graphical partitioning program that runs under LinuxPPC.

To start Perldisk from the LinuxPPC installer, simply click the Partition Drives button. Perldisk will start.

The main perldisk window is divided into three areas:



WARNING: YOU CAN DESTORY DATA WITH THIS PROGRAM!
Partitions:

#	Type	Name	Blocks	Base	Size
1	Apple_partition_map	Apple	63	1	
2	Apple_Driver43	Macintosh	54	64	
3	Apple_Driver43	Macintosh	74	118	
4	Apple_Driver_ATA	Macintosh	54	192	
5	Apple_Driver_ATA	Macintosh	74	246	
6	Apple_Patches	Patch Partition	512	320	
7	Apple_HFS	untitled	8451248	832	4.0G

Drive Info:

Drive	Total Size	Free	Used
/dev/hda	8452079	0	100.0%

Partition list

Command buttons

Drive info

- A list of partitions on the currently selected disk
- Command buttons
- Drive info

Take a look at the screenshot above to see an example startup window for Perldisk.

Partition List

The large partition list displays all of the partitions on the currently selected disk.

Command Buttons

The command buttons provide Perldisk's commands.

- *Add:* Create new partitions. You must have a free space partition selected to add new partitions. (See illustration on next page.)
- *Change Name:* The Change Name command is controlled by a button and pop-up menu containing the four commonly used partition names. To select a name, click on the pop-up menu, and choose the name that you want to use. To cause the pt's name to be changed, you must then click the Change Name button.

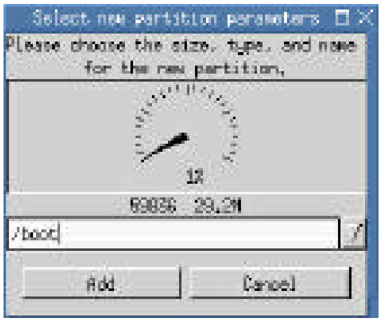
If you want to use a name that is not provided by the pop-up menu, you can type it in the name field, and click the Change Name button to commit the change. For example, you could type /opt in the name field, and click the Change Name button to rename a partition /opt.

- *Convert Type:* Toggles the partition type between Apple_HFS, Apple_UNIX_SVR2, and Linux_PPC. (Apple_UNIX_SVR2 and Linux_PPC are the same formats.) LinuxPPC Inc. recommends using the Linux_PPC type for installing LinuxPPC.

• *Delete:* Deletes the selected partition, turning it into free space. Once a partition has been turned in to free space, you can create new partitions with it. Be VERY careful with this command!

• *About:* Displays information about Perldisk, including legal liability information.

• *Quit:* Quits Perldisk. May prompt you to choose between Destructive Save and Safely Exit if changes have been made. (See Quitting Perldisk.)



Select new partition parameters

Please choose the size, type, and name for the new partition.

59836 29.2M

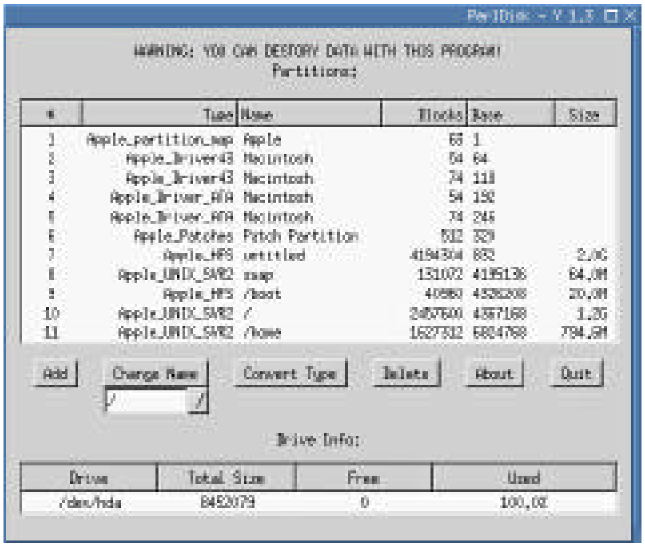
/boot

Add partition window

Drive Info

This panel displays all volumes attached to the computer, except for CD-ROM drives and other read-only media. Floppy drives are also not displayed.

If you have more than one hard disk in your computer, you can toggle between the different volumes by clicking on the drive names.



Perldisk - V 1.3

WARNING: YOU CAN DESTORY DATA WITH THIS PROGRAM!

Partitions:

#	Type	Name	Blocks	Base	Size
1	Apple_partition_map	Apple	63	1	
2	Apple_Driver43	Macintosh	54	64	
3	Apple_Driver43	Macintosh	74	118	
4	Apple_Driver_ATA	Macintosh	54	192	
5	Apple_Driver_ATA	Macintosh	74	246	
6	Apple_Patches	Patch Partition	512	320	
7	Apple_HFS	untitled	4194304	832	2.0G
8	Apple_UNIX_SVR2	swap	131072	4195136	64.0M
9	Apple_HFS	/boot	40960	4326208	20.0M
10	Apple_UNIX_SVR2	/	2457600	4367168	1.2G
11	Apple_UNIX_SVR2	/home	1627312	6824768	794.6M

/

Drive Info:

Drive	Total Size	Free	Used
/dev/hda	8452079	0	100.0%

Final window: finished configuration

Selecting Volumes

You can select the volume to modify by clicking the drive name in the Drive Info field. The drive name is displayed as the drive's hardware name, such as /dev/hda for IDE drives, and /dev/sda for SCSI drives. If you have multiple drives, they will be listed in this field in alphabetic order. IDE drives are listed before SCSI drives.

Modifying volumes

When you make a change to a disk partition, the changes are not immediately saved. No changes to the disk will be changed until you quit Perldisk with the "Destructive Save" option. If you choose to Safely Exit, no changes will be saved.

Deleting partitions

Deleting a partition in Perldisk is simple. Click on the partition you wish to delete, and click the Delete button. The partition will be converted into free space.

You should only delete partitions of the types Apple_HFS, Apple_UNIX_SVR2, and Linux_PPC. Do not delete any other types of partitions, especially Apple_Patches, Apple_Driver, or Apple_partition_map.

Warning: If you delete every Apple_HFS partition on a hard disk, it will become impossible to boot into the Mac OS. If you wish to install only LinuxPPC on a computer, then this will not be a problem. However, if you wish to have both the Mac OS and LinuxPPC installed, you will need at least one Apple_HFS partition to install the Mac OS on to. That partition can be reformatted as HFS Extended later on by the Mac OS.

Adding new partitions

Note: To create new disk partitions, you must have a free space partition on the disk. If you do not have any free space partitions (listed as Apple_Free Extra), you cannot create new partitions. There currently are no utilities that can resize Apple HFS Extended partitions without first erasing the disk.

To create a partition from free space:

- Click on the free space (Apple_Free Extra) partiton.
- Click the Add button. The partition sizer window will appear. Using the dial, you can change the partition's size. The default setting is 100% of the free space partition. (Future versions of Perldisk will allow you to manually type the amount of space.)
- Type in the name of the partition, or select an existing option from the pop-up menu. /, /boot, /home and swap are pre-defined partition names.
- Click the Add button to commit the changes, or Cancel to abort. Remember, no changes will actually be made until you exit Perldisk with the Destructive Save option.

Changing partition types

You can change a partition to Apple_HFS, Apple_UNIX_SVR2, or Linux_PPC by clicking the Convert Type button in the main Perldisk window. Click the button until the partition is changed to the desired type. Note that the partition will need to be formatted by an appropriate utility before it can be used. If you are installing LinuxPPC for the first time, the installer can format the partition for you. If you create an Apple_HFS partition, use the Mac OS to format the partition.

The new partition will be created as a Linux_PPC partition. If you want to change its name or type, you can do so now.

Changing partition names

To change the name of a partition, click on it. Then select a name from the pop-up menu, or manually type the name of the partition.

Quitting Perldisk

Quitting without saving changes: If you have made changes to a disk in Perldisk but do not wish to save your changes, click the Quit button, and then click the Safely Exit button. Perldisk will quit without modifying the drive map, leaving your disk untouched.

Quitting and saving changes to the drive: To quit Perldisk and save the changes to the drive, click the Quit button, and then click the Destructive Save button. You should then reboot the computer to ensure the changes are saved.

Using pdisk

Pdisk is a text-based partitioning program that runs within Linux. If you prefer, you can use it instead of Perldisk and Drive Setup; if you created your partitions with Drive Setup using a version older than Mac OS 9, you will need to use pdisk to convert those partitions to A/UX.

There is also a Mac OS port of pdisk that runs as a command-line-interface application available at ftp://ftp.linuxppc.org/pub/linuxppc/linuxppc-R4/RedHat/tools/. If you have problems using the installer's version of pdisk, you can try using the Mac OS version.

If you feel comfortable with partitioning, you can skip using Drive Setup and use pdisk to directly create A/UX partitions, but remember that pdisk does not write Mac OS-specific drivers for HFS partitions it creates.

Here is a basic list of pdisk commands:

Command	*Function*
?	help
p	prints the partition table
i	initialize the partition map (erases the old map)
c	create a new Apple_UNIX_SVR2 partition
C	create a new user-defined partition type (Apple_UNIX_SVR2 or Apple_HFS)
d	delete a partition (turns it into "Extra" space)
w	write the new partition table (permanently saves changes)
q	quit (does not automatically save changes)

To change a partition type, you must first delete the partition (thus creating free space) by typing “d [#]”, where [#] is the number of the partition to be deleted. (Note the starting point – also called “base” – of the partition and its length before continuing.) To find the partition number, type in “L” and hit Enter.

Next, you must create a new partition by typing “c” (lower case). Enter the first block (the “base” number) for the extra space that you just created, then enter the length in blocks. Now enter a name for the partition. This should be in lower case and match the directory type you want to reside there – root should be written with a single slash /, otherwise “swap”, “home”, “usr” and so on (without the quote marks).

We suggest that you check your partition table on display often, so you know you didn't make a mistake. If you did make a mistake, you can simply delete the partitions that you've made and create new ones. Changes are not permanent until you write the partition. Remember to write your partition table when you're done, otherwise, nothing happens.

When you’re finished, type “w” and hit Enter to write the partition table, then type “y” to confirm. You can now quit the program by typing “q” and hitting Enter.

For a more detailed description of using pdisk and fdisk, see Appendix D.

Additional notes about the Installer and partitioning

Installer handling of HFS partitions You may wish to have the computer automatically mount HFS partitions during Linux's startup process. This is possible by adding a "mount point" in the installer during the Mount and Format Partitions phase of installation.

However, the installer does not cor rectly write the partition type (HFS) in the /etc/fstab file. It writes it as type ext2. You will need to manually change the partition type in the /etc/fstab file.

If you want to edit the file and are not familiar with Linux/UNIX text editors, you may want to use the pico editor, a simple command-line text editor, or the gedit editor, a GUI-based text editor that is installed with the default LinuxPPC installation. You need to be logged in as root to edit the file.

If you have Perl experience and wish to help find a solution, please examine the /usr/X11/R6/bin/xinstaller.pl program. Suggestions/fixes can be sent to bresen@linuxppc.com.

Installation setup

Where to get the installation files

The files you need for installation can be broken down into two categories:

- Installation files required for all operating systems
- Specific files for your system

The easiest way to install is to get the LinuxPPC CD-ROM, which includes all the files you need. For Mac users, the bootable CD-ROM takes advantage of BootX and LinuxPPC Live to bring you to an X Windows-based installer at the click of an icon. If you have the LinuxPPC 2000 CD-ROM, you can skip this chapter and go on to the next chapter on installation. If you choose to download the files, they must be downloaded in binary (raw) format. Remember that it must be a HFS drive, since HFS+ drives are not yet supported. The complete download is nearly 500 MB.

LinuxPPC 2000 CD-ROM

The LinuxPPC CD-ROM can be ordered securely at https://order.linuxppc.com/. Purchasing the CD-ROM helps support the LinuxPPC Project. LinuxPPC, Inc. uses portions of the proceeds from CD orders to support LinuxPPC developers and to provide systems to commercial developers so that they can provide ports to LinuxPPC. Funds are also used to support the LinuxPPC.org web site and ftp sites.

If you have the CD already, congratulations! Then you can skip much of the next section.

Download sites

The main FTP site for LinuxPPC 2000 is at ftp://ftp.linuxppc.com/. The following is a list of some mirror sites for LinuxPPC – please use a site that is geographically close to you. The most up-to-date list of mirrors is located at http://www.linuxppc.org/getlinux/mirrors/.

North American Mirrors:
CA - ftp://ftp.cdrom.com/pub/linux/linuxppc/
IL - ftp://uiarchive.uiuc.edu/pub/systems/linux/linuxppc/
MA - ftp://rufus.w3.org/linux/linuxPPC/
MI - ftp://ftp.eecs.umich.edu/pub/linux/linuxppc
MO - ftp://wuarchive.wustl.edu/systems/linux/linuxppc/
NC - ftp://sunsite.unc.edu/pub/Linux/distributions/linux-ppc/
TN - ftp://sunsite.utk.edu/pub/linux/LinuxPPC/
WI - ftp://mirror.doit.wisc.edu/pub/mirrors/linux/distributions/linuxppc/
WI - ftp://dev.linuxppc.org/pub/linuxppc/
CAN - ftp://ftp.linuxberg.com/pub/distributions/

European Mirrors:
AT - ftp://gd.tuwien.ac.at/opsys/linux/linuxppc/
CH - ftp://sunsite.cnlab-switch.ch/mirror/linuxppc/
DE - ftp://ftp.apfel.de/pub/LinuxPPC/
DE - ftp://ftp.uni-bremen.de/pub/linux/dist/linuxppc/
ES - ftp://ftp.fi.upm.es/pub/mirrors/linux/linuxppc/
FI - ftp://ftp.funet.fi/pub/mirrors/ftp.linuxppc.org/
FR - ftp://ftp.lip6.fr/pub/linux/linuxppc/

NO - ftp://mac.pvv.ntnu.no/pub/linux/linuxppc/
SE - ftp://ftp.karen.hik.se/pub/linux/linuxppc/
UK - ftp://sunsite.doc.ic.ac.uk/Mirrors/ftp.linuxppc.org/pub/linuxppc/

Australian Mirrors:
- ftp://ftp.au.linuxppc.org/pub/linux/linuxppc/
- ftp://mirror.dstc.edu.au/pub/linuxppc/
- ftp://the.ausmac.net/pub/mac/Linux/LinuxPPC/
- http://the.ausmac.net/ftp/Linux/LinuxPPC/

Japanese Mirrors:
- ftp://mirror.nucba.ac.jp/mirror/linuxppc/
- http://mirror.nucba.ac.jp/mirror/linuxppc/
- ftp://ftp.ring.gr.jp/pub/linux/linuxpp/
- http://www.ring.gr.jp/pub/linux/linuxpp/
- ftp://sunsite.sut.ac.jp/pub/archives/linux/linuxppc/
- http://sunsite.sut.ac.jp/pub/archives/linux/linuxppc/
- ftp://ftp.ccex.miyazaki-u.ac.jp/pub/ftp.linuxppc.org/
- ftp://ppc.linux.or.jp/pub/mirrors/LinuxPPC/

A note about FTP with the Mac OS

You can use your favorite FTP program to download the files, but for Mac OS users we recommend using Fetch or Transmit to download binaries. Netscape and Internet Explorer should be able to download the necessary files as well, but be sure that they aren't downloaded in text format. Netscape and IE users should configure their browsers so that file types with extensions rpm, coff, tgz, and gz are downloaded in binary. In Netscape, this is in the Preferences > Navigator > Applications setting. In IE, this is in the Preferences > Receiving Files > File Helpers setting. You can also set this in the Internet control panel on Mac OS 8.5 and later (select User Mode “Advanced”, then click the Advanced tab > File Mappings).

You can get Fetch from the Info-Mac hyperarchive:

http://hyperarchive.lcs.mit.edu/HyperArchive/Archive/comm/inet/fetch-303.hqx

Necessary files for all systems

You only need to follow the steps in this section if you will be downloading all the installation files onto your hard drive for installation. If you are performing the installation directly through FTP or NFS in the installer, you can go on the next section on "Specific Files for your System." Again, those performing an installation via the LinuxPPC CD-ROM can skip on to the next chapter on using the installer.

Remember that to download the installation files you'll need around 500 MB of free drive space. The bulk of the LinuxPPC installation is in RPM format, approximately 420 MB if you download everything. RPM stands for Red Hat Package Manager, and is the format used for installation of most LinuxPPC programs. RPM packages for LinuxPPC are designated by the .ppc.rpm extension (these can also be used by MkLinux).

To set up, you will need a folder/directory called "software" in the root level of your hard drive. For Mac users, double-click on the icon of your hard drive. Create a new folder in the hard drive window, and rename it software.

Next, create a new folder/directory called "RedHat" in the "software" folder, and then create folders called "base", "cksums", "instimage", "maps", and "RPMS" inside the "RedHat" folder. If you use Mac OS, you should end up with something like the image to the right.

Next, go to one of the mirrors above and into the linuxppc-2000/RedHat directory. I will use ftp.cdrom.com and the /pub/linux/linuxppc/linuxppc-2000/RedHat/ directory as an example.

First you will need to copy the debug.log, ppc, and redhat.image.gz files into the "RedHat" folder on your hard drive. If you're using Fetch, the easiest thing to do would be to command-click the individual files to select them and drag them all onto the RedHat folder on your drive.

For the remaining files in each of the base, cksums, instimage, maps, and RPMS directories, it's safest to go into each directory individually, select all, and copy them all into the individual folders of your hard drive. We do this rather than copy the directory itself because the files in each directory are often symbolic links to files located elsewhere on the mirror and may not get resolved properly by your FTP client. For example, go into the base directory, select all files, and copy them into the base folder on your hard drive (in Fetch, you can use command-A to select all inside the base directory and drag-and-drop onto the base folder). Repeat this process for the remaining directories in the RedHat folder. Remember the RPMS folder contains about 420 MB worth of rpm's. I will be updating this guide in the future to show you how to customize the installation so you only download the rpms you need.

Workaround for limitations of the Mac OS 31-character filename limit: Some rpm files that you download may contain more than 31 characters in their filenames. As a result, your FTP client will shorten these filenames so that they fit under Mac OS's 31-character filename limitation. Different FTP clients will shorten the filenames in different ways - Fetch simply cuts off the last few characters of the filename, whereas Anarchie retains the last four characters (.rpm), cuts off some of the middle characters and replaces them with an ellipses (...). As a result, the X-based installer may not properly find these files. The workaround is to download a modified hdlistinfo file and replace the one located inside the software > RedHat > base folder. You can download a modified hdlistinfo for Fetch at

http://www.linuxppc.org/userguide/hdlistinfo-fetch.gz

which you should ungzip, rename to hdlistinfo, and replace the existing hdlistinfo located in software > RedHat > base folder. A similar hdlistinfo is available at

http://www.linuxppc.org.userguide/hdlistinfo-anarchie.gz

for users that downloaded the RPMS with Anarchie. Special thanks to LinuxPPC users Drew Thoeni and Fabian Ille for providing these files. Note that the Red Hat installer does not have this problem with Mac OS-shortened filenames.

Finally, you need to create a folder/directory called "install" in the root level of your hard drive. Then copy the "maps" folder into it. For Mac users, the easiest way to do this is to hold down the option key, click-and-hold onto "maps," and drag it onto the "install" icon. You should end up with the files build.master, install.master, order.master, and sort.master inside.

Specific files for your system

Specific files for each system can be found the linuxppc-2000/install directory in any of the mirrors above. See the section below that pertains to your system.

Mac OS systems Once you get to the install directory, select the linuxppc.live.gz and LinuxPPC_2000_installer.bin files as well as the Mac OS.utils directory and drag it into your hard drive (on some servers, linuxppc.live.gz is named live.filesystem.gz instead).

The linuxppc.live.gz file contains the LinuxPPC Live file system, which provides the X Windows-based installer. It is the recommended method of installation for LinuxPPC 2000, although you can use a Red Hat-style installer instead if you'd like to save yourself the download. The boot-up scripts will revert to the Red Hat installer if a live file system cannot be found. The CD-ROM will use the X-based installer be default. You can also enter "redhat" in the kernel arguments when you use BootX or Open Firmware to launch the installer (more about this later).

Next, open the Mac OS.utils folder and drag-and-drop the linuxppc.live.gz file onto MacGzip - this will unzip the live file system. After unzipping is complete, rename the linuxppc.live file to live.filesystem. Remember that it must be on a HFS drive.

Next, double click the LinuxPPC 2000 Installer icon.

This will install ramdisk.image.gz (a ram disk image that the installer first uses) into the System Folder, a Linux Kernels folder into the System Folder, the BootX extension in the Extensions folder, BootX Settings in the Preferences folder, and the BootX application (labeled as "Boot LinuxPPC") onto your desktop. Inside the Linux Kernels folder is a file called LinuxPPC Standard - this is the LinuxPPC kernel, also called vmlinux. It's the core of the operating system, analogous to the System file in Mac OS. More pre-compiled kernels can be found in the linuxppc-2000/kernels directory in the LinuxPPC mirrors. If you have problems booting, you might want to try one of the kernels located there (there are special kernels for the Blue & White G3's, Apple Network Servers, and machines with Adaptic 2940 UltraWide SCSI cards there). You can download these kernels and place them inside the Linux Kernels folder (they must be ungzipped first).

After running the Installer, you will be given the option to reboot. Go ahead and do so - after the Mac OS splash screen, you'll get the BootX screen, which will already be configured properly. Go on to the next chapter.

BeBox, CHRP, and PReP System Setup Currently the main part of this guide has little specific information for installing LinuxPPC on BeBox, CHRP, or PReP systems. We hope to have more specific information in future versions of the User's Guide. Take a look at Appendices F, G and H for more specific information.

You can find the system-specific files for these systems in the linuxppc-2000/install/PREP-CHRP-BeBox/ directory.

- BeBox - one of the best sources of information for LinuxPPC on the BeBox is at a website by William R. Sowerbutts:
http://www.guru.dircon.co.uk/belinux/status.html

- CHRP - CHRP system users can refer to a web site by Geert Uytterhoeven:
http://www.cs.kuleuven.ac.be/~geert/PowerPC/Install/

- PReP - Kazunori Aoshima has created a web site for installation onto PReP machines at:
http://ppc.linux.or.jp/~aoshimak/install.html

Now that you have all the files you need, you can start with installation.

Directly booting into LinuxPPC 2000

Using the bootable CD-ROM

Bootable Machines Virtually every Apple PowerMac and PowerMac clone is now bootable from the LinuxPPC CD-ROM. NuBus-based PowerMac 6100, 7100, 8100, their clones and derivative Performa and Workgroup Server models cannot boot with this CD. Only machines on the supported hardware list can boot from the CD. Generally, if the computer is able to boot LinuxPPC, it will be able to boot from the CD. This includes everything from the PowerMac 4400 and higher, all G3, G4, and iMacs. iBook, PowerBook 2400, 3400 and PowerBook G3 (all models) are bootable. Most Performas except the 6360 and 6400 cannot boot from the CD or use LinuxPPC at all.

Booting from the CD On supported computers, insert the LinuxPPC 2000 CD-ROM, and restart the computer. Immediately hold down the C key. The computer will detect the CD-ROM and begin booting from it. You can let go of the C key when you see the Penguin logo appear on your screen. You can also use the Startup Disk control panel. Some models may automatically boot from the CD if it is present in the CD-ROM drive at system startup.

The computer will boot into the LinuxPPC Installer. On slower machines, this may take up to two minutes. On fast G3 and G4 systems, it can take as little as a minute.

Setting up a machine to boot directly into Linux: New World ROM-based PowerMacs

New World ROM-based PowerMacs are the iMac, Blue & White PowerMac G3, PowerMac G4, iBook, and bronze-keyboard PowerBook G3 ("Lombard"). If you have an earlier machine, please skip to “Booting with Old World ROMs”.

To set up your computer to boot directly into Linux, you will need to create an HFS partition during the Linux installation process. The LinuxPPC installer provides the tools to do this.

1. Have a hard disk or high-capacity removable disk ready to be installed on. (This means that you are able to completely erase it without incurring any data loss.)

2. Boot up from the LinuxPPC CD.

3. Click on the Partition Disks button.

4. Using the Perldisk utility (see *Partitioning your hard disk*), set up the following partitions:

Name	*Size*	*Partition Type*
/	500+ MB	Apple_UNIX_SVR2
/boot	20 MB	Apple_HFS
swap	16 - 128 MB	Apple_UNIX_SVR2

Important: The root (/) partition must be named / (a forward slash character) or /root, and be of the type Apple_UNIX_SVR2, and the /boot partition must be an Apple_HFS partition for this system to work.

5. After you have finished partitioning, save your changes by clicking Quit, and Destructive Save. If you made some mistakes, and don't want to save your changes, click Safely Exit. You can run Perldisk as many times as you want and not loose any data so long as you click the Safely Exit button.

6. Set up the partitions and install LinuxPPC, as described in "Installing LinuxPPC" later in this manual. During the installation process, the installer will place a System Folder on the /boot partition. This System Folder is not a copy of the Mac OS, but rather a "fake" System Folder that tricks the Mac into booting into LinuxPPC.

7. Once the installation is complete, reboot into the Mac OS from a Mac OS CD-ROM, or a different hard drive with the Mac OS installed on it.

8. Edit the yaboot.conf file (Boot LinuxPPC/System Folder/yaboot.conf) with a text editor. You will need to change a few lines in it to make it boot from the /boot partition.

The first lines are:

```
init-message = "\nWelcome to LinuxPPC!\nHit <TAB> for boot options.\n\n"
timeout = 50
default = install

image = cd:,vmlinux.2.2.15pre3
label = linuxppc
novideo
```

Change the line:

```
default = install
```

to:

```
default = linuxppc
```

Next, change the line

```
image = cd:,vmlinux.2.2.15pre3
```

to:

```
image = hdX:,vmlinux.2.2.15pre3
```

where X is the partition number of the Linux root partition. If the root partition is the eighth partition on the hard disk, you would type hd8. A finished file would look like this:

```
init-message = "\nWelcome to LinuxPPC!\nHit <TAB> for boot options.\n\n"
timeout = 50
default = linuxppc

image = hd8:,vmlinux.2.2.15pre3
label = linuxppc
novideo
```

9. Specifying boot parameters in yaboot

To specify special boot parameters such as ATI Rage 128 frame buffer drivers or single user mode, you can add extra kernel arguments to the yaboot.conf file with the append command. For instance, to specify single user mode (used to do maintenance or repair on a damaged system), you would add this line to the block used for booting:

```
image = hd8:,vmlinux.2.2.15pre3
label = linuxppc
novideo
append = " single"            # This is the new line.
```

You can have lines like

```
append = " adb_buttons=111,119
video=aty128fb:vmode:17,cmode:24,mclk:63"
```

in the yaboot.conf file.

Please check http://www.linuxppc.com/support/updates/product/ for updates on this subject. (Special thanks to Martin Costabel for his insights on yabooting.)

Setting up a machine to boot directly into Linux: Old World ROM-based PowerMacs

The Old World ROM machines are the PowerMacs made prior to the iMac and Blue & White PowerMac G3 computers. These basically are the "beige" PowerMac models, up to and including the original desktop and minitower PowerMac G3 computers.

To set up your computer to boot directly into Linux, you will need to create an HFS partition during the Linux installation process. The LinuxPPC installer provides the tools to do this.

1. Have a hard disk or high-capacity removable disk ready to be installed on. (This means that you are able to completely erase it without incurring any data loss.)

2. Boot up from the LinuxPPC CD.

3. Click on the Partition Disks button.

4. Using the Perldisk utility, set up the following partitions:

Name	*Size*	*Partition Type*
/	500+ MB	Apple_UNIX_SVR2
/boot	20 MB	Apple_HFS
swap	16 - 128 MB	Apple_UNIX_SVR2

Important: The root (/) partition must be named / (a forward slash character) or /root, and be of the type Apple_UNIX_SVR2, and the /boot partition must be an Apple_HFS partition for this system to work.

5. After you have finished partitioning, save your changes by clicking Quit, and Destructive Save. If you made some mistakes, and don't want to save your changes, click Safely Exit. You

can run Perldisk as many times as you want and not loose any data so long as you click the Safely Exit button.

6. Set up the partitions and install LinuxPPC, as described in "Installing LinuxPPC" later in this manual. During the installation process, the installer will place a System Folder on the /boot partition. This System Folder is not a copy of the Mac OS, but rather a "fake" System Folder that tricks the Mac into booting into LinuxPPC.

Once the installation is complete, reboot into the Mac OS from a Mac OS CD-ROM, or a different hard drive with the Mac OS installed on it.

7. Reinstall the Mac OS, or if you don't want the Mac OS on your machine, boot up from a Mac OS CD-ROM. You need to "bless" the fake System Folder on the Boot LinuxPPC drive for the direct booting to work. The fake System Folder needs to be touched once by the Mac OS, as outlined in step 8.

8. When you reboot into the Mac OS, open the Boot LinuxPPC volume that appears on your desktop. Then open the System Folder on the partition. Opening it will cause the Mac OS to "bless" it as a real System Folder.

9. Open the Startup Disk control panel, and select the Boot LinuxPPC partition as the startup disk. Reboot, and your machine will reboot into LinuxPPC!

Rebooting into the Mac OS

There aren't any simple ways to automatically choose between booting into the Mac OS when an Old World machine is set up to boot directly into Linux. Here are three ways to do it:

1. Move the fake "System" file from the fake System Folder on /boot.

```
mv /boot/System\ Folder/System ../
```

2. Zap the PRAM by holding down Command-Option-P-R when turning on the machine or restarting.

3. Using perldisk, change the partition type of the /boot partition to a type other than Apple_HFS.

Please check http://www.linuxppc.com/support/updates/product/ for updates on this subject as they become available.

Alternatives to booting directly into LinuxPPC 2000

You can still use the BootX software (provided on this CD) with LinuxPPC 2000. For new users, this may be preferable to trying to get the yaboot system to work. For instructions on installing and using BootX, please read the user guide PDF file that is included on this CD.

Using yaboot will become easier with time and development. Please check http://www.linuxppc.com/support/updates/product/ for updates on this subject.

Notes

The installer should boot from the CD in about two minutes (less on very fast machines). Machines with 32 MB of RAM or less may experience slow booting or operation in the installer. If you are an experienced user, you can help speed up the installer by manually creating and mounting a swap partition.

The /etc/fstab file is editable while the computer is booted from the CD's RAM disk, so you will be able to add it to the fstab even though the computer is booted from CD.

For more information and updates to the boot software, please see our product updates section at http://www.linuxppc.com/updates/product/.

Installing LinuxPPC

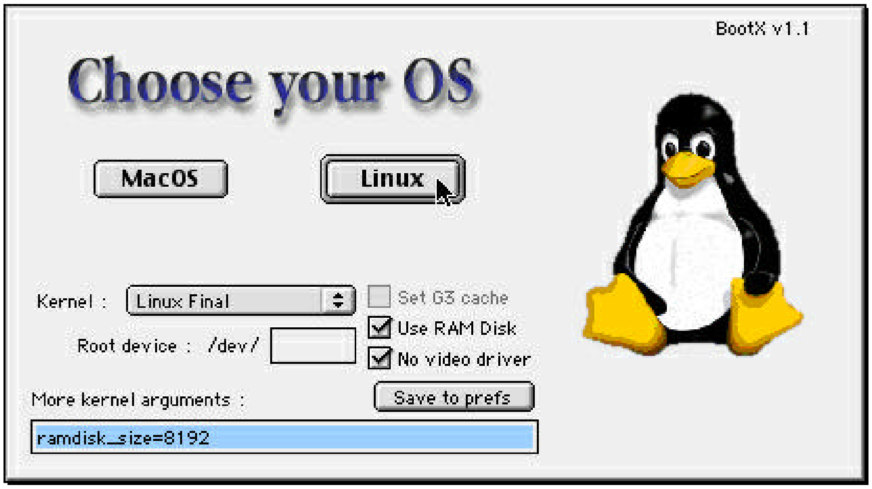
If you are installing from CD-ROM, insert the LinuxPPC CD into the drive. You should run the LinuxPPC 2000 Installer, which will place the required files onto your hard drive (refer to the previous chapter to see what files are installed). If you have an older CD-ROM without the LinuxPPC 2000 Installer, you can download it from the linuxppc-2000/install directory of any of the mirrors listed in the previous chapter. Otherwise, you can accomplish the same thing as the installer by doing the following: place the BootX settings from the CD into your Preferences folder, copy the Linux Kernels folder and ramdisk.image.gz file from the CD into your System Folder, unstuff the BootX_1.2b3.sit file located in the macosutils folder of the CD, move the BootX extension from the unstuffed BootX folder into your Extensions folder, and copy the Install LinuxPPC icon onto your desktop.

BootX (Mac OS only)

Users that ran the LinuxPPC 2000 Installer will be greeted with the BootX screen upon reboot.

It will start an automatic boot countdown - you can disable the countdown by hitting tab. Click on the Linux button to continue loading the installer.

The BootX window



Kernel : Linux Final

Root device : /dev/

☐ Set G3 cache

☑ Use RAM Disk

☑ No video driver

More kernel arguments :

ramdisk_size=8192

Notes on using BootX:

- You can use the tab key to toggle between Mac OS and Linux.
- The Kernel: menu allows you to select which kernel to boot from the Linux Kernels folder inside the System Folder
- The "Root device" space is left blank for now. After you have installed LinuxPPC, you would input the location of your root partition in this space.
- The "Set G3 cache" option allows you to set-up the state of the cache on G3 systems using G3 Cache Profiler by PowerLogix. Note that the G3 cache should already be configured properly for Apple G3 systems, but the cache control can be useful for systems with G3 upgrade cards. G3 Cache Profiler can be downloaded from http://www.powerlogix.com/support/software.html.
- The "Use RAM Disk" option is only necessary the first time you boot the installer - it loads the ramdisk.image.gz located on your hard drive or CD.

- The "No video driver" option uses Mac OS initialized video (unaccelerated). For now it is recommended that you keep this enabled - you can try disabling it and passing video settings to the kernel once you are more comfortable with LinuxPPC.
- Clicking on the "Save to prefs" button will store these settings so that you can select your default OS and store your other settings
- The "More kernel arguments" field includes specific information that you would like to pass to the kernel - for now you should leave it with the default setting (this setting increase the amount of ram that the kernel sets aside for the ram disk). Again, this is only necessary for initial installation and can be left blank once you have installed. This is also the location where you can specify video settings, the run level you boot into, and other options. More about this later. If you want to use the Red Hat style installer, you should append " redhat" (without the quotes) to the end of the argument.

Loading the kernel

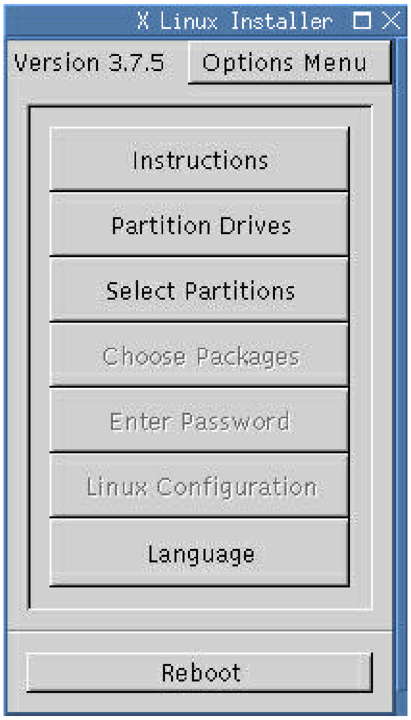
As the kernel loads, the screen will turn black and you will see the Linux penguin in the corner. Text will scroll by on the screen as the kernel proceeds to load.

The X Windows-based installer

The installer will then check the CD-ROM for the live.filesystem file so that it can load the X-based installer. If you are not installing from CD-ROM, the installer will look through your hard drive partitions (except for HFS+ partitions) for live.filesystem. If it's loading from CD, it will take a few minutes. Loading from the hard drive will take just a few seconds.

After the installer loads, you will be greeted by the installer window shown here:

The X Linux Installer's main interface is on the left side of the screen. Start with the "Instructions" button. Although it states that it cannot format HFS partitions, you can go into Perldisk by clicking on “Partition drives” (or go to pdisk through an Xterm window, if you prefer) and change the partitions types from HFS to A/UX. One other note: in the lower left-hand corner you'll notice an icon with the word "output" above it - this is a minimized Xterm window that displays the output from the installer. If you'd like to monitor installation progress, you can open up this window by double-clicking on the icon. It is particularly useful if you are having problems and would like to troubleshoot.



The initial installer window

Partition your hard drive

If you haven’t already done so, you should partition your hard drive now. You can do this in the LinuxPPC 2000 Installer using Perldisk. See the description of how to do this in *Partitioning your drive with Perldisk* above.

If you prefer, you can also use pdisk, a text-based partitioning tool. To open pdisk, click the mouse somewhere on the background of the screen - this will bring up the menu on the right.

Select XTerm - this will bring up a command-line Xterm window where you can type in "pdisk /dev/hda" followed by return to bring up pdisk (substitute /dev/hda with the actual name of your hard drive, typically /dev/hda for an internal

IDE drive or /dev/sda for the internal SCSI). Second internal IDE hard drives would be /dev/hdb if it's a slave drive or /dev/hdc if it's on the second bus. Secondary SCSI drives would be /dev/sdb, /dev/sdc, etc. based on the order that the kernel finds them on the SCSI bus. On a PowerBook G3, the right expansion module is /dev/hdc, and the left module is /dev/hde.

Here are the relevant commands in pdisk that you'll need:

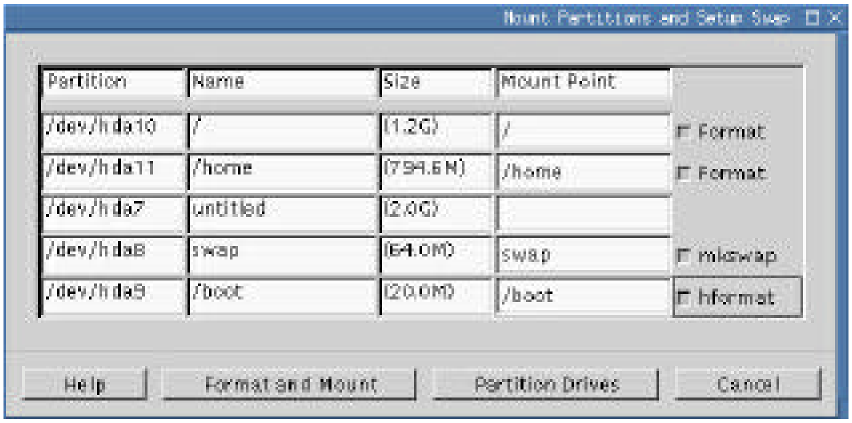
```
?       help
p       print out your partition map to screen
d[#]    deletes partition number [#] and converts it to free space
        Example: d 5 deletes partition number 5
n       creates a new Apple_UNIX_SVR2 (A/UX)partition
w       writes the partition map to disk
q       quit – does not automatically save changes. You must use w first
```

You should create at least two Linux (A/UX) partitions: one named /root and another named /swap. Root (also written as a slash "/") is where the Linux system files reside; swap is for virtual memory. Swap should be about 50 MB for a single-user computer; for servers and other multi-user computers, you should have about twice as much swap as you have RAM, but each swap partition can only be up to 128 MB. If you need more swap space, you can have more than one swap partition. For a full install, /root should be at least 1 GB. For more information, see *Partitioning your hard drive* above.

Select partitions

Now click on the "Select Partitions" button to bring up the "Mount partitions and setup swap" window. In this example I only want to install onto /dev/hde7 as my root partition and /dev/hda8 as my swap space (the partitions at /dev/hda10 and /dev/hda11 in the example are from a separate installation and aren't used here).

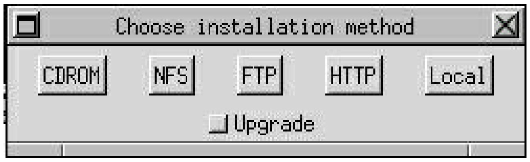
The Mount Partitions window



Partition	Name	Size	Mount Point	
/dev/hda10	/	(1.2G)	/	☐ Format
/dev/hda11	/home	(794.6M)	/home	☐ Format
/dev/hda7	untitled	(2.0G)		
/dev/hda8	swap	(64.0M)	swap	☐ mkswap
/dev/hda9	/boot	(20.0M)	/boot	☐ hformat

So under the Mount Point column I entered "/" for my root partition and indicated that I wanted that partition formatted by clicking on the button next to it. Write down the ID of this partition – you will need it later in BootX to specify the location of root. Swap was already entered since the installer recognized "swap" as the name of the partition. If you have any other partitions that you'd like to use as /home, /usr, etc. then select those partitions and type in the name of the directory you'd like that partition to mount at (for example "/home"). When finished, click on the "Format and Mount" button. It will take a few seconds to perform the operation.

Choose method



Next, you can select the "Choose Method" button. If you started the installer with the LinuxPPC CD in the drive, this button will not appear since the installer has already defaulted to a CD-ROM installation. The same is true if you downloaded all the installation files to the hard drive as described in the previous chapter. Otherwise, you will get the following window after clicking the "Choose Method" button:

If you choose CD-ROM, make sure that you have the CD inside the drive. The NFS, FTP, and HTTP selections will require that you set up your network settings:

You can get these settings from your original operating system, talking to your system admin, or by playing with the ifconfig, netstat, and route utilities in an Xterm window (the kernel will do its best to find out your network settings on its own). If your network uses DHCP, you can have your machine query for a DHCP address by going into an Xterm window and typing in dhcpcd. This will give you your IP address and put your name server in /etc/resolv.conf (type in "cat /etc/resolv.conf" to view them).

After configuring networks settings for a NFS, FTP, or HTTP installation, you'll get a window that looks something like this (I'm using NFS as an example):

Type in the address of the server followed by the directory that contains the RedHat directory. For example, if I were to perform an FTP installation from the ftp.cdrom.com mirror, I would input the following:

```
ftp.cdrom.com
pub/linux/linuxppc/linuxppc-2000
```

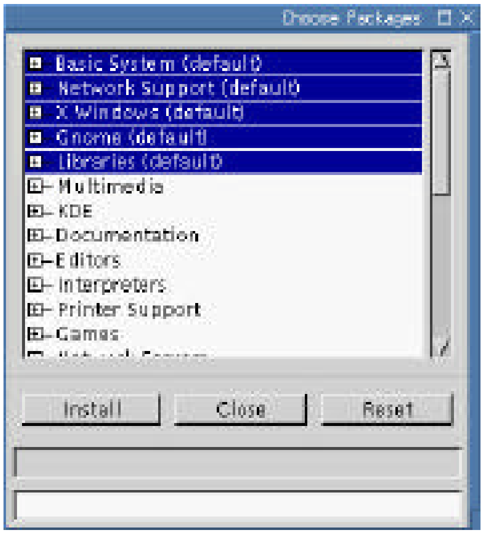
Notice that I did not put a / before "pub/linux/linuxppc/linuxppc-2000" nor is there a / at the end. It would look like this when I was finished:

Then click "Ok". The installer should query the site and download the necessary files for you to proceed to the next step, choosing the packages.

Choose packages

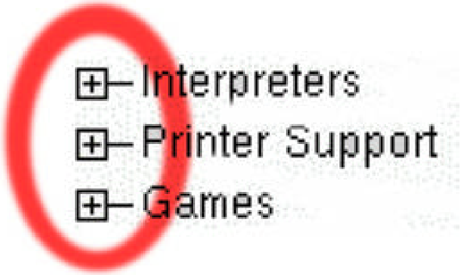
The installer should then bring up the Choose Packages window automatically - otherwise click on the button from the main menu window.

The default installation will require 350 MB of drive space - if you choose everything, it will consume nearly 1 GB. Then click on Install after selecting the files you want. (I will be updating this guide later to include details on different packages). You'll get a progress bar that indicates how installation is going. If you'd like more details, you can double click the "output" icon to get an Xterm window that displays each RPM being installed.

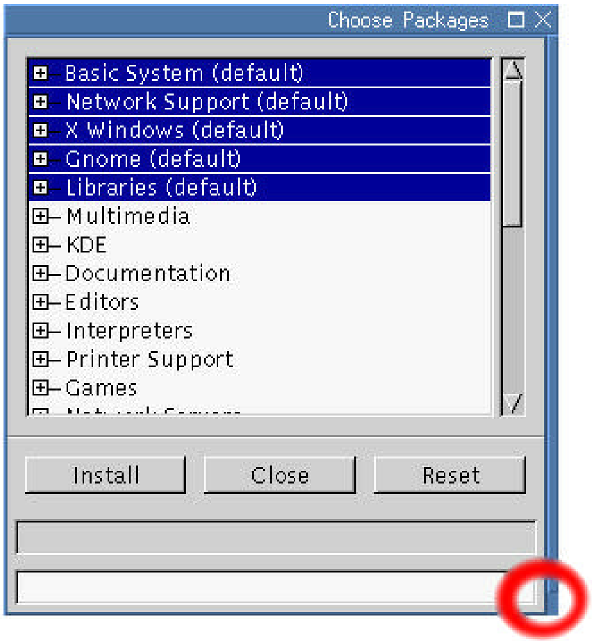


Choose packages window

Note that you can click on the plus symbols to expand or collapse the package categories to customize the installation, as illustrated below. You can also click on the resize tab on the bottom right of the window (circled in red) to enlarge or resize the window.



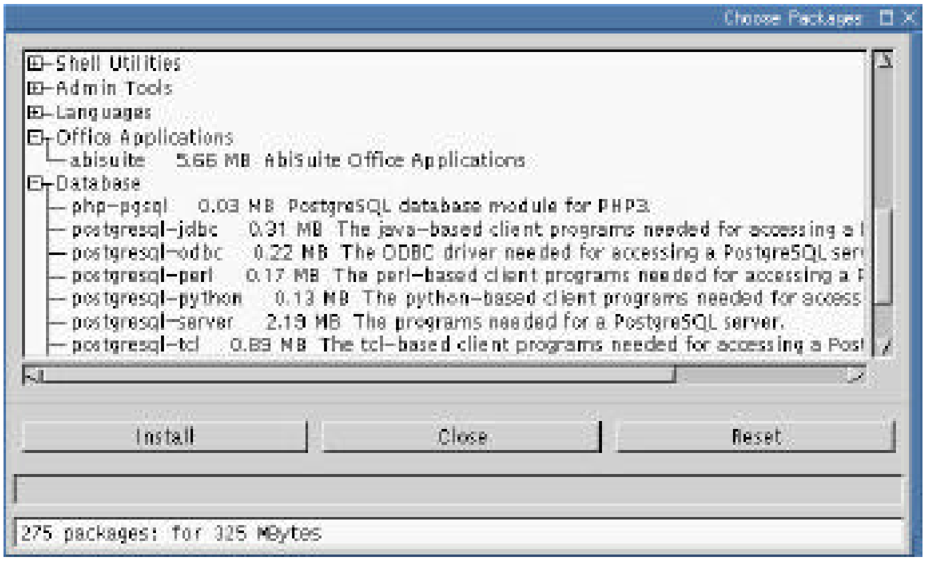
Click on the plus symbols (circled here) to expand ot collapse package categories.



```
⊞ Basic System (default)
⊞ Network Support (default)
⊞ X Windows (default)
⊞ Gnome (default)
⊞ Libraries (default)
⊞ Multimedia
⊞ KDE
⊞ Documentation
⊞ Editors
⊞ Interpreters
⊞ Printer Support
⊞ Games
```

Click on the lower right corner of the window to resize it.

Choose packages window (expanded)

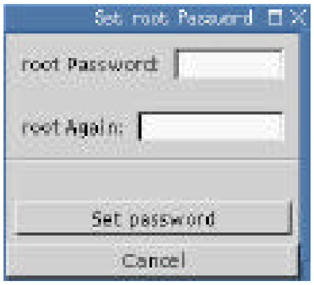


```
⊞ Shell Utilities
⊞ Admin Tools
⊞ Languages
⊟ Office Applications
   └ abisuite    5.66 MB  AbiSuite Office Applications
⊟ Database
   ─ php-pgsql    0.03 MB  PostgreSQL database module for PHP3.
   ─ postgresql-jdbc    0.31 MB  The java-based client programs needed for accessing a I
   ─ postgresql-odbc    0.22 MB  The ODBC driver needed for accessing a PostgreSQL ser
   ─ postgresql-perl    0.17 MB  The perl-based client programs needed for accessing a P
   ─ postgresql-python    0.13 MB  The python-based client programs needed for access
   ─ postgresql-server    2.19 MB  The programs needed for a PostgreSQL server.
   ─ postgresql-tcl    0.83 MB  The tcl-based client programs needed for accessing a Post

275 packages: for 325 MBytes
```

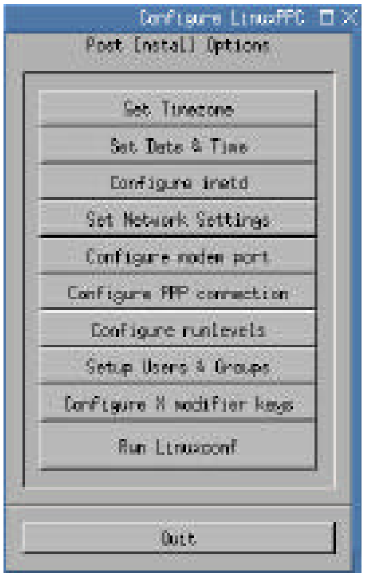
Set root password

The next step is to enter the password for root (the superuser of the machine). Don't use any dictionary words, since this is the most critical aspect of security on your system.



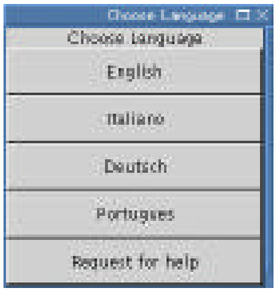
Configure LinuxPPC

Now you can set up the basic configuration for your LinuxPPC system using the window at the right. While you do not need to configure everything in the list, it would be a good idea to do so now.



Choose language

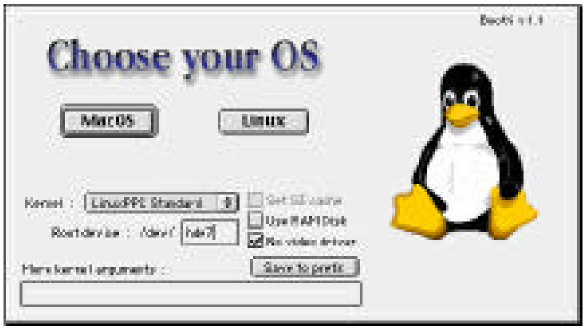
You can choose a default language setting with the “Language” button as illustrated here.



Congratulations!

Installation of LinuxPPC 2000 is complete. Click on the "Finished" button to read some notes, then click “Reboot”. If you use Mac OS, it will reboot into Mac OS and into the BootX screen, like the example shown below. Hit Tab to disable the countdown so that you can set up BootX to boot LinuxPPC 2000.

Under the root device field, enter the partition name that you had set as / during installation. Uncheck the "Use RAM Disk" button and clear the "More kernel arguments" field. Hit tab so that the Linux button is highlighted to make it your default OS, then click on the "Save to prefs" button. Then hit return to boot into LinuxPPC 2000.



Kernel : LinuxPPC Standard

Root device : /dev/ hda7

□ Set OF colors

□ Use RAM Disk

☑ No video driver

More kernel arguments :

An example BootX configuration

Getting started with LinuxPPC 2000

If you're new to LinuxPPC or Linux in general, this section is for you. Here we have assembled some information about basic Linux commands and configuration tips to help get you started using Linux effectively.

OK, I've booted into Linux. Now what?

Congratulations! Now you're ready to take the first step. You should be seeing a login window. If this is your first time booting into Linux, type in "root" for the username and your root password (the one you chose during the installation – hopefully you haven't forgotten it already!) as the password. Then click on "Login". You should now be logged in and see your desktop, assuming you chose to boot into X Windows (most likely yes).

Create users. The first thing you should do is create a new user for yourself. You should avoid working on your computer as root, since you can cause serious damage to the system if you don't know what you're doing or aren't careful. To do this, open a Terminal window by clicking on the Gnome menu (the one with the footprint), if you installed GNOME, or the K menu, if you installed KDE. (GNOME is the default desktop that is installed with LinuxPPC 2000.) Or right-click with your mouse (if you have a multi-button mouse) on the desktop, or point the mouse at the desktop and hit Option-3 if you have a single-button mouse, to see a pop-up menu, including "New terminal...".

There are several terminals you can use, depending solely on your preference. There are xterm, Eterm (with Enlightenment), rxvt, GNOME Terminal and others.

In the terminal window, use the command "adduser" to create a new user for yourself. Choose a username; note that this is case-sensitive ("John" is different from "john"). Then type in "passwd [username]" (in this example, "passwd john") and enter a new password. Then log out (in GNOME, click on the Gnome menu and choose "Log out...") and log in again as yourself.

To remove a user, you must be root. To remove the user "haXXor" from your computer, type in "deluser haXXor".

If you do need to do system maintenance later, you can change users without logging out first. To become Mary, for example, you can type "su mary" in a terminal window (also called an "Xterm window") and enter Mary's password. To become root, type in "su" (with no username) and enter the root password. You will have to be root to do most configuration and maintenance tasks in LinuxPPC.

The GNOME Desktop

The GNOME desktop should be immediately familiar to any computer user coming from Mac OS or Windows. The GNOME menu – the button with the footprint – is like the Apple menu or Start button. This is where you can find most of your installed programs as well as configuration tools and so on. The bar in which the menu resides is the GNOME Panel, which can contain useful applets like clocks, system monitors, pagers, and more. To manipulate files, double-click the "Home directory" icon – this starts Midnight Commander, which is similar to the Macintosh Finder or Windows Explorer in purpose and functionality.

The KDE Desktop

If you chose to install KDE instead of GNOME, you'll have the KDE desktop. This works ver y much like GNOME, but looks slightly different.

Where do I find stuff?

Linux has a general structure in the directories (folders) that it tries to follow. Generally speaking, this is the way files are stored under LinuxPPC:

/home/yourname	Your home directory (also written ~/)
/home/httpd/html	Web pages (if Apache web server is installed)
/usr	Unix System Resources
/var	Logs and other similar information
/etc	System configuration files
/usr/bin	Global binaries (programs for general use)
/usr/local/bin	Local binaries (programs for local use)
/usr/X11/bin	Programs for X Windows
/opt	Optional software; some programs stored here
/usr/src	Source code for compiling goes here
/tmp	Temporary data
/sbin	System binaries

Tips for basic Linux commands

Here is an overview of tips for using basic Linux commands. You can type these in any command line, if you are in the shell (text-only interface) or if you start a Terminal window.

Backgrounding apps If you are using a command to start an X application (that is, a program with a graphic interface that runs under X Windows), type the command name with an ampersand “&” after the command, like this:

```
commandname &
```

This starts up the program so that it doesn’t hog your terminal window – you can still use the terminal while the program is running. Otherwise, the terminal is “in use” by the program and can’t be used while that program is running.

Auto-complete commands Another tip with the terminal: if you partially type a pathname or command name, you can hit the Tab key to complete it (if it exists and is available to you). If more than one item starts with the characters you typed, you’ll hear a beep; hit Tab twice quickly to see a list of the possible matches. Otherwise the name will magically complete itself, and you can hit Enter/Return to execute the command.

Piping commands You can “pipe” commands into each other. This means that you can take the output from one command and pass it to another command for further processing. To do this, type a command (but don’t hit Enter yet), then type the “pipe” key (|) and then enter the next command. Now hit the “Enter” key. For example:

```
ps ax | grep httpd
```

The command “ps ax” lists all currently running processes on the computer; “grep httpd” searches for the text string “httpd”. By piping ps’s output to grep, we can search for what processes are running that are called httpd (the Apache webserver).

List of basic Linux commands

Without further ado, here is a list of basic Linux commands and their meanings:

Command	*Description*	*Example*
adduser	Create new user	`adduser [username]`
cd	Change directory	`cd [pathname]`
clear	Clear screen	`clear`
cp	Copy	`cp /path/filename /newpath/newname`
df	Disk free space (in blocks)	`df`
df -h	Disk free space (in MB)	`df -h`
du	Disk usage (in blocks)	`du`
du -h	Disk usage (in MB)	`du -h`
exit	User logout	`exit`
ftp	Text-based FTP client	`ftp [servername or IP address]`
help	Calls short summary about a command	`[command] --help`
httpd	Starts Apache webserver	`httpd`
kill	Force-quits a program The “pid” is found using the command “ps ax”	`kill [pid]`
ln -s	Creates a symbolic link Similar to a Mac alias or Windows shortcut	`ln -s [linkname] [link destination]`
ls	List directory contents	`ls [directory name]`
ls -al	Lists all directory contents This includes “invisible” files, i.e. files with a “.” beginning their names	`ls -al [directory name]`
man	Opens manual page for a command	`man [command]`
mkdir	Create directory	`mkdir [directory name]`

Command	*Description*	*Example*
more	Read text file Press “Q” to quit the program, spacebar to page through the file	`more [filename]`

mount	Mount a disk or filesystem
	The mount point must exist;
	if need be, a mount point
	can be created with
	`mkdir /mnt/mountpoint_name`
	Examples:
	`mount -r -t iso96600 /dev/cdrom /mnt/cdrom`
	This mounts an ISO-formatted CD-ROM at /mnt/cdrom
	`mount -t hfs /dev/hda5 /mnt/mac_files`
	This mounts a Mac partition on an IDE drive at /mnt/mac_files
		`mount [options] [format] [mount point]`

mv	Move/rename a file	`mv /path/oldname /newpath/newname`
passwd	Create/change password	`passwd [username]`
ping	Sends a data packet to a remote computer to see if network connection is working	`ping [IP address or hostname]`
ps	Show running processes	`ps ax`
pwd	Show current directory location	`pwd`
rm	Remove (erase) a file or directory with `rm -r directory_name`	`rm [filename]`
shutdown	Shut down or reboot	
	To restart	`shutdown -r now`
	To shut down	`shutdown -h now`
startx	Start X Windows	`startx`
tar	Archives or unpacks data	
	Unpacks a gzip file	`tar xzfv filename.tar.gz`
	Creates an archive from the folder ./data	`tar czfv data.tar.gz ./data`
telnet	Remote login on another computer	`telnet [IP address or hostname]`
vi	Starts vi, a text editor	`vi [filename]`

Installing software with RPM

If you want to install software, LinuxPPC 2000 includes the Red Hat Package Manager (RPM). There are several ways to use RPM.

The basic method is to get a PPC binary RPM package (named “filename.ppc.rpm”) from an FTP server ot Web server. It must be a PPC package; packages for other processors won’t work (like i386 or Alpha). To install a package, issue the following command in a terminal window:

```
rpm -ivh /path/to/package/packagename.ppc.rpm
```

If an older version of the package is already installed, use this command:

```
rpm -Uvh /path/to/package/packagename.ppc.rpm
```

To install directly from an FTP server, use the commands above in this form:

```
rpm -Uvh ftp://ftp.someserver.com/path_to_package/packagename.ppc.rpm
```

Another option is to get a source RPM. A source RPM package contains uncompiled source code, which you must compile for your computer. Use this command set to compile and install source RPMs:

```
rpm --rebuild /path/to/package/packagename.src.rpm
rpm -ivh /usr/src/redhat/RPMS/ppc/packagename.ppc.rpm
```

To find out if a package is already installed, use the query function of RPM:

```
rpm -q packagename
```

To remove an installed package, use this command:

```
rpm -e packagename
```

If you are using GNOME and have GnoRPM installed, you can also right-click on an RPM package (identified by a special RPM icon) and do all of the above steps with the commands in the pop-up menu. (Single-button mice can emulate a right-click by pointing at an icon and pressing Option-3.)

For more about RPM, type “man rpm”.

Getting more help with LinuxPPC

If you need more help than this User's Guide provides, there are plenty of places online to get it. The Linux community is large and growing fast; there is almost certainly information out there to help you solve your problem.

The first place to start is the LinuxPPC mailing lists, which you can find at

http://lists.linuxppc.org/

You can search the list archives there for help, and you can also join the list and ask questions there (though we do ask that you search the archives first).

There is also a Usenet newsgroup, reachable via news server or Deja.com's website. Try visiting news:comp.os.linux.powerpc to get started.

There are also plenty of web pages with useful information. We suggest the following links:

http://www.linuxppc.com/	LinuxPPC, Inc.'s website (updates, news, etc.)
http://www.linux.org/	Linux International's website
http://www.linuxppc.org/	LinuxPPC.org's website; links to other information
http://sunsite.unc.edu/LDP/	Linux Documentation Project
http://www.dartmouth.edu/cgi-bin/cgiwrap/jonh/lppc/faq.pl	Pmac-Linux FAQ-O-Matic
http://www.imaclinux.net/	iMac-centric Linux information
http://www.linuxmac.net/	News and information about Linux for Macs
http://www.maclinux.de/	German Mac/Linux website
http://www.linuxnewbie.org/	Discussion forums and articles for newbies
http://www.linuxhelp.net/	Howtos, tips and guides to running Linux
http://www.gnome.org/	Documentation and software for GNOME
http://www.kde.org/	Documentation and software for KDE
http://www.ppc.kernel.org/	Information about the PowerPC port of Linux

Naturally this only scratches the surface of the available websites about Linux, but these ought to get you on your way.

If you purchased a CD-ROM of LinuxPPC 2000, you can also contact support@linuxppc.com for direct support information.

Appendix A: Boot Variables with Open Firmware

Some PowerMacintosh and other computers can boot LinuxPPC 2000 using Open Firmware, instead of using BootX or yaboot (or other booting methods). This section lists some common OpenFirmware boot variables, which you can use to set up your computer to boot directly into LinuxPPC 2000.

Open Firmware (OF) is used to build the device tree and boot the operating system. Open Firmware boot variables can be modified in three ways:

- with the Boot Variables program in Mac OS (PowerMacintosh G3's should use Multibooter)
- with setenv within Open Firmware
- with nvsetenv within LinuxPPC.

The official Open Firmware website is at http://playground.sun.com/1275/home.html. Apple technotes 1061, 1062, and 1063 may also be useful. Also see the /Linuxppc/OpenFirmware category at the FAQ-O-Matic.

To get into Open Firmware, you will need to change some of your boot variable settings. From Mac OS, this can be done with the Boot Variables application. Boot Variables does not work on Powerbook 2400 and 3400. However, this does not matter in these machines since you can go directly into OF at boot-up by holding down the cmd-option-o-f key combination at startup until OF's white screen appears.

Those machines that can run Rhapsody can use Multibooter instead of Boot Variables. It's available from ftp://ftp.apple.com/devworld/Rhapsody/UsefulStuff, and is actually preferable to Boot Variables in Apple G3 machines since it applies some patches to OF.

You can interface directly with Open Firmware using two basic methods:

- with your monitor and keyboard
- viewing Open Firmware on your monitor is only possible on versions of Open Firmware which are able to initialize the video based on settings in the output-device boot variable

To interact with Open Firmware with your keyboard and monitor, set input-device to kbd, output-device to the setting for your machine (if available). Set auto-boot to false. Then write and reboot – you'll get into Open Firmware's interface, which is in Forth. Remember that only these machines with a properly configured, supported output-device can view OF output on their monitor. Refer to Appendix B for output-device settings for different machines.

- with a second computer, a serial cable, and a terminal program like zterm (available for download from hyperarchive). This should work on all models which have an available serial port.

To talk to OF from a second computer using Zterm, you will have to set input-device and output-device to ttya. if you're using the modem port (ttyb for the printer port). Connect the serial cable from the respective serial port to the terminal machine (machine with Zterm).

Launch Zterm in the terminal machine and change connection settings so that Data Rate = 38400, Data Bits = 8, Parity = none, and Stop Bits = 1. Disable auto-boot in the host machine and then write and reboot – you should soon see Open Firmware output in Zterm. *Thanks to Seth Paskin for sending us a guide on this method.*

To get into Open Firmware while auto-boot is enabled, reboot the machine and hold down the Command-Option-O-F key combination before the startup chime until the OF screen appears.

To get back into Mac OS, you can enter bye at the OF prompt and it will begin booting Mac OS. You can also reset your boot variable settings by zapping your parameter ram (PRAM) by holding down the Command-Option-P-R key combination during and after the start-up chime. This will also reset many of the settings in your control panels in Mac OS.

Some useful commands in OF (remember that it is case sensitive):

devalias	lists the aliases you can use for your devices
dev / ls	prints out your device tree
dev [device name] .properties	lists some data about a certain device
words	lists "commands" for the current device
boot [device]	tries to boot off that device
bye	boots you into Mac OS
printenv	prints your nvram settings
setenv [variable] [setting]	sets your nvram settings
set-defaults	reset Open Firmware settings
eject fd	ejects floppy disk
reset-all	reboots the computer
shut-down	shuts down the system

Useful Open Firmware boot variable options

Variable	*Setting*
auto-boot?	True / false true makes OF boot the boot-device automatically. False drops you into OF's interface during boot-up (requires setting the input-device and output-device) Alternatively, you can leave auto-boot? True and hold down the cmd-option-o-f key combination to get dropped into OF's interface. On the Powerbook G3 Series, you might have to use the right option key to get this to work.
boot-device	Defines what device you will boot off of. Includes Mac OS ROM, SCSI, IDE, and floppy

boot-file	May be used to define what kernel to use when booting, or pass specific options to the kernel such as the location of root and the video mode resolution for Linux to use. The format for vmode would be vmode=X , where X is the vmode setting. Examples include:
	vmode=5 (640x480 @ 60 Hz)
	vmode=6 (640x480 @ 67 Hz)
	vmode=10 (800x600 @ 60 Hz)
	vmode=11 (800x600 @ 72 Hz)
	vmode=12 (800x600 @ 75 Hz)
	vmode=13 (832x624 @ 75 Hz)
	vmode=14 (1024x768 @ 60 Hz)
	vmode=15 (1024x768 @ 72 Hz)
	vmode=16 (1024x768 @ 75 Hz)
	vmode=18 (1152x870 @ 75 Hz)

`vmode=19` (1280x960 @ 75 Hz)
`vmode=20` (1280x1024 @ 75 Hz)
** note that your video chipset may support some modes differently*

`input-device`

`kbd, ttya, or ttyb`
This allows you to input data to OF using the keyboard, modem port (with a second computer), or
printer port (using a second computer).
It is usually a good idea to set this so that you can input commands to OF, even if you can't see its output.

`Output-device`

`ttya, ttyb,`or the OF address for your video chipset on your machine, if supported
This only needs to be set if you want to use
OF's interface. Refer to Appendix B for a listing of each machine.

`Load-base`

For most machines you should leave it at its default value. For the Apple 5400, 5500, 6360, 6400, 6500, 20th Anniversary(?), Umax C500 and C600, and Power Computing Powerbase you will need to set it to 100000. For Apple G3 computers (desktops, minitowers, and PB G3 Series), it may be necessary to set load-base to 1000000 to prevent problems with gzip. On some G3 desktops/minitowers, you may find it necessary to try setting load-base to 4000, 60000, 100000, or 1000000 to successfully boot/install.
Note that load-base cannot be set with Multibooter, so run Multibooter first if you need to, then Boot Variables to set load-base (or use setenv in OF).

`Boot-command`

set to whatever special command OF should get upon bootup (usually boot). If you're using a SCSI drive and you're getting a lot of DEFAULT CATCH errors during bootup, this may be due to OF trying to boot off the drive before it is ready. Add `begin ['] boot catch 1000 ms cr again`to your boot-command so that OF will repeatedly try booting the SCSI drive until it works

You can also do things such as create a boot-up menu (Appendix E)

Appendix B: Open Firmware output-device Settings

Some models do not support monitor output from Open Firmware, so if you need to see OF's output you'll need to use a second computer and a terminal program, as described in Appendix A: Open Firmware and Boot Variables.

Make sure that output-device is not set to ttya if you have a modem connected to it, as modems sometimes interfere with Open Firmware (the same is true for ttyb if your modem is on the printer port)

For machines with add-on pci cards, you can try the following:

- ATI Xclaim 3d: /bandit/ATY,XCLAIM3D or pci1/ATY,XCLAIM3D or pci2/ATY,XCLAIM3d
- ixMicro TwinTurbo: /bandit/IMS,tt128mb or pci1/IMS,tt128mb or pci2/IMS,tt128mb (use pci1 and pci2 only if your machine has 2 pci buses, such as PM 9500)

Company	*Computer*	*output-device*
Apple	20th Anniversary	OF does not sync with built-in monitor
	PB 2400	default settings
	PB 3400	default settings
	PB G3 (original)	default settings
	PB G3 Series	default settings
	4400	screen
	5400	screen (works for floppy boot only - for scsi/IDE, set to ttyb)
	5500	/bandit/ATY,264GT-B*
	6360, 6400	screen (works for floppy boot only - for scsi/IDE, set to ttyb)
	6500	/bandit/ATY,264GT-B
	7200	no screen output available
	7300, 7500, 7600	/chaos/control**
	8200	no screen output available?
	8500, 8600	/chaos/control** or /bandit/ATY,XCLAIM or /bandit/ATY,264VT
	9500	/chaos/control** or /bandit/ATY,XCLAIM or /bandit/ATY,264VT or pci2/ATY,mach64

Company	*Computer*	*output-device*
Apple (cont'd)	9600 (ATI)	/chaos/control** or /bandit/ATY,XCLAIM or /bandit/ATY,264VT
	9600 (TwinTurbo)	/bandit@F4000000/IMS,tt128mb8A or /bandit@F2000000/IMS,tt128mb8A
	G3 (ATI Rage II)	screen or /pci/ATY,XCLAIMVR_II or /pci/ATY,mach64_3DU
	G3 (Rage Pro)	screen or /pci/ATY,XCLAIMVRPro or /pci/ATY,mach64_3DUPro
Motorola (and clones)	3000, 4000	screen
	5000, 5500 (ATI)	/bandit/ATY,RAGEII_M or /bandit/ATY,264GT
	5000, 5500 (Twin Turbo)	/bandit/IMS,tt128mb ?
Power Computing	Powerbase, Powercenter Pro	/bandit/ATY,mach64_3D_pcc or /bandit/ATY,264gt
	Powercenter	/bandit/ATY,264gt (some early models can't initialize the screen)
	Powercurve	?
	Powertower, Powertower Pro	pci1/IMS,tt128mb8 or pci2/IMS,tt128mb8
	Powerwave	/bandit/ATY,XCLAIM (200 MHz model may use /bandit/ATY,Mach64 (?)
UMAX	Apus 2000	screen
	C500, C600	no screen output available
	J700	/bandit/IMS,tt128mb
	S900	/bandit/IMS,tt128mb8 or /bandit/IMS,tt128mb

* - for the Powermac 5500, OF display may appear dim You can try using the following in the boot-command to brighten it up. *Thanks to Jens Ch. Restemeier*

```
0f3000030 000ffcf76 over ! 4 + 0f02f7f7a over ! 4 + 072ff3e00 swap ! boot
```

** - OF initialization of video using /chaos/control/ is inconsistent (OF version 1.0.5). Alan Mims sent in a fix for OF 1.0.5 to the Linux-pmac mailing list, which is in Appendix C.

Appendix C: Fixes for Apple OpenFirmware 1.0.5

[from the linux-pmac mailing list]

Alan Mimms, Sun, 17 Aug 1997 19:08:19 +0000

Hi.

I have finally had enough with the buggy screen and disk drivers in Apple's Open Firmware on PowerMac 7200, 7500, 7600, 8500, 9500, and probably some others I'm forgetting. So I have written some NVRAMRC based patches to Open Firmware to hack around the bugs enough for my purposes anyway. I hope the hacks may help you too.

Start up and break into your Open Firmware -- Cmd-Opt-O-F during the boot beep held down until the "user interface" for Open Firmware comes up on your screen or, if you're smart, your serial port. The banner printed by Open Firmware shows the Open Firmware version. These patches ONLY apply to Open Firmware version 1.0.5. Other versions will crash if these patches can be applied at all!

Type at the prompt:

```
nvedit
```

and hit control-L to see the cryptic stuff that is part of your Open Firmware's startup sequence. This stuff patches various bugs in the ROM. Hit control-N enough times that you no longer see a new line of gobbledegook every time several times in a row. This means you're at the bottom. Either paste the following into a terminal session (NOTE: must be at least a dozen or so ms delay between characters and maybe 100ms between lines to work right!) or else enter the following lines very very carefully:

```
dev /bandit/gc/via-cuda
' write value &W
: -&We &W swap - execute ;
: P1 4D8 -&We false 548 -&We ;
&W FC + ' P1 Blpatch
: P2 0C 2 ms ;
&W E0 + ' P2 Blpatch device-end

: wBoot
begin
   boot-device ['] $boot catch drop
   ." -Waiting for boot-device" cr
   d# 500 ms
key? Until
;
```

Note that ALL whitespace above except for the line indentation is REQUIRED. FORTH is a very very very strange "language". It may be safe to leave the line indentation as I have it above when pasting if you wish. It's wasteful but who cares?

At the end of this laborious typing (or pasting) session hit control-C (yes that's right: control-C is the end of editing session character in Open Firmware). Then type at the prompt

```
nvstore
```

to save the changed NVRAMRC variable into NVRAM.

The first block fixes a bug in the via-cuda driver in which not enough time is given for the device to settle when it is told to set the video controller's clocks up.

The second block defines a FORTH word that can be used in place of the normal boot-command contents to wait for the disk to spin up before attempting to really boot. This avoids the standard "black screen the first time you power on the computer each day" problem.

Then type at the prompt

```
setenv boot-command wBoot
```

This sets up the default command executed on auto-boot (normally on power on) to run the above disk spinup waiting hack.

FYI: the reason I know about this stuff is that I worked for about two years as the Copland booting guy. Sheesh…

Please don't inundate me with a bazillion questions about Open Firmware. I have a real job and it takes 60+ hours a week of my time. I get PAID to do it. I just did this to fix MY PPC Linux box and I wanted to help you folks out a teeny bit if I could. No, I don't work at Apple anymore. Cancelling Copland was the last straw. I worked there more than nine years. (sigh)

Happy trails.

Appendix D: Partitioning with pdisk and fdisk

The Red Hat Installer includes pdisk for you to partition drives (the installer refers to using fdisk, but it is actually linked to a version of pdisk). It is capable of partitioning both SCSI and IDE drives and should work with drives from any manufacturer. The only difference between using fdisk (pdisk) in the installer and pdisk in Mac OS or within Linux is the syntax that each uses for its commands.

The version of pdisk that runs as a Mac OS application was ported by the MkLinux team and also uses a command line interface. They have provided documentation on using this version of pdisk. These can be found at ftp://ftp.linuxppc.org/pub/linuxppc/linuxppc-R4/RedHat/tools/. Note that the partition naming scheme that they use (/dev/scsi0.1, /dev/ata0.0) is different than what LinuxPPC uses or what Open Firmware uses.

Each hard disk requires a partition map as the first partition. Disks used for Mac OS also require partitions which contain drivers and other patches. The rest of the partition map consists of the hard drive partitions – typically, this will initially be one large Apple_HFS partition dedicated to Mac OS.

Once partitions have been created, they are independent of any other partitions on the same disk. Each partition can be changed to a different type or reformatted without affecting any other partitions sharing the same drive. Unfortunately, splitting partitions will not protect them from hard drive failure.

In general, a Linux user coming from Mac OS will be concerned with two partition types here: Apple_HFS for Mac OS, and Apple_UNIX_SVR2 for Linux-pmac. While fdisk/pdisk can create Apple_HFS partitions, it cannot write the proper Mac OS drivers so that they can be used for Mac OS. It also will not format the partitions. Mac OS applications such as Disk First Aid, HD setup, and Drive Setup can properly initialize these partitions once they have been created in fdisk/pdisk. These partitions can later be formatted into HFS or HFS+ by the Mac OS.

Using fdisk/pdisk

Reminder: partitioning usually results in making all old data on your drive inaccessible, so remember to back-up first!

The Red Hat installer will drop you directly into fdisk (pdisk) after you select which disk to partition.

Otherwise, if you're calling up pdisk from a shell prompt, you'd use the syntax pdisk /dev/hdX (where X is a, b, c, etc.) if you're editing an IDE drive or pdisk /dev/sdX for a SCSI drive.

Fdisk and pdisk commands

fdisk	*pdisk*	*function*
?	?	help
p	p	prints the partition table
I	I	initialize the partition map (erases the old map)
n	c	create a new Apple_UNIX_SVR2 partition
N	C	create a new user-defined partition type (Apple_UNIX_SVR2 or Apple_HFS)
d	d	delete a partition (turns it into "Extra" space)
w	w	write the new partition table (permanently saves changes)
q	q	quit (does not automatically save changes)

The first thing to do when you start-up fdisk/pdisk is to print your partition map (with p). It will look something like this:

```
/dev/hda
    #: type                  name                length          base ( size )
    1: Apple_partition_map   Apple               63      @       1
    2: Apple_Driver43        Macintosh           64      @       64
    3: Apple_Driver_ATA      Macintosh           64      @       128
    4: Apple_Patches         Patch Partition     512     @       192
    5: Apple_HFS             untitled            5008048 @       704 ( 2.4G)

Block size=512, Number of Blocks=5008751
DeviceType=0x0, DeviceId=0x0
Drivers-1:
@ 64 for 18, type=0x1
2: @ 128 for 30, type=0x701
```

The numbers under the # sign are the partition numbers. The length is the size of the partition in blocks (typically, 1 block = 512 bytes), and the base is the block number where the partition starts. Since each block is 512 bytes, it takes 2048 blocks to make up one megabyte. Take partition #5 for example - its length is 5008048: dividing by 2048 gives you 2445.3 MB, shown in the "size" column as 2.4G.

Partitions can only be created within free space (type = Apple_Free). These are made by deleting existing partitions, using the d [partition #] command. Once you have free space, you can create new partitions within it using the c or C commands in pdisk (n or N commands in fdisk). Using C/N requires that the user specify what type of partition is being created; otherwise using c/n creates Apple_UNIX_SVR2 partitions.

When creating new partitions, pdisk/fdisk will ask you for the first block of the partition, its length in blocks, and the name of the partition. You can define the first block in one of two ways:

- by its actual location as specified in the 'base' column of the partition map, or
- by the partition number itself, followed by the letter 'p.'

For example, the first block of partition 5 in the map above can be defined as 704 or 5p.

The length in blocks of the partition can also be be defined one of two ways:

- by the number of blocks the partition occupies, or
- the size of the partition in terms of kilobytes, megabytes, and gigabytes followed by the letter k, m, or g, respectively. If you define it in terms of k, m, or g, the numbers must be integers (no decimal points).

If you wanted to create a 2.5 megabyte partition, you could use 5120 as the length in terms of blocks (2048 x 2.5) or you could use 2560k (= 1024 x 2.5). Using 2.5m would not work, as the decimal point will throw off pdisk/fdisk.

One additional note: if you want to enter a partition name that has spaces in it, you need to put quotes (") around it when you enter it.

Example use of pdisk

I think easiest way to learn how to partition is to see for yourself how it is done – with that in mind, I'll go through the steps on splitting the one large Apple_HFS partition at partition 5 into four partitions:

1. Mac OS partition, type Apple_HFS, approx. 2 gigabytes
2 .root partition, type Apple_UNIX_SVR2, 60 MB
3. swap partition, type Apple_UNIX_SVR2, 60 MB
4. usr partition, type Apple_UNIX_SVR2, 280 MB

User input is displayed in red . Notes are *italicized* and should not be entered into pdisk. Pdisk output is shown in blue .

```
Command (? For help):d 5
Command (? For help):N
First block:704  (you can also use 5p instead)
Length in blocks:4188848  (you can also use 2g instead)
Name of partition:Mac OS
Type of partition:Apple_HFS
Command (? For help):p

/dev/hda
#: type                    name              length        base ( size )
1: Apple_partition_map     Apple             63     @      1
2: Apple_Driver43          Macintosh         64     @      64
3: Apple_Driver_ATA        Macintosh         64     @      128
4: Apple_Patches           Patch Partition   512    @      192
5: Apple_HFS               Mac OS                   4188848 @      704 ( 2.0G)
6: Apple_Free              Extra             819200 @      4189552 (400.0M)

Block size=512, Number of Blocks=5008751
DeviceType=0x0, DeviceId=0x0
Drivers-1:
@ 64 for 20, type=0x1
2: @ 128 for 31, type=0x701

Command (? For help):n
First block:4189552  (you can also use 6p instead)
Length in blocks:122880  (you can also use 60m instead)
Name of partition:root
Command (? For help):p

/dev/hda
#: type                    name              length        base ( size )
1: Apple_partition_map     Apple             63     @      1
2: Apple_Driver43          Macintosh         64     @      64
3: Apple_Driver_ATA        Macintosh         64     @      128
4: Apple_Patches           Patch Partition   512    @      192
5: Apple_HFS               Mac OS                   4188848 @      704 ( 2.0G)
6: Apple_UNIX_SVR2         root              122880 @      4189552 ( 60.0M)
7: Apple_Free              Extra             696320 @      4312432 (340.0M)

Block size=512, Number of Blocks=5008751
DeviceType=0x0, DeviceId=0x0
```

```
Drivers-1:
@ 64 for 20, type=0x1
2: @ 128 for 31, type=0x701

Command (? For help):n
First block:4312432  (you can also use 7p instead)
Length in blocks:122880  (you can also use 60m instead)
Name of partition:swap
Command (? For help):p

/dev/hda
#: type                     name                length        base ( size )
1: Apple_partition_map      Apple               63      @     1
2: Apple_Driver43           Macintosh           64      @     64
3: Apple_Driver_ATA         Macintosh           64      @     128
4: Apple_Patches            Patch Partition     512     @     192
5: Apple_HFS                Mac OS              4188848 @     704 ( 2.0G)
6: Apple_UNIX_SVR2          root                122880  @     4189552 ( 60.0M)
7: Apple_UNIX_SVR2          swap                122880  @     4312432 ( 60.0M)
8: Apple_Free               Extra               573440  @     4435312 (280.0M)

Block size=512, Number of Blocks=5008751
DeviceType=0x0, DeviceId=0x0
Drivers-1:
@ 64 for 20, type=0x1
2: @ 128 for 31, type=0x701

Command (? For help):n
First block:4435312  (you can also use 8p instead)
Length in blocks:573440  (you can also use 280m instead)
Name of partition:usr
Command (? For help):p

/dev/hda
#: type                     name                length        base ( size )
1: Apple_partition_map      Apple               63      @     1
2: Apple_Driver43           Macintosh           64      @     64
3: Apple_Driver_ATA         Macintosh           64      @     128
4: Apple_Patches            Patch Partition     512     @     192
5: Apple_HFS                Mac OS                      4188848 @     704 ( 2.0G)
6: Apple_UNIX_SVR2          root                122880  @     4189552 ( 60.0M)
7: Apple_UNIX_SVR2          swap                122880  @     4312432 ( 60.0M)
8: Apple_UNIX_SVR2          usr                 573440  @     4435312 (280.0M)

Block size=512, Number of Blocks=5008751
DeviceType=0x0, DeviceId=0x0
Drivers-1:
@ 64 for 20, type=0x1
2: @ 128 for 31, type=0x701
```

```
Command (? For help):w
Writing the map destroys what was there before is that okay? [n/y]y:
The partition table has been altered!
Command (? For help):q
```

Print your partition table to display often, so you know you didn't make a mistake. If you did make a mistake, you can simply delete the partitions that you've made and create new ones. Changes are not permanent until you write the partition. Remember to write your partition table when you're done, otherwise, nothing happens.

Appendix E: Open Firmware Boot Menu

OF Boot Menu by Andi Payn

Andi Payn came up with a cool little Open Firmware program that you can put into the boot-command option in Boot Variables to make it give you an OS boot selector. Here is a copy of Andi's posting about it to the linux-pmac mailing list:

- - - - - - -

Subject: Open Firmware boot menu!
Date: Sat, 19 Jul 1997 18:17:58 +0000
From: andi payn [payn@null.net]
To: Multiple recipients of list (linux-pmac@arvidsjaur.anu.edu.au)

I know I'm not the only one here with a roommate/kid/whatever who can't seem to figure out how to properly boot the Mac as a Mac and ends up ruining everything by trying... And I know some people envy the boot menu that you get with MkLinux (and BeOS).

In the true spirit of Linux-pmac, of course, we want a boot menu that doesn't require loading Mac OS.

(If you really wanted to do that, here's what you'd have to do: (1) Write an extension for the Mac OS that displays an OS Chooser dialog. If the user chooses Mac OS, unload yourself nicely; if the user chooses Linux, rewrite the boot variables to boot to Linux, then reboot. (2) Change your Linux startup so you nvsetenv the boot variables back to boot to Mac OS. Or just change your shutdown script so it changes boot variables first...)

There are two ways to get something to run at startup in OpenFirmware. The first, and probably superior, is to write an nvramrc file and point OF at it. While I can see where, in BootVariables, I should be setting things, I haven't been able to make this work. I'll mention the details of what I tried later, and any help would make me very happy, but for now, let's assume that we can't make that work. There's still another possibility: The boot command. Some people have had to modify their boot command to wait for their drive to spin up before booting and so forth. I modified mine to display a boot menu.

Now, my Forth skills are very rusty, and my OF knowledge is still pretty rudimentary, and I just slapped this together. Hopefully people will improve on my work.

The major limitation here is that you have a limited amount of space to use. I didn't count it, but I'm guessing it's 255 characters (it took me a few errors and freezes to realize it was throwing away the end of the string if it got too long).

What does it look like when I boot my computer? Well, I see a menu that looks like this:

```
<M>ac
<L>inux
<O>F
```

Then the computer starts slowly counting down in hex from 100. If I hit L (without the shift key!) during the countdown, it boots Linux/pmac. If I hit O, it gives me the usual OpenFirmware startup info (BYE to boot Mac OS, BOOT to continue, etc.) If I hit M, or anything else for that matter, or wait for it to finish counting, it boots Mac OS.

Actually, I think you have to hold down the key for a second or so or it won't work. Anyway, it's not beautiful, but it works. My roommate can figure out how to hit the M key, and if he can't, he can just wait.

You probably want to write your script in SimpleText and copy it (or just copy mine from this email) and paste it into boot-variables. Or you'll want to write a brief shell script in emacs or vi that does it. Either way, remember to keep it all on one line...

So to anyone else who used to enter 1-line program contests in Applesoft Basic, this should be a nice bit of nostalgia.

Here's the program, formatted to look pretty (that's a relative term):

```
" <M>ac" cr ."<L>inux" cr ." <O>F" cr
variable i 100 i !
begin
    key?
    if
        key
        dup ascii o = if throw then
        ascii l = if boot then
        bye
    then
    i @ dup . 1- dup i ! 0<> while
    1000000 begin 1- dup 0<> while repeat drop
repeat
forget i
bye
```

For those with some Forth knowledge, but not too much, I'll explain. The first line displays the menu.

Then we create our loop index variable, i, and set it to 0x100. We're going to loop through the "begin..repeat" loop 255 times (someone who can remember how to use a +loop could make this simpler and save us a few characters we could spend elsewhere) or until someone hits a key.

Each time through the loop, we first check to see if a key's been pressed with the "key?" command.

Unfortunately, in OF I think it returns true only if a key is currently being held down, rather than if there's a key in the buffer (which is why you have to hold the key down).

If a key's pressed, read it (with the "key" command). Then check it against lower-case o. If it matches, throw. Since we're not catching, OF will catch it and dump the user into the OF screen.

Probably not the prettiest way to do this, but the only thing I could think of. Then check to see if it's an L, and if so, boot. Note that your boot-device and boot-file have to be set up right for this to work (you could use eval to pass parameters to boot, but you'll be wasting characters you don't have to waste...).

Otherwise, another key's been pressed, so boot Mac OS with "bye." If you want to change the logic so invalid keys are ignored, change the line above to "dup ascii l = if boot then" and change this line to "ascii m = if bye then" and hopefully that won't push you over 255 characters. If you need to squeeze a few out, change the first line to:

```
   " <M>ac <L>inux <O>F" cr
```

Now, if no key's been pressed (or a key's been pressed and it's invalid), we're going to display the index, decrement it, and check it. If it's still non-zero, the loop keeps going (that's what the "while" does in this begin..repeat loop).

The next line, the "body" of the loop, just loops 0x1000000 times. That's about 16 million, and should take your computer somewhere in the vicinity of a second (depending on your computer). You can adjust this if it's off (actually, it's pretty far off for my computer...). Or, better yet, you could use a command that's made for waiting. I know there must be one, I just don't remember what it is. As a guess, I'd say "1000 wait" to wait for 1000ms... Play around with it.

After the outer loop's done, and we haven't quit, we know that the user isn't paying attention. So we clean up (probably not necessary, if you're looking for characters to save you could kill the "forget i" line) and boot Open Firmware ("bye").

Here's the whole thing on one line, ready to paste into BootVariables:

```
" <M>ac" cr ."<L>inux" cr ." <O>F" cr variable i 100 i ! begin key? if key dup ascii
o = if throw then ascii l = if boot then bye then i @ dup . 1- dup i ! 0<> while
1000000 begin 1- dup 0<> while repeat drop repeat forget i bye
```

Enjoy it!
- - - - - - -

It's pretty effective, but there are a few problems I had with his configuration:

- there should be a '.' at the beginning of the command (right before " <M>ac") otherwise that text won't show up in OF
- it should be '." <L>inux"' rather than '."<L>inux"' (note the space between the quote and <L>). Otherwise, OF barfs.

When booting to Linux, the residuals of the menu command after the outer loop gets fed to the second stage QUIK loader. Normally, my settings in boot-file get sent to quik, but this isn't happening with the menu command. This confuses linux bootup on my StarMax 4160. (in other-words, quik gets sent the option 'forget i bye' instead of the usual '/vmlinux root=/dev/hda5' that I use to boot off the IDE).

I can get around this in two ways:

- manually type in '/vmlinux root=/dev/hda5' when quik asks for the bootup image. This gets to be a pain.
- Fix /etc/quik.conf so that it knows what to do if it sees the 'forget I bye'. I decided to get rid of the 'forget i' part on the command - it doesn't seem to affect anything, so now all that gets sent to the quik loader is 'bye'. All I had to do was add 'alias = bye' to my quik.conf, and it loads off of 'bye'. So now my quik.conf on my IDE root looks like this:

```
# partition = X is only necessary if root isn't the 1st bootable partition
partition = 5
timeout = 0
default = linux
image = /vmlinux
    label = linux
    root = /dev/hda5
    alias = bye
# switch bye to whatever commands comes 1st after the outer loop of the menu
# command (like 'forget'). I haven't tested it though.
```

I also got rid of the OF option because the throw wasn't working correctly. I haven't tried to figure out why yet.

So now my boot-command looks like this:

```
." Please select:" cr ." <m>ac" cr ." <l>inux" cr ." & hold down the key
for a sec." cr variable i 9 i ! begin key? if key dup ascii l = if boot then
bye then i @ dup . 1- dup i ! 0<> while 700000 begin 1- dup 0<> while repeat
drop repeat bye
```

Finally, Harry Eaton has a variation on the boot menu that he included in his fix for OF on the G3 (ftp://ftp.linuxppc.org/users/harry/fixg3.tgz):

```
." Select" cr ." (m)ac" cr ." (l)inux" cr ." (o)f" cr a 0 do 50000 begin
key? if drop leave then 1- dup 0 <> while repeat drop i . loop key? if key
dup 6d = if bye then 6f = if quit then then bootr
```

Appendix F: Information for BeBox Installation

This page on BeBox installation courtesy of William R Sowerbutts <btg@guru.dircon.co.uk> http://www.guru.dircon.co.uk/belinux/install.html

LinuxPPC

Linux for the BeBox is based on the LinuxPPC distribution. They have some specific information for installing LinuxPPC on the BeBox here.

Chuck Shotton has written up some installation scripts to make life easier when installing. They're detailed on his site.

A Brief Warning

Note that the entire contents (each and every partition) of the hard disk you choose to install the standard LinuxPPC distribution on will be erased; you will require two hard disks if you wish to use both BeOS and LinuxPPC on the same machine. This fact was not well documented on the LinuxPPC web site when I installed it. This is not true if you use the patched kernels detailed on this page, however.

Installing LinuxPPC and BeOS

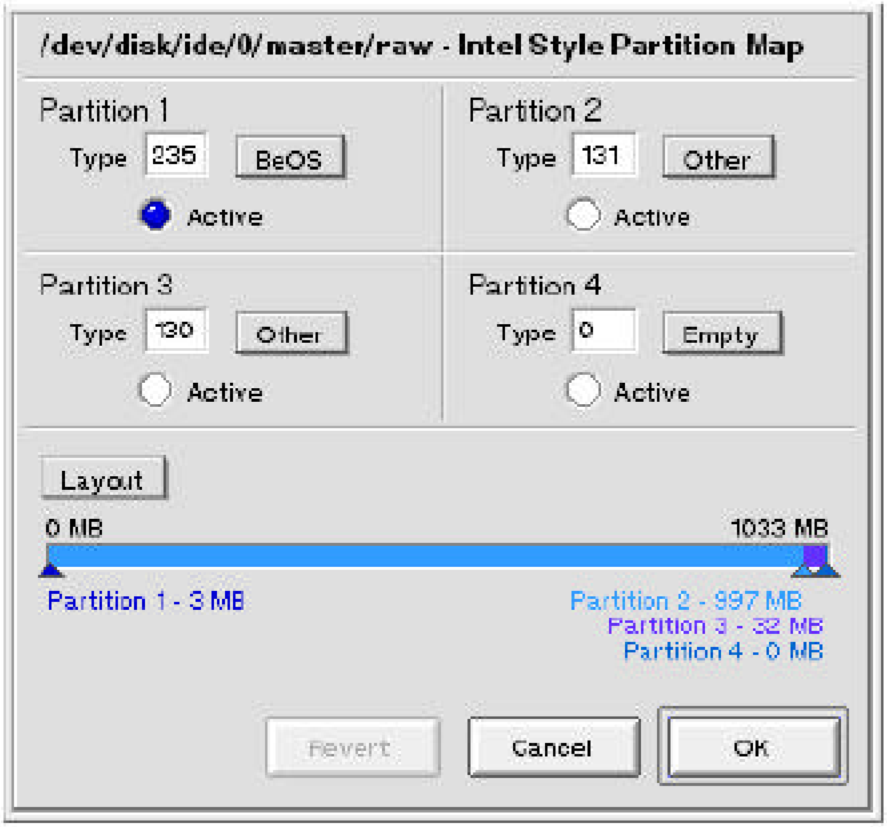
Your mission, should you choose to accept it, is to persuade LinuxPPC and BeOS to share a partition map. The problem is that LinuxPPC and BeOS use different partition maps by default. Since BeOS is the "native" operating system of the BeBox, and since we have the means to change the way LinuxPPC behaves but not the way BeOS behaves, we have to change our partitioning scheme to match theirs.

BeOS supports two partition mapping schemes; Intel and Apple style. LinuxPPC supports Intel style partition mapping, so we'll use that one. Since none of the LinuxPPC partitioning software will write out correctly formatted partition maps (yet), you MUST do your partitioning under BeOS. You will require a blank hard disk to install on. Our target partition layout is this:

Partition	Type	Contents	Size
1	BeOS (0xEB)	Bootstrap for booting LinuxPPC	At least 3Mb
2	Linux Native (0x83, 131 decimal)	Your root Linux filesystem	Most of the disk
3	Linux Swap (0x82, 130 decimal)	Swap space for Linux	About 32Mb
4	Unpartitioned (0x00)	Unused slot	0Mb

If you want to have BeOS itself on the same hard drive, use the fourth partition. Do not attempt to use the first partition. It won't work. You'll have to install BeOS on the fourth partition and make it the default partition to boot from (there's a Be application for doing that in Preferences somewhere, I'm sure).

Boot the BeOS and load the Drive Setup application. Right click on the hard drive you wish to use for Linux, then choose "Setup", "Partition", "Intel" . A dialog will pop up. Adjust the settings to match the partition layout given above. For reference, this is how my 1Gb Linux drive looks under Drive Setup:Format (initialise) the first partition on this disk as a BeOS partition. Give it a name like "LinuxPPC bootstrap".



/dev/disk/ide/0/master/raw - Intel Style Partition Map

Partition 1
Type 235 BeOS
● Active

Partition 2
Type 131 Other
○ Active

Partition 3
Type 130 Other
○ Active

Partition 4
Type 0 Empty
○ Active

0 MB 1033 MB

Partition 1 - 3 MB
Partition 2 - 997 MB
Partition 3 - 32 MB
Partition 4 - 0 MB

Find a LinuxPPC bootstrap floppy disk, and place it in the floppy drive on your BeBox. I recommend you use the very latest one, which is available here. Details on writing it to a floppy are further down this page.

Open the "Disks" window, and from the "File" menu choose "Mount Settings". Click the "Mount all disks now" button. Open your Linux bootstrap disk, and create a folder named "system" on it. Open the folder named "system" that is on the floppy disk, and copy the file "kernel" from the floppy disk to the folder named "system" on your newly formatted hard drive. That drive will now load the LinuxPPC bootstrap if you try to boot from it. Now you will need to download some more things (sorry!). The kernel has had to be patched to work with the Be style partition mapping. There are several files available, all by anonymous FTP from ftp.halfast.com in the /pub/be-linux-devel/ directory. This is a list of the files (just click one to download it).

Filename	*Contents*	*Size*
980610-kernel-patches.tgz	This archive file contains the various files I've patched. For the partition mapping, it's drivers/block/genhd.c. There are also the results of my fiddling with the interrupt handling code in here.	19Kb
980610-bootstrap.gz	This is the latest version of the bootstrap floppy disk. Uncompress and write this image out to a floppy disk. Leave the floppy disk in the drive and reboot your BeBox to begin loading LinuxPPC.	30Kb

Filename	*Contents*	*Size*
980610-setup.gz	This is a compressed kernel image of the kernel you require to install LinuxPPC. You must have this file to install with a Be-style partition map. Uncompress it before writing it to a floppy disk.	1322Kb
980610-boot-ide.gz	Once you have installed LinuxPPC you must use a second kernel to boot for normal operation. This is a compressed kernel image of the kernel you require if your root disk is on IDE. You must have this file to run Linux with a Be-style partition map. Uncompress it before writing it to a floppy disk.	478Kb
980610-boot-scsi.gz	Once you have installed LinuxPPC you must use a second kernel to boot for normal operation. This is a compressed kernel image of the kernel you require if your root disk is on SCSI. You must have this file to run Linux with a Be-style partition map. Uncompress it before writing it to a floppy disk.	478Kb

Performing the installation

Download the latest bootstrap from the above list. Uncompress and write it out to a formatted 1.44Mb 3.5" floppy disk with a command like this:

```
inferno:~/belinux$ gzip -dc 980610-bootstrap.gz > /dev/fd0
```

Label that floppy disk Bootstrap. You will need this disk to prepare the hard drive as a bootstrap (see above). Insert a new one and repeat the process for the setup disk. Change the disk again, and then write out your boot disk - make sure you choose the correct disk, depending on whether the disk you are installing on is IDE or SCSI. Both disks include drivers for IDE and SCSI, but one disk expects to find the root filesystem on the first IDE disk, whereas the other expects to find it on the first SCSI disk. You cannot presently install on disks other than the first SCSI disk (well you can, but you'd need to patch the kernel first).

Place the "Setup" disk in your floppy drive, and reboot the BeBox. Hold down the shift key as soon as the Be logo appears (or even from before it appears, if you can). You should get a list of the BeOS filesystems in your system. One of them should be the bootstrap you prepared earlier. If you cannot see it, choose the bottom option, which will cause the system to rescan the SCSI and IDE busses looking for disks. If you still cannot see it after a few seconds, something has gone wrong. Did you copy the bootstrap code to "system/kernel" on that disk, as described above?

If you still cannot get it to work, something is seriously wrong. Contact me and tell me what you've done.

When you've found the correct disk to boot from, highlight it and press return. You should see the video card change to text mode and the words "PowerPC Linux for BeBox bootstrap" followed by a massive wadge of status and debug information. If you just get a black screen, reboot and try again. If you get no change at all, you have an unsupported video card. Sorry about that. We're actively working to add support for more video cards, even as you read this.

With the "Setup" floppy in your floppy drive, hit return. It's okay to boot with this floppy in the drive, the boot ROM won't try to load an operating system from it ;)

The system will load the kernel from the floppy disk. As it loads, the two rows of lights on either side of the BeBox will "fill up" . After the kernel has loaded, the screen will clear again and you'll be confronted with the "Linux/PowerPC Load:" prompt. Just hit return at this point, and wait as the installation kernel and RAM disk (slowly) decompresses.

Finally, the screen will clear once more and Linux will boot into single user mode from the RAM disk built into the setup kernel image. You should get a "#" prompt. During the "Partition check" you should see "{BeFix}" printed for the disk you plan to install on (either sda or hda). If you don't, abandon your installation now and contact me!

Type "crdisk-net" and hit return to begin the installation. Choose "Diskettes" as your installation method. Enter "/dev/fd1" as the device name for your floppy. Make sure you enter the correct device name to install onto - use "/dev/sda" for the first SCSI disk, and "/dev/hda" for the first IDE disk. Name "dev/hda2" or "dev/sda2" as the root partition, and "dev/hda3" or "dev/sda3" as the swap partition. When it asks you for the options for cfdisk, erase the "-z" from the command line! Do NOT repartition your hard drive when it tells you to! It will start cfdisk for you, you must quit it without allowing it to alter your partition table. To do this, move the cursor over to "Quit" using the right arrow key, then press the return key. You will need to download, decompress, and write some files out to floppy disk to complete the installation. These are all found on the LinuxPPC FTP site.

Once you've completed the installation you'll get the "#" prompt again. Reboot the BeBox and start the bootstrap as before. Instead of using the "Setup" disk to load the kernel, however, use one of the "boot-ide" or "boot-scsi" disks. You may now install RedHat PowerPC RPMs to get a working LinuxPPC installation.

Page last modified: Fri Jun 12 21:11:34 1998

Be Install Scripts

Below are some installation scripts for the BeBox, courtesy of Chuck Schotten. These can also be found at his website at http://www.biap.com/Linux/be.html

This is a quick and dirty page describing 3 scripts for use in installing the LinuxPPC 98 CD distribution on BeBox hardware. That CD installer will not work with the BeBox, hence the need for these scripts.

This page assumes you have a basic working knowledge of Unix-like operating systems and can create directories, edit files, etc. The scripts seem to work, but sometimes kernel crashes do occur during execution. If this happens, you should modify the scripts to resume where you left off, rather than rerunning the complete script.

These scripts are based on information found at Robert Currey's site (http://www.halcyon.com/curreyr/BeLinux.html). What is important is that they attempt to document the minimal packages that need to be installed and the basic order in which they should be installed. If the script execution fails, you may be better off running the individual commands by hand.

Finally, you should start with the 2.0.30 kernel available from ftp.linuxppc.org. These scripts should be run after performing a floppy-based install as described at that site. A network-based install that uses the TAPE_FILE3.gz file system will probably not allow X Windows to be installed later.

Install script 1

This script creates a mount point for the CD, mounts it, and installs the basic libraries and utility functions necessary to continue.

```
Mkdir /mnt/cd
mount -o ro /dev/sr0 /mnt/cd
cd /mnt/cd/RedHat/RPMS
rpm -ihv --nodeps glibc-0*
rpm -ihv ldconf*
rpm -ihv --nodeps glibc-info*
rpm -ihv --nodeps glibc-static*
rpm -ihv --nodeps libc-5*
rpm -ihv --nodeps libc-static*
rpm -ihv --nodeps libg++-2*
rpm -ihv --nodeps libgr-2*
rpm -ihv --nodeps libgr-progs*
rpm -ihv --nodeps libjpeg-*
rpm -ihv --nodeps libpng-0*
rpm -ihv --nodeps libtermcap-2*
rpm -ihv --nodeps zlib-1*
rpm -ihv --nodeps pwdb-0*
rpm -ihv --nodeps perl-*
rpm -ihv --nodeps ncurses-*
rpm -ihv --nodeps nvi-*
echo ""
echo "Now edit the /etc/ld.so.conf file and add the following lines:"
echo "/lib"
echo "/usr/lib"
echo "/usr/local/lib"
echo "/usr/X11/lib -- only necessary if you plan on installing X"
echo ""
echo "Next, run ldconfig as follows:"
echo "/sbin/ldconfig -v -f /etc/ld.so.conf"
echo ""
echo "Finally, reboot the BeBox (shutdown -r now) and continue with the"
echo "second install script, be2."
Echo ""
```

Install script 2

This script installs additional tools, networking functions, and documentation.

```
Mount -o ro /dev/sr0 /mnt/cd
cd /mnt/cd/RedHat/RPMS
rpm -ihv --nodeps binu*
rpm -ihv --nodeps less-*
rpm -ihv --nodeps setup-*
rpm -ihv --nodeps etcskel-*
rpm -ihv --nodeps rootfiles-*
rpm -ihv --nodeps bash-*
rpm -ihv --nodeps diffutils-*
rpm -ihv --nodeps e2fsprogs-*
rpm -ihv --nodeps textutils-*
rpm -ihv --nodeps findutils-*
rpm -ihv --nodeps cpio-*
rpm -ihv --nodeps fileutils-*
rpm -ihv --nodeps gawk-3*
rpm -ihv --nodeps sed-2*
rpm -ihv --nodeps cracklib-*
rpm -ihv --nodeps pam-0*
rpm -ihv --nodeps sh-utils-*
rpm -ihv --nodeps util-linux-*
rpm -ihv --nodeps rpm-2*
rpm -ihv --nodeps rpm-*
rpm -ihv --nodeps passwd-0*
rpm -ihv --nodeps termcap-9*
rpm -ihv --nodeps groff-1*
rpm -ihv --nodeps man-1*
rpm -ihv --nodeps man-pages*
rpm -ihv --nodeps SysVinit-*
rpm -ihv --nodeps slang-*
rpm -ihv --nodeps newt-*
rpm -ihv --nodeps tcp_wrappers*
rpm -ihv --nodeps net-tools-*
rpm -ihv --nodeps netkit*
rpm -ihv --nodeps MAKEDEV*
rpm -ihv --nodeps sysklogd*
rpm -ihv --nodeps telnet*
rpm -ihv --nodeps wu*
rpm -ihv --nodeps lynx*
rpm -ihv --nodeps adduser*
/sbin/ldconfig -v -f /etc/ld.so.conf
echo ""
echo "Reboot the BeBox to complete the basic installation. (shutdown -r now)"
```

X Windows Install script

This script installs the basic X Windows client functionality. You likely won't be able to run a X server on the local BeBox console with these files, but the packages below will make it possible to talk to the BeBox via X with a server running on a separate platform.

```
Mount -o ro /dev/sr0 /mnt/cd
cd /mnt/cd/RedHat/RPMS
rpm -ihv --nodeps --force xinitrc-*
rpm -ihv --nodeps --force X11R6.3-0*
rpm -ihv --nodeps --force X11R6.3-100dpi-fonts*
rpm -ihv --nodeps --force X11R6.3-75dpi-fonts*
rpm -ihv --nodeps --force X11R6.3-devel*
rpm -ihv --nodeps --force X11R6.3-fonts*
rpm -ihv --nodeps --force X11R6.3-libs*
rpm -ihv --nodeps --force X11R6.3-man*
/sbin/ldconfig -v -f /etc/ld.so.conf
```

Additional Info

If you have additional information to contribute, suggestions, or changes, please send e-mail to <cshotton@biap.com>.

Appendix G: Information for CHRP Installation

Installation information on CHRP systems is courtesy of Geert Uytterhoeven. This information can also be found at his website at http://www.cs.kuleuven.ac.be/~geert/PowerPC/Install/

This section discusses the basic installation of Linux/PPC on the CHRP LongTrail.

Since the floppy drive isn't working yet under Linux, the easiest installation method is using a root file system on NFS. Make sure you have a second computer available that can act as a RARP/BOOTP and NFS server. Alternatively you can format a hard disk on another machine (msdos partitioning), install the root filesystem and move the disk to your CHRP machine.

Files

You can download the following files from this page:
http://www.cs.kuleuven.ac.be/~geert/PowerPC/Install/

Linux kernels

There are two kernels available, one with support for ADB keyboards and mice, and another with support for a PS/2 keyboard and mouse (you can't have both ADB and PS/2 support in the same kernel (yet)). Both kernels support DEC21041 Ethernet and ATI video boards. If you have a different video board, it will still work, as long as Open Firmware knows how to initialize the board.

ZImage.adb (http://www.cs.kuleuven.ac.be/~geert/PowerPC/Install/zImage.adb)
Kernel image with support for ADB keyboards and mice
.config.adb (http://www.cs.kuleuven.ac.be/~geert/PowerPC/Install/.config.adb)
Kernel configuration for the above kernel image
System.map.adb (http://www.cs.kuleuven.ac.be/~geert/PowerPC/Install/System.map.adb)
System.map for the above kernel image
zImage.ps2 (http://www.cs.kuleuven.ac.be/~geert/PowerPC/Install/zImage.ps2)
Kernel image with support for a PS/2 keyboard and mouse
.config.ps2 (http://www.cs.kuleuven.ac.be/~geert/PowerPC/Install/.config.ps2)
Kernel configuration for the above kernel image
System.map.ps2 (http://www.cs.kuleuven.ac.be/~geert/PowerPC/Install/System.map.ps2)
System.map for the above kernel image

Root filesystem

chrproot.tar.gz (http://www.cs.kuleuven.ac.be/~geert/PowerPC/Install/chrproot.tar.gz)

This is a modified version of TAPE_FILE3.gz, which supports the new console device minor. Extract this archive on your NFS server.

Kernel source patches

chrp-2.1.72-19971223.diff.gz (http://www.cs.kuleuven.ac.be/~geert/PowerPC/Install/chrp-2.1.72-19971223.diff.gz)

brings your 2.1.72 kernel source tree to the same level as the `bleeding edge' source tree at vger.rutgers.edu (dated December 22, 1997).

Booting Linux

Open Firmware booting is nice: just copy the kernel to a msdos formatted floppy and type

```
boot a:<kernel-name> root=/dev/nfs
```

The kernel has support for both RARP and BOOTP to find out it's IP address (and hostname). It will mount (using NFS).

```
/tftpboot/<ip-address>
```

or

```
/tftpboot/<hostname>
```

(if BOOTP supplied a hostname) as its root file system. If you want to disable RARP and BOOTP, add

```
`ip=off'
```

to the boot command. My /etc/bootptab looks like

```
callisto:\
:hn:ht=ethernet:vm=rfc1048:\
:ha=0080c85af85b:
```

callisto is the name of my CHRP machine, 0080c85af85b is the hardware address of my Ethernet board.

If Open Firmware doesn't support your video board, you can still boot Linux using a serial terminal, connected to one of the 9 pin D-SUB serial ports, as the console, by adding

```
console=ttyS0
```

or

```
console=ttyS1
```

to the boot command. The first things the boot loader says are:

```
Boot device: <boot-device> File and args: <file-and-args>
chrpboot starting
gunzipping… done
start address = 0x9000100c
copying OF device tree… done
instantiating rtas… done
```

The copy process takes about 10-15 seconds.

After this you should see the well known Linux penguin logo and the kernel boot messages, and you'll be thrown into a single user shell. Then you can partition your hard disk and copy the root file system to it.

Good luck!

Miscellaneous

- If you don't specify a root file system, the default is /dev/sda2.
- Of course you can put kernels on hard disk too. Just create a msdos formatted partition (e.g. on /dev/sda1) and copy your kernels to it. I use

```
boot scsi/disk@6,0:1,zImage ip=off
```

to boot from the file zImage on the first partition of the SCSI disk at unit 6 LUN 0.

- IDE disks are not yet supported. Besides, IDE sucks! SCSI lives!

Appendix H: Information for PReP Installation

Installation information on PreP systems is courtesy of Cort Dougan. This information can also be found at his website at http://linuxppc.cs.nmt.edu/getit.html

How to get the system

What you need to get for the PReP systems. See Paul Mackerras' info for what you need for the PowerMac systems using this kernel. The current pmac archive is here (ftp://samba.anu.edu.au/pub/linux-pmac/). We are slowly merging to one binary for pmac/prep/chrp.

If you have problems: email cort@cs.nmt.edu with questions

Files you need for the RedHat based install method

RedHat-style installer

RedHat-style installer (ftp://linuxppc.cs.nmt.edu/pub/linuxppc/install/installer.prep)

Just dd or raw-write this image onto a floppy and boot it

RedHat-style packages

RedHat-style PPC packages (ftp://linuxppc.cs.nmt.edu/pub/linuxppc/install/)

You can install these with the install program by allowing it to ftp them during the install or you can retrieve them now.

Boot image – for after you have installed your system

Zimage (ftp://linuxppc.cs.nmt.edu/pub/linuxppc/kernel-images/zImage)

Appendix I: HFS/HFS+ Compatibility

Linux is currently incompatible with HFS Extended partitions.

What this means

Linux cannot read from or write to HFS Extended partitions. However, you can have the BootX extension/app and kernel files on an HFS Extended partition. BootX, a Mac OS application, automatically "knows" how to read/write on HFS Extended volumes.

What you can't do

- Mount an HFS Extended volume in LinuxPPC.
- Copy files to or from an HFS Extended volume.

The Mac OS's ability to read/write HFS Extended volumes is unaffected by having LinuxPPC installed on a Macintosh computer.

What you can do

- Keep BootX and the Linux kernels on an HFS Extended volume/partition.
- Run Mac-on-Linux to access Mac OS files. Mac-on-Linux can use HFS Extended volumes, since it's running the Mac OS inside a protected memory space in LinuxPPC.
- Run the netatalk AppleShare IP server to serve files to Macs with HFS Extended volumes.
- Serve FTP, and any other network service to Macs with HFS Extended volumes.

Additional notes

Mac-on-Linux

The Mac-on-Linux (MOL) Mac OS runtime environment is provided, but we are not providing technical support for it. Free, community support for MOL is available from our new Mac-on-Linux mailing list. To join the list, please visit the LinuxPPC mailing lists web site.

http://lists.linuxppc.org/

Installer handling of HFS partitions

You may wish to have the computer automatically mount HFS partitions during Linux's start-up process. This is possible by adding a "mount point" in the installer during the Mount and Format Partitions phase of installation.

However, the installer does not cor rectly write the partition type (HFS) in the /etc/fstab file. It writes it as type ext2. You will need to manually change the partition type in the /etc/fstab file.

If you want to edit the file and are not familiar with Linux/UNIX text editors, you may want to use the pico editor, a simple command-line text editor, or the gedit editor, a GUI-based text editor that is installed with the default LinuxPPC installation. You need to be logged in as root to edit the file.

If you have Perl experience and wish to help find a solution, please examine the /usr/X11/R6/bin/xinstaller.pl program. Suggestions/fixes can be sent to bresen@linuxppc.com.

Apple PowerMac G4 Compatibility

All the Apple PowerMac G4 models shipping as of January 2000 can boot and run LinuxPPC. Certain features, such as sound, Ethernet, and ease of installation are in various stages of development. Please check <http://www.linuxppc.com/support/updates/product/> for current information on the status of the G4.

Apple Blue & White PowerMac G3 Compatibility

Users of the Blue & White PowerMac G3 computers should visit Robert Shaw's Blue G3 Linux site at <http://www.linuxppc.org/blueg3/>. This site has late-breaking news, tips and advice for Blue G3 users.

Thanks for using LinuxPPC 2000!

We at LinuxPPC, Inc. would like to thank you for choosing to use LinuxPPC 2000. Your support is important to keep Linux – especially for the PowerPC – going strong. Buying CDs and other items from LinuxPPC, Inc. also benefits open source developers, since LinuxPPC, Inc. supports them both financially and logisitcally. Supporting companies and developer groups making PowerPC-native software for Linux also helps in the long term.

We hope you enjoy using LinuxPPC 2000 and become an active, supportive member of the Linux community. Open source works – but only if you help out.