Red Hat Linux 4.2

The Official Red Hat Linux Users Guide

The Red Hat Development Team:

Donald Barnes

Marc Ewing

Michael Fulbright

Michael K. Johnson

Jim Knoble

Michael Maher

Hunter Matthews

Clarence Smith, Jr.

Erik Troan

Red Hat Software, Inc.

ISBN: 1-888172-65-7

Revision 4.2, April 1997.

Red Hat Software, Inc.
3203 Yorktown Ave. Suite 123
Durham, NC 27713
(919) 572-6500
`redhat@redhat.com`
`http://www.redhat.com`

Contents

Preface

What is Red Hat Linux?

Red Hat Linux is a complete operating system based on the Linux kernel, which was originally created by Linus Torvalds. Linux continues to be developed by a worldwide team, led by Linus, working over the Internet and is still evolving forward rapidly. The currently stable 2.0.x kernel (on which Red Hat Linux 4.2 is based) demonstrates the remarkable talent and creativity of the Linux development community as well as the value of the cooperative development model. Please see Appendix F, *Red Hat and Freely Distributable Software*, for more on this subject.

Since the release of version 4.0, Red Hat Linux runs on three leading computing platforms: Intel compatibles PCs, Digital Alpha computers, and Sun SPARC equipment. Our unified source tree and the benefits of RPM (Red Hat Package Management) technology enable us to keep Red Hat Linux releases for these three platforms concurrent, which in turn enables our users to manage and port software between these platforms with minimum effort.

We are pleased to note the growing number of applications being made available on the Linux operating system, several of these from Red Hat Software. We make our versions of the Linux OS available by unrestricted FTP from our site and many mirror sites on the Internet. Red Hat Linux is also available on CD-ROM. For current information on our product offerings and links to other Linux resources please check Red Hat Software's web site at `http://www.redhat.com`.

On most systems, Red Hat Linux is easy to install: the installation program can walk you through the process in as little as 15 minutes. The system itself is very flexible: you can install and uninstall individual software "packages" with only a few mouse clicks. It is easy to maintain: package installations can be verified and corrected, and packages can be installed and uninstalled simply and reliably. Furthermore, Red Hat Linux is easy to administer: a set of graphical administration tools bundled into a "control-panel" reduce the hassle of everyday system administration. Complete source code is provided for the freely distributable components of the system.

An Overview of This Book

This manual is organized in the following fashion:

Chapter 1, *Quick Start* contains enough information for experienced users to perform a basic installation of Red Hat Linux/Intel without much trouble. We recommend that new Linux users read through the manual before installing to get a good understanding of the system.

Chapter 2, *Installing Red Hat Linux/Intel* contains detailed instructions for installing Red Hat Linux on an Intel or compatible PC.

Chapter 3, *Installing Red Hat Linux/Alpha* contains instructions for installing Red Hat Linux on a Digital Alpha.

Chapter 4, *Installing Red Hat Linux/SPARC* contains instructions for installing Red Hat Linux on a Sun SPARC machine.

Chapters 5 – 10 explain how to configure various services once you have installed Red Hat Linux, how to find documentation on your system, and how to use the various system management and administration tools which accompany Red Hat Linux. They also include an explanation of what's special about your Red Hat Linux system, including where special files live and how to shut down Red Hat Linux.

Appendices contain extra information about Red Hat Linux, including a manual for the MetroLink X server included with Red Hat Linux/Intel, the Red Hat Linux Frequently Asked Questions (FAQ), etc.

Thanks and Acknowledgments

We would like to thank all our beta testers for entrusting their systems to early versions of Red Hat Linux and for taking the time to submit bug reports from the front, especially those who have been with Red Hat since the "Halloween" release and earlier. We would also like to thank Linus Torvalds and the hundreds of developers around the world for creating, truly, one of the wonders of distributed development.

Thanks again, and welcome to the wonderful world of Linux!

The Red Hat Development Team

Chapter 1

Quick Start

This chapter summarizes how to install or upgrade Red Hat Linux/Intel; it is intended for users who are already experienced at installing Linux. If this chapter does not provide enough information, please read Chapter 2, *Installing Red Hat Linux/Intel*.

1.1 The Red Hat Linux Errata

Occasionally, we discover problems with a release of Red Hat Linux, some of which may affect installation. When we discover a problem, we post information about the problem and our solution in the Red Hat Linux Errata, which we make available in several ways:

- via the World Wide Web: `http://www.redhat.com/errata`
- via electronic mail: `errata@redhat.com`

Please note that the solution to a problem listed in the Errata may require that you use a different boot or install disk image than the ones which are mentioned below or which come with your Red Hat Linux CD-ROM.

1.2 Getting Red Hat Linux

There are several ways to get Red Hat Linux; the easiest way is on CD-ROM. This may not be practical if you don't have a CD-ROM drive, so we have made it freely available via FTP; you can download it from any of the Red Hat mirrors listed in Appendix E, *Red Hat Linux FAQ*.

These mirror sites are provided for those who want to install via FTP or want package upgrades. It is possible to download the distribution to a hard drive and install from there, but this is highly impractical for most and is beyond the scope of this document.

To purchase Red Hat Linux on CD-ROM, please contact Red Hat Software at `http://www.redhat.com`.

1.3 Making Floppies

If you have the boot and supplemental diskettes shipped with Official Red Hat Linux 4.2, please skip to Section 1.4 below.

If your version of Red Hat Linux/Intel did not include a boot diskette and supplemental diskette, you need to create them. If there is a README file on your CD-ROM (or on the FTP site), please read it now; it contains the latest information on Red Hat Linux 4.2, some of which may be different from what you are about to read.

The boot and supplemental disk images are in the `images` directory on the CD-ROM and the FTP site. If you are doing a CD-ROM or NFS installation you will only need to make a boot diskette. All other installation methods require the supplemental diskette as well. You can use `dd` under Linux or `rawrite.exe` (found in the `dosutils` directory) under DOS to write each disk image file to a formatted high density (1.44 MB) 3.5 inch floppy.

1.4 Installing

To install Red Hat Linux, insert the boot floppy into drive A, and insert the CD into your CD-ROM drive (for a CD-ROM install) or connect to your network (for an FTP or NFS install). Then reboot your machine and answer all questions as they are presented. If you are installing from PCMCIA, FTP or Hard Disk, insert the supplemental disk when prompted.

1.5 Upgrading from a Prior Version

The installation process for Red Hat Linux 4.2 includes the ability to upgrade from prior versions of Red Hat Linux (2.0, 2.1, 3.0.3, 4.0, and 4.1) which are based on RPM technology. Upgrading your system installs the modular 2.0.x kernel as well as updated versions of the packages which are installed on your machine. The upgrade process preserves existing configuration files using a `.rpmsave` extenstion (e.g., `sendmail.cf.rpmsave`) and leaves a log telling what actions it took in `/tmp/upgradelog`. As software evolves, configuration file formats can change, so you should carefully compare your original configuration files to the new files before integrating your changes.

To upgrade a Red Hat Linux system, you must use a boot floppy (and possibly a supplemental floppy), just as if you were performing a full install. After selecting the medium to install from (and setting up networking information if it's required), the installation procedure prompts you to either **Install** or **Upgrade**; select **Upgrade**. This upgrade procedure is the functional equivalent of running the `upgrade` script in prior versions of Red Hat Linux.

Please note that some upgraded packages may "depend" on other packages which may not be installed on your system. The upgrade procedure takes care of these *dependencies*, but it may need to install additional packages in order to satisfy them.

1.6 Why Can't I Run Directly off the CD?

Some earlier versions of Red Hat Linux came with a "live filesystem" on the CD-ROM; this allowed users to run Red Hat Linux directly off the CD-ROM without first installing it on their hard drive.

Unfortunately, with the modular Linux-2.0.x kernel, adding "live" boot capabilities has become too complex and is no longer supported. We apologize for the inconvenience.

Chapter 2

Installing Red Hat Linux/Intel

This chapter explains in detail how to install or upgrade Red Hat Linux on Intel and compatible PCs. This chapter is organized as follows:

Section 2.1, ***The Red Hat Linux Errata*****,** contains important pointers to the most current information about Red Hat Linux/Intel 4.2.

Section 2.2, ***Before You Begin*****,** describes some of the things you need to know before you actually install Red Hat Linux.

Section 2.3, ***Preparing to Install*****,** tells how to create installation floppies (if were not included with your copy of Red Hat Linux) and explains how to use the installation program.

Section 2.4, ***Step-By-Step Installation*****,** gives step-by-step instructions for completing the entire installation or upgrade process.

2.1 The Red Hat Linux Errata

Occasionally, we discover problems with a release of Red Hat Linux, some of which may affect installation. When we discover a problem, we post information about the problem and our solution in the Red Hat Linux Errata, which we make available in several ways:

- via the World Wide Web: `http://www.redhat.com/errata`
- via electronic mail: `errata@redhat.com`

Please note that the solution to a problem listed in the Errata may require that you use a different boot or install disk image than the ones which are mentioned below or which come with your Red Hat Linux CD-ROM.

2.2 Before You Begin

Before you begin installing Red Hat Linux, you should collect some information about your system:

- Be familiar with the hardware you have installed in your machine, including:
  - **hard drive(s)**: number and size; if you have more than one, it's helpful to know which one is first, second, etc.
  - **memory**: amount of RAM.
  - **CD-ROM**: its interface type (IDE, SCSI, or other interface), and, for non-IDE, non-SCSI CD-ROMs, the make and model number.
  - **SCSI adapter**: make and model number.
  - **network card**: make and model number.
  - **mouse**: type (serial, PS/2, or bus mouse), protocol (Logitech, Microsoft, MouseMan, etc.), and number of buttons; also, for serial mice, the com port it is connected to.

  Also, if you will be installing the X Window System, you should be familiar with the following:

  - **your video card**: make and model number or video chipset, amount of video RAM.

  - **your monitor**: make and model number, allowable range of horizontal and vertical refresh rates.

  You can find most of the above information in the documentation accompanying your system or from your system's vendor or manufacturer.

  Please Note: Current information about what hardware is supported by Red Hat Linux/Intel is available via Red Hat Software's World Wide Web site at `http://www.redhat.com/hardware`.

- If you will be connected to a network, be sure you know your IP address, netmask, gateway IP address, name server IP addresses, domain name, and hostname. If you don't know these values, ask your network administrator.

- If Red Hat Linux/Intel will coexist on your machine with OS/2, you must create your disk partitions with the OS/2 partitioning software—otherwise, OS/2 may not recognize the disk partitions. During the installation, do not create any new partitions, but do set the proper partition types for your Linux partitions using the Linux `fdisk`.

- If you plan to use LILO (the LInux LOader) to boot your Red Hat Linux system, see Section 2.4.19, *LILO Installation*, for guidelines that might affect your partitioning plans.

Please read all of the installation instructions *before* starting; this will prepare you for any decisions you need to make and should eliminate potential surprises.

2.2.1 Installation Methods

You can install or upgrade Red Hat Linux/Intel via any of several basic methods. Depending on the method you use, you need either one or two formatted high-density (1.44 MB) 3.5-inch diskettes.

Installing from CD-ROM or via NFS requires only a boot diskette. Installing from a hard drive, via FTP, from an SMB volume, or from a PCMCIA device (including PCMCIA-based CD-ROMs) requires both a boot diskette and a supplemental diskette. Section 2.3.1 below explains how to create boot and supplemental diskettes.

CD-ROM If you have a Red Hat Linux CD and a boot diskette you will need a supported CD-ROM drive and either a 3.5 inch floppy drive or an operational installation of MS-DOS on your machine in order to run the installation utility. If a boot diskette did not accompany your CD, you will need access to a computer running either Linux or MS-DOS to create a boot diskette from the CD.

NFS If you wish to install over a network, you will need to mount the Red Hat Linux CD-ROM on a machine that supports ISO-9660 file systems with Rock Ridge extensions. The machine must also support NFS. Export the CD-ROM file system via NFS. You will need to have nameservices configured, or know the NFS server's IP address, and the path to the exported CD-ROM.

FTP For an FTP install, you must have a boot disk and supplemental disk. You will need to have a valid nameserver configured or the IP address of the FTP server you will be using. You will also need the path to the root of the Red Hat Linux directory on the FTP site.

SMB Shared Volume If you wish to install from an SMB shared volume, you will need to mount the Red Hat Linux CD-ROM on a Microsoft Windows NT or Windows 95 server that supports shared volumes. You will need to have nameservices configured, or know the server's IP address; you will also need the name of the shared volume containing the Red Hat Linux CD-ROM and the account number and password to use to access the volume.

Hard Drive To install Red Hat Linux from a hard drive you will need the same boot and supplemental disks used by the FTP install. You must first create a `RedHat` directory at the top level of your directory tree. Everything you will install should be placed in that directory. First copy the `base` subdirectory, then copy the packages you want to install to another subdirectory called `RPMS`. You can use available space on an existing DOS partition or a Linux partition that is not required in the install procedure (for example, a partition that would be used for data storage on the installed system).

If you are using a DOS filesystem, you may not be able to use the full Linux filenames for the `RPM` packages. The installation process does not care what the filenames look like, but it is a good idea that you keep track of them so you will know what you are installing.

PCMCIA If your CD-ROM, ethernet card, or local hard disk is connected to a PCMCIA adapter, you must install with PCMCIA support. You need a supported PCMCIA controller and a supported PCMCIA SCSI adapter or ethernet card. Installing via PCMCIA requires the use of a supplemental diskette.

2.2.2 A Note About Kernel Drivers

During installation of Red Hat Linux, there are some limits placed on the filesystems and other drivers supported by the kernel. However, after installation there is support for all file systems available under Linux. At install time the modularized kernel has support for (E)IDE devices, (including ATAPI CD-ROM drives), SCSI adapters, and network cards. Additionally, all mice, SLIP, CSLIP, PPP, PLIP, FPU emulation, console selection, ELF, SysV IPC, IP forwarding, firewalling and accounting, reverse ARP, QIC tape and parallel printers, are supported.

After the installation is complete you may want to rebuild a kernel that includes support only for your hardware. See Chapter 5, Section 5.1 for information on how to build a customized kernel.

2.2.3 Disk Partitions

To install Red Hat Linux, you must have disk space available for it to “live” in. This disk space needs to be separate from the disk space used by other operating systems you may have installed on your computer (e.g., MS-DOS, OS/2, a different version of Linux, or another well-known operating system which we won’t name). The way to make disk space available is by dividing it into *partitions*.

You may wish to install Red Hat Linux on its own hard disk, or even on a computer which contains no other operating system. In that case, you can use the Red Hat Linux installation system to create the disk partitions you need.

Alternatively, you may wish to install Red Hat Linux on a disk which already contains software or data from a different operating system; how to create the disk partitions you need depends on what other operating system is present.

Disk Partitions and MS-DOS

Most MS-DOS systems, as they come from the vendor, have one hard disk which contains one large partition. In order to install Red Hat Linux, you need to make that partition smaller, and then create a partition for Red Hat Linux in the space that's left. There are two ways to accomplish this:

Destructive Repartitioning This procedure destroys any data in the disk partition(s) you are resizing; you should make a complete, reliable backup of everything you wish to keep from the disk you are repartitioning. Then, use the MS-DOS disk partitioning utility, called `fdisk`, to delete the large partition and create a smaller MS-DOS partition. Restore your data to the new partition from your backup.

Non-Destructive Repartitioning This procedure is not supposed to destroy data in the disk partition(s) you are resizing; however, we recommend making reliable backups of data you wish to keep in any case. You can use the `fips` utility, included on the Red Hat Linux CD in the `dosutils` directory, to resize your MS-DOS partition; we highly recommend reading the `fips` documentation, located the the `fipsdocs` subdirectory, before doing so.

After repartitioning, you can use the Red Hat Linux installation system to create partitions for your Red Hat Linux system as described in *Step-By-Step Installation*.

Disk Partitions and Other Operating Systems

Information on creating disk partitions to be compatible with other operating systems is available in several HOWTOs and Mini-HOWTOs, available on the Red Hat Linux CD in the `doc/HOWTO` and `doc/HOWTO/mini` directories.

How Many Partitions?

If you wish to, you can install Red Hat Linux in a single large partition. However, we recommend the following (**Please Note:** If you plan to install all the software packages accompanying Red Hat Linux, you may need to use even larger partitions):

A swap partition for virtual memory. If your computer has 16 MB of memory or less, you must create a *swap partition*; even if you have more memory, a swap partition is recommended. The size of your swap partition should be at least 16 MB or the same as the amount of memory in your computer, whichever is larger.

A root partition to be mounted as `/` (the root directory) when your Red Hat Linux system boots; it only needs to contain things necessary to boot your system, as well as system configuration files. 50–80 MB works well for most systems.

A `/usr` partition, where much of the software on a Red Hat Linux system lives; this partition should be 200–500 MB, depending on how many packages you plan to install.

A `/home` partition, where users' *home directories* go; the size of `/home` depends mostly on how many users you plan to have on your Red Hat Linux system and what they might store in their home directories.

Additionally, you may wish to create any of the following:

A `/tmp` partition for temporary files. This is a good idea for larger, multiuser systems or network server machines.

A `/usr/src` partition if you wish to install a large amount of the source code included with Red Hat Linux.

A `/usr/local` partition to hold things you wish to keep separate from the rest of your Red Hat Linux system, such as software that is not available as an RPM package.

2.3 Preparing to Install

2.3.1 Creating the Boot and Supplemental Floppies

Your Red Hat Linux/Intel CD set may include a boot diskette and supplemental diskette. If so, you may skip to Section 2.3.2. If not, you must create them.

You need only a boot diskette if you are installing Red Hat Linux from CD-ROM or via NFS. You also need a supplemental diskette if you are installing from a hard disk, via FTP, or from a PCMCIA device.

To create the boot and supplemental floppies, you need two blank, formatted, high-density (1.44 MB), 3.5-inch diskettes. The diskette *images* are located in the `images` directory on the Red Hat Linux CD or the FTP site where you got Red Hat Linux. Each image is a "snapshot" of the appropriate floppy diskette, which you need to transfer onto an actual floppy; you can do this using either MS-DOS or Linux.

Making Floppies Under MS-DOS

To make floppies under MS-DOS, use the `rawrite` utility included on the Red Hat Linux CD in the `dosutils` directory. First, label a formatted 3.5-inch floppy "Boot Diskette" or something similar and insert it into the floppy drive. Then, use the following commands (assuming your CD is drive `d:`):

```
d:
cd \images
\dosutils\rawrite.exe
```

`rawrite` first asks you for the filename of a diskette image; enter `boot.img`. Then it asks for a diskette drive to write the image to; enter `a:`. If you need to make a supplemental floppy, label a second diskette and then run `rawrite` again, using `supp.img`.

Making Floppies Under Linux

To make floppies under Linux, you must have permission to write to `/dev/fd0` (the 3.5-inch floppy drive). Label a blank, formatted diskette "Boot Diskette" or something similar and insert it into the floppy drive (but don't mount it). After mounting the Red Hat Linux CD, change to the `images` directory and use the following command:

```
dd if=boot.img of=/dev/fd0 bs=1440k
```

If you need to make a supplemental floppy, label a second diskette, insert it in the floppy drive, and then use the following command:

```
dd if=supp.img of=/dev/fd0 bs=1440k
```

Installing Without Using a Boot Floppy

Alternatively, if you have MS-DOS installed on your computer, you can boot the installation system directly from the CD without making floppies. To do this, use the following commands (assuming your CD is drive `d:`):

```
d:
cd \dosutils
autoboot.bat
```

2.3.2 A Note About Virtual Consoles

The Red Hat Linux installation system is more than simply the dialog boxes it presents as it guides you through the installation procedure. In fact, the installation system presents several different kinds of diagnostic messages while running, and it gives you a means to enter commands from a shell prompt. It presents the installation dialogs, shell prompt, and messages on five *virtual consoles* which you can switch between using a single keystroke. These virtual consoles can be very helpful if you encounter a problem while installing Red Hat Linux; messages displayed on the install log or system log can help to pinpoint the problem. Please see Figure 2.1 for a listing of the virtual consoles, the keystrokes to switch to them, and their contents.

Console	Keystroke	Contents
1	Alt-F1	installation dialog
2	Alt-F2	shell prompt
3	Alt-F3	install log (messages from install program)
4	Alt-F4	system log (messages from kernel, etc.)
5	Alt-F5	other messages

Figure 2.1: Virtual Console Information

2.3.3 Using the Dialog Boxes

You can navigate around the installation dialogs using a simple set of keystrokes. In most dialog boxes there is a *cursor* or *highlight* which you can move using the arrow keys (←, →, ↑, ↓), Tab, and Alt-Tab. A summary of movement commands is presented at the bottom of each screen.

To "press" a button (such as **OK**), Tab to the button and press Space. To select an item from a list of items, move the highlight to the item you wish to select and press Enter. To select any number of items from a *checklist*, move the highlight to the item you wish to select and press Space to select an item. To deselect an item, press Space a second time.

Pressing F12 accepts the current values and proceeds to the next dialog; it is usually equivalent to pressing the **OK** button.

Please Note: Do not press random keys during the installation process; it may result in unpredictable behavior.

2.4 Step-By-Step Installation

2.4.1 Starting the Installation System

Booting

To begin installing Red Hat Linux, insert the boot floppy you just created into the floppy drive and reboot the machine (if you have MS-DOS installed on your machine, you can start installing directly from the CD-ROM; see Section 2.3.1 above). The `boot:` prompt appears; press Enter to boot. Watch the boot messages to see whether the Linux kernel detects your hardware. If it does not properly detect your hardware, you may need to reboot and enter some options at the `boot:` prompt, e.g.:

```
boot: linux hdc=cdrom
```

Press Enter to boot using those options. If you do need to specify some boot options, please make note of them; they will be needed later.

Please Note: Because Red Hat Linux supports installation on many different types of hardware, many drivers (including those for SCSI adapters, network cards, and many CD-ROMs) are not built into the Linux kernel used during installation; rather, they are available as *modules* and loaded as you need them during the installation process. You will have the chance to specify options for such drivers at the time they are loaded, and in fact these drivers will ignore any options you specify for them at the `boot:` prompt. Please see Section 2.2.2 above for more information about drivers built into the installation system.

Beginning the Installation

After booting, the installation process begins by asking if you are using a color monitor (see Figure 2.2). **Yes** is the default; if you can't distinguish a cursor or highlight press Tab once, and then Enter to proceed. Next it presents a welcome message; press Enter to begin the installation. If you wish to abort the installation process, you may simply eject the boot floppy now and reboot your machine.

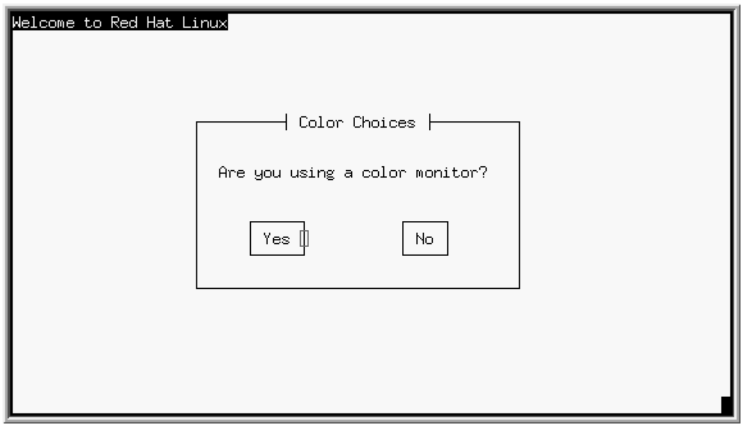


Figure 2.2: Beginning the Installation

2.4.2 Selecting a Keymap

After the welcome dialog, the installation program gives you an opportunity to select a keyboard map (see Figure 2.3). Select the keymap you wish to use on your Red Hat Linux system and press Enter; the keymap you select will be loaded automatically both for the remainder of the installation process and each time you boot your Red Hat Linux system.

If you wish to change your keymap after you have booted your Red Hat Linux system, you may use the `/usr/sbin/kbdconfig` command.

Figure 2.3: Selecting a Keymap

2.4.3 PCMCIA Support

Next, the installation program asks you whether you need PCMCIA support. If you are using a PCMCIA device for installation (for example, a PCMCIA ethernet card for installing via NFS, or PCMCIA SCSI card for installing from CD) you should answer **Yes**; the installation program prompts you to insert the supplemental diskette. You should wait until the next dialog appears before inserting your PCMCIA card.

2.4.4 Selecting an Installation Method

You can install Red Hat Linux via any of four basic methods (see Section 2.2.1, *Installation Methods*), some of which require the use of a supplemental diskette. To summarize:

CD-ROM If you have a CD-ROM drive with the Red Hat Linux CD; does not require a supplemental diskette.

NFS If you are installing from an NFS server which is exporting either the Red Hat Linux CD or a mirror of Red Hat Linux from the Red Hat FTP site; does not require a supplemental diskette.

Hard Drive If you copied the Red Hat Linux files to one of your local hard drives; requires a supplemental diskette.

FTP If you are installing directly from an FTP server; requires a supplemental diskette.

SMB image If you are installing from a Microsoft Windows NT or Windows 95 shared volume; requires a supplemental diskette.

The next dialog asks what type of installation method you wish to use (see Figure 2.4). Highlight the appropriate choice and select **OK**, or press Enter. If you select **FTP**, or **hard drive**, and you're not using PCMCIA support, the installation program prompts you to insert the supplemental diskette.

CD-ROM Type

If you choose to install Red Hat Linux/Intel from CD-ROM, the installation program prompts you for the type of CD-ROM:

IDE (ATAPI) Most CD-ROMs available for Intel-based PCs are IDE (also known as ATAPI) CD-ROMs. The install program checks for a CD-ROM on all available IDE devices before giving up.

SCSI Select this if your CD-ROM is attached to a supported SCSI adapter; the installation program asks you to choose a SCSI driver. Choose the driver that most closely resembles your adapter. You may specify options for the driver if necessary; however, most drivers detect your SCSI adapter automatically.

Other If your CD-ROM is neither an IDE nor a SCSI CD-ROM, it's an "other". The installation program presents a list of drivers for supported CD-ROMs; choose a driver and, if necessary, specify any driver options.

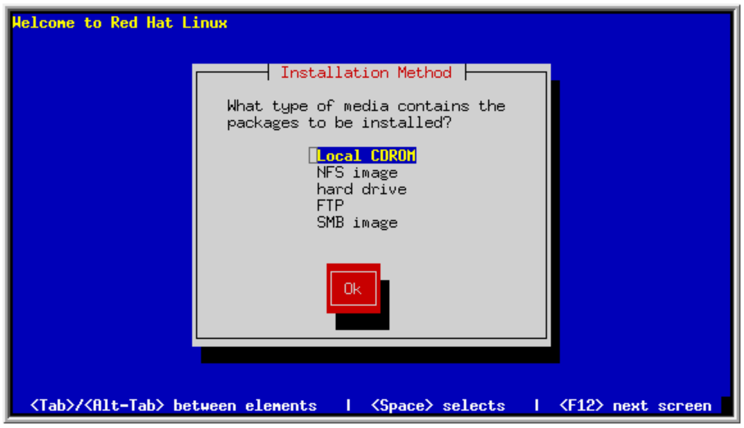


Figure 2.4: Selecting an Installation Method

2.4.5 If You Are Installing via NFS

If you are not installing Red Hat Linux via NFS, you may skip to Section 2.4.6 below.

Choosing a Network Driver

If you are installing via NFS, you must have a supported network card installed in your computer. The installation program prompts you to choose the driver that supports your network card and to specify any options necessary for the driver to locate and recognize the card. Most of the time, the driver can locate the card automatically by *probing*; however, it is possible that probing your hardware might cause your machine to hang. If this happens, reboot the machine, start the installation again, and specify additional driver options when you get to this step (see Appendix A).

Configuring TCP/IP Networking

After the installation program has found your network card, it presents several dialogs for configuring TCP/IP networking. This information is only used during the installation process; you will be able to configure permanent networking information later. This feature allows greater flexibility if you want to install from a network other than the one where the machine will "live". Figure 2.5 contains example networking information.

Field	**Example Value**
IP Address	10.0.2.15
Netmask	255.255.255.0
Default Gateway	10.0.2.254
Primary Nameserver	10.0.2.1
Domain Name	redhat.com
Hostname	pooh.redhat.com

Figure 2.5: Sample Networking Information

The first dialog asks you for IP and other network addresses (see Figure 2.6). Enter the **IP address** you are using during installation and press Enter. The installation program attempts to guess your **Netmask** based on your IP address; you may change the netmask if it is incorrect. Press Enter. The installation program guesses the **Default gateway** and **Primary nameserver** addresses from your IP address and netmask; you may change them if they are incorrect. Choose **OK** to continue.

The second dialog prompts you for a domain name, a hostname, and other networking information (see Figure 2.7). Enter the **Domain name** for your system and press Enter; the installation program carries the domain name down to the **Host name** field. Enter the hostname you are using in front of the domain name to form a fully qualified domain name (FQDN). If your network has more than one nameserver, you may enter IP addresses for additional nameservers in the **Secondary nameserver** and **Tertiary nameserver** fields. Choose **OK** to continue.

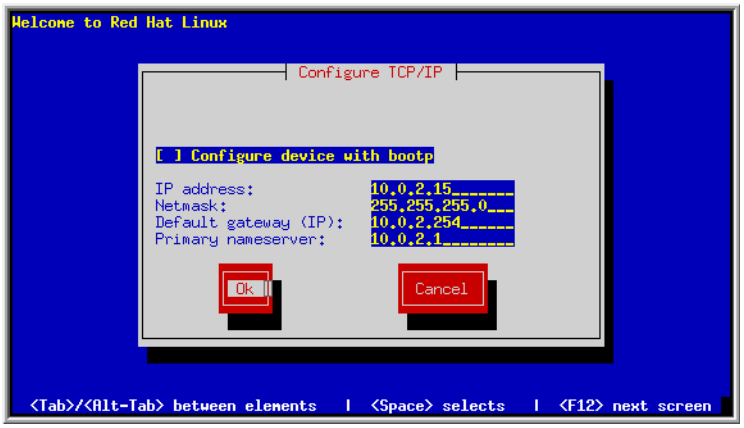


Figure 2.6: Configuring TCP/IP

NFS Server Information

The next dialog requests information about the NFS server (see Figure 2.8). Enter the name or IP address of your NFS server, and the name of the exported directory that contains the Red Hat Linux CD. For example, if the NFS server has the Red Hat Linux CD mounted on `/mnt/cdrom`, enter `/mnt/cdrom` in **Red Hat directory**. If the NFS server is exporting a mirror of the Red Hat Linux installation tree instead of a CD, enter the directory which contains the `RedHat` directory. For example, if your NFS server contains the directory `/mirrors/redhat/i386/RedHat`, enter `/mirrors/redhat/i386`.

2.4.6 Upgrading or Installing

After you choose an installation method (and, if you're installing via NFS or SMB volume, after you configure networking information), the installation program prompts you to either *install* or *upgrade* (see Figure 2.9).

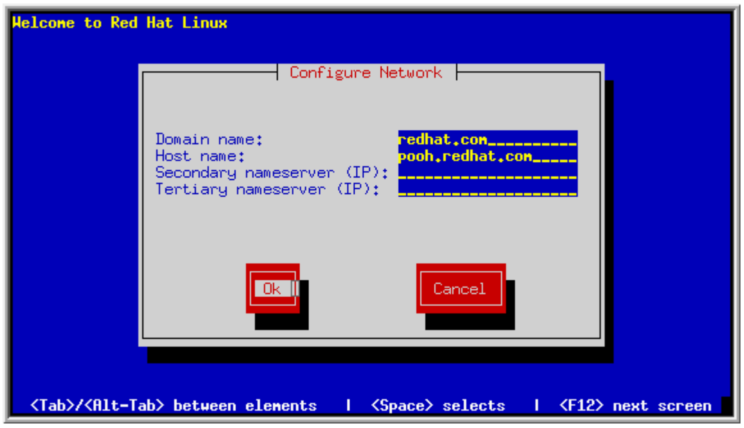


Figure 2.7: Configuring Networking

Installing

You usually install Red Hat Linux on a clean disk partition or set of partitions, or over another installation of Linux. Installing Red Hat Linux over another installation of Linux (including Red Hat Linux) does not preserve any of the prior installation.

If you wish to perform a full install, choose **Install**.

Upgrading

The installation process for Red Hat Linux 4.2 includes the ability to upgrade from prior versions of Red Hat Linux (2.0, 2.1, 3.0.3, 4.0, and 4.1) which are based on RPM technology. Upgrading your system installs the modular 2.0.x kernel as well as updated versions of the packages which are installed on your machine. The upgrade process preserves existing configuration files using a `.rpmsave` extenstion (e.g., `sendmail.cf.rpmsave`) and leaves a log telling what actions it took in `/tmp/upgradelog`. As software evolves, configuration file formats can change, so you should carefully compare your original configuration files to the new files before integrating your changes.

If you wish to upgrade your Red Hat Linux/Intel system, choose **Upgrade**.

Figure 2.8: Installing via NFS

Please Note: Some upgraded packages may "depend" on other packages which may not be installed on your system. The upgrade procedure takes care of these *dependencies*, but it may need to install additional packages in order to satisfy them.

2.4.7 SCSI Support

After you choose whether to perform an upgrade or a full install, the installation program asks you whether you have any SCSI adapters. If you choose **Yes**, the following dialog presents a list of SCSI drivers. Choose the driver that most closely resembles your SCSI adapter. The installation program then gives you an opportunity to specify options for the SCSI driver you selected; most SCSI drivers detect your hardware automatically, however.

2.4.8 If You Are Installing via FTP or SMB

If you are not installing Red Hat Linux via FTP or via an SMB volume, you may skip to Section 2.4.9 below.

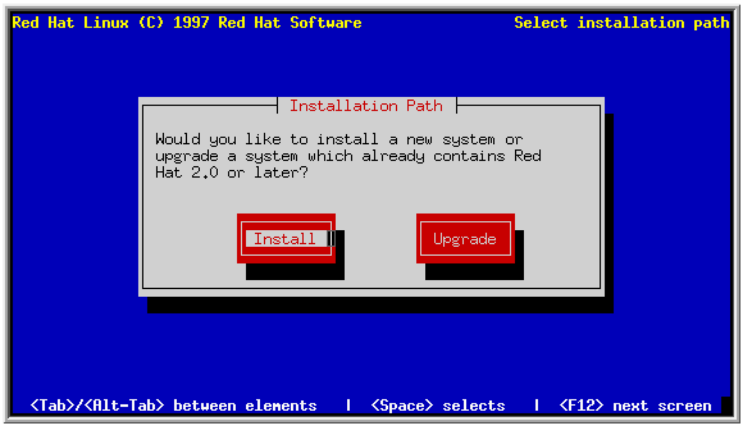


Figure 2.9: Upgrading or Installing

Choosing a Network Driver

If you are installing via FTP or via an SMB volume, you must have a supported network card installed in your computer. The installation program prompts you to choose the driver that supports your network card and to specify any options necessary for the driver to locate and recognize the card. Most of the time, the driver can locate the card automatically by *probing*; however, it is possible that probing your hardware might cause your machine to hang. If this happens, reboot the machine, start the installation again, and specify additional driver options when you get to this step (see Appendix A).

Configuring TCP/IP Networking

After the installation program has found your network card, it presents several dialogs for configuring TCP/IP networking. This information is only used during the installation process; you will be able to configure permanent networking information later. This feature allows greater flexibility if you want to install from a network other than the one where the machine will "live". Figure 2.5 contains example networking information.

The first dialog asks you for IP and other network addresses (see Figure 2.6). Enter the **IP address** you are using during installation and press

Enter. The installation program attempts to guess your **Netmask** based on your IP address; you may change the netmask if it is incorrect. Press Enter. The installation program guesses the **Default gateway** and **Primary nameserver** addresses from your IP address and netmask; you may change them if they are incorrect. Choose **OK** to continue.

The second dialog prompts you for a domain name, a hostname, and other networking information (see Figure 2.7). Enter the **Domain name** for your system and press Enter; the installation program carries the domain name down to the **Host name** field. Enter the hostname you are using in front of the domain name to form a fully qualified domain name (FQDN). If your network has more than one nameserver, you may enter IP addresses for additional nameservers in the **Secondary nameserver** and **Tertiary nameserver** fields. Choose **OK** to continue.

FTP Server Information

If you're installing via FTP, the next dialog requests information about the FTP server (see Figure 2.10). Enter the name or IP address of the FTP site you are installing from, and the name of the directory there which contains the `RedHat` directory for your architecture. For example, if the FTP site contains the directory `/pub/mirrors/redhat/i386/RedHat`, enter `/pub/mirrors/redhat/i386`. If you are not using anonymous FTP, or if you need to use a proxy FTP server (if you're behind a firewall, for example), another dialog requests the FTP account or proxy information.

SMB Server Information

If you're installing via SMB volume, the next dialog requests information about the SMB server (see Figure 2.11). Enter the name or IP address of your SMB server, the name of the shared volume which contains the Red Hat Linux CD, and the account name and password for the volume.

2.4.9 Creating Partitions for Red Hat Linux

After choosing whether to use SCSI support (and, if you're installing via FTP, after you configure networking information), the installation program presents a list of the hard disks it can find on your system and prompts you

Figure 2.10: Installing via FTP

to set aside at least one disk partition for Red Hat Linux. If you need to create one or more partitions, select the disk you wish to partition and press Enter; the installation program starts the Linux `fdisk` utility. Please see Section 2.2.3 for recommendations about partitions for Red Hat Linux.

Using `fdisk`

`fdisk` includes online help which is terse but useful. Here are a few tips:

- To list the current partition table, use the `p` command. To add a new partition, use `n`.
- Linux `fdisk` creates partitions of type `Linux native` by default. When you create a swap partition, don't forget to change it to type `Linux swap` using the `t` command.
- Linux allows up to four (4) partitions on one disk. If you wish to create more than that, one of the four may be an *extended* partition, which in turn contains one or more *logical* partitions. Since it acts as a container, the extended partition must span all of the space that the logical partitions will take up.
- It's a good idea to write down which partitions (e.g., `/dev/hda2`) are meant for which filesystems (e.g., `/usr`) as you create each one.

Figure 2.11: Installing via SMB volume

- **Please Note:** None of the changes you make take effect until you save them and exit `fdisk` using the `w` command. You may quit `fdisk` at any time without saving changes using the `q` command.

Numbering Partitions

Linux refers to disk partitions using a combination of letters and numbers which may be confusing to inexperienced users. Here is a summary:

The letters denote the device the partition is on, e.g., `/dev/hda` (the first IDE hard disk) or `/dev/sdb` (the second SCSI disk).

The number denotes the partition. The first four (primary or extended) partitions are numbered `1` through `4`. Logical partitions start at `5`. E.g., `/dev/hda3` is the third primary or extended partition on the first IDE hard disk; `/dev/sdb6` is the second logical partition on the second SCSI hard disk.

Changing the Partition Table

When you are finished partitioning your disks, press **Done**; you may see a message indicating that the installation program needs to reboot. This is a normal occurrence after changing a disk's partition data; it usually happens if you created, changed, or deleted any extended partitions. After you press **OK**, your machine will reboot. Follow the same installation steps you did up until **Partitioning Disks**; then simply choose **Done**.

2.4.10 Initializing Swap Space

After creating partitions for Red Hat Linux, the installation program looks for swap partitions. If it finds any, it asks whether you want to initialize them. Select the partition(s) you wish to initialize as swap space using Space; if you wish to check the partitions for bad blocks, make sure the **Check for bad blocks during format** box is checked. Choose **OK**.

If the installation program can't find your swap partition, make sure you have set the partition type to `Linux swap`; see Section 2.4.9 above.

2.4.11 Filesystem Configuration

Since Red Hat Linux may be distributed across several disk partitions, you need to choose which disk partitions go with which *filesystem*.

Selecting the Root Partition

If you are performing a full install, the installation program presents a list of the Linux partitions it can find and prompts you to choose a *root partition*. The root partition (or *root filesystem*) is the one which holds everything necessary to boot your Red Hat Linux system; it is mounted on / (the *root directory*) when your system boots. Select the root partition you wish to use and press Enter.

If you are performing an upgrade, the installation program tries to find your root partition automatically; if it finds one, it goes on to the next step.

Other Partitions

The next dialog presents a list of most of the disk partitions on your system (including MS-DOS and other partitions Linux can read), giving you the opportunity to assign other partitions to different parts of the filesystem. The partitions you assign will be automatically mounted when your Red Hat Linux system boots. Select the partition you wish to assign and press Enter (or choose **Edit**); then enter the *mount point* for that partition, e.g., `/usr` (see Figure 2.12).

Figure 2.12: Filesystem Configuration

Additionally, Red Hat Linux allows you to mount NFS volumes when your system boots; this allows directory trees to be shared across a network. Choose **Add NFS**; then enter the hostname of the NFS server, the path to the NFS volume, and the local mount point for that volume (see Figure 2.13).

Formatting Partitions

The next installation dialog presents a list of partitions to format. Most partitions should be formatted, especially if they are newly created; however, partitions such as `/home` or `/usr/local` do not need to be formatted if they contain data you wish to keep. Select each partition to format and press Space. If you wish to check for bad blocks while formatting each

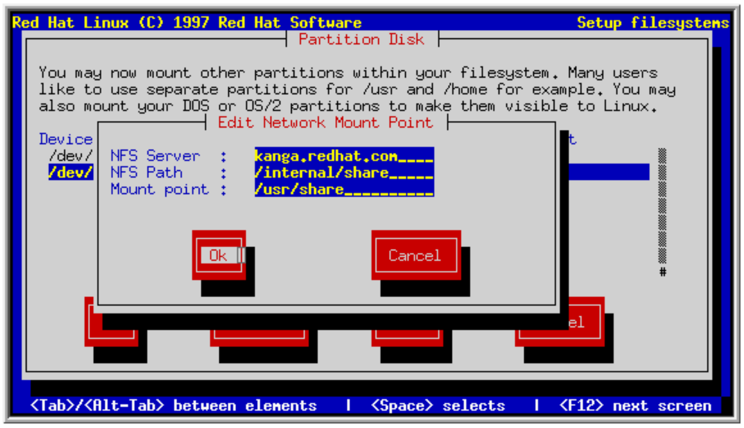


Figure 2.13: Adding an NFS Mount

filesystem, select **Check for bad blocks during format**. Press **OK**.

2.4.12 Selecting and Installing Packages

After your partitions are configured and formatted, you are ready to select packages to install. You can select *system components*, which group packages together according to function, select individual packages, or select a combination of the two.

Selecting System Components

System components group packages together according to the function your Red Hat Linux system should be able to perform: for example, **C Development**, **Networked Workstation**, or **Web Server**. Select each component you wish to install and press Space. Selecting **Everything** installs all packages included with Red Hat Linux (see Figure 2.14).

If you wish to select or deselect individual packages as well as system components, choose **Select individual packages**.

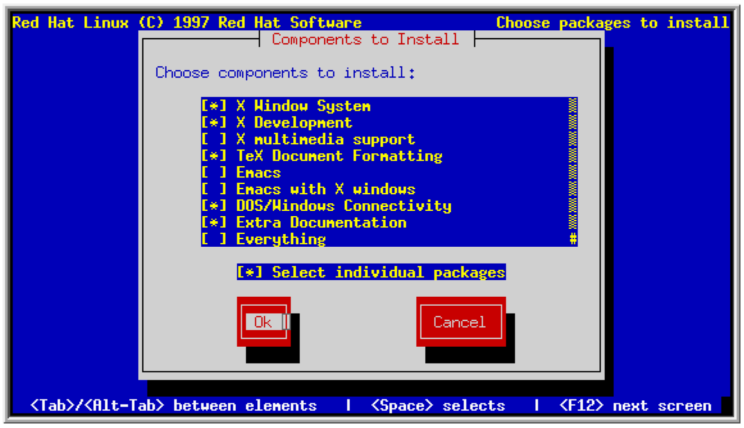


Figure 2.14: Selecting System Components

Selecting Individual Packages

After selecting the components you wish to install, you may select or deselect individual packages. The installation program presents a list of the package groups available; select a group to examine and press Enter. The installation program presents a list of the packages in that group, which you may select or deselect using Space. You may view a detailed description of a package by pressing F1 (see Figure 2.15). When you are finished selecting individual packages, press **OK** in the **Select Group** dialog.

Please Note: Some packages (such as the kernel and certain libraries) are required for every Red Hat Linux system and are not available to select or deselect.

Package Dependencies

Many software packages, in order to work correctly, depend on other software packages or libraries to be installed on your system. For example, many of the graphical Red Hat system administration tools require the `python` and `pythonlib` packages. To make sure your system has all the packages it needs in order to be fully functional, Red Hat Linux checks *dependencies* each time you install or uninstall software packages.

Figure 2.15: Selecting Packages

After you have finished selecting packages to install, the installation program checks the list of selected packages for dependencies. If any package requires another package which you have not selected to install, the program presents a list of these *unresolved dependencies* and gives you the opportunity to resolve them (see Figure 2.16). If you simply press **OK**, the program will resolve them automatically by adding all required packages to the list of selected packages.

Package Installation

After all package dependencies are resolved, the installation program begins to install packages; it leaves a complete log of what it installs in `/tmp/install.log` on your Red Hat Linux system.

2.4.13 Configuring a Mouse

To configure your mouse, choose the mouse type that best matches your mouse and press Enter. If you have a 2 button mouse and want to emulate the third (middle) button by clicking the left and right buttons at the same time, choose **Emulate 3 Buttons**.

If you choose a serial mouse, the next dialog prompts you for the serial

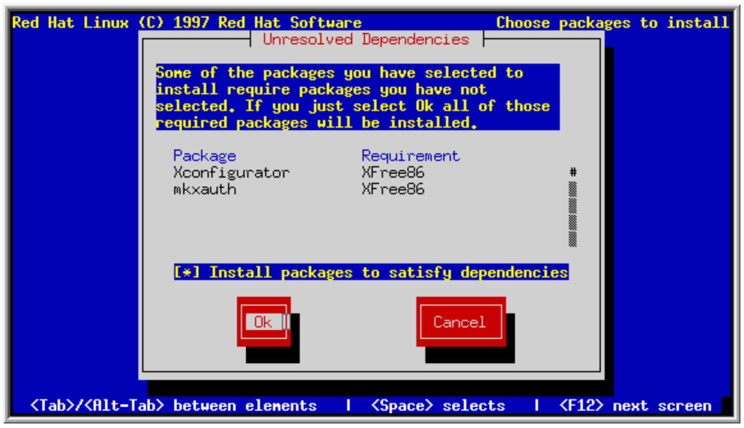


Figure 2.16: Unresolved Dependencies

(COM) port your mouse is connected to. Choose the correct port and press Enter.

Please Note: If your mouse is connected to a *PS/2 mouse port*, you should select **PS/2 Mouse**, no matter what kind of mouse it is.

If you wish to change your mouse configuration after you have booted your Red Hat Linux system, you may use the `/usr/sbin/mouseconfig` command.

2.4.14 Configuring X Windows

After setting up your mouse, if you installed the X Windows packages, you have the opportunity to configure your X server. If you did not choose to install the X Window System, you may skip to Section 2.4.15 below.

Configuring the Metro-X Server

If you are installing from an Official Red Hat Linux/Intel CD, you may install one copy of MetroLink's Metro-X server, a high-quality commercial X server for Linux. Please see the MetroLink license (paper copy) accompanying your Red Hat Linux CD for license information. If you choose to use Metro-X, the install program launches the Metro-X configuration utility `configX`; please see Chapter 5, Section 5.3 for more information on how to use `configX`.

Configuring an XFree86 Server

If you wish to use XFree86 (because Metro-X may not support your video card, because you have already installed Metro-X on another machine, because Metro-X was not included with your copy of Red Hat Linux/Intel, or because you simply prefer to use XFree86), the install program launches the `Xconfigurator` utility, which is described in detail in Chapter 5, Section 5.2.

`Xconfigurator` first presents a list of video cards. Select your video card from the list and press Enter. If your video card does not appear on the list, XFree86 may not support it; however, if you have technical knowledge about your card, you may choose **Unlisted Card** and attempt to configure it anyway.

Once you have selected your video card, the installation program installs the appropriate XFree86 server, and `Xconfigurator` presents a list of monitors. If your monitor appears on the list, select it and press Enter. Otherwise, select **Custom**. If you do select **Custom**, `Xconfigurator` prompts you to select the horizontal sync range and vertical sync range of your monitor (these values are generally available in the documentation which accompanies your monitor, or from your monitor's vendor or manufacturer).

Please Note: It is not recommended to select a monitor "similar" to your monitor unless you are certain that the monitor you are selecting does not exceed the capabilities of your monitor. If you do so, it is possible you may overclock your monitor and damage or destroy it.

Next, `Xconfigurator` prompts you for the amount of video memory installed on your video card. If you are not sure, please consult the documentation accompanying your video card. It ought not to damage your video card by choosing more memory than is available, but the XFree86 server may not start correctly if you do.

If the video card you selected might have a video clockchip, `Xconfigurator` presents a list of clockchips. The recommended choice is **No Clockchip Setting**, since XFree86 can automatically detect the proper clockchip in most cases.

Finally, `Xconfigurator` prompts you to select the video modes you wish to use; select one or more modes by pressing Space. `Xconfigurator` then writes a configuration file containing all of your choices to `/etc/X11/XF86Config`.

2.4.15 Configuring Networking

After assisting you in configuring an X server, the installation program gives you an opportunity to configure (or reconfigure) networking. If you are installing from CD-ROM or from a local hard disk, the installation program asks if you want to configure networking. If you choose **No**, your Red Hat Linux system will be a standalone workstation. If you choose **Yes**, you may configure networking as described below.

If you are installing Red Hat Linux via NFS or FTP, you will have already configured temporary networking information. The install program offers you three choices:

- Keep the setup you have been using to install with; all the networking information you entered previously becomes part of your system's permanent configuration.
- Reconfigure networking; the installation program presents the network configuration dialogs described below with the values you used during installation filled in as defaults. Choose this if your system will be installed on a network other than the one you used to install Red Hat Linux.
- Don't set up networking at all; your system will not have networking configured. Choose this if you installed your system over a network, but it will be used as a standalone workstation.

Network Configuration Dialogs

After the installation program has found your network card, it presents several dialogs for configuring your sytem's permanent TCP/IP networking. Figure 2.5 contains example networking information.

The first dialog asks you for IP and other network addresses (see Figure 2.6). Enter the **IP address** you are using during installation and press Enter. The installation program attempts to guess your **Netmask** based on your IP address; you may change the netmask if it is incorrect. Press Enter. The installation program guesses the **Default gateway** and **Primary nameserver** addresses from your IP address and netmask; you may change them if they are incorrect. Choose **OK** to continue.

The second dialog prompts you for a domain name, a hostname, and other networking information (see Figure 2.7). Enter the **Domain name** for your system and press Enter; the installation program carries the domain name down to the **Host name** field. Enter the hostname you are using in front of the domain name to form a fully qualified domain name (FQDN). If your network has more than one nameserver, you may enter IP addresses for additional nameservers in the **Secondary nameserver** and **Tertiary nameserver** fields. Choose **OK** to continue.

2.4.16 Configuring a Printer

After you have set up networking, the installation program asks whether you would like to configure a printer. If you choose **Yes**, a series of dialogs will assist you in configuring your printer. The following information applies to all printers:

- **Printer Connection**: Where the printer is connected—*local*, *remote*, or *LAN-Manager*.
  - **Local printer**: A *local* printer is connected directly to the system you're installing on.
  - **Remote printer**: A *remote lpd* printer is connected to another machine on the network which runs `lpd` (the print server used on Linux and many other flavors of Unix).
  - **LAN-Manager printer**: A *LAN-Manager* printer is connected to another machine on a LAN-Manager-style (SMB) network.
- **Queue**: The name of the printer queue you are creating.
- **Spool Dir**: The directory where print jobs are stored while they are being printed.

- **Type of Printer**: The make and model of your printer; if your printer doesn't appear on the list, you may have luck by choosing a printer make and model which is similar to yours.
- **Paper Size**: The size of paper your printer uses.
- **Resolution**: The resolution, in dots per inch, you wish your printer to use.
- **Color**: If you have a color printer, the color depth you wish your printer to use.

Local printers also require the following information:

- **Printer Device**: The printer port your printer is attached to.

Remote LPD printers also require the following information:

- **Remote Host**: The name or IP number of the host the remote printer is connected to.
- **Remote Queue**: The name of the remote printer queue.

LAN Manager printers also require the following information:

- **LAN Manager Host**: The name of the host the SMB printer is connected to.
- **LAN Manager IP**: The IP number of the host the SMB printer is connected to; this is optional.
- **Share Name**: The name of the shared SMB printer.
- **Username**: The username you need to use to access the SMB printer (typically `guest` for Windows servers, or `nobody` for `samba` servers).
- **Password**: The password you ened to use to access the SMB printer (typically blank).

When you have finished configuring a printer, the installation program will ask you to confirm the choices you made. After you confirm them, you may configure another printer, or you may continue with the installation.

2.4.17 Configuring the Clock

Next, the installation program presents a dialog to help you configure your Red Hat Linux system's timezone.

If you wish to set the hardware (CMOS) clock to GMT (Greenwich Mean Time, also known as UTC, or Coordinated Universal Time), select **Hardware clock set to GMT**. Setting your hardware clock to GMT means your Red Hat Linux system will properly handle daylight savings time, if your timezone uses it. Most networks use GMT.

Select the timezone your system will be operating in from the list, and press Enter.

If you wish to change your timezone configuration after you have booted your Red Hat Linux system, you may use the `/usr/sbin/timeconfig` command.

2.4.18 Setting a Root Password

After you have set up your system's timezone, the installation program prompts you to set a *root password* for your system; this is the password you will use to log into your Red Hat Linux system for the first time.

The root password must be at least six characters long; the password you type is not echoed to the screen. You must enter the password twice; if the two passwords do not match, the installation program will ask you to enter them again.

You ought to make the root password something you can remember, but not something that is easy for someone else to guess. Your name, your phone number, `qwerty`, `password`, `root`, `123456`, and `anteater` are all examples of poor passwords. Good passwords mix numerals with upper and lower case letters and do not contain dictionary words: `Aard387vark` or `420BMttNT`, for example. Remember that the password is case-sensitive. Write down this password and keep it in a secure place.

Please Note: The *root* user (also known as the *superuser*) has complete access to the entire system; for this reason, logging in as the root user is best done only to perform system maintenance or administration, so that critical system files are not inadvertently changed, moved, or deleted. Please see

Section 8.1 for instructions on how to add a user for yourself after you reboot your system.

2.4.19 Installing LILO

In order to be able to boot your Red Hat Linux system, you usually need to install LILO (the LInux LOader). You may install LILO in one of several places:

The Master Boot Record (MBR) is the recommended place to install LILO, unless another operating system loader (e.g., System Commander or OS/2's Boot Manager) is already installed there. When your machine boots, LILO starts and presents the `boot:` prompt; you can boot Red Hat Linux or any other operating system you configure LILO to boot (see below).

The first sector of your root partition is recommended if you are already using another boot loader on your system (such as OS/2's Boot Manager); then you can setup that boot loader to start LILO and boot Red Hat Linux.

The first sector of a floppy diskette is an alternative either of the above options; to boot Red Hat Linux, insert the LILO diskette into your machine's floppy drive and reboot. LILO starts from the floppy and presents the `boot:` prompt.

Select the location you wish to install LILO and press **OK**. If you do not wish to install LILO, press **Skip**.

Limitations of LILO

LILO is subject to some limitations imposed by most PC BIOSes. Specifically, most BIOSes can't access more than two hard drives and they can't access beyond cylinder 1023 (the 1024th cylinder) of any drive. Some more recent BIOSes do not have these limitations.

All the things which LILO needs to access at boot time (including the kernel) are located in the `/boot` directory of your root partition.

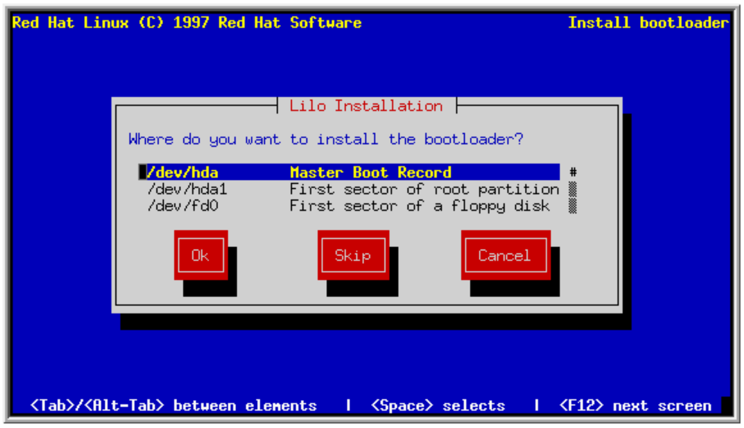


Figure 2.17: Installing LILO

Here are the "rules" of where `/boot` must live if you want to install LILO on a hard drive:

1. If you have 2 IDE (or EIDE) drives, `/boot` must live on one of them. This also includes any IDE CD-ROM drives on your primary controller. If you have one IDE hard drive, and one IDE CD-ROM on the primary controller, `/boot` must live on the hard drive.
2. If you have one IDE (or EIDE) drive and one or more SCSI drives, `/boot` must live either on the IDE drive or the SCSI drive that is at ID 0. No other SCSI IDs will work.
3. If you have SCSI only, `/boot` must live on a drive at ID 0 or ID 1. No other SCSI IDs will work.
4. In any case, `/boot` must live below cylinder 1023. For IDE (or EIDE) drives, this generally means below the 512 MB mark; for SCSI drives, it's generally below the 1 GB mark.

Adding Options to the LILO Boot Command Line

Finally, the installation program will ask if you wish to add default options to the LILO boot command. Any options you enter will be passed to the

Linux kernel every time it boots. If you have an LBA drive, check **Use linear mode**. Press **OK** when finished.

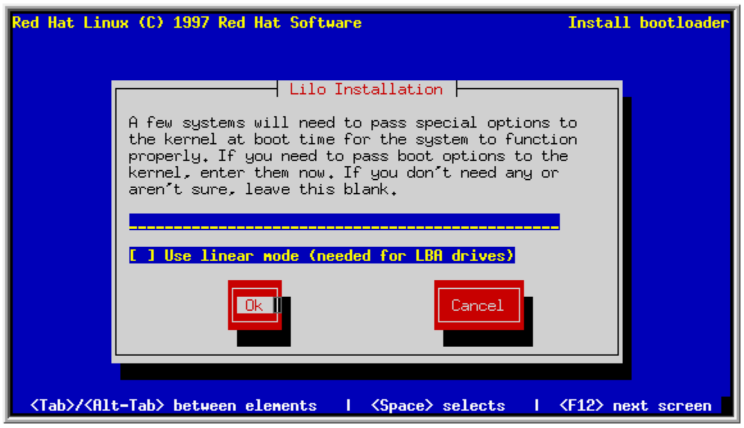


Figure 2.18: LILO options

Alternatives to LILO

If you do not wish to use LILO to boot your Red Hat Linux system, there are a few alternatives:

LOADLIN can load Linux from MS-DOS; unfortunately, it requires a copy of the Linux kernel (and an initial ram disk, if you have a SCSI adapter) to be available on an MS-DOS partition. The only way to accomplish this is to boot your Red Hat Linux system using some other method (e.g., from LILO on a floppy) and then copy the kernel an MS-DOS partition. `LOADLIN` is available from `ftp://sunsite.unc.edu/system/Linux-boot/` and `sunsite`'s various mirror sites.

SYSLINUX is an MS-DOS program very similar to `LOADLIN`; it is also available from `ftp://sunsite.unc.edu/system/Linux-boot/` and `sunsite`'s various mirror sites.

Some commercial bootloaders, such as `System Commander`, are able to boot Linux (but may still require LILO to be installed in your Linux root partition).

2.4.20 Finishing Up . . . or Starting Up

After you have completed LILO installation, the installation program will reboot your system. Please don't forget to remove the installation floppy from the diskette drive.

At the `boot:` prompt, press Enter and watch your Red Hat Linux system boot. When it presents the `login:` prompt, enter `root`; at the `Password:` prompt, type the root password you set for your machine in Section 2.4.18.

Chapter 3

Installing Red Hat Linux/Alpha

3.1 Introduction

These are brief installation instructions for the Red Hat Linux/Alpha 4.2 release for Digital Equipment Corporation's Alpha series of processors. This chapter describes the differences between installing Red Hat Linux/Alpha and Red Hat Linux/Intel. For further details on the installation procedure, please see Chapter 2, *Installing Red Hat Linux/Intel*.

3.2 Supported Hardware

Red Hat Linux/Alpha supports a variety of hardware based on the Alpha processor and the PCI bus. Platforms on which this release is known to work include:

1. AlphaPC64 (Cabriolet, Aspen Telluride)
2. AxpPCI33 (Noname)
3. EB64+ (Aspen Alpine)

4. EB66 (NekoTech Mach 1)
5. EB66+
6. Jensen (DEC PC 150, 2000 model 300, Cullean)
7. Universal Desktop Box (UDB, sometimes called Multia). This is just a noname board and is treated as such for the rest of this document.
8. AlphaStation 200, 250, 255, 400 (Avanti machines)
9. EB164 (Aspen Avalanche, Timberline, Summit; Microway Screamer)
10. Platform 2000 machines from Kinetics
11. PC164 machines (Durango)
12. Alcor type machines (AlphaStation 500, 600; Maverick, Brett)
13. Alpha-XL
14. Alpha-XLT (XL 300, XL 366)
15. Mikasa type machines (AlphaServer 1000), 1000A not supported)

All of those platforms except the Jensen include an NCR 810 SCSI controller, though BusLogic PCI SCSI controllers (other than the FlashPoint), the Adaptec AHA2940 SCSI controller, and Qlogic 1020 ISP controllers are also supported. The Jensen design uses an AHA1740 SCSI controller and is supported.

NE2000, DE422, and DE4x5 (PCI) Ethernet cards are supported. This includes the UDB's internal Ethernet hardware. Token ring support is also included in the kernels.

The X Window System should work on any machine with a S3 based video card except the Jensen machines. There is a server available from `ftp://ftp.azstarnet.com` for S3 cards in `/pub/linux/axp/jensen`. Most of the cards from Orchid and Number 9 will work. Most Diamond Stealth cards are also supported. Digital TGA cards (based on the DC21030 chip) are supported in 8 bit mode, and an X server for Mach64 cards is also provided.

More information on Alpha hardware is available in Appendix C.

3.3 Installation Overview

Installing Red Hat Linux on an Alpha system is slightly more complex than installing Red Hat Linux/Intel, mostly due to differences in machine architecture. For most systems, the main steps to a successful install are:

1. Install MILO, the Linux/Alpha miniloader, onto your machine (or onto a floppy disk); see Section 3.4, *Installing MILO.*
2. Create a kernel boot disk and a ramdisk from images available on the Red Hat CD; see Section 3.5, *Making Boot and Ramdisk Floppies.*
3. Use MILO to boot the appropriate Linux kernel; see Section 3.6.1, *Booting the Kernel Floppy.*
4. Load and run the Red Hat Linux/Alpha installation program; see Section 3.6.2, *Starting the Install.*

To perform these steps, you'll need three (3) blank, formatted 3.5-inch floppies.

3.4 Installing MILO

For most systems, the best method of booting Red Hat Linux/Alpha is using the MILO, the Linux/Alpha miniloader. You may either load MILO directly from your system's firmware, or use the ARC console to load MILO as an alternate "operating system". Details on these procedures are in the MILO HOWTO, which is available as Appendix D.

The full MILO tree is available on the Red Hat CD in the `milo/` directory. It contains the latest documentation and MILO images available when this CD was manufactured (later versions of MILO are regularly made available at `gatekeeper.dec.com`). The `milo/images/` directory also contains floppy disk images that can be used with the ARC console to boot directly into MILO. They contain MILO and `linload.exe`.

3.4.1 Making a MILO Floppy

To make a MILO floppy, you'll need to use either the `dd` command on a Linux (or Digital Unix) system, or the `rawrite.exe` program (on the Red Hat CD) on a Microsoft Windows NT or MS-DOS system. To use `dd`, first mount the CD (e.g., under `/mnt/cdrom/`), choose the appropriate image for your system from `milo/images/`, and:

```
cd /mnt/cdrom
dd if=milo/images/your-image.img of=/dev/fd0
```

To use `rawrite`, choose the appropriate image for your system, and (assuming your CD-ROM is drive `d:`):

```
d:
cd milo/images
\dosutils\rawrite.exe
```

`rawrite` first asks you for the name of a diskette image; enter the name of the MILO image for your system (e.g., `noname-udb.img`). Then it asks for a diskette drive to write the image to; enter `a:`.

Label the floppy "MILO floppy".

3.4.2 What if MILO doesn't work?

If MILO is not supported on your machine, you need to boot the floppy or the Red Hat CD directly from the SRM console. Information on doing this is available from the Red Hat Software web site at
`http://www.redhat.com/linux-info/alpha/faq`.

Please Note: If you are having trouble with MILO, the first thing to try is a newer MILO.

3.5 Making Boot and Ramdisk Floppies

3.5.1 Making a Boot Floppy

After you've made a MILO floppy, you need to make a boot floppy containing the appropriate Linux/Alpha kernel for your machine. The kernel images are on the Red Hat CD in the `images/` directory; also there is a `README` file containing more information about the kernel images.

Available boot images include:

1. `cab.img` (AlphaPC64, Cabriolet)
2. `noname.img` (AxpPCI33, Noname, Universal Desktop Box)
3. `eb64p.img` (EB64+, Aspen Timberlines)
4. `eb66.img` (EB66)
5. `eb66p.img` (EB66+)
6. `jensen.img` (Jensens)
7. `avanti.img` (AlphaStation 200, 250, 400)
8. `xl.img` (Alpha XL)
9. `xlt.img` (Alpha XTL)
10. `eb164.img` (EB164 based machines)
11. `p2000.img` (Platform 2000)
12. `pc164.img` (Alcor based machines)
13. `alcor.img` (Alcor based machines)
14. `mikasa.img` (Mikasa based machines)

All of these images should work on machines with either VGA or TGA based graphics boards.

To make a kernel floppy, use either `dd` or `rawrite`. For example, using `dd` (under Linux or Unix):

```
cd /mnt/cdrom
dd if=images/your-image.img of=/dev/fd0
```

Using `rawrite` (under Microsoft Windows NT or MS-DOS):

```
d:
cd images
\dosutils\rawrite.exe
```

`rawrite` first asks you for the name of a diskette image; enter the name of the kernel image for your system (e.g., `noname.img`). Then it asks for a diskette drive to write the image to; enter `a:`.

Label the floppy "Kernel floppy".

3.5.2 Making a Ramdisk Floppy

The ramdisk image is located on the Red Hat CD as `images/ramdisk.img`. You can use `dd` or `rawrite` to write the image to a diskette, much as for the MILO floppy and the boot floppy. Using `dd` (under Linux or Unix):

```
cd /mnt/cdrom
dd if=images/ramdisk.img of=/dev/fd0
```

Using `rawrite` (under Microsoft Windows NT or MS-DOS):

```
d:
cd images
\dosutils\rawrite.exe
```

`rawrite` first asks you for the name of a diskette image; enter `ramdisk.img`. Then it asks for a diskette drive to write the image to; enter `a:`.

Label the floppy "Ramdisk floppy".

3.6 Installing

3.6.1 Booting the Kernel Floppy

To boot the Linux/Alpha kernel from the diskette you made above, use MILO. See Appendix D, *Alpha Miniloader Howto*, Section D.5 for detailed information about how to load MILO from your MILO floppy. Boot to the `MILO>` prompt, insert your kernel floppy in the first floppy drive, and give MILO the following command:

```
boot fd0:vmlinux.gz root=/dev/fd0 load_ramdisk=1 prompt_ramdisk=1
```

MILO reads the Linux kernel from your boot disk and runs it.

If you are using the SRM console to load the kernel instead of MILO, you need to give the following arguments to the kernel:

```
load_ramdisk=1 prompt_ramdisk=1
```

3.6.2 Starting the Install

After the kernel boots, it prompts for a ramdisk. Insert the ramdisk floppy into the floppy drive and press Enter.

3.6.3 Issues Specific to Linux/Alpha

Disk Partitioning

Red Hat Linux/Alpha supports two varieties of hard drive partitioning:

- **Standard `fdisk`-style partitioning**: this has the usual oddities (such as extended partitions) and is done in the same way as on Intel platforms (using Linux `fdisk`. This is the partitioning style you must use if you are using the ARC console.
- **BSD-style disk-labelling**: It is normal to make the first partition on a disk-labelled hard drive the swap partition. We don't recommend using BSD disk-labelling unless it is absolutely necessary.

XFree86 and TGA

If you have an Alpha with a TGA card and you choose to install XFree86, the installation program prompts you to choose a video card. Choose **Unlisted Server**; the installation program displays a list of available servers. Choose the TGA server.

3.6.4 Finishing Up

After the install is finished and your system is fully configured, you will be asked to reset your computer. This indicates that your system has been successfully installed.

3.7 Booting your Red Hat Linux/Alpha system

To boot Red Hat Linux from MILO, you must use the `boot` command. The command differs slightly depending on where your root partition is. For example, if your root partition is the second partition on your first SCSI hard drive, you would boot as follows:

```
boot sda2:vmlinux.gz root=/dev/sda2
```

If instead your root partition is the third partition on your second IDE drive, you would use this command:

```
boot hdb3:vmlinux.gz root=/dev/hdb3
```

For information on these commands and on automating the boot process, please read the *MILO HOWTO* in Appendix D.

Chapter 4

Installing Red Hat Linux/SPARC

These are brief installation instructions for the Red Hat Linux/SPARC 4.2 release for the SPARC family of processors. This chapter describes the differences between installing Red Hat Linux/SPARC and Red Hat Linux/Intel. For further details on the installation procedure, please see Chapter 2.

4.1 Hardware

Red Hat Linux/SPARC is known to function on the following hardware:

- sun4c architecture machines (IPC, SS1, etc)
- sun4m architecture machines (Classic, SS5, SS10, etc)
- bwtwo, cg3, cg6, TCX framebuffers (24 bit on the TCX)
- cg14 framebuffer (in cg3 mode)
- SCSI and Ethernet on all of the above
- type 4 and type 5 keyboards and mice

- external SCSI drives
- CD-ROM drives (external and internal)
- SCSI/Ether SBUS expansion cards
- Any original equipment Sun monitor for the above framebuffers

A partial list of *unsupported* hardware:

- VME based sun4m machines, such as the 4/690
- sun4d (SS1000, SS2000)
- sun4u (UltraSparc1, UltraSparc2, EnterPrise UltraSparc servers)
- sun4 architecture
- Eurocard sun4c machines, called the 'sun4e' under Solaris2.4 and others, are not supported. These are VME-bus sun4c machines, to which the kernel has not been ported yet.

4.2 Booting from Floppy

4.2.1 Making a Boot Floppy

If your SPARC system has a floppy drive, you can boot the Red Hat Linux installation program from a floppy diskette. The boot floppy image is located in the `images/` directory on your Red Hat CD.

To create the boot floppy, you can use `rawrite.exe` under MS-DOS (located on your Red Hat Linux/SPARC CD in the `dosutils` directory), or you can use `dd` under Linux or Unix. Under MS-DOS, you would enter the following commands (assuming the CD-ROM is drive `d:`):

```
d:
cd \images
\dosutils\rawrite.exe
```

`rawrite` first asks you for the name of a diskette image; enter `boot.img`. Then it asks for a diskette drive to write the image to; enter `a:`.

Under Linux or Unix, mount the CD, change to the directory it is mounted under, and use the following command:

```
dd if=images/boot.img of=/dev/fd0
```

Please Note: Do not use `/dev/rfd0` under Solaris; it will not work. Use `/dev/fd0` instead.

4.2.2 Booting

Insert the boot diskette, and at the PROM prompt enter `boot floppy`. You will be presented with a SILO prompt. If you wish to do your install using a ramdisk (it will use about 2M of RAM and is recommended if you have 8M or more of RAM), you can simply press Enter at the SILO prompt. If you wish to do an NFS-rooted install, you will need to supply some paramters here. The formula is as follows:

```
linux nfsroot=nfs.server.IP.address:/path/to/RH/image
```

For this to work you will need a `rarp` entry or a `bootp` entry on your network that will give your SPARC its IP address. This is covered in Section 4.3 below.

4.3 Booting from a Network

If you do not have floppy drive, and your PROM is less than version 2, you will need to do a network boot. This is fairly easy to do, but you will need a rarp entry on a server on your local network to tell the target SPARC what its IP address is. This can be done on a Linux system that has RARP built into the kernel (or inserted as a module), using a command of this form:

```
rarp -s ip.address.of.sparc hw:address:of:sparc
```

You also need an ARP entry in most cases; for example:

```
arp -s ip.address.of.sparc hw:address:of:sparc
```

Now you need to setup a TFTP server on your local network to serve the kernel to the target SPARC. This should be the same machine that is doing the `rarp` above. Under Red Hat Linux, this can be done by editing `/etc/inetd.conf`. You need an entry that looks similar to:

```
tftp dgram udp wait root /usr/sbin/tcpd in.tftpd /home/tftp
```

You can now copy the `kernels/vmlinux` to `/home/tftp` on your server and create a symbolic link so that the PROM can get the proper file. This symbolic link is conditional based on the IP address and architecture of the target SPARC. You need to make the link be the hex value of your IP address "dot" the architecture. For example, to create a symbolic link for `flanders.redhat.com`, a sun4m machine with IP address 199.183.24.232, you would enter the command:

```
cd /home/tftp
ln -s vmlinux C7B718E8.SUN4M
```

At this point you should be able to go to the PROM on your SPARC and type:

```
boot net nfsroot=nfs.server.IP.address:/path/to/RH/image
```

and it should get its rarp request from the TFTP server fulfilled and start loading the kernel via TFTP. The kernel should then boot and start the install.

After booting the installation disk you will be greeted by a welcome screen with a single button; press Enter to begin the installation.

Now, choose the medium you will be installing from; NFS, Hard Disk or FTP. Installing from hard disk will work if you have an existing partition with an EXT2 filesystem only; installation from UFS is not yet implemented.

4.4 Network Installation

You can also do an FTP install by TFTPing an entire install image (kernel and root filesystem). You will want to do the same as the above example, except for the nfsroot and the kernel link. Instead of using `kernels/vmlinux`, use `images/tftpboot.img`. Put it in `/home/tftp` and make the symlink:

```
cd /home/tftp
ln -s tftpboot.img C7B718E8.SUN4M
```

Please remember to change the IP address and architecture based on your own network numbers and machine architecture.

Now all you need to do is a `boot net` from the SPARC, and it should TFTP everything over, boot a kernel, and start the install program. You can then do an FTP install from a server near you, or you can do an NFS, hard drive, or CD-ROM install if you choose.

4.5 Installation Using a Serial Terminal

You can also install Red Hat Linux/SPARC 4.2 using a serial terminal; any terminal which emulates a VT100 ought to work fine. Boot from the floppy, and at the SILO prompt enter:

```
linux serial
```

The installation program runs on the first serial port at `9600,8,N,1` (9600 baud, 8 bits, no parity, 1 stop bit). The installation program can run in color, if your serial terminal supports color.

At any prompt dialog during a serial installation (any dialog with an **OK** button), you can press Ctrl-Z to start a subshell; to return to the installation program, enter `exit` at the shell prompt.

When the installation is complete, simply boot normally, and Red Hat Linux should come up on your serial terminal.

4.6 Installation: Selecting System Components

(For an explanation of system components, please see Chapter 2, Section 2.4.12.)

Important! When selecting system components to install for Red Hat Linux/SPARC, choosing to install **Everything** without choosing **Select individual packages** could result in an unbootable Red Hat Linux system. The last kernel to be installed will be a multi-processor kernel, which will fail to work on sun4c machines. Additionally, the default X server will end up as `XsunMono`, which may not work for most people. If you do wish to install **Everything**, choose **Select individual packages** as well. Then, deselect all the kernels except for one you wish to install (`kernel-sparc` works for most) and all X servers except the one for your hardware (`XsunMono` for bwtwo video cards, `Xsun` for all 8-bit displays, and `Xsun24` for the supported 24-bit displays).

4.7 SILO Configuration

SILO configuration is nearly identical to LILO configuration, see section 2.4.19 for details.

4.8 Partitioning

There is one additional step in partitioning a hard drive for a SPARC. You must create the third partition of every disk as type `Whole Disk` spanning from cylinder 0 to the end of the disk. It shouldn't be touched, but should exist. You can still create other partitions just as you normally would.

Additionally, the first partition on a disk should not be a swap partition, *unless* the swap partition starts on cylinder 1.

4.9 X Windows

There is no mouse or X windows config on the SPARC. The 8 bit server is installed by default. If you want to use the 24 bit server (for TCX cards), you need to install the `Xsun24` package. If you only have a bwtwo framebuffer and want to increase performance, install the `XsunMono` package. Installing both of the above requires you to restart X windows. Also, you can uninstall the `Xsun` package if you use one of the above servers.

Chapter 5

Post Installation Configuration

After installation your system is fully functional, but you may still want to do some additional configuration. This chapter describes some of the various system configuration tasks you may need to do.

5.1 Building a Custom Kernel

With the introduction of modularization in the Linux 2.0.x kernel there have been some significant changes in building customized kernels. In the past you were required to compile support into your kernel if you wanted to access a particular hardware or filesystem component. For some hardware configurations the size of the kernel could quickly reach a critical level. To require ready support for items that were only occasionally used was an inefficient use of system resources. With the capabilities of the 2.0.x kernel, if there are certain hardware components or filesystems that are used infrequently, driver modules for them can be loaded on demand. For information on handling kernel modules see Chapter 8, Section 8.6.

5.1.1 Building a modularized kernel

Only Red Hat Linux/Intel and Red Hat Linux/SPARC support modular kernels; Red Hat Linux/Alpha users must build a monolithic kernel (see Section 5.1.3 below).

These instructions provide you with the knowledge required to take advantage of the power and flexibility available through kernel modularization. If you do not wish to take advantage of modularization, please see Section 5.1.3 for an explanation of the different aspects of building and installing a monolithic kernel. It is assumed that you have already installed the `kernel-headers` and `kernel-source` packages and that you issue all commands from the `/usr/src/linux/` directory.

It is important to begin a kernel build with the source tree in a known condition. Therefore, it is recommended that you begin with the command `make mrproper`. This will remove any configuration files along with the remains of any previous builds that may be scattered around the source tree. Now you must create a configuration file that will determine which components to include in your new kernel. Depending upon your hardware and personal preferences there are three methods available to configure the kernel.

- `make config` An interactive text program. Components are presented and you answer with `Y` (yes), `N` (no), or `M` (module).
- `make menuconfig` A graphic, menu driven program. Components are presented in a menu of categories, you select the desired components in the same manner used in the Red Hat Linux installation program. Toggle the tag corresponding to the item you want included; **Y** (yes), **N** (no), or **M** (module).
- `make xconfig` An X Windows program. Components are listed in different levels of menus, components are selected using a mouse. Again, select **Y** (yes), **N** (no), or **M** (module).

Please Note: In order to use `kerneld` (see Section 8.6 for details) and kernel modules you must answer **Yes** to **kerneld support** and **module version (CONFIG_MODVERSIONS) support** in the configuration.

Please Note: If you are building a Linux/Intel kernel on (or for) a machine that uses a “clone” processor (for example, one made by Cyrix or AMD), it is recommended to choose a **Processor type** of **386**.

If you wish to build a kernel with a configuration file (`/usr/src/linux/.config`) that you have already created with one of the above methods, you can omit the `make mrproper` and `make config` commands and use the command `make dep` followed by `make clean` to prepare the source tree for the build.

The next step consists of the actual compilation of the source code components into a working program that your machine can use to boot. The method described here is the easiest to recover from in the event of a mishap. If you are interested in other possibilities details can be found in the Kernel-HOWTO or in the `Makefile` in `/usr/src/linux` on your Linux system.

- Build the kernel with `make boot`.
- Build any modules you configured with `make modules`.
- Move the old set of modules out of the way with:

  ```
  rm -rf /lib/modules/2.0.29-old
  mv /lib/modules/2.0.29 /lib/modules/2.0.29-old
  ```

  Of course, if you have upgraded your kernel, replace `2.0.29` with the version you are using.
- Install the new modules (even if you didn't build any) with `make modules_install`.

If you have a SCSI adapter and made your SCSI driver modular, build a new `initrd` image (see Section 5.1.2; note that there are few practical reasons to make the SCSI driver modular in a custom kernel).

In order to provide a redundant boot source to protect from a possible error in a new kernel you should keep the original kernel available. Adding a kernel to the LILO menu is as simple as renaming the original kernel in `/boot`, copying the new kernel to `/boot`, adding a few lines in `/etc/lilo.conf` and running `/sbin/lilo`. Here is an example of the default `/etc/lilo.conf` file shipped with Red Hat Linux:

```
boot=/dev/hda
map=/boot/map
```

```
install=/boot/boot.b
prompt
timeout=100
image=/boot/vmlinuz
        label=linux
        root=/dev/hda1
        read-only
```

Now you must update `/etc/lilo.conf`. If you built a new initrd image you must tell LILO to use it. In this example of `/etc/lilo.conf` we have added four lines at the bottom of the file to indicate another kernel to boot from. We have renamed `/boot/vmlinuz` to `/boot/vmlinuz.old` and changed its label to `old`. We have also added an `initrd` line for the new kernel:

```
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
prompt
timeout=100
image=/boot/vmlinuz
        label=linux
        initrd=/boot/initrd
        root=/dev/hda1
        read-only
image=/boot/vmlinuz.old
        label=old
        root=/dev/hda1
        read-only
```

Now when the system boots and you press Tab at the LILO `boot:` prompt two choices will be shown;

```
LILO boot:
linux    old
```

To boot the new kernel (`linux`) simply press Enter, or wait for LILO to time out. If you want to boot the old kernel (`old`), simply enter `old` and press Enter.

Here is a summary of the steps;

- `mv /boot/vmlinuz /boot/vmlinuz.old`
- `cp /usr/src/linux/arch/i386/boot/zImage /boot/vmlinuz`
- edit `/etc/lilo.conf`
- run `/sbin/lilo`

You can begin testing your new kernel by rebooting your computer and watching the messages to ensure your hardware is detected properly.

5.1.2 Making an initrd image

An `initrd` image is needed for loading your SCSI module at boot time. The shell script `/sbin/mkinitrd` can build a proper `initrd` image for your machine if the following conditions are met:

- The loopback block device is available.
- The `/etc/conf.modules` file has a line for your SCSI adapter; for example:

  ```
  alias scsi_hostadapter BusLogic
  ```

To build the new `initrd` image, run `/sbin/mkinitrd` with parameters such as this:

```
/sbin/mkinitrd /boot/newinitrd-image 2.0.12
```

where `/boot/newinitrd-image` is the file to use for your new image, and `2.0.12` is the kernel whose modules (from `/lib/modules`) should be used in the `initrd` image (not necessarily the same as the version number of the currently running kernel).

5.1.3 Building a monolithic kernel

To build a monolithic kernel you follow the same steps as building a modularized kernel with a few exceptions.

- When configuring the kernel only answer **Yes** and **No** to the questions (don't make anything modular).
- Omit the steps:

```
make modules
make modules_install
```

- Edit the file `/etc/rc.d/rc.sysinit` and comment out the line `depmod -a` by inserting a "#" at the beginning of the line.

5.2 X Window Configuration with XFree86

There are four methods for configuring XFree86 on your machine: `Xconfigurator`, `XF86Setup`, `xf86config`, or by hand. `Xconfigurator`, `XF86Setup`, and `xf86config` are functional equivalents and should work equally well. If you are unsure of anything in this process, a good source of additional documentation is:

```
http://www.xfree86.org
```

`Xconfigurator` is a full-screen menu driven program that walks you through setting up your X server. `XF86Setup` is a graphical setup program distributed with XFree86 which requires the VGA 16-color server in order to run. `xf86config` is a line oriented program distributed with XFree86. It isn't as easy to use as `Xconfigurator`, but it is included for completeness. If these utilities fail to provide a working `XF86config` file, you may have an unsupported card or you may need to write the config file by hand. Usually the former is the case, so check and make sure your card is supported before attempting to write the config file yourself. If your card is not supported by XFree86 you may wish to consider using a commercial X server, such as the Metro-X server which accompanies Official Red Hat Linux/Intel. If you have questions about whether or not your video card is supported you can check out `http://www.xfree86.org` for information on XFree86 and `http://www.metrolink.com` for information on Metro-X.

5.2.1 The X Server

Provided you selected the proper video card at install time, you should have the proper X server installed. When later running `Xconfigurator` or `xf86config`, you need to make sure you select the same video card or the autoprobe will fail.

If you think you installed the wrong X server for your video card, you will have to install the correct one before it can be configured. For instance, if the CD is mounted on `/mnt/cdrom`, and you need to install the S3 server, enter the following commands:

```
cd /mnt/cdrom/RedHat/RPMS
rpm -ivh XFree86-S3-3.1.2-1.i386.rpm
ln -sf ../../usr/X11R6/bin/XF86_S3 /etc/X11/X
```

This will install the S3 server and make the proper symbolic link.

5.2.2 Xconfigurator

To configure X Windows you must first select your video card. Scroll down the list of supported cards until you locate the card in your machine. Section 5.2.2 may help you determine the video server that matches your hardware. If your card is not listed it may not be supported by XFree86. In this case you can try the last card entry on the list (Unlisted Card) or a commercial X Windows server, such as Metro X. The next step is to select your monitor. If your monitor is not listed you can select one of the generic monitor entries or "Custom" and enter your own parameters. Custom monitor configuration is recommended only for those who have a sound understanding of the inner workings of CRT displays. The average user should probably use one of the generic selections from the list. After selecting a monitor you need to tell Xconfigurator how much video memory you have. Move the highlight to the appropriate list entry and then press Enter or F12 to continue. For the next step it is recommended that you select the default (No Clockchip Setting) entry, but experienced users may want to select a specific clockchip.

Selecting your Server

If you are unsure what chip-set you have, the best way to find out is usually to look at the card. Figure 5.1 lists which chip-sets and boards require which servers. Pick the one that best matches your hardware.

Server	Chipset
VGA16	All VGA boards with 256K (16 color only)
SVGA	Trident 8900 & 9400, Cirrus Logic, C & T, ET4000, Others
Mach8	ATI boards using the Mach8 chip-set
Mach32	ATI boards using the Mach32 chip-set
Mach64	ATI boards using the Mach64 chip-set
8514	IBM 8514/A Boards and true clones
S3	#9 Boards, most Diamonds, some Orchids, Others
AGX	All XGA graphics boards
P9000	Diamond Viper (but not the 9100) and Others
W32	All ET4000/W32 cards, but not standard ET4000's

Figure 5.1: X Servers

Finishing Up

If later you want to increase your refresh rate for your monitor, you can edit the config file by hand or you can run `Xconfigurator` again and pick a monitor from our list that more closely matches the specs of your monitor.

The final configuration step consists of selecting the video modes that you want to include in your XF86config file. Use the arrow keys to move the cursor up and down the list under each color depth (8, 16 and 24 bit). Use the Spacebar to select individual resolutions and the Tab key to move between color depth fields. When you have selected the video modes you want to use move the cursor to the "OK" button and press Enter, or use the F12 shortcut. An information screen will give you the most current information on selecting video modes, starting and stopping the X server.

5.3 X Window Configuration with Metro-X

By special arrangement, a license to install a single copy of the Metro-X enhanced X server is included with the Official Red Hat Linux/Intel CD. Please note that, unlike the XFree86 server (also included), Metro-X is commercially licensed software. You should review and accept its license terms before installing it. Metro-X supports many video cards that are not supported by XFree86. The configuration utility can be run in graphical mode (`configX`) or text mode (`configX.curses`).

5.3.1 System Requirements

8 megabytes of RAM minimum; 16 megabytes are recommended. The base server package requires 12 megabytes of disk space.

5.3.2 configX.curses

`configX.curses` is a simple graphical configuration tool for configuring Metro-X. The configX.curses utility does not provide support for configuring multiple screens or choosing the number of buttons on your mouse. If you need these facilities you must use configX. To run `configX.curses` type the command:

```
configX.curses
```

The opening screen has information on Metro Link, Inc. and brief instructions on what you will need to configure Metro-X. Press the Enter key to begin the configuration. The screen will display the current configuration information at the top and the configuration menu at the bottom. Use the arrow keys to move the highlight up and down the menu and the Enter key to select the highlighted item.

Graphics Card

When you select **Graphics Card** from the configuration menu, a list of supported cards will be displayed. Select your video card as above. After selecting your video card you will be asked to select a video mode from a list of defaults. The resolution used is a matter of personal preference. The smaller the numbers are the larger the items on the display will be. Some combinations of video cards and video modes support virtual screens larger than the physical display. If your combination is one of these, you will be asked to select the virtual screen size.

Mouse

Configuring your mouse with Metro-X requires that you know what type of mouse you have and how it is connected to your computer. When you select a mouse type, the default device will be provided in the configuration window. If you are in doubt, accept the default and make adjustments later if needed.

Exiting

Selecting **Save and Exit** from the menu will save any changes you have made to the Metro-X configuration and exit the `configX.curses` program.

Selecting **Quit without Saving** from the menu will discard any changes you have made to the Metro-X configuration and exit the `configX.curses` program.

5.3.3 configX

Metro-X can also be configured for your system through a graphical utility called *ConfigX*, which is used to specify the mouse, keyboard, monitor, graphics card, color depth, and screen resolution(s) you are using.

Start-up

1. Login as root and type the following command: `/usr/bin/X11/configX`
2. The explanation of special keys appears. (This message only appears prior to the initial configuration of Metro-X, when no Xconfig file exists.)

3. Press [Enter] to continue.

Special Keys

When Metro-X is first installed, it doesn't yet recognize your mouse. Special keys must be used to move around the screens and make your selections during the initial configuration process.

Key	Function
[Tab]	Moves forward from field to field.
[Shift]-[Tab]	Moves backward from field to field.
[↑] [↓] [←] [→]	Changes the value in the current field.
[Enter]	Sets/unsets a toggle button.
[Page Down]	Switches to the next configuration page.
[Page Up]	Switches to the previous configuration page.
[Esc]	Cancels out of the current screen or pop-up.
[Alt-F]	Jumps to the File pull-down menu.
[Alt-S]	Jumps to the Screen pull-down menu.
[Alt-C]	Jumps to the Configuration pull-down menu.

Figure 5.2: Function of Special Keys

Mouse configuration

If the default mouse is not correct for your system:

1. Press [↓] to display the list of mice.
2. Press [↑] or [↓] to scroll to your mouse entry, then press [Enter] to select it.
3. Press [Tab] to move to the next field.
4. Press [↓] to display the list of mouse buttons. (The default entry is based on your mouse selection, and is usually the appropriate choice.)

5. Press $\boxed{uparrow}$ or $\boxed{downarrow}$ to scroll to your button entry, then press $\boxed{\textsf{Enter}}$ to select it.
6. Press $\boxed{\textsf{Tab}}$ to move to the next field.
7. If you need to change the path for your mouse device, edit it here.
8. Press $\boxed{\textsf{Tab}}$ to move to the next field.

Keyboard configuration

Metro-X includes international keyboard support, which allows you to select the keyboard appropriate for your language and country.

1. The initial keyboard setting is *Default: Default Console Mapping*, which will work with most keyboards.
2. Press $\boxed{\downarrow}$ to display the list of keyboards.
3. Press $\boxed{\uparrow}$ or $\boxed{\downarrow}$ to scroll to your keyboard entry, then press $\boxed{\textsf{Enter}}$ to select it.
4. Press $\boxed{\textsf{Tab}}$ to move to the next field.

Monitor configuration

The default monitor setting is *Generic 640x480 @ 60Hz 13 inch.*

1. Press $\boxed{\downarrow}$ to display the list of monitors.
2. Press $\boxed{\uparrow}$ or $\boxed{\downarrow}$ to scroll to your monitor entry, then press $\boxed{\textsf{Enter}}$ to select it.
3. Press $\boxed{\textsf{Tab}}$ to move to the next field.
4. If you need to change the screen width (mm), edit it here.
5. Press $\boxed{\textsf{Tab}}$ to move to the next field.
6. If you need to change the screen height (mm), edit it here.
7. Press $\boxed{\textsf{Tab}}$ to move to the next field.

Screen Saver

Some graphics cards and monitors support the Display Power Management System (DPMS), which minimizes the power used by the monitor when user input ceases. If your hardware has this feature, the phases of DPMS can be controlled in this section. The typical DPMS setup has a cascade effect, with the various phases being activated in succession with your choice of delay times between each.

- X Screen Saver - Preliminary phase to DPMS, which blanks the screen but provides no power reduction.
- Standby Mode - First phase of DPMS, which provides moderate power reduction by disabling the horizontal sync signal to the monitor.
- Suspend Mode - Second phase of DPMS, which provides significant power reduction by disabling the vertical sync signal to the monitor.
- Monitor Off - Final phase of DPMS, which provides maximum power reduction by disabling both the horizontal and vertical sync signals to the monitor.

1. In the *X Screen Saver* Delay field, press Enter to toggle this option on or off.
2. If toggled on, press Tab to move to the *Minutes* field, then use ← or → to set the value.
3. Press Tab to move to the next field.
4. In the *Standby Mode Delay* field, press Enter to toggle this option on or off.
5. If toggled on, press Tab to move to the Minutes field, then use ← or → to set the value.
6. Press Tab to move to the next field.
7. In the *Suspend Mode Delay* field, press Enter to toggle this option on or off.
8. If toggled on, press Tab to move to the *Minutes* field, then use ← or → to set the value.

9. Press Tab to move to the next field.
10. In the *Monitor Off Delay* field, press Enter to toggle this option on or off.
11. If toggled on, press Tab to move to the *Minutes* field, then use ← or → to set the value.

Graphics Card Configuration

Press Page Down to move to the second configuration screen, which controls the graphics card, the number of colors, the virtual resolution, and the physical resolution.

Each graphics card configuration consists of five elements:

- The graphics card (make and model).
- The number of colors supported on screen simultaneously (color depth).
- The virtual resolution of the screen (width x height).
- The physical resolution of the screen (width x height).
- The screen refresh rate (Hz).

Non-interlaced modes give a better screen appearance than refresh rates which are labeled "Interlaced". The highest refresh rate that your monitor supports will give the best image. At higher screen resolutions a higher refresh rate is particularly important to prevent the screen from visibly flickering. The highest refresh rate supported by the selected monitor is automatically displayed. **Graphics Card**

1. The default graphics card setting is *IBM VGA.*
2. Press ↓ to display the list of graphics cards.
3. Press ↑ or ↓ to scroll to your graphics card entry, then press Enter to select it.
4. Press Tab to move to the next field.

Colors

Not all graphics cards can support all color depths (number of colors). Only the options for the color depths supported by your card will be sensitive in this field. The other options will be greyed out.

- Only *one* color option can be selected (due to design limitations of X).
- In some cases the color options available will be limited by the physical resolution(s) you choose, based on the capabilities of your graphics card. Not all color depths are available at high resolutions.
  1. The default setting is at the lowest value for your card.
  2. Decide which available color setting you want to use.
  3. Press Tab to move to that color setting, then press Enter to select it. (The previous setting will be cancelled.)
  4. Press Tab to move to the next field.

Virtual Resolution

The *Virtual Resolution* setting allows you to have an on-screen work area that is actually larger than your physical screen. Windows can extend beyond the screen's physical boundaries and are viewed by scrolling into the extended areas. Only the options for the virtual resolutions supported by your card will be sensitive in this field. The other options will be greyed out. If you don't want to use this feature, select the *Disabled* option.

- Only *one* virtual resolution can be selected.
- In some cases the virtual-resolution options available will be limited by the physical resolution(s) you choose, since the virtual resolution must always be greater than the highest physical resolution.
  1. The default setting for virtual resolution is *Disabled.*
  2. Decide which available virtual resolution you want to use.
  3. Press Tab to move to that setting, then press Enter to select it. (The previous setting will be cancelled.)
  4. Press Tab to move to the next field.

Physical Resolution

Not all graphics cards and monitors can support all physical resolutions. Only the options for the resolutions supported by your hardware will be sensitive in this field. The other options will be greyed out.

- *Multiple* physical resolutions can be selected. You can switch between these resolutions while running Metro-X by pressing a hot-key combination:
  - Press Ctrl-Alt-+ to change to the next resolution.
  - Press Ctrl-Alt-- to change to the previous resolution.
- If you want only *one* resolution, make sure only *one Physical Resolution* toggle button is selected.
- In some cases the physical resolution(s) you choose may limit the selections you can make in *Colors* and *Virtual Resolution*, depending on the capabilities of your graphics card.
  1. The default setting is at the lowest value for your card.
  2. Decide which available resolution(s) you want to use.
  3. Press Tab to move to the setting, then press Enter to select it.
  4. Press Tab to move to the corresponding field to set the screen refresh rate (Hz) for that resolution, then use ← or → to select a value from the frequency options available. The default setting is the highest frequency your monitor supports, and is usually the appropriate choice.
  5. Repeat this procedure for each physical resolution you want to set.

Multiple-Screen Configuration

Metro-X supports up to four screens which can all be controlled simultaneously with a single keyboard and mouse. This allows you to run many applications without overlapping windows, and is particularly useful for image-processing work. For each screen used, a separate configuration must be created for that screen's monitor and graphics card. See section B.1.4 for graphics card requirements.

Adding a Screen

Metro-X defaults to a single-screen setup. The first screen configured is labeled *Screen 0*. When you add a second screen, all of its configuration default settings will duplicate the current settings of Screen 0. So if your hardware is the same on all screens, you can save time by configuring Screen 0 before adding the other screens. Similarly, as more screens are added, the settings of whichever one is currently active when you add another screen will determine the default settings of the new screen.

1. Press Alt-S to display the Screen pull-down menu.
2. Type *A* for *Add.*
3. The *Monitor* and *Graphics Card* icons for Screen 0 become outlines, and two full-color icons appear for Screen 1.
4. Refer to the procedures described earlier in this chapter if any changes need to be made to the new screen's configuration.
5. Use the Page Down and Page Up keys to move to the configuration pages of each screen.

Deleting a Screen

When a screen is deleted, all screens with higher screen numbers immediately drop down one slot. For instance: You currently have screens 0, 1, and 2 configured and you delete screen 1. That screen and all of its configuration information disappears. Screen 2 then becomes screen 1, but retains its own configuration. You are left with screens 0 and 1.

1. Use the Page Down or Page Up key to activate the set of icons for the screen you want to delete.
2. Press Alt-S to display the *Screen* pull-down menu.
3. Type *D* for *Delete.*
4. One set of *Monitor* and *Graphics Card* icons disappears. Higher-numbered screens (if any) immediately fill the gap of the deleted screen.

Screen Layout

The Screen Layout window lets you specify how the multiple screens are arranged in your workspace so mouse movement from screen to screen will be appropriate.

1. Press Alt-S to display the Screen pull-down menu.
2. Type *L* for *Layout.*
3. The *Screen Layout* window appears.
4. Press ← and → to view the various layout combinations for your number of screens.
5. When the layout you want is displayed, press Enter to select it.

Font Path

If needed, you can edit your font path from within the Metro-X Configuration Utility. This will be necessary if you are using a font server instead of or in addition to your existing fonts.

1. Press Alt-C to display the *Configuration* pull-down menu, then press Enter.
2. The *Font Path* window appears.
3. Edit the file as necessary for your system. Metro-X looks for fonts in the order in which the directories are listed in this file. If you are using a font server, enter this as the first line:

   ```
   tcp/<fontserver machine>:7000
   ```

   If your port number is not the default `7000`, change it to the appropriate port number for your system.
4. Press Tab to move to and amongst the bottom buttons, then press Enter to select one. If the information is correct, select the *OK* button. If you want to disregard the changes you've made, select the *Cancel* button. If you've edited your font path and want to go back to the original settings, select the *Load Defaults* button.

Save and Exit

1. Press Alt-F to display the File pull-down menu.
2. Type *S* for *Save and Exit.*
3. Your changes are saved, the Metro-X Configuration Utility closes, and your system is now ready to run Metro-X!

Quit Without Saving

1. Press Alt-F to display the File pull-down menu.
2. Type *Q* for *Quit.*
3. A precautionary dialog box appears: *Save current configuration before exiting?*
4. Press Tab to move to the *No* button, then press Enter to confirm that you want to exit the Metro-X Configuration Utility without saving your changes.

For more details on features of Metro X and the format of the `Xconfig` file see Appendix B.

5.4 Mouse configuration

To configure your mouse (or reconfigure your mouse after installation) enter the command `/usr/sbin/mouseconfig`. Scroll down the list with the arrow keys until your mouse type is highlighted. If you have *any* mouse connected to a PS/2 style port you should select **PS/2** as your mouse type. If you have a mouse connected to a mouse port on an ATI video card you should select **ATI Bus Mouse**. If you have a 2 button mouse and want to emulate the third (middle) mouse button with a simultaneous click of the left and right mouse buttons use Tab to move the cursor to the **Emulate 3 Bottons** checkbox and press Spacebar to toggle the selection.

5.5 Sendmail

A default `sendmail.cf` file will be installed in `/etc/`. The default configuration should work for most SMTP-only sites. It will *not* work for UUCP sites; you will need to generate a new `sendmail.cf` if you need to use UUCP mail transfers. To generate a new `sendmail.cf`, you will need to install `m4` and the `sendmail` source package. Read the `README` file in the sendmail sources for more details on creating `sendmail` configuration files. Also, O'Reilly & Associates publishes a good sendmail reference entitled *sendmail* by Bryan Costales.

One common sendmail configuration is to have a single machine act as a mail gateway for all the machines on your network. For instance, at Red Hat Software we have a machine `mail.redhat.com` that does all our mail. On that machine we simply need to add the names of machines for which `mail.redhat.com` will handle mail to `/etc/sendmail.cw`. Here is an example:

```
# sendmail.cw - include all aliases for your machine here.
torgo.redhat.com
poodle.redhat.com
devel.redhat.com
```

Then on the other machines, `torgo`, `poodle`, and `devel`, we need to edit `/etc/sendmail.cf` to "masquerade" as `mail.redhat.com` when sending mail, and to forward any local mail processing to `redhat.com`. Find the `DH` and `DM` lines in `/etc/sendmail.cf` and edit them thusly:

```
# who I send unqualified names to (null means deliver locally)
DRmail.redhat.com

# who gets all local email traffic
DHmail.redhat.com

# who I masquerade as (null for no masquerading)
DMredhat.com
```

With this type of configuration, all mail sent will appear as if it were sent from `redhat.com`, and any mail sent to `torgo.redhat.com` or the other hosts will be delivered to `mail.redhat.com`.

Please be aware that if you configure your system to masquerade as another any email sent from your system to your system will be sent to the machine you are masquerading as. For example, in the above illustration, log files that are periodically sent to `root@poodle.redhat.com` by the `cron` daemon would be sent to `root@mail.redhat.com`.

5.6 Controlling Access to Services

As a security measure, most network services are managed by a protective program called a *TCP wrapper*. The protected services are those listed in `/etc/inetd.conf` that specify `/usr/sbin/tcpd`. `tcpd` can allow or deny access to a service based on the origin of the request, and the configuration in `/etc/hosts.allow` and `/etc/hosts.deny`. By default Red Hat Linux allows all service requests. To disable or limit services you can edit `/etc/hosts.allow`. Here is an example `/etc/hosts.allow` file:

```
ALL: redhat.com .redhat.com
in.talkd: ALL
in.ntalkd: ALL
in.fingerd: ALL
in.ftpd: ALL
```

This configuration allows all connections from `redhat.com` and `*.redhat.com` machines. It also allows talk, finger, and ftp requests from all machines.

`tcpd` allows much more sophisticated access control, using a combination of `/etc/hosts.allow` and `/etc/hosts.deny`. Read the tcpd(8) and hosts_access(5) man pages for complete details.

5.7 Anonymous FTP

Setting up anonymous FTP is simple. All you need to do is install the `anon-ftp` rpm package (which you may have already done at install time). Once it is installed, anonymous FTP will be up and running.

There are a few files you might wish to edit to configure your FTP server.

`/etc/ftpaccess` This file defines most of the access control for your ftp server. Some of the things that you can do are: set up logical "groups" to control access from different sites, limit the number of simultaneous FTP connections, configure transfer logging, and much more. Read the ftpaccess man page for complete details.

`/etc/ftphosts` The ftphosts file is used to allow or deny access to certain accounts from various hosts. Read the ftphosts man page for details.

`/etc/ftpusers` This file lists all the users that are *not* allowed to ftp into your machine. For example, `root` is listed in `/etc/ftpusers` by default. That means that you can not ftp to your machine and log in as root. This is a good security measure, but some administrators prefer to remove `root` from this file.

5.8 World Wide Web Server

If you installed the Apache web server (from the `apache` package), then your Web service is already up and running! Just point your web browser at `http://localhost`.

The default page shown is `/home/httpd/html/index.html`. You can edit this file (or completely replace it) to your liking. All the CGI programs, icons, and html pages are stored in `/home/httpd`, but this can be changed in the apache configuration files, all of which are stored in `/etc/httpd/conf/`. Logs of all httpd activity are kept in `/var/log/httpd/`. Setting up your web site is as easy as adding your own HTML pages and links to the `/home/httpd/html/` directory. For more information on customizing your web server we recommend a reference such as *HTML: The Definitive Guide* by Chuck Musciano & Bill Kennedy, published by O'Reilly & Associates.

5.9 NFS Configuration

NFS stands for Network File System, and is a way to share files between machines as if they were on your local hard drive. Linux can be both an NFS server and an NFS client, which means that it can *export* filesystems to other systems, and *mount* filesystems exported from other machines.

5.9.1 Mounting NFS Filesystems

Use the `mount` command to mount a NFS filesystem from another machine:

```
mount porky:/mnt/export /mnt/local
```

In this command, `porky` is the hostname of the NFS fileserver, `/mnt/export` is the filesystem that porky is exporting, and `/mnt/local` is a directory on my local machine where we want to mount the filesystem. After the `mount` command runs (and if we have the proper permissions from `porky`) we can enter `ls /mnt/local` and get a listing of the files in `/mnt/export` on `porky`.

5.9.2 Exporting NFS Filesystems

The file that controls what filesystems you wish to export is `/etc/exports`. Its format is:

```
directory       hostname(options)
```

the `(options)` are optional. For example:

```
/mnt/export     speedy.redhat.com
```

would allow `speedy.redhat.com` to mount `/mnt/export`, but:

```
/mnt/export     speedy.redhat.com(ro)
```

would just allow `speedy` to mount `/mnt/export` read-only.

Each time you change `/etc/exports`, you need to tell the NFS daemons to examine it for new information. One simple way to accomplish this is to just stop and start the daemons:

```
/etc/rc.d/init.d/nfs stop
/etc/rc.d/init.d/nfs start
```

The following will also work:

```
killall -HUP rpc.nfsd rpc.mountd
```

See the following man pages for more details: nfsd(8), mountd(8), and exports(5). Another good reference is *Managing NFS and NIS Services* by Hal Stern, published by O'Reilly & Associates.

5.10 Running Programs at Boot Time

The file `/etc/rc.d/rc.local` is executed at boot time, after all other initialization is complete, and whenever you change runlevels. You can add additional initialization commands here. For instance, you may want to start up additional daemons, or initialize a printer. In addition, if you require serial port setup, you can edit `/etc/rc.d/rc.serial`, and it will be executed automatically at boot time.

The default `/etc/rc.d/rc.local` simply creates a nice login banner with your kernel version and machine type.

5.11 Color-ls Setup

`color-ls` is now part of the standard `ls` command. To get colorized directory listings, simply issue the command (in `bash`):

```
alias ls='ls --color=tty'
```

5.12 NIS/NYS Setup

The basic content of this section was donated by Grant Taylor. It was edited for clarity and space. Other content was contributed by users on the `redhat-list` mailing list.

This text attempts to describe the yp, NIS, and NIS+ support offered in Red Hat Linux products.

- What is yp? NIS? NIS+? NYS? These questions are answered in the NIS-HOWTO. It is included with Red Hat Linux in the `ldp` package and on the CD. If you install the `ldp` package, you will find the NIS-HOWTO in `/usr/doc/HOWTO`.

The quick answer is that while yp ("Yellow Pages") is the phone book in Britain, NIS is the traditional RPC-based implementation for sharing passwd, group, hosts, services, and other useful things between many machines. NIS+ is a snazzier, and more secure implementation of the same idea. NYS is the public domain version of NIS.

- What *is* offered? Red Hat ships NYS client code in its `libc` and plain yp/NIS client programs like `ypcat` and `ypbind`. We ship a plain NIS server daemon, since the NYS (aka NIS+) server daemon is not yet ready for prime-time. Our `ypserv` does not appear to have the `tcp_wrapper`-style `/etc/hosts.allow/deny` checking compiled in.

- How do I get `ypbind` to work properly? How do I set up a client workstation? The NYS code in `libc` does it all. There is no need to run `ypbind`. You probably *do* need to run `domainname` (apparently for the benefit of of plain NIS client programs). You must configure which maps are read from where in `/etc/nsswitch.conf`, and which NIS server and domain you are in using `/etc/yp.conf`. `nsswitch.conf` is well commented, and `yp.conf` will look like this:

  ```
  domain foo
  ypserver foo.bar.com
  ```

  Do not put the usual `+::::` lines at the end of `passwd` or `group`.

- How do I run `ypserv` (master)? Red Hat ships a plain NIS (a.k.a. yp) server. It works just like any other `ypserv`—modify the `Makefile` in `/var/nis` to taste, perhaps move the source files into `/var/nis/src`, or perhaps not, run `make` to make the databases, and run `ypserv`. Further documentation is in `/usr/doc/ypserv*/*`.

- Where are `yppasswd` and `yppasswdd`? They are not included in Red Hat Linux, but you can obtain the regular NIS distribution and install it yourself without too much hassle. It is available from the places listed in the NIS HOWTO. You have to modify the `pwupdate` script that comes with it, even though the README claims it works with the NYS Makefile we ship. We also don't ship a `rpcsrv/yppasswd.h`, although we do have the `.x` file. The `yppasswdd` package comes with both so it doesn't matter much. The `NYS` define should be used, but don't use the `-lnsl` library since it is built into our `libc`.

- This information is miserably incomplete! There is already an NIS-HOWTO that describes how to set things up far better than this document does or should. It is also included with Red Hat Linux in the `ldp` package.

Chapter 6

Finding Documentation

Red Hat Linux includes thousands of pages of online documentation to help you learn how to use the system. The man pages, info documents, and plain text files included provide information on almost every aspect of Linux. For more help, Red Hat also includes all of the documentation produced by the Linux Documentation Project electronically.

6.1 On Line Help

When you are looking for general help on commands and error messages, the best place to start is right on your system.

6.1.1 Man Pages

Most every command on your system has an associated "man" page. This is documentation that you can get to instantly should you have questions or problems. If you were having trouble with the command `ls`, you could enter `man ls`. This will bring up the man page for `ls`. The man page is viewed through the `less` program, so all of the options to `less` will work while in a man page. Some important key strokes are:

- `q` to quit

- Enter to page down line by line
- Space to page down page by page
- b to page back up by one page
- / followed by a string and Enter to search for a string
- n to find the next occurrence of the previous search

Sometimes viewing man pages isn't too friendly on line. Providing you have a working printer, you can print man pages as well. If you don't have postscript printing capability and just want to print ASCII, you can print man pages with:

```
man COMMAND | lpr
```

If you do have a postscript printer, you will probably want to print with:

```
man -t COMMAND | lpr
```

In both of those commands substitute "COMMAND" for the command you are trying to get help for.

Also, sometimes things have more than one man page. Here is a table of what is located where:

Section	**Contents**
1	user commands
2	system calls
3	library calls
4	devices
5	file formats
6	games
7	miscellaneous
8	system commands
9	kernel internals

So, let's say that you want to see the man page for `swapon`. You do `man swapon`. You will actually get the man page for the system call `swapon(2)`, which is the function you use in a C program to turn swap on. Unless you are writing your own program to do it, this probably isn't what you want. So, using the chart above, you can see that what you want is probably a "system command" and is located in section 8. You can then do `man 8 swapon`. All of this is because `man` searches the man directories in order, and then returns when it finds the first match.

You can also search the man pages for strings. You do this using `man -k string_to_search_for`. This won't work, however, unless the makewhatis database has been created. Under Red Hat Linux, this is done by a cron job overnight. If you don't leave your system running overnight the database won't get created. If that is the case, run the following command as the `root` user:

```
/etc/cron.weekly/makewhatis.cron
```

Once you've done that, you could enter `man -k swapon`. That command would return:

```
# man -k swapon
swapon, swapoff (2) - start/stop swapping to file/device
swapon, swapoff (8) - enable/disable devices and files for \
                      paging and swapping
```

So you can see that there are pages in section 2 and 8 both referring to swapon (and swapoff in this case).

6.1.2 Package Documentation

Many packages of software have `README` files and other documentation as part of the source package. Red Hat Linux uses a standard place to install those documents so that you don't have to install the sources to look at the documents. All of those documents are stored in subdirectories of `/usr/doc`. The subdirectory depends on the package. Each package that has extra documentation will create a directory called packagename-version-releasenumber. For example, the `tin` package might be version

1.22 and release number 2. The path to its documentation would be `/usr/doc/tin-1.22-2`.

For the most part, the documents in this directory are ASCII. You can usually view them with `more filename` or `less filename`. This is nice, but what if you want to see if there is documentation for a specific command or file and you don't know the package it came from? It doesn't matter! You can simply enter:

```
rpm -qdf /etc/sendmail.cf
```

This will report all the documentation from the package containing the file `/etc/sendmail.cf`. Commands like this are covered more in depth in the RPM-HOWTO , available from `www.redhat.com`.

Also, what if it's a command you need help with and the man page isn't good? You could do something like:

```
rpm -qdf `which COMMAND`
```

Again, "COMMAND" is the actual command you need help with. This will work only when the command is on your path.

6.1.3 HOWTOs and FAQs

Most of the contents of the Linux Documentation Project (LDP) are available in `/usr/doc` on your system.

`/usr/doc/HOWTO` contains the ASCII versions of all the available HOWTOs at the time we pressed the CD-ROM. They are gzipped, so you have to use `gunzip` to unzip them or use a command like:

```
zcat HAM-HOWTO.gz | more
```

The latter will work, but is a bit less flexible than unzipping and then using `more`. However, unzipping also requires more disk space unless you re-gzip the document when done.

`/usr/doc/HOWTO/mini` contains the ASCII versions of all the available mini-HOWTOs. They are not compressed and can be viewed with `more` or `less`.

`/usr/doc/HTML` contains the HTML versions of all the HOWTOs and the *Linux Installation and Getting Started* guide. To view things here, just use a WWW browser like `redbaron` from Red Hat Software, or `arena` . You would do something like:

```
cd /usr/doc/HTML
arena index.html
```

`/usr/doc/FAQ` contains ASCII version (and some HTML versions) of some popular FAQs, including the RedHat-FAQ. All of them can be viewed using `more` or `less`.

6.1.4 The "locate" Command

When you don't know the full name of a command or file, but need to find it, you can usually find it with `locate`. `locate` uses a database to find all files on your system. Normally, this database gets built from a cron job every night. This won't happen, however, if your machine isn't booted into Linux all the time. So, if that is the case, you may occasionally want to run the following command:

```
/etc/cron.daily/updatedb.cron
```

You will need to be root on your system when doing that. That will allow `locate` to work properly.

So, if you know you need to find all the "finger" files, you could run:

```
locate finger
```

It should return something like:

```
/usr/bin/finger
/usr/lib/irc/script/finger
/usr/man/man1/finger.1
/usr/man/man8/in.fingerd.8
/usr/sbin/in.fingerd
```

One thing to note, however, is that `locate` not only returns hits based on file name, but also on path name. So if you have a `/home/djb/finger/` directory on your system, it would get returned along with all files in the directory.

6.1.5 "info" Pages

While `man` is the most ubiquitous documentation format, `info` is much more powerful. It provides hypertext links to make reading large documents much easier and many features for the documentation writer. There are some very complete `info` documents on various aspects of Red Hat (especially the portions from the GNU project).

To read `info` documentation, use the `info` program without any arguments. It will present you with a list of available documentation. If it can't find something, it's probably because you don't have the package installed that includes that documentation. Install it with RPM and try again.

If you're comfortable using emacs, it has a built in browser for `info` documentation. Use Ctrl-h Ctrl-i to see it.

The `info` system is a hypertext based system. Any highlighted text that appears is a link leading to more information. Use Tab to move the cursor to the link, and press Enter to follow the link. Pressing p returns you to the previous page, n moves you to the next page, and u goes up one level of documentation. To exit `info`, press Ctrl-x Ctrl-c (control-x followed by control-c).

The best way to learn how to use `info` is to read the info documentation on it. If you read the first screen that `info` presents you'll be able to get started.

6.2 Help from the Internet Community

6.2.1 Red Hat Mailing Lists

If you can't find help for your problem on line and you have WWW access, you should see `http://www.redhat.com/support/mailing-lists/`. Here you can search the archives of the redhat-list. Many questions have already been answered there.

The subscription address for each of our lists is:

```
redhat-list-request@redhat.com
redhat-digest-request@redhat.com
redhat-announce-list-request@redhat.com
redhat-devel-list-request@redhat.com
rpm-list-request@redhat.com
axp-list-request@redhat.com
sparc-list-request@redhat.com
```

To subscribe, send mail to the address of the list you want to subscribe to with `subscribe` in the `Subject:` line.

To unsubscribe, send mail to the address of the list you want to unsubscribe from with `unsubscribe` in the `Subject:` line.

Then to send mail to the list, you just send it to the address above without the `-request` in the name. The only exception to this is the `redhat-digest` which is only a digest version the `redhat-list`. In that case the address is `redhat-list@redhat.com`.

6.2.2 USENET Newsgroups

Another good source of help is the `comp.os.linux` hierarchy on USENET. If you are familiar with news, you should check it out.

Chapter 7

Package Management with RPM

Red Hat **P**ackage **M**anager (RPM), is an open packaging system available for anyone to use, and works on both Red Hat Linux as well as other Linux and UNIX systems. Red Hat Software encourages other vendors to take the time to look at RPM and use it for their own products. RPM is distributable under the terms of the GPL.

For the end user, RPM provides many features that make maintaining a system far easier than it has ever been. Installing, uninstalling, and upgrading RPM packages are all one line commands, and all the messy details are taken care of for you. RPM maintains a database of installed packages and their files, which allows you to perform powerful queries and verification of your system. During upgrades RPM handles configuration files specially, so that you never lose your customizations – a feature that is impossible with straight `.tar.gz` files.

For the developer, RPM allows you to take source code for software and package it into source and binary packages for end users. This process is quite simple and is driven from a single file and optional patches that you create. This clear delineation of "pristine" sources and your patches and build instructions eases the maintenance of the package as new versions of the software are released.

At the time of this writing, a book is in the works that fully documents RPM, including: building and using packages, configuring RPM, using the RPM library in your programs, the RPM file format, and more. For more information contact Red Hat Software or check our web page at `http://www.redhat.com`.

7.1 RPM Design Goals

Before trying to understand how to use RPM, it helps to have an idea of what the design goals are.

Upgradability With RPM you can upgrade individual components of your system without completely reinstalling. When you get a new release of an operating system based on RPM (such as Red Hat Linux), you don't need to reinstall your machine (as you do with operating systems based on other packaging systems). RPM allows intelligent, fully-automated, in-place upgrades of your system. Configuration files in packages are preserved across upgrades, so you won't lose your customizations.

Powerful Querying RPM is also designed to have powerful querying options. You can do searches through your entire database for packages or just certain files. You can also easily find out what package a file belongs to and where it came from. The files an RPM package contains are in a compressed archive, with a custom binary header containing useful information about the package and its contents, allowing you to query individual packages quickly and easily.

System Verification Another powerful feature is the ability to verify packages. If you are worried that you deleted an important file for some package, simply verify the package. You will be notified of any anomalies. At that point, you can reinstall the package if necessary. Any configuration files that you modified are preserved during reinstallation.

Pristine Sources A crucial design goal was to allow the use of "pristine" software sources, as distributed by the original authors of the software. With RPM, you have the pristine sources along with any patches that were used, plus complete build instructions. This is a big advantage for several reasons. For instance, if a new version of a program comes out, you don't necessarily have to start from scratch to get it to compile. You can look at the patch to see what you *might* need to do. All the compiled-in defaults, and all of the changes that were made to get the software to build properly are easily visible this way.

This goal may only seem important for developers, but it results in higher quality software for end users too. We would like to thank the folks from the BOGUS distribution for originating the pristine source concept.

7.2 Using RPM

RPM has five basic modes of operation (not counting package building): installing, uninstalling, upgrading, querying, and verifying. This section contains an overview of each mode. For complete details and options try `rpm --help`, and see the rpm man page and the (previously mentioned) RPM book.

7.2.1 Installing

RPM packages typically have file names like `foo-1.0-1.i386.rpm`, which includes the package name (`foo`), version (`1.0`), release (`1`), and architecture (`i386`). Installing a package is as simple as:

```
$ rpm -ivh foo-1.0-1.i386.rpm
foo                           ####################################
```

As you can see, RPM prints out the name of the package (which is not necessarily the same as the file name, which could have been `1.rpm`), and then prints a succession of hash marks as the package is installed, as a sort of progress meter.

Installing packages is designed to be simple, but you can get a few errors:

Package Already Installed

If the package is already installed, you will see:

```
$ rpm -ivh foo-1.0-1.i386.rpm
foo                         package foo-1.0-1 is already installed
error: foo-1.0-1.i386.rpm cannot be installed
```

If you really want to install the package anyway, you can use `--replacepkgs` on the command line, which tells RPM to ignore the error.

Conflicting Files

If you attempt to install a package that contains a file that has already been installed by another packages, you'll see:

```
# rpm -ivh foo-1.0-1.i386.rpm
foo            /usr/bin/foo conflicts with file from bar-1.0-1
error: foo-1.0-1.i386.rpm cannot be installed
```

To cause RPM to ignore that error, use `--replacefiles` on the command line.

Unresolved Dependency

RPM packages can "depend" on other packages, which means that they require other packages to be installed in order to run properly. If you try to install a package for which there is such an unresolved dependency, you'll see:

```
$ rpm -ivh bar-1.0-1.i386.rpm
failed dependencies:
        foo is needed by bar-1.0-1
```

To handle this error you should install the requested packages. If you want to force the installation anyway (a bad idea since the package probably will not run correctly), use `--nodeps` on the command line.

7.2.2 Uninstalling

Uninstalling a package is just as simple as installing:

```
$ rpm -e foo
```

Notice that we used the package *name* "`foo`", not the name of the original package *file* "`foo-1.0-1.i386.rpm`".

You can encounter a dependency error when uninstalling a package if some other installed package depends on the one you are trying to remove. For example:

```
$ rpm -e foo
removing these packages would break dependencies:
        foo is needed by bar-1.0-1
```

To cause RPM to ignore that error and uninstall the package anyway (which is a bad idea since the package that depend on it will probably fail to work properly), use `--nodeps` on the command line.

7.2.3 Upgrading

Upgrading a package is almost just like installing.

```
$ rpm -Uvh foo-2.0-1.i386.rpm
foo                        ####################################
```

What you don't see above is the fact that RPM automatically uninstalled any old versions of the `foo` package. In fact you may want to always use `-U` to install packages, since it works fine even when there are no previous versions of the package installed.

Since RPM performs intelligent upgrading of packages with configuration files, you may see a message like:

```
saving /etc/foo.conf as /etc/foo.conf.rpmsave
```

This means that your changes to the configuration file may not be "forward compatible" with the new configuration file in the package, so RPM saved your original file, and installed a new one. You should investigate and resolve the differences between the two files as soon as possible to ensure that your system continues to function properly.

Since upgrading is really a combination of uninstalling and installing, you can encounter any errors from those modes, plus one more: If RPM thinks you are trying to upgrade to a package with an *older* version number, you will see:

```
$ rpm -Uvh foo-1.0-1.i386.rpm
foo       package foo-2.0-1 (which is newer) is already installed
error: foo-1.0-1.i386.rpm cannot be installed
```

To cause RPM to "upgrade" anyway, use `--oldpackage` on the command line.

7.2.4 Querying

Querying the database of installed packages is accomplished with `rpm -q`. A simple use is `rpm -q foo` which will print the package name, version, and release number of the installed package foo:

```
$ rpm -q foo
rpm-2.0-1
```

Instead of specifying the package name, you can use the following options with `-q` to specify what package(s) you want to query. These are called *Package Specification Options.*

- `-a` queries all currently installed packages.
- `-f <file>` will query the package owning <file>.
- `-F` is the same as `-f` except it takes filenames via stdin (e.g. `find /usr/bin | rpm -qF`).
- `-p <packagefile>` queries the package <packagefile>.
- `-P` is like `-p` except it takes package filenames from stdin (e.g. `find /mnt/cdrom/RedHat/RPMS | rpm -qP`).

There are a number of ways to specify what information to display about queried packages. The following options are used to select the information you are interested in. These are called *Information Selection Options.*

- `-i` displays package information such as name, description, release, size, build date, install date, vendor, and other miscellaneous information.
- `-l` displays the list of files that the package "owns".
- `-s` displays the state of all the files in the package. There are only two possible states, normal and missing.
- `-d` displays a list of files marked as documentation (man pages, info pages, README's, etc).
- `-c` displays a list of files marked as configuration files. These are the files you change after installation to adapt the package to your system (sendmail.cf, passwd, inittab, etc).

For those options that display file lists, you can add `-v` to your command line to get the lists in a familiar `ls -l` format.

7.2.5 Verifying

Verifying a package compares information about files installed from a package with the same information from the original package. Among other things, verifying compares the size, MD5 sum, permissions, type, owner and group of each file.

`rpm -V` verifies a package. You can use any of the *Package Selection Options* listed for querying to specify the packages you wish to verify. A simple use is `rpm -V foo` which verifies that all the files in the foo package are as they were when they were originally installed. For example:

- To verify a package containing particular file:

  ```
  rpm -Vf /bin/vi
  ```

- To verify ALL installed packages:

  ```
  rpm -Va
  ```

- To verify an installed package against an RPM package file:

  ```
  rpm -Vp foo-1.0-1.i386.rpm
  ```

> This can be useful if you suspect that your RPM databases are corrupt.

If everything verified properly there will be no output. If there are any discrepancies they will be displayed. The format of the output is a string of 8 characters, a possible "c" denoting a configuration file, and then the file name. Each of the 8 characters denotes the result of a comparison of one attribute of the file to the value of that attribute recorded in the RPM database. A single "." (period) means the test passed. The following characters denote failure of certain tests:

5 MD5 checksum

S File size

L Symbolic link

T File modification time

D Device

U User

G Group

M Mode (includes permissions and file type)

If you see any output, use your best judgment to determine if you should remove or reinstall the package, or somehow fix the problem.

7.3 Impressing Your Friends with RPM

RPM is a very useful tool for both managing your system and diagnosing and fixing problems. The best way to make sense of all the options is to look at some examples.

- Let's say you delete some files by accident, but you aren't sure what you deleted. If you want to verify your entire system and see what might be missing, you would enter:

```
rpm -Va
```

If some files are missing, or appear to have been corrupted, you should probably either re-install the package or uninstall, then re-install the package.

- Let's say you run across a file that you don't recognize. To find out which package owns it, you would enter:

  ```
  rpm -qf /usr/X11R6/bin/xjewel
  ```

  The output would look like:

  ```
  xjewel-1.6-1
  ```

- We can combine the above two examples in the following scenario. Say you are having problems with `/usr/bin/paste`. You would like to verify the package that owns that program but you don't know which package that is. Simply enter:

  ```
  rpm -Vf /usr/bin/paste
  ```

  and the appropriate package will be verified.

- If you are using a program and want to find out more information about it, you can enter the following to find out what documentation came with the package that "owns" that program (in this case `ispell`):

  ```
  rpm -qdf /usr/bin/ispell
  ```

  The output would be:

  ```
  /usr/man/man4/ispell.4
  /usr/man/man4/english.4
  /usr/man/man1/unsq.1
  /usr/man/man1/tryaffix.1
  /usr/man/man1/sq.1
  /usr/man/man1/munchlist.1
  /usr/man/man1/ispell.1
  /usr/man/man1/findaffix.1
  /usr/man/man1/buildhash.1
  /usr/info/ispell.info.gz
  /usr/doc/ispell-3.1.18-1/README
  ```

- You find a new koules RPM, but you don't know what it is. To find out some information on it, enter:

  ```
  rpm -qip koules-1.2-2.i386.rpm
  ```

  The output would be:

  ```
  Name        : koules        Distribution: Red Hat Linux Colgate
  Version     : 1.2                 Vendor: Red Hat Software
  Release     : 2               Build Date: Mon Sep 02 11:59:12 1996
  Install date: (none)          Build Host: porky.redhat.com
  Group       : Games           Source RPM: koules-1.2-2.src.rpm
  Size        : 614939
  Summary     : SVGAlib action game; multiplayer, network, sound
  Description :
  This arcade-style game is novel in conception and excellent in
  execution.  No shooting, no blood, no guts, no gore.  The play is
  simple, but you still must develop skill to play.  This version
  uses SVGAlib to run on a graphics console.
  ```

- Now you want to see what files the koules RPM installs. You would enter:

  ```
  rpm -qlp koules-1.2-2.i386.rpm
  ```

  The output is:

  ```
  /usr/man/man6/koules.6
  /usr/lib/games/kouleslib/start.raw
  /usr/lib/games/kouleslib/end.raw
  /usr/lib/games/kouleslib/destroy2.raw
  /usr/lib/games/kouleslib/destroy1.raw
  /usr/lib/games/kouleslib/creator2.raw
  /usr/lib/games/kouleslib/creator1.raw
  /usr/lib/games/kouleslib/colize.raw
  /usr/lib/games/kouleslib
  /usr/games/koules
  ```

These are just several examples. As you use the system you will find many more uses for rpm.

7.4 Other RPM Resources

For more information on RPM, check out the man page, the help screen (`rpm --help`), and the RPM documents available at

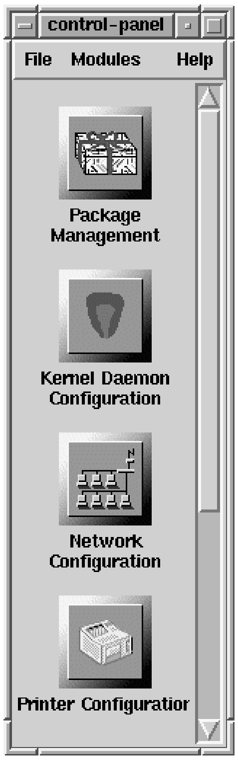
```
ftp://ftp.redhat.com/pub/redhat/code/rpm
```

As mentioned earlier there will also be a comprehensive RPM book available soon. When it is published there will be information on our web site, `http://www.redhat.com`.

There is also a mailing list for discussion of RPM related issues, called `rpm-list@redhat.com`. The list is archived on `http://www.redhat.com/support/mailing-lists/`. To subscribe, send mail to `rpm-list-request@redhat.com` with the word `subscribe` in the subject line.

Chapter 8

Control Panel



The control panel is a launching pad for a number of different system administration tools. They make your life easier by letting you configure things without remembering configuration file formats and awkward command line options.

To start the control-panel, start the X Window System as root with `startx` and type `control-panel` in an xterm. You will need to be root to run the control-panel tools successfully. You can do this as well if you already have X running as a normal user. Just type `su -c control-panel` and then type the root password when prompted. If you plan to do other tasks as root, you could type `su -` followed by the root password when prompted. You will then be given a root shell. Here, you will need to type `DISPLAY=:0 control-panel &` to get the control panel to work. The "DISPLAY" part tells the control panel to use your display. This syntax works only for a Bourne-compatible shell like bash. If you run tcsh, you would enter `setenv DISPLAY :0; control-panel &`.

Double clicking on an icon starts up a tool. Please note that you are not prevented from starting two instances of any tool, but doing so is a very bad idea because you may try to edit the same files in two places and end up overwriting your own changes. If you do accidentally start a second copy of a tool, you should quit it immediately. Also, do not manually edit any files managed by the control-panel tools while the tools are running. Similarly, do not run any other programs that may change those files while the tools are running.

8.1 User and Group Configuration

The tool shown in Figure 8.1 manages the users and groups on your system. In a sense, it is a graphical editor for `/etc/passwd` and `/etc/group`, and for `/etc/shadow` if you have enabled shadow passwords. With this tool you can add and remove users and groups, set login shells, full names, home directories, uid, gids, etc.

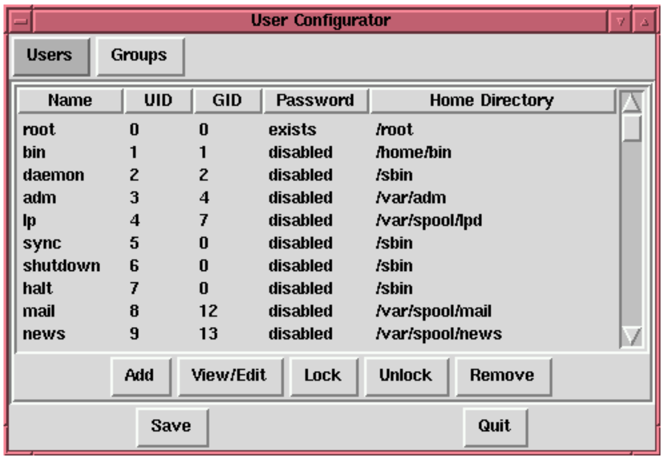


Figure 8.1: User and Group Configuration Panel

8.1.1 Adding or Editing a User

The same interface is used to add and edit users. To add a user, click on the **Add** button in the main window. A dialog box will pop up like the one in figure 8.2 will appear, with some defaults already filled in. To edit an existing user, double-click on the user in the main window, or click on the user and then on the **View/Edit** button.

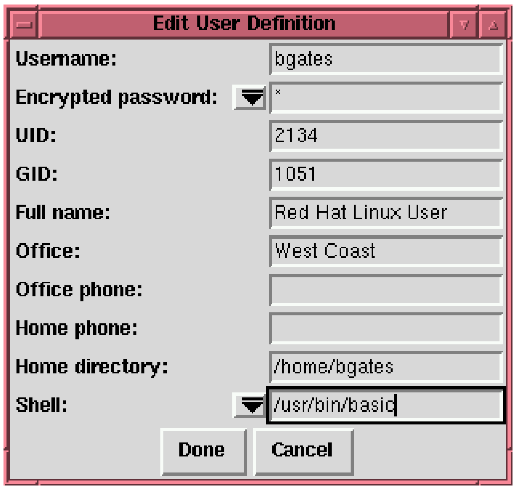


Figure 8.2: Add User Dialog Box

First enter the user name. This is *not* the user's first and last name; it is the id they will use to log on to the system. Do not include any spaces or colons, and do not enter more than 8 characters. Press the enter key when you are finished, and some of the other fields will be filled in with default values. You may change them if you wish, but it is not necessary. You can, at your option, enter the user's full name, office, and phone numbers. The login shell can be changed from the default (normally `/bin/bash`) either by choosing a selection from the drop-down menu or by simply typing in the shell like the other fields.

You need to do something about the password. There are five options on the **Encrypted Password** menu: Original, Change, No Password, Lock, and Unlock. No Password is a very bad option — with no password anyone can login to your system using this id. Choosing Lock will prevent anyone from logging in with this id but keep any password already assigned so that it can be unlocked later; Unlock will unlock a previously locked password. Original is blank (No Password) when you are adding a new user — the same dialog box is used for editing existing users, when Original is the user's original password. Usually you will choose Change. A small dialog box will appear where you will need to enter a password, and then confirm it by typing it a second time. Click on **Done** in the password entry window to confirm that you have type the password correctly.

The numbers that the tool provides for **UID** and **GID** will work. If you change them, you are on your own; the tool is designed to assume that if you change them, you know what you are doing and want to change them. If you don't know what UID or GID stand for, leave them alone.

If you have configured shadow passwords on your system you will have a Shadow Management button that will allow you to configure password and account expiry. By default (fields shown as blank have the default value), no expiry will be done. If you are unfamiliar with shadow password account management, you will probably wish to accept the default. Click on **Done** when you are done editing the expiry information.

When you are done editing the user, click on **Done** in the Edit User Definition dialog box to add the user.

If this user does not already have a home directory (and they won't unless you have already created the home directory), one will be created.

8.1.2 Removing a User

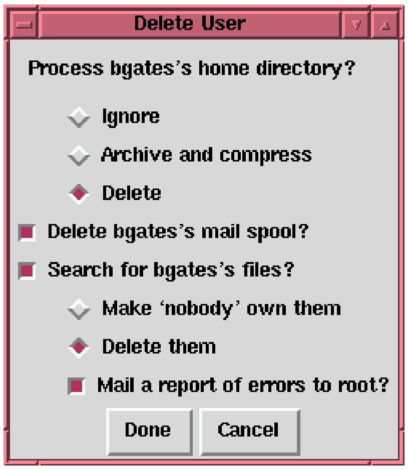


Figure 8.3: Delete User Dialog Box

To remove a user, select a user in the main window by clicking on it, and then click on **Remove**. You will be presented with the dialog box presented in Figure 8.3. There are three types of action you may take. You may ignore the user's home directory, leaving it alone, you can archive and compress it into a `.tar.gz` file, or you can delete it completely. You can delete the user's mail spool if you like. These first two actions will take place immediately after you confirm the deletion, before the tool accepts more input. You can search for files owned by the user on parts of the filesystem other than the user's home directory and either give ownership to the user `nobody` or delete them. If you choose to search for the user's files, you have the option of mailing a report of the errors to root. This is important because the searching will be done in the background as you continue to work; it may take a long time. To find files owned by `nobody`, execute the following command:

```
find / \( -group nobody -o -user nobody \) -print
```

Deactivating a User

Sometimes you may just want to temporarily *remove* a user, with the intention of reinstating the user later. The **Lock** and **Unlock** functions handle this. When you deactivate a user, the user's password is locked by prepending a '`*`' to it, preventing the user from logging in. You also have the option of "collapsing" the user's home directory with `tar` and `gzip`, which can save some space. When you reactivate a user, the password locking is removed, and if the home directory was collapsed, it is expanded back to normal. When you lock a user, you are presented with the same options as when you delete a user, but you are unlikely to want to delete any files.

8.1.3 Creating a New Group

To create a new group, first switch to group editing mode by selecting the **Groups** tab at the top of the main window. Clicking on **Add** brings up a dialog box to specify the group details. First enter a name for the new group. Just like a user name, the group name should have no spaces or colons, and should be no longer than 8 characters. You can set a group password; if you do, users who are not members of the group may assume group membership by using the newgrp program and providing it with the password. To add members to the group, enter their names in the User list field. When you have selected all the group members, click **Done**

8.1.4 Editing an Existing Group

To edit an existing group, double-click on the group in the main window, or click on the group and then on **View/Edit**. The same interface is used to edit an existing group as to create a new group.

8.2 File System Configuration

The file system configuration tool shown in Figure 8.4 allows you to easily examine and manipulate file system mount points, types, options, etc. It is very useful for manipulating a large number of file systems. You probably don't want to do this unless you are an experienced system administrator. If you do know what you are doing, it should be fairly straightforward as to how to mount, unmount, and add devices.

The **Reload** entry in the **FSM** menu causes the file system configuration tool to re-load `/etc/fstab` from your hard drive. If you edit `/etc/fstab` by hand while the file system configuration tool is running (which you shouldn't do), you probably want to reload.

To use the buttons along the bottom of the window, select a filesystem in the main window, and click on one of the buttons. The buttons perform the following functions:

- **Info**: Displays information on the filesystem, including the device, partition type, filesystem type, mount point and options, comment, size, percent used, etc.
- **Check**: Performs a filesystem check (`fsck`) on the partition. You can do this only on *unmounted* partitions — if the partition is mounted, you will get an error and you will have to unmount it first.
- **Mount**: Mounts the selected filesystem.
- **Unmount**: Unmounts the selected filesystem.

- **Format**: Creates a new filesystem on the selected partition. This will erase all data on the selected partition! You can do this only on *unmounted* partitions — if the partition is mounted, you will get an error and you will have to unmount it first.
- **Edit**: Brings up a dialog box where you can edit the mount point, mount options, comment, etc.

8.2.1 Adding NFS Mounts

To mount a filesystem via NFS, select **Add Mount** from the **NFS** menu. A dialog box will appear and you will have to fill in the following values:

- **Device**: Enter the host name and path, separated by a colon. For example, `foo.bar.com:/usr/exported` indicates the `/usr/exported` directory on `foo.bar.com`.
- **Mount Point**: Enter the directory on your machine where you want to mount the NFS filesystem. For example, `/mnt/foo`.
- **Options**: Enter the mount options for this filesystem. The default is `soft,intr,rw`. The `rw` means the filesystem is read-write, and `soft,intr` are options that make your system a little more resilient when the remote server goes down. See the mount man page for a complete list of available options.
- **Comment**: This optional field can be used to store a small comment.

After filling everything out properly, click on **OK**. At this point the entry is made in your `/etc/fstab`, but the filesystem is not actually mounted. To mount it, select it in the main window and click on **Mount**.

8.3 Printer Configuration

The printer configuration tool (`printtool`) maintains `/etc/printcap`, print spool directories, and print filters. The filters allow you to print many different types of files, including:

- plain text (ASCII) files

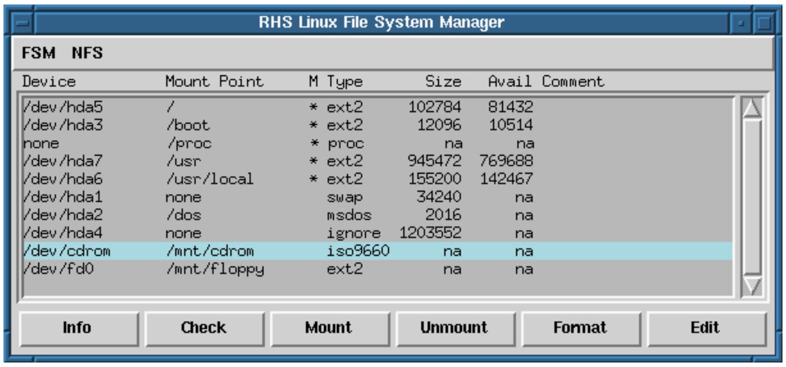


Figure 8.4: File System Configuration Panel

- PostScript files
- TEX `.dvi` files
- GIF, JPEG, TIFF, and other graphics formats
- RPMs

Simply printing a GIF or RPM file using `lpr` does "the right thing".

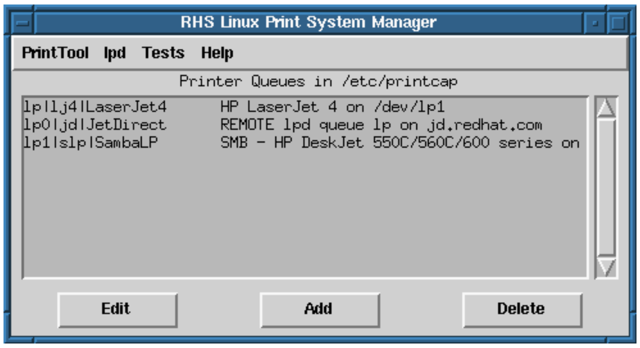


Figure 8.5: Print Tool

In order to create a new *print queue*, choose **Add**. Then, select what type of printer is being added. There are 3 types of print queues which can be configured with printtool:

- **Local**: print queues are for printers attached to a printer or serial port on your Red Hat Linux system.
- **Remote**: print queues are attached to a different system which you can access over a TCP/IP network.
- **SMB**: print queues are attached to a different system which uses LAN-Manager-type (SMB) networking.

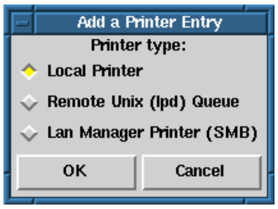


Figure 8.6: Selecting a Printer Type

After you choosing the printer type, a dialog requests further information about the print queue. All types of print queues require the following information:

- **Queue Name**: What the queue will be called. Multiple names can be specifed with the | (pipe) character separating entries.
- **Spool Directory**: This is the directory on the local machine where files are stored before printing occurs. Be careful to not have more than one printer queue use a given spool directory.
- **File Limit**: Maximum size print job accepted, in kilobytes (1 kb = 1024 bytes). A size of 0 indicates no limit should be imposed.
- **Input Filter**: Filters convert printed files into a format the printer can handle. Press **Select** to choose the filter which best matches your printer.

  In addition to configuring print queue able to print graphical and PostScript output you can configure a *text-only* printer, which will only print plain ASCII text. Most printer drivers are also able to print ASCII text without converting it to PostScript first; simply choose **Fast text printing** when you configure the filter. **Please Note:** This only works for non-PostScript printers.

- **Suppress Headers**: Check this if you don't want a header page to be printed at the beginning of each print job.

For *local* printers, the following are also required:

- **Printer Device**: Usually `/dev/lp1`, the name of the port which the printer is attached to. Serial printers are usually on `/dev/ttyS?` ports. You will need to manually configure serial parameters.

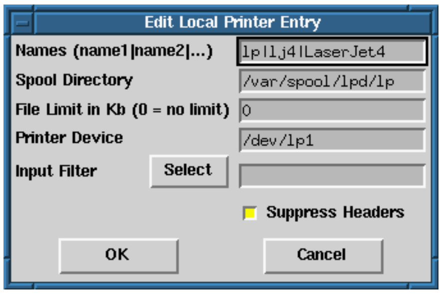


Figure 8.7: Adding a Local Printer

For *remote* printers, fill in the following information:

- **Remote Host**: Hostname of the remote machine hosting the printer.
- **Remote Queue**: Name of the queue to print to on the remote machine.

The remote machine must be configured to allow the local machine to print on the desired queue. Typically `/etc/hosts.lpd` controls this.

For *SMB* printers, fill in the following information:

- **Hostname of Printer Server**: Name of the machine to which the printer you want to use is attached.
- **IP number of Server**: The IP address of the machine to which the printer you want to use is attached; this is optional.

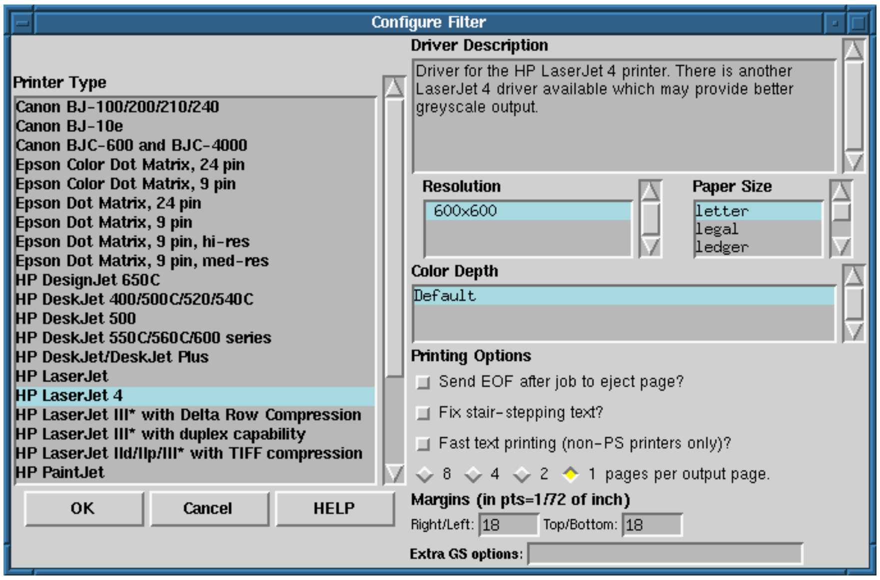


Figure 8.8: Configuring a Print Filter

- **Printer Name**: Name of the printer on which you want to print.
- **User**: Name of user you must login as to access the printer (typically `guest` for Windows servers, or `nobody` for `samba` servers).
- **Password**: Password (if required) to use the printer (typically blank). Someone should be able to tell you this if you do not already know it.

Please Note: If you need to use a username and password for an SMB (LAN Manager) print queue, they are stored unencrypted in a local script and must be passed on the command line to the smbclient program. Thus, it is possible for another person to learn the username and password. It is therefore recommended that the username and password for use of the printer not to be the same as that for a user account on the local Red Hat Linux system, so that the only possible security compromise would be unauthorized use of the printer. If there are file shares from the SMB server, it is recommended that they also use a different password than the one for the print queue.

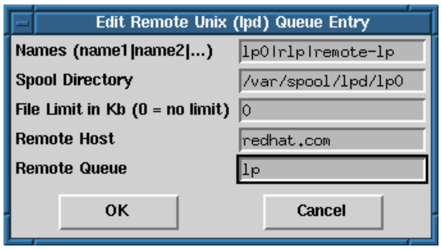


Figure 8.9: Adding a Remote Printer

After you have added your print queue, you may need to restart the printer daemon (`lpd`). To do so, choose **Restart lpd** from the **lpd** menu.

You may print a *test page* for any print queue you have configured. Select the type of test page you would like to print from the **Tests** menu.

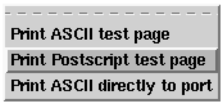


Figure 8.10: Printing a Test Page

8.4 Network Configuration

The network configuration tool (`netcfg`) shown in Figure 8.11 is designed to allow easy manipulation of parameters such as IP address, gateway address, and network address, as well as name servers and `/etc/hosts`.

Network devices can be added, removed, configured, activated, deactivated and aliased. Ethernet, arcnet, token ring, pocket (ATP), PPP, SLIP, PLIP and loopback devices are supported. PPP/SLIP/PLIP support works well on most hardware, but some hardware setups may exhibit unpredictable behavior. When using the Network Configuration Tool click **Save** to write your changes to disk, to quit without making any changes select **Quit**.

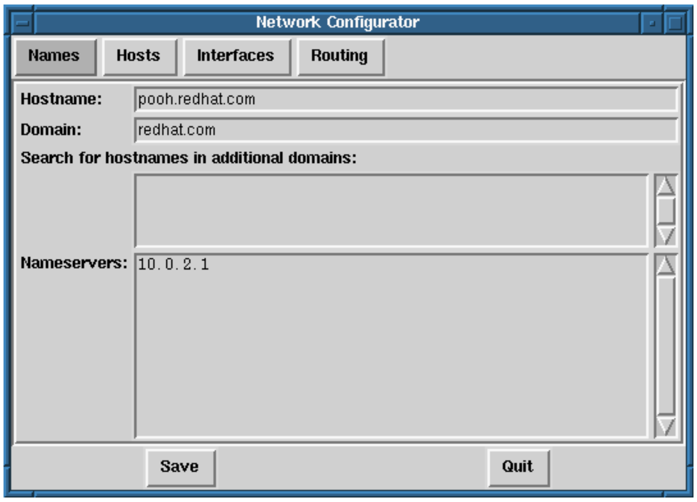


Figure 8.11: Network Configuration Panel

8.4.1 Managing Names

The **Names** panel of the Network Configuration tool serves two primary purposes: setting the hostname and domain of the computer, and determining which name server will be used to look up other hosts on the network. The Network tool is not capable of configuring a machine as a nameserver. To edit a field or add information to a field simply click on the field with the left mouse button and type the new information.

8.4.2 Managing Hosts

In the **Hosts** management panel you have the ability to add, edit, or remove hosts from the `/etc/hosts` file. Adding or editing an entry involves identical actions. An edit dialog box will appear, simply type the new information and click **Done** when you are finished. See Figure 8.12 for an example.

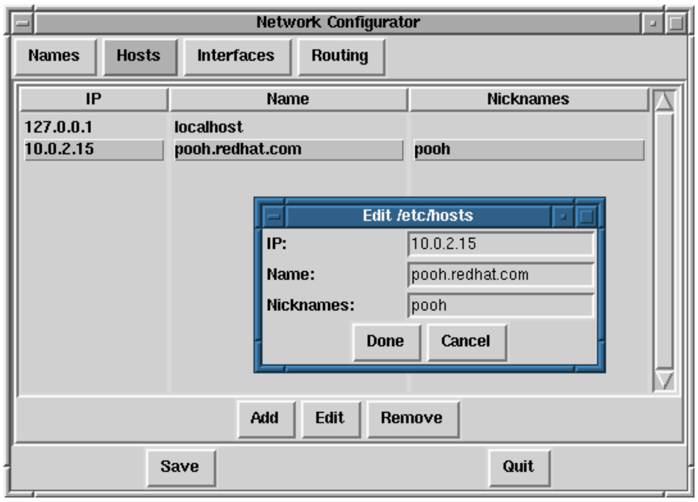


Figure 8.12: Adding/Editing Hosts

8.4.3 Adding a Networking Interface

If you have added a networking interface to your machine since installing Red Hat Linux, or you didn't configure your ethernet card at install time, you can configure it with a few clicks of a mouse.

Please Note: You may need to configure `kerneld` to load a driver for the network interface you are adding (e.g., `eth0`); see Section 8.6 for more information.

Begin adding an interface by clicking on **Interfaces** in the main panel. This will bring up a window of configured devices with a row of available options, see figure 8.13.

To add a device, first click the **Add** button then select the type of interface you want to configure from the box that appears (See Figure 8.14).

PPP Interface

Adding a PPP interface can be as simple as supplying the phone number, login name and password in the **Create PPP Interface** dialog shown in

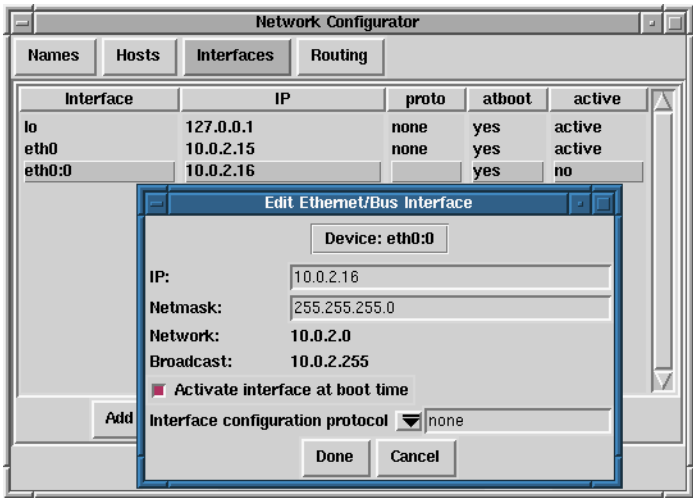


Figure 8.13: Configured Interfaces

Figure 8.14: Choose Interface Type

Figure 8.15. If you need to use PAP authentication for your PPP connection, choose **Use PAP authentication**. In many cases some degree of customization will be needed to establish a PPP connection. Choosing the **Customize** button will allow you to make changes to the hardware, communication, and networking settings for the PPP interface.

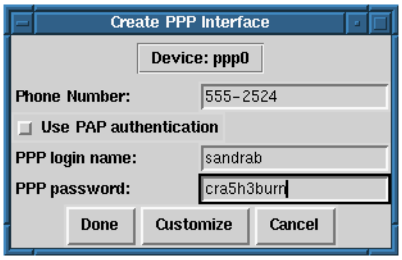


Figure 8.15: Create PPP Interface

SLIP Interface

In order to configure a SLIP interface you must first supply a phone number, login name, and password. This will supply the initial parameters for the chat script needed to establish a SLIP connection. When you choose **Done**, a dialog titled **Edit SLIP Interface** appears that enables you to further customize the hardware, communication and networking parameters for your SLIP interface.

PLIP Interface

To add a PLIP interface to your system you only have to supply the IP address, the remote IP address, and the Netmask. You can also select if you want to activate the interface at boot time.

Ethernet, Arcnet, Token Ring and Pocket Adaptor Interfaces

If you are adding an ethernet, arcnet, token ring or pocket adapter to your computer you will need to supply the following information:

- **Device**: This is determined by netconfig based on the devices already configured.
- **IP Address**: Enter an IP address for your network device.
- **Netmask**: Enter the network mask for your network device.

  The network and broadcast addresses are calculated automatically based on the IP address and netmask you enter.
- **Activate interface at boot time**: If you want the device to be configured automatically when your machine boots select this by clicking on the box.
- **Allow any user to (de)activate interface**: Check this if you want any user to be able to activate or deactivate the interface.
- **Interface configuration protocol**: If you have a BOOTP or DHCP server on your network and would like to use it to configure the interface, choose the appropriate option; otherwise, choose **none**.

After providing the configuration information for your new device, click **Done**. The device should appear in your **Interfaces** list as an inactive device. (The active column should have a label of **no**). To activate the new device, first select it with a mouse click and then choose on the **Activate** button. If it does not come up properly, you may need to reconfigure it by choosing on **Edit**.

8.4.4 Managing Routes

In the Routes management screen you have the ability to add, edit, or remove static networking routes. Adding or editing an entry involves identical actions, just like the Hosts panel. An edit dialog box will appear, simply type the new information and click **Done** when you are finished. See figure 8.16 for an example.

8.5 Time and Date

The time machine allows you to change the time and date by clicking on the appropriate part of the time and date display and clicking on the arrows to change the value.

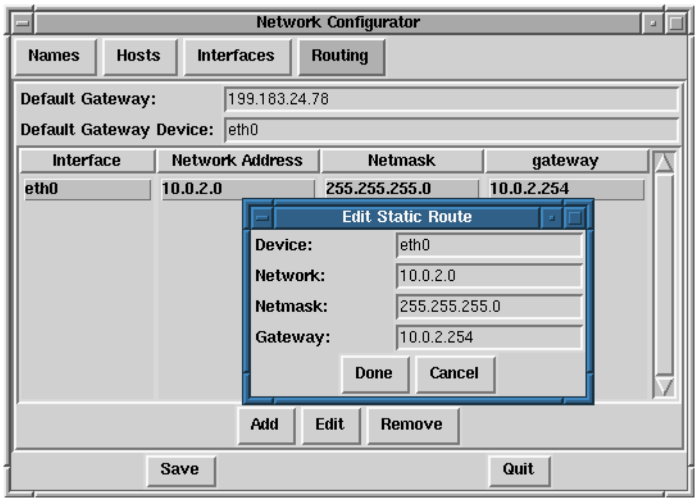


Figure 8.16: Adding/Editing Routes

The system clock is not changed until you click on the **Set System Clock** button.

Click on **Reset Time** to set the time machine time back to that of the system.

Please Note: Changing the time can seriously confuse programs that depend on the normal progression of time, and could possibly cause problems. Try to quit as many applications and processes as possible before changing the time or date.

8.6 Kernel Daemon Configuration

As discussed in Chapter 5, Red Hat Linux includes `kerneld`, the Kernel Daemon, which automatically loads some software and hardware support into memory as it is needed, and unloads it when it is no longer being used.

The tool shown in Figure 8.17 manages the configuration file for kerneld. While kerneld can load some things, such as filesystems, without explicit configuration, it needs to be told what hardware support to load when it is presented with a generic hardware request.

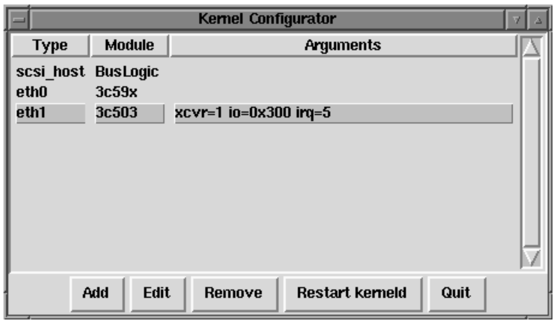


Figure 8.17: Kernel Module Management

For instance, when the kernel wants to load support for ethernet, kerneld needs to know which ethernet card you have, and if your ethernet card requires special configuration, it needs to know about that, too.

8.6.1 Changing Module Options

To change the options being given to a module when it is loaded, click on the line to select it, then click the **Edit** button. kernelcfg will bring up a window which looks like Figure 8.18 The options which kernelcfg knows about them (normally all available options) each have their own field. Normally, you will want to ignore the **Other arguments** field. Some modules normally take no arguments; just in case, they have an **Arguments** field which allows you to enter configuration information.

8.6.2 Changing Modules

To change which module gets invoked to provide a generic service, such as an ethernet card or SCSI host adapter module, you need to delete the old one and add a new one. To delete a module, select it by clicking on it, then click on **Delete**. Then click on **Add** to add the new module, as explained in the following section.

If you have changed your SCSI controller (`scsi_hostadapter`), remember

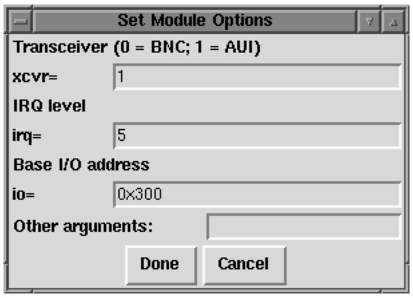


Figure 8.18: Editing Module Options

to make a new initial ramdisk with the `/sbin/mkinitrd` command as documented in section 5.1.2.

8.6.3 Adding Modules

To add a module of any type, click on the **Add** button. You will be presented with a dialog box (Figure 8.19) asking you to choose a module type. Ethernet is `eth`, Token Ring is `tr`, SCSI controllers are `scsi_hostadapter`, and so on. Click **OK** to continue to

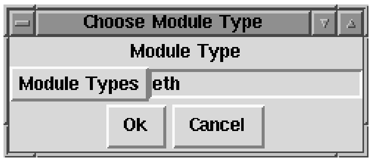


Figure 8.19: Adding a module

the next dialog box. If there is more than one module which can be used for the module type you have chosen, you will be presented with a dialog box (Figure 8.20) which asks which module you want to use, and may also ask for specifics about the type of module; for ethernet, for example, you need to choose from `eth0`, `eth1`, etc. When you are done, click **OK** again to continue to specify any module options in the next dialog box (Figure 8.20), which is the same as the dialog for editing a module.

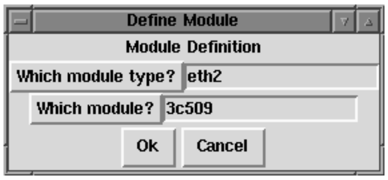


Figure 8.20: Selecting from available modules

8.6.4 Restarting Kerneld

The changes that you make with the Kernel Daemon Configuration tool will be made in the `/etc/conf.modules` file, which kerneld reads whenever it is started. Once you have made changes, you can restart kerneld by clicking on the **Restart kerneld** button. This will **not** cause any modules which are currently in use to be reloaded, it will only notify kerneld to use the configuration when it loads more modules in the future.

Chapter 9

Glint

Red Hat provides a graphical tool to aid in package installation and removal. It's called glint (Graphical Linux INstallation Tool) and runs under the X Windows System. It allows easy installation, uninstallation, upgrading, querying, and verification of packages. The interface is similar to the one found in many popular file managers and is simple to use.

Operations are performed in glint by selecting the packages to operate on and then selecting the operation to perform via pushbuttons. Installing a package places all of the components of that package on your system. Uninstalling one removes all traces of the package except for configuration files you have modified. Upgrading a package installs the newly available version and uninstalls all other versions that were previously installed. This allows quick upgrading to the latest releases of packages.

The query operation lets you examine the details of both installed or available package. You can view the description of the package, where and when it was built, the files in the package, and other attributes. All of the configuration and documentation components of each package are clearly marked as such to reduce the time you spend looking for them.

Using glint to perform all of these operations is the same as using rpm to do them from the command line. However, the graphical nature of glint often makes these operations easier to perform.

The normal way to handle glint is to display the available packages and files, select the ones you want to operate on, and then press a button or choose a menu item that performs the operation. For instance, you can install several packages with a few button clicks.

9.1 Starting glint

To start glint, simply run `glint &` from any X terminal window. That will bring up a window that looks like the one in figure 9.1. Any user can use glint to query and verify packages, but if you need to install, uninstall, or upgrade packages be sure to run glint as root.

There are two main parts to the glint interface. The first, on the left, allows you to browse and select the packages installed on your system. The right side contains buttons that manipulate the selected packages.

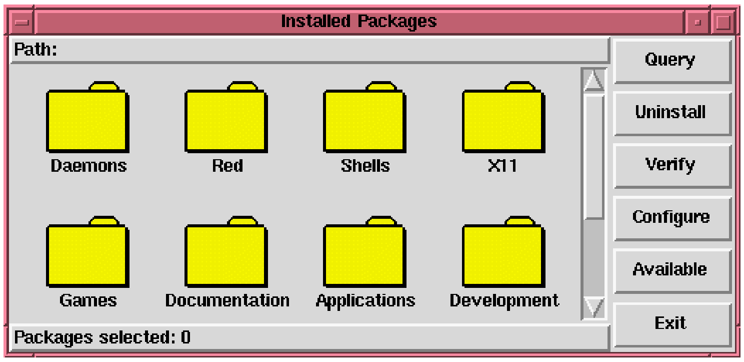


Figure 9.1: Main Glint Window

9.2 The Package Display

Each folder icon in glint represents a group of packages. Each group can contain other groups, which allows for flexible locations of packages. Groups are used to place packages that perform similar functions in similar locations. For example, Red Hat includes many application programs such as editors and spreadsheets. All of the text based ones appear in the "Applications" group. Inside of that group, there is another grouping for all of the editors that are shipped.

By convention, groups are written in the same way as UNIX paths. The top most group is written first, and subsequent groups follow with a slash separating the group names. This means that an X-based drawing program appears in the X11/Applications/Graphics group.

To view the packages and subgroups within a group, double click the left mouse button on a group's folder icon. The window then changes to show what that package contains. The top line of the package display shows which group you're currently looking in, as well as the groups leading to the current one. To return to the previous group you were looking at, double click on the "Back" folder, which is always in the upper left hand corner of the folder area (though it often gets scrolled away).

If you'd like to examine a subgroup in a new window, double click the middle mouse button on its folder. If your mouse has only two buttons, click both. This will create a new window with that group in it.

9.2.1 Context Sensitive Menus

Pressing the right mouse button on any icon in the package window brings up a small, *context sensitive* menu. The exact items it contains depends on exactly where you press it. They all contain options to select or deselect the item, and many let you install, uninstall, query, upgrade, or verify the item you clicked on. There's more information on how to do these things later.

To choose an item from a context sensitive menu, press and hold the right mouse button on a icon. While still holding the right button down, move the mouse pointer over the item you'd like to select (which will then become highlighted). Release the right mouse button to select that item and make the menu disappear.

9.2.2 Selecting Packages

To select a single package, click the left mouse button on it. You'll notice a thin border appear around the package's icon (as shown in figure 9.2) which shows that it's currently selected. To unselect it, click the left mouse button on it and the border will disappear. The number of packages currently selected is always displayed at the bottom of the window. A group's folder icon displays the number of packages inside of that group have been selected, or `All` if all of them have been.

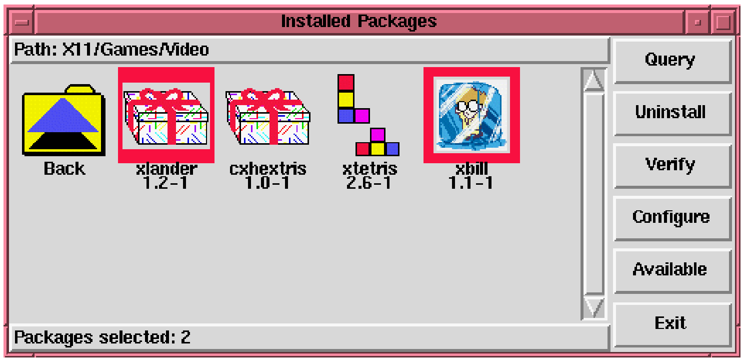


Figure 9.2: Selecting Packages in Glint

The context sensitive menu for a package also allows easy selection and unselection. Using the `select` and `unselect` options on a package's icon selects or unselects that package, while those options on a group's folder icon select and unselect all of the packages in that group. Using these menu options makes selecting groups of packages much quicker than selecting each package individually.

9.2.3 Viewing Available Packages

To see what packages are available for you to install, choose the `Available` pushbutton from any glint window. After a few moments, a new window, like the one shown in figure 9.3, will appear. The differences in the title and buttons indicate that this window is listing packages you may install. Navigating through these packages and selecting them is the same as in the other glint windows.

If you get an error message from glint saying that it can't find any RPMs, see the section below on Configuration.

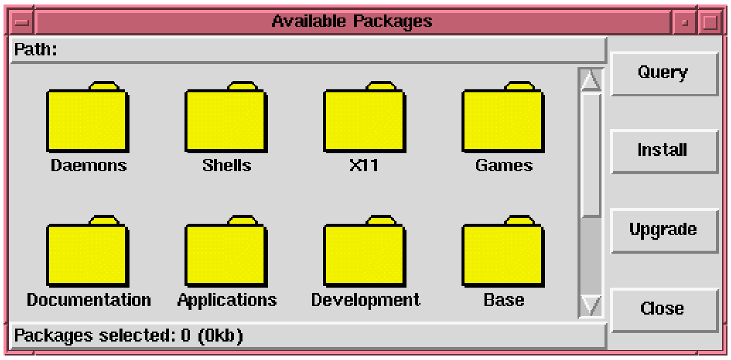


Figure 9.3: Available Window

9.3 Configuration

The only configuration information glint needs is the path to new RPMs. When you're using your Red Hat Linux CD-ROM, this will probably be /mnt/cdrom/RedHat/RPMS, which is the default path for glint. If you download new RPMs from the Internet or want to install RPMs via a NFS mounted CD-ROM this path will probably be different for you.

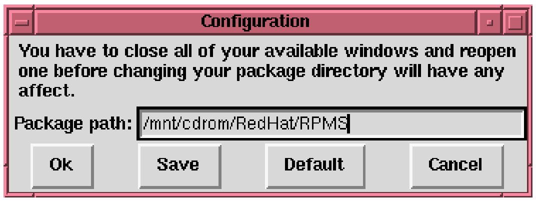


Figure 9.4: Configuration Window

To change this path, first be sure to close all of the windows listing available packages you may have open. Then choose the `Configuration` option from one of the remaining windows. This will open a dialog box like the one shown in figure 9.4. Here you can type the full path to the RPMs you'd like to look at. Choosing the `Save` button will save this path, making it the default for future glint sessions. The `Default` button restores the path to the one that glint used when it started.

After changing this path and closing the dialog box, you can use the `Available` button to view the packages available in the new location.

9.4 Package Manipulation

9.4.1 Querying Packages

The easiest way to query a single package or group is to use the query option from the icon's context sensitive menu. If you want to query a more diverse set of packages, select them all and use the `Query` button in one of the windows.

Using either of these methods creates a window like the one shown in figure 9.5. If you choose only one package, it will look a bit different however, so some of this won't apply.

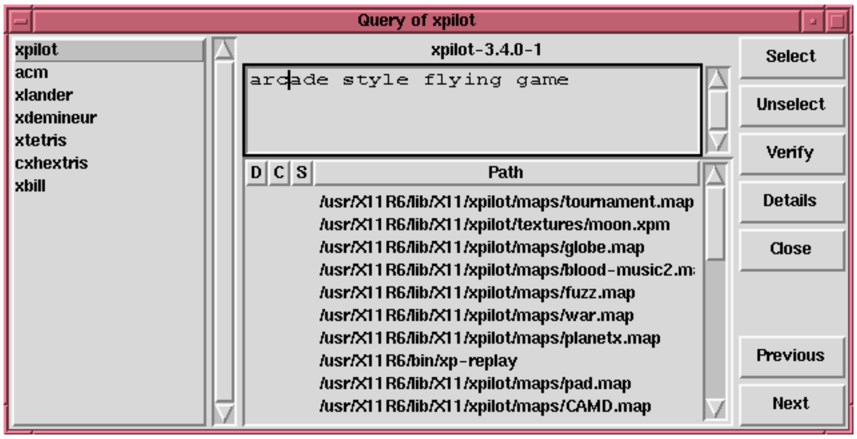


Figure 9.5: Query Window

On the very left of the window is a list of the packages that have been queried. Selecting one of them will change the information in the rest of the window. You may step through them in order by using the `Next` and `Previous` buttons on the right side of the window.

The name, version, and release of the current package are in the top middle of the query window. Immediately below this is the description of the package, which can be quite large. A scroll bar is there to let you read the whole thing.

Below the description is a list of the files contained in the package. Along with the full path to the file, the file list tells you a couple of other things. If a `D` appears to the left of the path, that file is a documentation file and would be a good thing to read. If a `C` appears there, then the file is a configuration file. A * means that the correct version of that file is not installed on your system. This can occur because a more recent version of a package was installed or because two packages contain different versions of the same file.

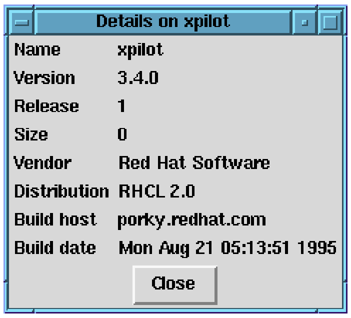


Figure 9.6: Query Details

More information on a package can be seen by clicking on the `Details` push-button. A window like the one in figure 9.6 will then appear. This lists more information about the package being displayed in the main query window. When you select a new package in the query window, the information in the details window will change to reflect your new choice.

You may also select, unselect, or verify a package while querying it by using the buttons provided. Click on the `Close` button when you are finished looking at the packages.

9.4.2 Verifying Packages

Verifying a package checks all of the files in the package to insure they match the ones present on your system. The checksum, file size, permissions, and owner attributes are all checked against the database. This check can be used when you suspect that one of programs files has become corrupted due to the installation of new programs.

Choosing the packages to verify is the same as choosing the packages to query. Select the packages and use the `Verify` button or choose the `Verify` entry from a context sensitive menu. A window opens like the one in figure 9.7.

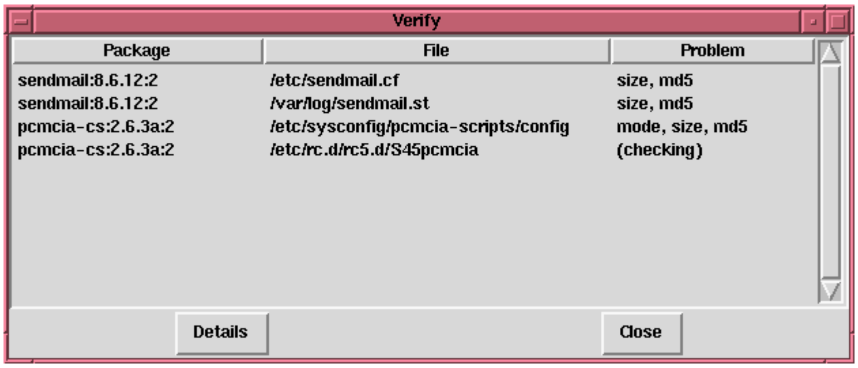


Figure 9.7: Verify Window

The three columns in this window describe the package with a problem in it, the file that has the problem, and a brief description of the discrepancies that were found. While the check is running, the current file being checked appears as the last element in the list, and the problem is listed as `(checking)`. A full list of the problems that can be found through verification appears in figure 9.8.

Problem	**Description**
missing	The file is no longer on your system
mode	permission bits have changed
size	file's size has changed
uid	owner's uid has changed
gid	owner's gid has changed
md5	the md5 checksum has changed
link	the file is a symlink to the wrong place

Figure 9.8: Possible Problems found by Verification

To get more information on the problems found with a file, double click on the file's path. A window like the one shown in figure 9.9 lists the expected and current values of the attributes that are amiss.

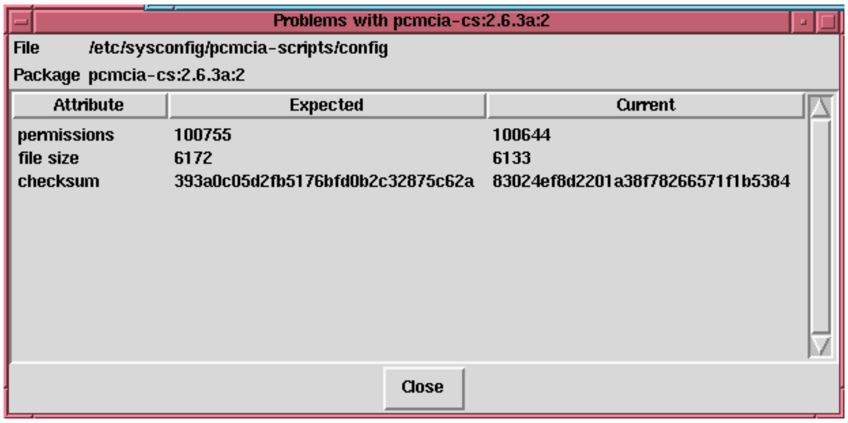


Figure 9.9: Verification Details

9.4.3 Installing New Packages

Installing new packages from glint is very simple. First look at the packages available for installation (see the section 9.2.3 for how to do this). You may select any number of these (and query them if you're not sure what they are) for installation (in the same manner you select packages for verification.) If you want to install a single package or group, the context sensitive menus provide a shortcut for doing so. Figure 9.10 shows a window with some packages selected for installation.

After you've begun the installation, a window appears like the one in figure 9.11. It tracks the progress of the installation so you'll know something is happening. The top bar shows how much of the current package (whose name is listed inside of it) has been installed while the bottom graph shows how much of the total installation has been finished. The number of packages, package sizes, and time estimates are continually updated.

If a problem occurs during the installation, a window will appear listing any errors that occurred. If this happens, you should correct the problems and then try again.

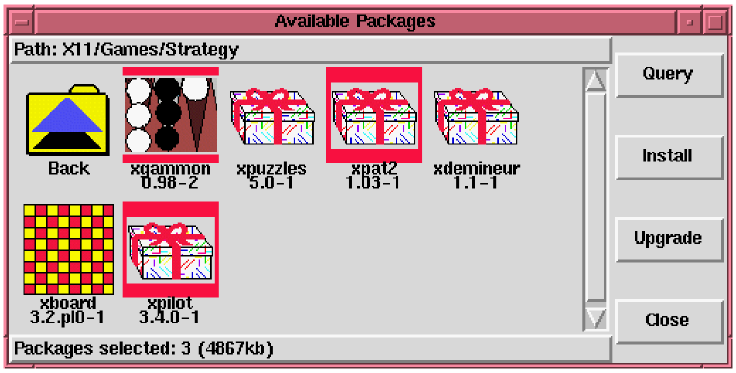


Figure 9.10: Packages Selected for Installation

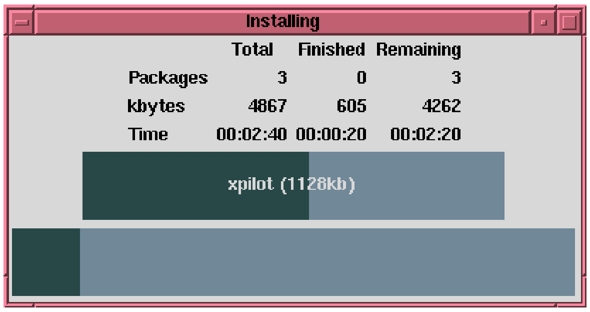


Figure 9.11: Installation Progress

After the installation has completed, the package and groups that have been installed are moved from the available window to the main glint window to show you that they have been successfully installed.

Upgrading Packages

When a new version of a package has been released, it is easy to install it on your system. Select the packages from the window of available packages in the same way you select packages for installation. Both the `Upgrade` button and the context sensitive menus will begin the upgrade.

During the upgrade, you'll see a progress indicator like the one for installing packages. When it's finished, the installed packages will appear in the the main glint windows and any old versions of the packages will be removed.

It is much better to use the upgrade option than to uninstall the old versions of a package and then install the new one. Using upgrade ensures that any changes you made to package configuration files get preserved properly, while doing it manually could cause those changes to be lost.

If you run out of disk space during an installation, the install will fail. However, the package which was being installed when the error occurred may leave some files around. To clean this up, reinstall the package after you've made more disk space available.

9.4.4 Uninstalling Packages

Uninstalling a package is not the same as upgrading one. When a package is uninstalled, any files it uses that are not needed by other packages on your system are removed. Changed configuration files are copied to `<filename>.rpmsave` so you can reuse them later.

Like verifying and querying packages, you can remove a package through the buttons on the right of the glint window or through a context sensitive menu. Remember that when you make a choice from a group's menu, the operation is performed on all of the packages in that group, so be careful!

Once you've begun the uninstall, glint asks for confirmation, showing a window like the one in figure 9.12. All of the packages that are about to be uninstalled are listed. You should look at them all to ensure you're not about to remove something you want to keep. Clicking the `Yes` button will start the uninstallation process. After it completes, the packages and groups that have been removed will disappear from any windows they were in.

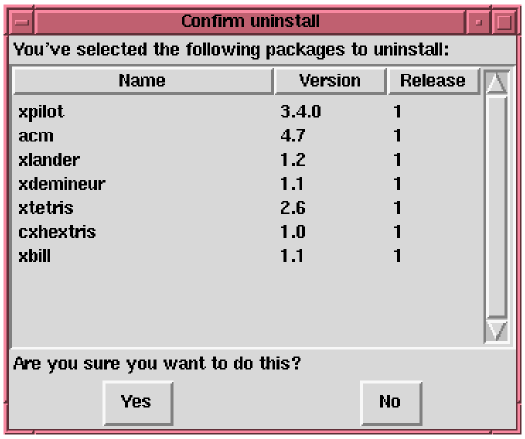


Figure 9.12: Uninstall Window

Chapter 10

System Administration

This chapter is an overview of the Red Hat Linux system. It will illustrate things that you may not know about the system and things that are somewhat different from other UNIX systems. Note that most system administration task are performed with the control-panel, which is covered in chapter 8.

10.1 Filesystem Structure

Red Hat Software is committed to the Linux File System Standard, a collaborative document that defines the names and locations of many files and directories. We will continue to track the standard to keep Red Hat compliant.

While compliance with the standard means many things, the two most important are compatibility with other compliant systems, and the ability to mount the `/usr` partition read-only. The `/usr` partition contains common executables and is not meant to be changed by users. Because of the, the `/usr` partition can be mounted from the CD-ROM or from another machine via read-only NFS. The current Linux Filesystem Standard (FSSTND) document is the authoritative reference to any FSSTND compliant filesystem, but the standard leaves many areas undefined or extensible. In this section we provide an overview of the standard and a description of the parts of the filesystem not covered by the standard.

The complete standard can be viewed at:

`http://www.pathname.com/fhs/`

10.1.1 Overview of the FSSTND

The directories and files noted here are a small subset of those specified by the FSSTND document. Check the latest FSSTND document for the most up to date and complete information.

The `/etc` Directory

The `/etc` directory is reserved for configuration files that are local to your machine. No binaries are to put in `/etc`. Binaries that were in the past put in `/etc` should now go into `/sbin` or possibly `/bin`.

The `X11` and `skel` directories should be subdirectories of `/etc`:

```
/etc
  |- X11
  +- skel
```

The `X11` directory is for X11 configuration files such as `XF86Config`. The `skel` directory is for "skeleton" user files, which are files used to populate a home directory when a user is first created.

The `/lib` Directory

The `/lib` directory should contain only those libraries that are need to execute the binaries in `/bin` and `/sbin`.

The `/sbin` Directory

The `/sbin` directory is for executables used only by the root user, and only those executables needed to boot and mount `/usr` and perform system recovery operations. The FSSTND says:

> "/sbin typically contains files essential for booting the system in addition to the binaries in /bin. Anything executed after /usr is known to be mounted (when there are no problems) should be placed in /usr/sbin. Local-only system administration binaries should be placed into /usr/local/sbin."

At a minimum, the following programs should be in `/sbin`:

```
clock, getty, init, update, mkswap, swapon, swapoff,
halt, reboot, shutdown, fdisk, fsck.*, mkfs.*, lilo,
arp, ifconfig, route
```

The `/usr` Directory

The `/usr` directory is for files that are shareable across a whole site. The `/usr` directory usually has its own partition, and it should be mountable read only. The following directories should be subdirectories of `/usr`:

```
/usr
  |- X11R6
  |- bin
  |- dict
  |- doc
  |- etc
  |- games
  |- include
  |- info
  |- lib
  |- local
  |- man
  |- sbin
  |- share
  +- src
```

The `X11R6` directory is for the X Window System (XFree86 on Red Hat Linux), bin is for executables, doc is for random non-man-page documentation, etc is for site-wide configuration files, include is for C header files, info is for GNU info files, lib is for libraries, man is for man pages, sbin is for system administration binaries (those that do not belong in /sbin), and src is for source code.

The `/usr/local` Directory

The FSSTND says:

> "The /usr/local hierarchy is for use by the system administrator when installing software locally. It needs to be safe from being overwritten when the system software is updated. It may be used for programs and data that are shareable amongst a group of machines, but not found in /usr."

The `/usr/local` directory is similar in structure to the `/usr` directory. It has the following subdirectories, which are similar in purpose to those in the `/usr` directory:

```
/usr/local
      |- bin
      |- doc
      |- etc
      |- games
      |- include
      |- info
      |- lib
      |- man
      |- sbin
      +- src
```

The `/var` directory

Since the FSSTND requires that you be able to mount `/usr` read-only, any programs that write log files or need spool or lock directories probably should write them to the `/var` directory. The FSSTND says `/var` is for

> "... variable data files. This includes spool directories and files, administrative and logging data, and transient and temporary files."

The following directories should be subdirectories of `/var`:

```
/var
  |- log
  |- catman
  |- lib
  |- local
  |- named
  |- nis
  |- preserve
  |- run
  |- lock
  |- tmp
  +- spool
      |- at
      |- cron
      |- lpd
      |- mail
      |- mqueue
      |- rwho
      |- smail
      |- uucp
      +- news
```

System log files such as `wtmp` and `lastlog` go in `/var/log`. The `/var/lib` directory also contains the RPM system databases. Formatted man pages go in `/var/catman`, and lock files go in `/var/lock`. The `/var/spool` directory has subdirectories for various systems that need to store data files.

10.1.2 `/usr/local` in Red Hat Linux

In Red Hat Linux, the intended use for `/usr/local` is slightly different from that specified by the FSSTND. The FSSTND says that `/usr/local` should be where your store software that is to remain safe from system software upgrades. Since system upgrades from Red Hat Software are done safely with the RPM system and Glint, you don't need to protect files by putting them in `/usr/local`. Instead, we recommend you use `/usr/local` for software that is local to your machine.

For instance, let's say you have mounted `/usr` via read-only NFS from *beavis*. If there is a package or program you would like to install, but you are not allowed to write to *beavis*, you should install it under `/usr/local`. Later perhaps, if you've managed to convince the system administrator of *beavis* to install the program on `/usr`, you can uninstall it from `/usr/local`.

10.2 Special Red Hat File Locations

In addition to the files pertaining to the RPM system that reside in `/var/lib/rpm` (see chapter 7), there are two other special locations that are reserved for Red Hat Linux configuration and operation.

The control-panel and related tools put lots of stuff in `/usr/lib/rhs`. There is probably nothing here that you would want to edit. It is mostly small scripts, bitmaps and text files.

The other location, `/etc/sysconfig`, stores configuration information. The major users of the files in this directory are the scripts that run at boot time. It is possible to edit these by hand, but it would be better to use the proper control-panel tool.

10.3 Users, Groups and User-Private Groups

Managing users and groups has traditionally been tedious. Red Hat Linux has a few tools and conventions that make user and groups easier to manage, and more useful.

The easiest way to manage users and groups is through the Users and Groups module of the control-panel (see section 8 for details on the control-panel and section 8.1 for details on the Users and Groups module).

You can also use `adduser` to create a new user from the command line.

10.3.1 Standard Users

Table 10.3.1 lists the standard users set up by the installation process (this is essentially the `/etc/passwd` file). The group id (GID) in this table is the *primary group* for the user. See section 10.3.3 for details on how groups are used.

User	UID	GID	Home Directory	Shell
root	0	0	/root	/bin/bash
bin	1	1	/bin	
daemon	2	2	/sbin	
adm	3	4	/var/adm	
lp	4	7	/var/spool/lpd	
sync	5	0	/sbin	/bin/sync
shutdown	6	0	/sbin	/sbin/shutdown
halt	7	0	/sbin	/sbin/halt
mail	8	12	/var/spool/mail	
news	9	13	/var/spool/news	
uucp	10	14	/var/spool/uucp	
operator	11	0	/root	/bin/bash
games	12	100	/usr/local/games	
gopher	13	30	/usr/lib/gopher-data	
ftp	14	50	/usr/rhs/ftp	
nobody	99	99	/root	

Figure 10.1: Standard Users

10.3.2 Standard Groups

Table 10.3.2 lists the standard groups as set up by the installation process (this is essentially the `/etc/group` file).

10.3.3 User Private Groups

Red Hat Linux uses a user private group (UPG) scheme, which makes UNIX groups much easier to use. The UPG scheme does not add or change anything in the standard UNIX way of handling groups. It simply offers a new convention for handling groups. Whenever you create a new user, by default, he or she has a unique group. The scheme works as follows:

User Private Group Each user has it's own primary group, to which only it is a member.

Group	GID	Members
root	0	root
bin	1	root,bin,daemon
daemon	2	root,bin,daemon
sys	3	root,bin,adm
adm	4	root,adm,daemon
tty	5	
disk	6	root
lp	7	daemon,lp
mem	8	
kmem	9	
wheel	10	root
mail	12	mail
news	13	news
uucp	14	uucp
man	15	
games	20	
gopher	30	
dip	40	
ftp	50	ftp
nobody	99	
users	100	

Figure 10.2: Standard Groups

umask = 002 The traditional UNIX umask is 022, which prevents other users *and other members of a user's primary group* from modifying a user's files. Since every user has their own private group in the UPG scheme, this "group protection" is not needed. A umask of 002 will prevent users from modifying other users' private files. The umask is set in `/etc/profile`.

SGID bit on Directories If you set the SGID bit on a directory (with `chmod g+s` *directory*), files created in that directory will have their group set to the directory's group.

Most computing sites like to create a group for each major project and assign people to the groups they need to be in. Managing files traditionally has been difficult, though, because when someone creates a file it is owned by the primary group he or she belongs to. When a single person works on multiple projects, it becomes hard to make the files owned by the group that is associated with that project. In the UPG scheme, groups are automatically assigned to files on a project-by-project basis, which makes managing group projects very simple.

Let's say you have a big project called *devel*, with many people editing the devel files in a `devel` directory. Make a group called `devel`, `chgrp` the `devel` directory to `devel`, and add the all the devel users to the `devel` group. Now, all the devel users will be able to edit the devel files and create new files in the `devel` directory, and these files will always retain their `devel` group. Thus, they will always be edit-able by other devel users.

If you have multiple projects like *devel*, and users who are working on multiple projects, these users will never have to change their umask or group when they move from project to project. The SGID bit on each project's main directory "selects" the proper group.

Since each user's HOME directory is owned by the user and their private group, it is safe to set the SGID bit on the HOME directory. However, by default, files are created with the primary group of the user, so the SGID bit would be redundant.

User Private Group Rationale

Since the UPG scheme is new, many people have questions about it, and they wonder why it is necessary. The following is the rationale for the scheme.

- You'd like to have a group of people work on a set of files in say, the `/usr/lib/emacs/site-lisp` directory. You trust a few people to mess around in there, but certainly not everyone.
- So you enter:

  ```
  chown -R root.emacs /usr/lib/emacs/site-lisp
  ```

  and you add the proper users to the group.
- To allow the users to actually create files in the directory you enter:

  ```
  chmod 775 /usr/lib/emacs/site-lisp
  ```

- But when a user creates a new file it is assigned the group of the users default group (usually `users`). To prevent this you enter

  ```
  chmod 2775 /usr/lib/emacs/site-lisp
  ```

  which causes everything in the directory to be created with the "emacs" group.
- But the new file needs to be mode 664 for another user in the emacs group to be able to edit it. To do this you make the default umask 002.
- Well, this all works fine, except that if your default group is "users", every file you create in your home directory will be writable by everybody in "users" (usually everyone).
- To fix this, you make each user have a "private group" as their default group.

At this point, by making the default umask 002 and giving everyone a private default group, you can easily set up groups that users can take advantage of without doing any magic. Just create the group, add the users, and do the above chown and chmod on the group's directories.

10.4 User Authentication with PAM

Programs which give users access to privileges of any sort need to be able to authenticate the users. When you log into a system, you provide your name and password, and the login process uses those to authenticate the login—to verify that you are who you say you are. Other forms of authentication than passwords are possible, and it is possible for the passwords to be stored in different ways.

PAM, which stands for "Pluggable Authentication Modules", is a way of allowing the system administrator to set authentication policy without having to recompile programs which do authentication. With PAM, you control how the modules are plugged into the programs by editing a configuration file.

Most Red Hat Linux users will never need to touch this configuration file. When you use RPM to install programs that need to do authentication, they automatically make the changes that are needed to do normal password authentication. However, you may want to customize your configuration, in which case you need to understand the configuration file.

10.4.1 Modules

There are four types of modules defined by the PAM standard. `auth` modules provide the actual authentication, perhaps asking for and checking a password, and set "credentials" such as group membership or kerberos "tickets". `account` modules check to make sure that the authentication is allowed (the account has not expired, the user is allowed to log in at this time of day, etc.). `password` modules are used to set passwords. `session` modules are used once a user has been authenticated to make it possible for them to use their account, perhaps mounting the user's home directory or making their mailbox available.

These modules may be *stacked*, so that multiple modules are used. For instance, rlogin normally makes use of at least two authentication methods: if "rhosts" authentication succeeds, it is sufficient to allow the connection; if it fails, then standard password authentication is done.

New modules can be added at any time, and PAM-aware applications can then be made to use them. For instance, if you have a one-time-password calculator system, and you can write a module to support it (documentation on writing modules is included with the system), PAM-aware programs can use the new module and work with the new one-time-password calculators without being recompiled or otherwise modified in any way.

10.4.2 Services

Each program which uses PAM defines its own "service" name. The login program defines the service type `login`, ftpd defines the service type `ftp`, etc. In general, the service type is the name of the program used to **access** the service, not (if there is a difference) the program used to **provide** the service.

10.4.3 The Configuration Files

The directory `/etc/pam.d` is used to configure all PAM applications. (This used to be `/etc/pam.conf` in earlier PAM versions; while the `pam.conf` file is still read if no `/etc/pam.d/` entry is found, its use is deprecated.) Each application (really, each **service**) has its own file. A file looks like this:

```
#%PAM-1.0
auth      required  /lib/security/pam_securetty.so
auth      required  /lib/security/pam_pwdb.so shadow nullok
auth      required  /lib/security/pam_nologin.so
account   required  /lib/security/pam_pwdb.so
password  required  /lib/security/pam_cracklib.so
password  required  /lib/security/pam_pwdb.so shadow nullok use_authtok
session   required  /lib/security/pam_pwdb.so
```

The first line is a comment. Any line that starts with a `#` character is a comment. The next three lines stack up three modules to use for login authorization. The first line makes sure that *if* the user is trying to log in as root, the tty on which they are logging in is listed in the `/etc/securetty` file *if* that file exists. The second line causes the user to be asked for a password and the password checked. The third line checks to see if the file `/etc/nologin` exists, and if it does, displays the contents of the file, and if the user is not root, does not let him or her log in.

Note that all three modules are checked, *even if the first module fails*. This is a security decision—it is designed to not let the user know why their authentication was disallowed, because knowing why it was disallowed might allow them to break the authentication more easily. You can change this behavior by changing `required` to `requisite`; if any `requisite` module returns failure, PAM fails immediately without calling any other modules.

The fifth line causes any necessary accounting to be done. For example, if shadow passwords have been enabled, the `pam_pwdb.so` module will check to see if the account has expired, or if the user has not changed his or her password and the grace period for changing the password has expired.

The sixth line specifies that if the login program changes the user's password, it should use the `pam_pwdb.so` module to do so. (It will do so only if an `auth` module has determined that the password needs to be changed—for example, if a shadow password has expired.)

The final line specifies that the `pam_pwdb.so` module should be used to manage the session. Currently, that module doesn't do anything; it could be replaced (or supplemented by stacking) by any necessary module.

Note that the order of the lines within each file matters. While it doesn't really matter much in which order `required` modules are called, there are other *control flags* available. While `optional` is rarely used, and never used by default on a Red Hat Linux system, `sufficient` and `requisite` cause order to become important.

Let's look at the `auth` configuration for `rlogin`:

```
auth  required    /lib/security/pam_securetty.so
auth  sufficient  /lib/security/pam_rhosts_auth.so
auth  required    /lib/security/pam_pwdb.so shadow nullok
auth  required    /lib/security/pam_nologin.so
```

That looks *almost* like the `login` entry, but there's an extra line specifying an extra module, and the modules are specified in a different order.

First, `pam_securetty.so` keeps root logins from happening on insecure terminals. This effectively disallows all root rlogin attempts. If you wish to allow them (in which case we recommend that you either not be internet-connected or be behind a good firewall), you can simply remove that line.

Second, `pam_nologin.so` checks `/etc/nologin`, as specified above.

Third, if `pam_rhosts_auth.so` authenticates the user, PAM immediately returns success to rlogin without any password checking being done. If `pam_rhosts_auth.so` fails to authenticate the user, that failed authentication is ignored.

Finally (if `pam_rhosts_auth.so` has failed to authenticate the user), the `pam_pwdb.so` module performs normal password authentication.

Note that if you do not want to prompt for a password if the securetty check fails, you can change the `pam_securetty.so` module from `required` to `requisite`

10.4.4 Shadow Passwords

The default `pam_pwdb.so` module can support shadow passwords. In order to convert your system to shadow passwords, use these commands:

```
cd /etc
pwconv5
chmod 600 passwd- shadow-
```

The `pam_pwdb.so` module will automatically detect that you are using shadow passwords and make all necessary adjustments.

10.4.5 More Information

This is just an introduction to PAM. More information is included with the system in `/usr/doc/pam*`, including a *System Administrators' Guide*, a *Module Writers' Manual*, an *Application Developers' Manual*, and the PAM standard, DCE-RFC 86.0. In addition, documentation is available from the Red Hat web site, at `http://www.redhat.com/linux-info/pam/`.

10.5 The Boot Process, Init, and Shutdown

10.5.1 System V Init

This section is a brief description of the internals of the boot process. It basically covers in detail how the machine boots using SysV Init and the differences between the original init used in older Linux releases.

Init is the program that gets run by the kernel at boot time. It is in charge of starting all the normal processes that need to run at boot time. These include the gettys that allow you to log in, NFS daemons, FTP daemons, and anything else you want to run when your machine boots.

SysV Init is fast becoming the standard in the Linux world to control the startup of software at boot time. This is because it is easier to use and more powerful and flexible than the traditional BSD init.

SysV init also differs from BSD init in that the config files are in a subdirectory of `/etc` instead of residing directly in `/etc`. This directory is called `rc.d`. In there you will find `rc.sysinit` and the following directories:

```
init.d
rc0.d
rc1.d
rc2.d
rc3.d
rc4.d
rc5.d
rc6.d
```

`init.d` contains a bunch of scripts. Basically, you need one script for each service you may need to start at boot time or when entering another runlevel. Services include things like networking, nfs, sendmail, httpd, etc. Services do not include things like setserial that must only be run once and then exited. Things like that should go in `rc.local` or `rc.serial`.

If you want `rc.local`, it should be in `/etc/rc.d`. Most systems include one even though it doesn't do much. You can also include an rc.serial in `/etc/rc.d` if you need to do serial port specific things at boot time.

The chain of events is as follows:

- The kernel looks in several places for `init` and runs the first one it finds
- `init` runs `/etc/rc.d/rc.sysinit`
- `rc.sysinit` does a bunch of necessary things and then runs `rc.serial` (if it exists)
- `init` runs all the scripts for the default runlevel.
- `init` runs `rc.local`

The default runlevel is decided in `/etc/inittab`. You should have a line close to the top like:

```
id:3:initdefault:
```

From this, you'd look in the second column and see that the default runlevel is 3, as should be the case for most systems. If you want to change it, you can edit `/etc/inittab` by hand and change the 3. Be very careful when you are messing with the inittab. If you do mess up, you can fix it by rebooting and doing:

```
LILO boot:  linux single
```

This *should* allow you to boot into single user mode so you can fix inittab.

Now, how does it run all the right scripts? If you enter `ls -l` on `rc3.d`, you might see something like:

```
lrwxrwxrwx 1 root root 17  13:11 S10network -> ../init.d/network
lrwxrwxrwx 1 root root 16  13:11 S30syslog -> ../init.d/syslog
lrwxrwxrwx 1 root root 14  13:32 S40cron -> ../init.d/cron
lrwxrwxrwx 1 root root 14  13:11 S50inet -> ../init.d/inet
lrwxrwxrwx 1 root root 13  13:11 S60nfs -> ../init.d/nfs
lrwxrwxrwx 1 root root 15  13:11 S70nfsfs -> ../init.d/nfsfs
lrwxrwxrwx 1 root root 18  13:11 S90lpd -> ../init.d/lpd.init
lrwxrwxrwx 1 root root 11  13:11 S99local -> ../rc.local
```

What you'll notice is that there are no real "files" in the directory. Everything there is a link to one of the scripts in the init.d directory. The links also have an “S” and a number at the beginning. The “S” means to start this particular script and a “K” would mean to stop it. The number is there just for ordering purposes. Init will start all the services based on the order they appear. You can duplicate numbers, but it will only confuse you somewhat. You just need to use a two digit number only, along with an upper case “S” or “K” to start or stop the services you need to.

How does init start and stop services? Simple. Each of the scripts is written to accept an argument which can be “start” and “stop”. You can execute those scripts by hand in fact with a command like:

```
/etc/rc.d/init.d/httpd.init stop
```

to stop the httpd server. Init just reads the name and if it has a “K”, it calls the script with the “stop” argument. If it has an “S” it calls the script with a “start” argument. Why all these runlevels? Some people want an easy way to set up machines to be multi-purpose. I could have a “server” runlevel that just runs httpd, sendmail, networking, etc. Then I could have a “user” runlevel that runs xdm, networking, etc.

10.5.2 Init Runlevels

Generally, Red Hat Linux runs in run level 3—full multiuser mode. The following runlevels are used in Red Hat Linux:

0 Halt.

1 Single user mode.

2 Multiuser mode, without NFS.

3 Full multiuser mode.

6 Reboot.

If your machine gets into a state where it will not boot due to a bad `/etc/inittab`, or will not let you log in because you have a corrupted `/etc/passwd` or have simply forgotten your password, boot into single user mode by typing `linux 1` at the LILO boot prompt. A very bare system will come up and you will be given a shell from which you can fix things.

10.5.3 Shutting Down

To shutdown Red Hat Linux, issue the `shutdown` command. You can read the shutdown man page for complete details, but the two most common usages are:

```
shutdown -h now
shutdown -r now
```

Each will cleanly shutdown the system. After shutting everything down, the first will halt the machine, and the second will reboot.

Avoid running the `reboot` or `halt` commands directly in order to prevent damage to your filesystem.

10.6 Rescue Modes

When things go wrong, there are several ways to work on fixing them. However, they require that you understand the system well. This manual can't teach you what to do, but we will present the ways that you can use our products to get into rescue modes where you can use your own knowlege to rescue the system.

10.6.1 Through LILO

If your system boots, but does not allow you to log in when it has completed booting, you can use the `single` or `emergency` boot option. At the `LILO boot:` prompt, type `linux single` in order to boot in single-user mode. In single-user mode, your local filesystems will be mounted, but your network will not be activated. In emergency mode, almost nothing will be set up. Only the root filesystem will be mounted, and it will be mounted read-only.

10.6.2 Emergency Boot Floppies

The installation floppy set can also be used as a rescue disk set. When you boot the boot floppy, type `rescue` at the boot prompt (on non-Intel platforms, add `rescue` as an option to the kernel you are booting, for example, `linux rescue`), and the install process will ask a few questions, request that you insert the second disk, and set up shells on VC 1 and VC 2. The shell that is running is ash, a minimal bourne shell with no history or command-line editing.

The `PATH` environment variable is set so that once you have mounted your root partition on `/mnt`, binaries from your root partition will be run if they do not exist on the floppy. Also, the libraries in /mnt/lib and /mnt/usr/lib will be used if needed. You may wish to change your `PATH` to not include `/bin` and `/usr/bin`, leaving `/mnt/bin`, `/mnt/usr/bin`, etc., while you are working on repairing your system. If you do that, you will need to use `/bin/umount` to unmount `/mnt`.

Limited versions of important utilities are available before you mount `/mnt`:

`mount` The `mount` command does not require that the device node being mounted already exist, and doesn't require the full name of the device. It assumes ext2 filesystems unless told otherwise.

```
mount /dev/sda1 /mnt -t ext2
```

is functionally equivalent to

```
mount sda1 /mnt
```

`ash` The full version

`cat` Does not accept any options

`chmod` The `chmod` command does not accept symbolic mode names, only octal ones.

`cpio` The full version

`e2fsck` The full version

`fdisk` The full version

`gzip` The full version

`gunzip` The full version

`insmod` The full version

`ls` The full version

`lsmod` Does not accept any options

`mkdir` No GNU long options

`mke2fs` The full version

`mknod` The `mknod` command does not accept symbolic mode names, only octal ones.

`open` The full version

`rm` No GNU long options

`rmmod` The full version

`sh` Symlink to `ash`

Appendix A

Module parameters

This appendix is provided to illustrate *some* of the possible parameters that *may* be needed by certain driver modules. It should be noted that in most cases the use of additional parameters is not needed.

A.1 CD-ROM parameters

Hardware	**Parameter**
Mitsumi CD-ROM	`mcd=`*port,irq*
Sony CDU 31 or 33 CD-ROM	`cdu31a_port=`*base_addr* `cdu31a_irq=`*irq*
Aztech CD268	`aztcd=`*port*
SB Pro or 16 compatible	`sbpcd=`*io_addr,sb_pro_setting*
ATAPI/IDE CD-ROM Drives	`hd`*x*`=cdrom`

Examples of the above would be:

Configuration	**Example**
non-IDE Mitsumi CD-ROM on port 340, IRQ 11	`mcd=0x340,11`
Sony CDU 31 or 33 at port 340, no IRQ	`cdu31a_port=0x340 cdu31a_irq=0`
Aztech CD-ROM at port 220	`aztcd=0x220`
ATAPI CD-ROM, jumpered as master on 2nd interface	`hdc=cdrom`
Panasonic-type CD-ROM on a SoundBlaster at port 230	`sbpcd=0x230,1`

Please Note: Most newer Sound Blaster cards come with IDE interfaces. You do not need to use `sbpcd` parameters, only use `hd`*x* parameters.

A.2 SCSI parameters

Hardware	**Parameter**
SCSI Seagate ST0x	`controller_type=1` `base_address=`*shmemaddr* `irq=`*irq*
SCSI Future Domain TMC-8xx	`controller_type=2` `base_address=`*shmemaddr* `irq=`*irq*
SCSI Trantor T128	t128=*base_addr,irq*
SCSI NCR-5380 Based	ncr5380=*port,irq,dma_channel*
Adaptec 152x, SB SCSI	aha152x=*port,irq,scsi_id,reconnect,parity*
Adaptec 1542	bases=*base_addr*
Buslogic	buslogic=*base_addr*
Pro Audio Spectrum / Studio 16	pas16=*port,irq*

Some examples would be:

Configuration	**Example**
Adaptec AHA1522 at port 330, IRQ 11, SCSI ID 7	`aha152x=0x330,11,7`
Adaptec AHA1542 at port 330	`bases=0x330`
Future Domain TMC-800 at CA000, IRQ 10	`controller_type=2` `base_address=0xca000 irq=10`

When a parameter has commas, make sure you do not put a space after a comma.

A.3 Ethernet parameters

Most ethernet drivers accept parameters to specify a base IO address and an IRQ as follows:

`io=`*base_addr* `irq=`*irq*

For example, for a 3com 3c509 ethernet card located at IO address 210 (IO addresses are usually in hexadecimal) and IRQ 10, use the following parameters for the `3c509` driver:

```
io=0x210 irq=10
```

You can use multiple ethernet cards in one machine. If each card uses a different driver (e.g., 3c509 and a DE425), you simply need to add aliases (and possibly options) for each card to `/etc/conf.modules`; for example:

```
alias eth0 3c509
options 3c509 io=0x210 irq=10
alias eth1 de4x5
options de4x5 io=0
```

See Section 8.6 for more information.

However, if any two ethernet cards use the same driver (e.g., two 3c509's or a 3c595 and a 3c905), you will need to compile a custom kernel with the ethernet driver built in. In that case, you will need to use "classic" LILO `boot:` parameters of the form:

`ether=`*irq,base_addr,interface*

For example:

```
LILO boot: linux ether=10,0x210,eth0 ether=11,0x300,eth1
```

(For more information about using more than one ethernet card, see the *Multiple-Ethernet* mini-HOWTO.)

Appendix B

Metro X

B.1 Special Features

B.1.1 Hot-Key Exit

Metro-X supports a hot-key exit. This is useful for terminating the X server if the last client in your `.xinitrc` or `.xsession` does not provide a way to exit the session. The exit key combination is Ctrl-Alt-Backspace.

B.1.2 Hot-Key Resolution Switching

When configuring Metro-X you can select several screen resolutions. You can then change resolutions on the fly while running your X server by pressing the appropriate key combination. To switch to the next resolution, press Ctrl-Alt-+. To switch to the previous resolution, press Ctrl-Alt--.

B.1.3 Hardware Panning

Metro-X supports hardware panning. This lets you set the virtual resolution higher than the monitor's physical resolution. Windows can then extend beyond the screen's physical boundaries and can be viewed by scrolling into the extended area.

B.1.4 Multiple Screens

Metro-X supports up to four screens which can all be controlled simultaneously with a single keyboard and mouse. This allows you to run many applications without overlapping windows, and is particularly useful for image-processing work.

In a multiple screen configuration there is one X process, one mouse, and one keyboard - but several screens. The mouse/cursor can be moved between the different screens. Clients connect to the different screens with the syntax `-display :0.X` where X is the screen number.

Supported Cards

Use up to four of any one card from the following list:

- One or more Matrox MGA Impression Plus
- One or more Matrox MGA Ultima Plus
- One or more Matrox MGA Millennium

B.1.5 International Keyboards

Metro-X uses the X Keyboard extension to support various keyboards. The keyboard description contains information about the physical layout of a keyboard, the key codes that come from the keyboard, and the symbol information needed to map the keycodes into keysyms.

Metro-X provides many predefined configurations, sorted by language and country. These configurations are selected through *configX* (section 5.3.3.

B.2 Metro X Configuration File

B.2.1 Warning

The primary configuration file for the X server is `Xconfig`. If inappropriate changes are made to this file, the proper functioning of Metro-X will be impaired! Do not edit the `Xconfig` file unless absolutely necessary.

B.2.2 Xconfig

Metro-X looks for the `Xconfig` file in the following locations and in the following order:

```
$HOME/Xconfig
$HOME/.Xconfig
/etc/X11/Xconfig (on Linux only)
/usr/lib/X11/Metro/Xconfig
```

Most options are set with the configuration utility (`ConfigX`, section 5.3.3). Additional options can be added using an editor, and ConfigX will preserve these.

B.2.3 Keywords

Descriptions of the `Xconfig` keywords are listed below. They are alphabetized here for your convenience and any order dependencies are noted within the individual entries.

Options Set In Configuration Utility

The following list of `Xconfig` keywords are options controlled via the Metro-X Configuration Utility. Any changes to these options should be made through the utility to avoid accidental damage to the function of Metro-X.

`BitsPerPixel` <*number*> Some X drivers support packed 24 bits per pixel memory organization. This means that each pixel takes up three bytes of video memory. In this case, *configX* inserts the line: `BitsPerPixel 24`

`Buttons` <*number*> This specifies the number of buttons on the mouse. *This entry must immediately follow the Mouse entry in the Xconfig file.*

`Card` <*string*> This designates the model of the graphics card in use. This is only used by configX.

`Chipset` <*string*> This displays the chipset used by the server. Normally configX inserts the correct chipset into the Xconfig file.

`Clocks` <*number*> *[*<*number*>*...]* This contains a list of numbers specifying the frequencies associated with the clock generator on the card. The numbers are specified in MHz.

`Depth` <*number*> This references the color depth of the screen, which controls how many colors can be displayed. Different cards and drivers support different depth values: 4, 8, 16, 24, or 32. Depth 4 supports 16 colors and depth 32 supports 16 million colors.

`Dimensions` <*width*> <*height* > This keyword represents the dimensions of the screen in millimeters, which is passed to X clients upon request.

`DPMSoffDelay` <*number*> This controls the number of seconds after the last user input before the screen enters DPMS off mode. Off mode provides maximum power reduction. To activate this mode, the X server disables both the horizontal and vertical sync signals to the monitor.

`DPMSstandbyDelay`<*number*> This controls the number of seconds after the last user input before the screen enters DPMS standby mode. Standby mode provides minimal power reduction. To activate this mode, the X server disables the horizontal sync signal to the monitor.

`DPMSsuspendDelay`<*number*> This controls the number of seconds after the last user input before the screen enters DPMS suspend mode. Suspend mode provides significant power reduction. To activate this mode, the X server disables the vertical sync signal to the monitor.

`Driver`<*string*> This references the name of the driver, which usually corresponds to the name of the X server. For example:

```
Driver 8514
```

initiates the section for defining the 8514 driver.

`Emulate3Buttons` This entry is used when a two-button mouse is emulating a three-button mouse. This entry must immediately follow the Mouse entry in the Xconfig file.

`FontPath` <*string*> FontPath is followed by a comma-delimited string of font directories. A font directory is any directory which contains fonts and a fonts.dir file.

`HotKeyExit` This allows you to shut down the X server with the key sequence Ctrl - Alt - Backspace. If this keyword is not present, Ctrl - Alt - Backspace does nothing special.

`KbdDesc` <*string*> The quoted string must match one of the entries in the KeyboardInfo file.

`KbdDescFile` <*string*> The string describes the keyboard.

`LogFile` <*filename*> Errors from the X server are logged here.

`Modes` The entries following the Modes give display mode timing information. If more than one mode is specified, the server can switch between modes. See Modes (section B.2.4 for more details.

`Monitor` *"<string>"* The quoted string must match one of the entries in the MonitorInfo file. `Mouse`<*type*> *[device]* This describes the type of mouse in use. Type is a keyword that is OS dependant; the device parameter is usually required and is a file in the /dev directory.

`MouseDesc` *"<string>"* The quoted string must match one of the entries in the MouseInfo file.

`ScreenLayout`<*screen*> <*left_neighbor*> <*right_neighbor*> <*top_neighbor*> <*bottom_neighbor*> <*left_edge*> <*right_edge*> <*top_edge*> <*bottom_edge*> This defines how the mouse cursor behaves when it gets to the edge of the screen. There is one ScreenLayout entry for each configured screen. The *_neighbor fields tell which screen to move the mouse to. A value of -1 indicates that the mouse should stop at that edge. The *_edge fields tell at which edge of the corresponding *_neighbor screen the mouse will appear. Screens are numbered from 0 to 4. Edges are numbered as follows:

NOTE: Any changes made to this entry will be lost if configX is run.

`ScreenLayoutIndex`<*number*> This tells which of the predefined screen layouts was chosen in configX.

EDGES	NUMBER
ENTER_LEFT	1
ENTER_RIGHT	2
ENTER_UP	3
ENTER_DOWN	4

Figure B.1: ScreenLayout Edge Numbering

`Virtual`<*width*> <*height*> This designates the virtual width and height of the screen. The virtual width and height may be larger than the physical width and height if hardware panning is used.

`Weight`<*string*> This designates the RGB color specification, which is dependent on the depth value. When the depth value is 16, the options are:

```
R5:G5:B5 (default) or
R5:G6:B5
```

When the depth value is 32, the options are:

```
R8:G8:B8 (default) or
B8:G8:R8
```

When the depth value is 4 or 8, the weight value is ignored. `XscreenSaverDelay` This controls the number of seconds (with no input) before the screen is blanked.

Options Not Set in Configuration Utility

The following keywords control options not set through the configuration utility, and require manual editing of the Xconfig file.

`BackBuffer` This entry tells the X server to allocate off-screen memory for a BackBuffer.

`CursorAdjustX`

`CursorAdjustY` If the hardware cursor does not position correctly over the X Window screen, CursorAdjustX and CursorAdjustY allow you to specify a delta (number of pixels) to adjust the cursor left or right.

`CursorScreenOne` In a multiple screen configuration, this setting constrains the cursor to screen 0.

`NoBackingStore` This turns off backing store. Backing store is a server feature that allows programs to create complex drawings and have the server automatically handle exposure requests, thus preventing lengthy re-draw cycles. Backing store uses system memory to hold the window contents and can actually slow down the system if there is limited memory.

`NoCursorWrap` In a multiple screen configuration, this setting prevents the cursor from wrapping from screen 1 to screen 0.

`NoFontCache` This turns off font caching.

`NoKeyboard` This setting disallows keyboard input.

`NoLEDs` No LED processing is done by the server.

`NoPixmapCache` This turns off pixmap and stipple caching.

`NotPrimaryMonitor` No console processing is done for Virtual Terminal Support. This is normally used when X is run on a graphics card that is not driving the console.

`Option <string>` This specifies a special option passed to the server. It might specify the card model or a specific feature not typically used. Normally configX inserts the correct options in the Xconfig file.

`Option HardwarePex` Add this line to the Xconfig file to use the Hardware Z Buffer on the Matrox MGA Impression Plus graphics card. This file entry must immediately follow the Driver MGA entry.

`Option NoHardwareCursor` Add this line to disable hardware cursor support. This file entry must immediately follow the Driver entry.

`RemapCursorPad` This configuration causes the X server to send back different keycodes for the keypad arrow keys and cursorpad arrow keys. This is a minor violation of the X protocol, but it allows clients to differentiate between these keys.

`SnfFormat` <*number*> <*number*> <*number*> <*number*> Metro-X X11.5 can read X11.4 SNF (Server Natural Format) fonts. Since there is no standard for SNF fonts, the following four numbers can be adjusted to read most SNF fonts. The four number parameters represent the following values, normally built into the server at compile time:

```
BITMAP_BIT_ORDER
IMAGE_BYTE_ORDER
GLYPHPADBYTES
BITMAP_SCANLINE_UNIT
```

For example, to read SCO SNF fonts, add this line to the `Xconfig file:`

```
SnfFormat 1 0 1 1
```

Or, to read SNF fonts compatible with X11 R4 from Interactive Systems Corporation, add this line to the Xconfig file:

```
SnfFormat 1 1 4 4
```

`Terminate` Terminate causes the server to exit when the last client disconnects. Normally xinit kills the X server when the last client disconnects, before the server can reset.

`Trace <number>` This prints additional debugging information.

`VideoRam <number>` This specifies the amount of video memory on the graphics card, specified in terms of 1024-byte units (with one megabyte equal to a value of 1024). The X server usually detects how much memory is on the card.

`VisualClass <number>` This uses the default VisualClass from `/usr/include/X11/X.h`, and is based on the following table:

VISUAL CLASS	NUMBER
StaticGray	0
GrayScale	1
StaticColor	2
PseudoColor	3
TrueColor	4
DirectColor	5

Figure B.2: Visual Classes

On monochrome monitors the line

```
VisualClass 1
```

should be added to the Xconfig file.

B.2.4 Modes

The last section of the `Xconfig` file is the Mode database (`Modes`). It contains the VESA timing specifications for various resolutions. The default values work for most graphics cards, but they can be tweaked to center the image and adjust its size on the monitor. Most of the Metro-X servers use this database to configure the card for graphics mode. The numbers indicate which clock to use from the DisplayConfig file's CLOCKS line, and also show the horizontal and vertical timing values. More specifically, an example of the numbers are:

```
''1024x768''  65  1024  1092  1220  1344  768  786  791  810
```

The following table shows the definition of each Modes number from the prior example.

VALUE	Meaning
"1024X768"	Screen Resolution in Pixels
65	Pixel Clock (MHz)
1024	Pixels Displayed (H)
1092	Sync Start (H)
1220	Sync End (H)
1344	Total (H)
768	Pixels Displayed (V)
786	Sync Start (V)
791	Sync End (V)
810	Total (V)

Figure B.3: Modes Definitions

These numbers represent the ranges the electron beam traverses per one vertical refresh of the screen. They determine where the image is positioned on the screen, and what the refresh rate is.

Adjusting the Image Position

You can edit the Modes numbers if the screen image does not properly center or fill the monitor screen. Be sure to center the image before stretching or shrinking it. Any adjustments you make will go into effect after restarting the X server. **Centering the Image:** The `Pixels Displayed` numbers must match the desired physical screen resolution. The `Sync Start` and `Sync End` numbers must be increased or decreased (to move left or right; up or down) at the same time and by the same amount, and stay within the range of the `Displayed` and `Total` values.

- Image too far right: To adjust an image that is too far to the right, increase the `Horizontal Sync Start` and `Horizontal Sync End` values equally.
- Image too far left: To adjust an image that is too far to the left, decrease the `Horizontal Sync Start` and `Horizontal Sync End` values equally. For example, to adjust an image that is too far to the left, you could begin the process by adding 8 to the two horizontal timing numbers (`Sync Start` and `Sync End`). Make sure that the `Sync End` number does not exceed the `Horizontal Total` number. The same technique applies for vertical positioning:
- Image too low: To adjust an image that is too far to the bottom, decrease the `Vertical Sync Start` and `Vertical Sync End` values equally.
- Image too high: To adjust an image that is too far to the top, increase the `Vertical Sync Start` and `Vertical Sync End` values equally. For example, to adjust an image that is too high, you could begin the process by adding 8 to the two vertical timing numbers (`Sync Start` and `Sync End`). Make sure that the `Sync End` number does not exceed the `Vertical Total` number.

Stretching or Shrinking the Image: The `Pixels Displayed` numbers must match the desired physical screen resolution. The `Total` value is increased or decreased to stretch or shrink the image. In addition, to hold the center position, the `Sync Start` and `Sync End` numbers must be increased or decreased equally, based on one-half of the difference in the changed `Total` value. For instance, if you added 10 to the `Horizontal Total` value, you would also add 5 to the `Horizontal Sync Start` value and 5 to the `Horizontal Sync End` value.

- Image too narrow horizontally: To stretch an image that is too narrow horizontally (extra space appears on the left and right), increase the `Horizontal Total` by amount x, then add one-half of x to both the `Horizontal Sync Start` and `Horizontal Sync End` values.
- Image too narrow vertically: To stretch an image that is too narrow vertically (extra space appears on the top and bottom), increase the `Vertical Total` by amount x, then add one-half of x to both the `Vertical Sync Start` and `Vertical Sync End` values.
- Image too wide horizontally: To shrink an image that is too wide horizontally (the image extends beyond the left and right edges), decrease the `Horizontal Total` by amount x, then subtract one-half of x from both the `Horizontal Sync Start` and `Horizontal Sync End` values.
- Image too tall vertically: To shrink an image that is too tall vertically (the image extends beyond the top and bottom edges), decrease the `Vertical Total` by amount x, then subtract one-half of x from both the `Vertical Sync Start` and `Vertical Sync End` values.

Extensions

Extensions are configured in the `/usr/lib/X11/Metro/Extensions` file. Most extensions are dynamically loaded; that is, if they are needed they are loaded automatically without any additional manual configuration. An extension that is "ON" is loaded at server startup and never unloaded. An extension that is "DYN" is dynamically loaded on request by an X client `XQueryExtension` call. These extensions display with `XListExtensions` or `xdpyinfo`. An extension that is "OFF" can not be used, nor is it displayed with `XListExtensions` or `xdpyinfo`.

Special Notes

- `Multi-Buffering` and `DOUBLE-BUFFER` can not be dynamically loaded. They must be set to "ON" or "OFF".
- `GLX` (OpenGL) can be dynamically loaded, however `GLX` depends on `DOUBLE-BUFFER` being "ON". Therefore, to use `GLX`, set `DOUBLE-BUFFER` to "ON" and `GLX` to "DYN".

- `X3D-PEX` (PEX) can be dynamically loaded, however `X3D-PEX` depends on Multi-Buffering being "ON". Therefore, to use `X3D-PEX`, set `Multi-Buffering` to "ON" and `X3D-PEX` to "DYN".

Appendix C

Intro to Alpha Systems and Processors

This document is a brief overview of existing Alpha CPUs, chipsets and systems. It has something of a hardware bias, reflecting my own area of expertese. Although I am an employee of Digital Equipment Corporation, this is not an official statement by Digital and any opinions expressed are mine and not Digital's.

C.1 What is Alpha

"Alpha" is the name given to Digital's 64-bit RISC architecture. The Alpha project in Digital began in mid-1989, with the goal of providing a high-performance migration path for VAX customers. This was not the first RISC architecture to be produced by Digital, but it was the first to reach the market. When Digital announced Alpha, in March 1992, it made the decision to enter the merchant semicondutor market by selling Alpha microprocessors. Alpha is also sometimes referred to as Alpha AXP, for obscure and arcane reasons that aren't worth persuing. Suffice it to say that they are one and the same.

C.2 What is Digital Semiconductor

Digital Semiconductor (DS, `http://www.digital.com/info/semiconductor/`) is the business unit within Digital Equipment Corporation (Digital - we don't like the name DEC) that sells semiconductors on the merchant market. Digital's products include CPUs, support chipsets, PCI-PCI bridges and PCI peripheral chips for comms and multimedia.

C.3 Alpha CPUs

There are currently 2 generations of CPU core that implement the Alpha architecture:

- EV4
- EV5

Opinions differ as to what "EV" stands for (Editor's note: the true answer is of course "Electro Vlassic" See section C.12), but the number represents the first generation of Digital's CMOS technology that the core was implemented in. So, the EV4 was originally implemented in CMOS4. As time goes by, a CPU tends to get a mid-life performance kick by being optically shrunk into the next generation of CMOS process. EV45, then, is the EV4 core implemented in CMOS5 process. There is a big difference between shrinking a design into a particular technology and implementing it from scratch in that technology (but I don't want to go into that now). There are a few other wildcards in here: there is also a CMOS4S (optical shrink in CMOS4) and a CMOS5L.

True technophiles will be interested to know that CMOS4 is a 0.75 micron process, CMOS5 is a 0.5 micron process and CMOS6 is a 0.35 micron process.

To map these CPU cores to *chips* we get:

/21064-150,166/ EV4 (originally), EV4S (now)

/21064-200/ EV4S

/21064A-233,275,300/ EV45

/21066/ LCA4S (EV4 core, with EV4 FPU)

/21066A-233/ LCA45 (EV4 core, but with EV45 FPU)

/21164-233,300,333/ EV5

/21164A-417/ EV56

/21264/ You'll just have to wait and see...

The EV4 core is a dual-issue (it can issue 2 instructions per CPU clock) superpipelined core with integer unit, floating point unit and branch prediction. It is fully bypassed and has 64-bit internal data paths and tightly coupled 8Kbyte caches, one each for Instruction and Data. The caches are write-through (they never get dirty).

The EV45 core has a couple of tweaks to the EV4 core: it has a slightly improved floating point unit, and 16KB caches, one each for Instruction and Data (it also has cache parity). (Editor's note: Neal Crook indicated in a separate mail that the changes to the floating point unit (FPU) improve the performance of the divider. The EV4 FPU divider takes 34 cycles for a single-precision divide and 63 cycles for a double-precision divide (non data-dependent). In constrast, the EV45 divider takes typically 19 cycles (34 cycles max) for single-precision and typically 29 cycles (63 cycles max) for a double-precision division (data-dependent).)

The EV5 core is a quad-issue core, also superpipelined, fully bypassed etc etc. It has tightly-coupled 8Kbyte caches, one each for I and D. These caches are write-through. It also has a tightly-coupled 96Kbyte on-chip second-level cache (the Scache) which is 3-way set associative and write-back (it can be dirty). The EV4-¿EV5 performance increase is better than just the increase achieved by clock speed improvements. As well as the bigger caches and quad issue, there are microarchitectural improvements to reduce producer/consumer latencies in some paths.

The EV56 core is fundamentally the same microarchitecture as the EV5, but it adds some new instructions for 8 and 16-bit loads and stores (see Section C.8 Bytes and all that stuff.) These are primarily intended for use by device drivers. The EV56 core is implemented in CMOS6, which is a 2.0V process.

The 21064 was anounced in March 1992. It uses the EV4 core, with a 128-bit bus interface. The bus interface supports the 'easy' connection of an external second-level cache, with a block size of 256-bits (2 data beats on the bus). The Bcache timing is completely software configurable. The 21064 can also be configured to use a 64-bit external bus, (but I'm not sure if any shipping system uses this mode). The 21064 does not impose any policy on the Bcache, but it is usually configured as a write-back cache. The 21064 does contain hooks to allow external hardware to maintain cache coherence with the Bcache and internal caches, but this is hairy.

The 21066 uses the EV4 core and integrates a memory controller and PCI host bridge. To save pins, the memory controller has a 64-bit data bus (but the internal caches have a block size of 256 bits, just like the 21064, therefore a block fill takes 4 beats on the bus). The memory controller supports an external Bcache and external DRAMs. The timing of the Bcache and DRAMs is completely software configurable, and can be controlled to the resolution of the CPU clock period. Having a 4-beat process to fill a cache block isn't as bad as it sounds because the DRAM access is done in page mode. Unfortunately, the memory controller doesn't support any of the new esoteric DRAMs (SDRAM, EDO or BEDO) or synchronous cache RAMs. The PCI bus interface is fully rev2.0 compliant and runs at upto 33MHz.

The 21164 has a 128-bit data bus and supports split reads, with upto 2 reads outstanding at any time (this allows 100utilisation under best-case dream-on conditions, i.e., you can theoretically transfer 128-bits of data on every bus clock). The 21164 supports easy connection of an external 3-rd level cache (Bcache) and has all the hooks to allow external systems to maintain full cache coherence with all caches. Therefore, symmetric multiprocessor designs are 'easy'.

The 21164A was announced in October, 1995. It uses the EV56 core. It is nominally pin-compatible with the 21164, but requires split power rails; all of the power pins that were +3.3V power on the 21164 have now been split into two groups; one group provided 2.0V power to the CPU core, the other group supplies 3.3V to the I/O cells. Unlike older implementations, the 21164 pins are not 5V-tolerant. The end result of this change is that 21164 systems are, in general, not upgradeable to the 21164A (though note that it would be relatively straightforward to design a 21164A system that could also accommodate a 21164). The 21164A also has a couple of new pins to support the new 8 and 16-bit loads and stores. It also improves the 21164 support for using synchronus SRAMs to implement the external Bcache.

C.4 21064 performance vs 21066 performance

The 21064 and the 21066 have the same (EV4) CPU core. If the same program is run on a 21064 and a 21066, at the same CPU speed, then the difference in performance comes only as a result of system Bcache/memory bandwidth. Any code thread that has a high hit-rate on the *internal* caches will perform the same. There are 2 big performance killers:

1. Code that is write-intensive. Even though the 21064 and the 21066 have write buffers to swallow some of the delays, code that is write-intensive will be throttled by write bandwidth at the system bus. This arises because the on-chip caches are write-through.
2. Code that wants to treat floats as integers. The Alpha architecture does not allow register-register transfers from integer registers to floating point registers. Such a conversion has to be done via memory (And therefore, because the on-chip caches are write-through, via the Bcache). (Editor's note: it seems that both the EV4 and EV45 can perform the conversion through the primary data cache (Dcache), provided that the memory is cached already. In such a case, the store in the conversion sequence will update the Dcache and the subsequent load is, under certain circumstances, able to read the updated d-cache value, thus avoiding a costly roundtrip to the Bcache. In particular, it seems best to execute the stq/ldt or stt/ldq instructions back-to-back, which is somewhat counter-intuitive.)

If you make the same comparison between a 21064A and a 21066A, there is an additional factor due to the different Icache and Dcache sizes between the two chips.

Now, the 21164 solves both these problems: it achieve *much* higher system bus bandwidths (despite having the same number of signal pins - yes, I *know* it's got about twice as many pins as a 21064, but all those extra ones are power and ground! (yes, really!!)) and it has write-back caches. The only remaining problem is the answer to the question "how much does it cost?"

C.5 A Few Notes On Clocking

All of the current Alpha CPUs use high-speed clocks, because their microarchitectures have been designed as so-called short-tick designs. None of the sytem busses have to run at horrendous speeds as a result though:

- on the 21066(A), 21064(A), 21164 the off-chip cache (Bcache) timing is completely programmable, to the resolution of the CPU clock. For example, on a 275MHz CPU, the Bcache read access time can be controller with a resolution of 3.6ns
- on the 21066(A), the DRAM timing is completely programmable, to the resolution of the CPU clock (*not* the PCI clock, the CPU clock).
- on the 21064(A), 21164(A), the system bus frequency is a sub-multiple of the CPU clock frequency. Most of the 21064 motherboards use a 33MHz system bus clock.
- Systems that use the 21066 can run the PCI at any frequency relative to the CPU. Generally, the PCI runs at 33MHz.
- Systems that use the APECs chipset (see Section C.6) always have their CPU system bus equal to their PCI bus frequency. This means that both busses tends to run at either 25MHz or 33MHz (since these are the frequencies that scale up to match the CPU frequencies). On APECs systems, the DRAM controller timings are software programmable in terms of the CPU system bus frequency

Aside: someone suggested that they were getting bad performance on a 21066 because the 21066 memory controller was only running at 33MHz. Actually, it's the superfast 21064A systems that have memory controllers that 'only' run at 33MHz.

C.6 The chip-sets

DS sells two CPU support chipsets. The 2107x chipset (aka APECS) is a 21064(A) support chiset. The 2117x chipset (aka ALCOR) is a 21164 support chipset. There will also be 2117xA chipset (aka ALCOR 2) as a 21164A support chipset.

Both chipsets provide memory controllers and PCI host bridges for their CPU. APECS provides a 32-bit PCI host bridge, ALCOR provides a 64-bit PCI host bridge which (in accordance with the requirements of the PCI spec) can support both 32-bit and 64-bit PCI devices.

APECS consists of 6, 208-pin chips (4, 32-bit data slices (DECADE), 1 system controller (COMANCHE), 1 PCI controller (EPIC)). It provides a DRAM controller (128-bit memory bus) and a PCI interface. It also does all the work to maintain memory coherence when a PCI device DMAs into (or out of) memory.

ALCOR consists of 5 chips (4, 64-bit data slices (Data Switch, DSW) - 208-pin PQFP and 1 control (Control, I/O Address, CIA) - a 383 pin plastic PGA). It provides a DRAM controller (256-bit memory bus) and a PCI interface. It also does all the work required to support an external Bcache and to maintain memory coherence when a PCI device DMAs into (or out of) memory.

There is no support chipset for the 21066, since the memory controller and PCI host bridge functionality are integrated onto the chip.

C.7 The Systems

The applications engineering group in DS produces example designs using the CPUs and support chipsets. These are typically PC-AT size motherboards, with all the functionality that you'd typically find on a high-end Pentium motherboard. Originally, these example designs were intended to be used as starting points for third-parties to produce motherboard designs from. These first-generation designs were called Evaluation Boards (EBs). As the amount of engineering required to build a motherboard has increased (due to higher-speed clocks and the need to meet RF emission and susceptibility regulations) the emphasis has shifted towards providing motherboards that are suitable for volume manufacture.

Digital's system groups have produced several generations of machines using Alpha processors. Some of these systems use support logic that is designed by the systems groups, and some use commodity chipsets from DS. In some cases, systems use a combination of both.

Various third-parties build systems using Alpha processors. Some of these companies design systems from scratch, and others use DS support chipsets, clone/modify DS example designs or simply package systems using build and tested boards from DS.

The EB64: Obsolete design using 21064 with memory controller implemented using programmable logic. I/O provided by using programmable logic to interface a 486 <-> ISA bridge chip. On-board Ethernet, SuperI/O (2S, 1P, FD), Ethernet and ISA. PC-AT size. Runs from standard PC power supply.

The EB64+: Uses 21064 or 21064A and APECs. Has ISA and PCI expansion (3 ISA, 2 PCI, one pair are on a shared slot). Supports 36-bit DRAM SIMs. ISA bus generated by Intel SaturnI/O PCI-ISA bridge. On-board SCSI (NCR 810 on PCI) Ethernet (Digital 21040), KBD, MOUSE (PS2 style), SuperI/O (2S, 1P, FD), RTC/NVRAM. Boot ROM is EPROM. PC-AT size. Runs from standard PC power supply.

The EB66: Uses 21066 or 21066A. I/O sub-system is identical to EB64+. Baby PC-AT size. Runs from standard PC power supply. The EB66 schematic was published as a marketing poster advertising the 21066 as "the first microprocessor in the world with embedded PCI" (for trivia fans: there are actually 2 versions of this poster - I drew the circuits and wrote the spiel for the first version, and some Americans mauled the spiel for the second version)

The EB164: Uses 21164 and ALCOR. Has ISA and PCI expansion (3 ISA slots, 2 64-bit PCI slots (one is shared with an ISA slot) and 2 32-bit PCI slots. Uses plus-in Bcache SIMMs. I/O sub-system provides SuperI/O (2S, 1P, FD), KBD, MOUSE (PS2 style), RTC/NVRAM. Boot ROM is Flash. PC-AT-sized motherboard. Requires power supply with 3.3V output.

The AlphaPC64 (aka Cabriolet): derived from EB64+ but now baby-AT with Flash boot ROM, no on-board SCSI or Ethernet. 3 ISA slots, 4 PCI slots (one pair are on a shared slot), uses plug-in Bcache SIMMs. Requires power supply with 3.3V output.

The AXPpci33 (aka NoName), is based on the EB66. This design is produced by Digital's Technical OEM (TOEM) group. It uses the 21066 processor running at 166MHz or 233MHz. It is a baby-AT size, and runs from a standard PC power supply. It has 5 ISA slots and 3 PCI slots (one pair are a shared slot). There are 2 versions, with either PS/2 or large DIN connectors for the keyboard.

Other 21066-based motherboards: most if not all other 21066-based motherboards on the market are also based on EB66 - there's really not many system options when designing a 21066 system, because all the control is done on-chip.

Multia (aka the Universal Desktop Box): This is a very compact pedestal desktop system based on the 21066. It includes 2 PCMCIA sockets, 21030 (TGA) graphics, 21040 Ethernet and NCR 810 SCSI disk along with floppy, 2 serial ports and a parallel port. It has limited expansion capability (one PCI slot) due to its compact size. (There is some restriction on when you can use the PCI slot, can't remember what) (Note that 21066A-based and Pentium-based Multia's are also available).

DEC PC 150 AXP (aka Jensen): This is a very old Digital system - one of the first-generation Alpha systems. It is only mentioned here because a number of these systems seem to be available on the second-hand market. The Jensen is a floor-standing tower system which used a 150MHz 21064 (later versions used faster CPUs but I'm not sure what speeds). It used programmable logic to interface a 486 EISA I/O bridge to the CPU.

Other 21064(A) systems: There are 3 or 4 motherboard designs around (I'm not including Digital *systems* here) and all the ones I know of are derived from the EB64+ design. These include:

- EB64+ (some vendors package the board and sell it unmodified); AT form-factor.
- Aspen Systems motherboard: EB64+ derivative; baby-AT form-factor.
- Aspen Systems server board: many PCI slots (includes PCI bridge).
- AlphaPC64 (aka Cabriolet), baby AT form-factor.

Other 21164(A) systems: The only one I'm aware of that isn't simply an EB164 clone is a system made by DeskStation. That system is implemented using a memory and I/O controller proprietary to Desk Station. I don't know what their attitude towards Linux is.

C.8 Bytes and all that stuff

When the Alpha architecture was introduced, it was unique amongst RISC architectures for eschewing 8-bit and 16-bit loads and stores. It supported

32-bit and 64-bit loads and stores (longword and quadword, in Digital's nomenclature). The co-architects (Dick Sites, Rich Witek) justified this decision by citing the advantages:

1. Byte support in the cache and memory sub-system tends to slow down accesses for 32-bit and 64-bit quantities.
2. Byte support makes it hard to build high-speed error-correction circuitry into the cache/memory sub-system.

Alpha compensates by providing powerful instructions for manipulating bytes and byte groups within 64-bit registers. Standard benchmarks for string operations (e.g., some of the Byte benchmarks) show that Alpha performs very well on byte manipulation.

The absence of byte loads and stores impacts some software semaphores and impacts the design of I/O sub-systems. Digital's solution to the I/O problem is to use some low-order address lines to specify the data size during I/O transfers, and to decode these as byte enables. This so-called Sparse Addressing wastes address space and has the consequence that I/O space is non-contiguous (more on the intricacies of Sparse Addressing when I get around to writing it). Note that I/O space, in this context, refers to all system resources present on the PCI and therefore includes both PCI memory space and PCI I/O space.

With the 21164A introduction, the Alpha archtecture was ECO'd to include byte addressing. Executing these new instructions on an earlier CPU will cause an OPCDEC PALcode exception, so that the PALcode will handle the access. This will have a performance impact. The ramifications of this are that use of these new instructions (IMO) should be restricted to device drivers rather than applications code.

These new byte load and stores mean that future support chipsets will be able to support contiguous I/O space.

C.9 PALcode and all that stuff

This is a placeholder for a section explaining PALcode. I will write it if there is sufficient interest.

C.10 Porting

The ability of any Alpha-based machine to run Linux is really only limited by your ability to get information on the gory details of its innards. Since there are Linux ports for the E66, EB64+ and EB164 boards, all systems based on the 21066, 21064/APECS or 21164/ALCOR should run Linux with little or no modification. The major thing that is different between any of these motherboards is the way that they route interrupts. There are three sources of interrupts:

- on-board devices
- PCI devices
- ISA devices

All the systems use an Intel System I/O bridge (SIO) to act as a bridge between PCI and ISA (the main I/O bus is PCI, the ISA bus is a secondary bus used to support slow-speed and 'legacy' I/O devices). The SIO contains the traditional pair of daisy-chained 8259s.

Some systems (e.g., the Noname) route all of their interrupts through the SIO and thence to the CPU. Some systems have a separate interrupt controller and route all PCI interrupts plus the SIO interrupt (8259 output) through that, and all ISA interrupts through the SIO.

Other differences between the systems include:

- how many slots they have
- what on-board PCI devices they have
- whether they have Flash or EPROM

C.11 More Information

All of the DS evaluation boards and motherboard designs are license-free and the whole documentation kit for a design costs about $50. That includes all the schematics, programmable parts sources, data sheets for CPU and support chipset. The doc kits are available from Digital Semiconductor distributors. I'm not suggesting that many people will want to rush out and buy this, but I do want to point out that the information is available.

Hope that was helpful. Comments/updates/suggestions for expansion to Neal Crook, `neal.crook@reo.mts.digital.com`

C.12 References

`http://www.research.digital.com/wrl/publications/abstracts/TN-13.html`
Bill Hamburgen, Jeff Mogul, Brian Reid, Alan Eustace, Richard Swan, Mary Jo Doherty, and Joel Bartlett. *Characterization of Organic Illumination Systems*. DEC WRL, Technical Note 13, April 1989.

Appendix D

Alpha Miniloader Howto

D.1 Introduction

This document describes the Miniloader for Linux on Alpha AXP (MILO). This firmware is used to initialize Alpha AXP based systems, load and start Linux and, finally, provide PALcode for Linux.

D.1.1 Copyright

NO WARRENTY about the information in the HOWTO and I cannot be made liable for any consequences resulting from using the information in this HOWTO.

D.1.2 New Versions of this Document

The latest version of this document can be found in `ftp://gatekeeper.dec.com/pub/Digital/Linux-Alpha/Miniloader/docs` and David Mosberger-Tang is kind enough to include the html form of it in his excellent Linux Alpha FAQ site `http://www.azstarnet.com/ axplinux`.

D.2 What is MILO?

On Intel based PC systems, the BIOS firmware sets up the system and then loads the image to be run from the boot block of a DOS file system. This is more or less what MILO does on an Alpha based system, however there are several interesting differences between BIOS firmware and MILO, not least of which is that MILO includes and uses standard Linux device drivers unmodified. MILO is firmware, unlike LILO, which relies on the BIOS firmware to get itself loaded. The main functional parts of MILO are:

1. PALcode,
2. Memory set up code (builds page tables and turns on virtual addressing),
3. Video code (BIOS emulation code and TGA (21030)),
4. Linux kernel code. This includes real Linux kernel code (for example, the interrupt handling) and ersatz or mock Linux kernel,
5. Linux block device drivers (for example, the floppy driver),
6. File system support (ext2, MS-DOS and ISO9660),
7. User inteface code (MILO),

8. Kernel interface code (sets up the HWRPB and memory map for linux),

9. NVRAM code for managing environment variables.

The following paragraphs describe these functional parts in more detail.

PALcode can be thought of as a tiny software layer that tailors the chip to a particular operating system. It runs in a special mode (PALmode) which has certain restrictions but it uses the standard Alpha instruction set with just five extra instructions. In this way, the Alpha chip can run such diverse operating systems as Windows NT, OpenVMS, Digital Unix and, of course, Linux. The PALcode that MILO uses (and therefore Linux itself) is, like the rest of MILO, freeware. It is derived from Digital's Evaluation Board software example Digital Unix PALcode.. The differences between the different PALcodes are because of differences in address mapping and interrupt handling that exist between the Alpha chips (21066 based systems have a different I/O map to 21064+2107x systems) and different Alpha based systems.

For MILO to operate properly it needs to know what memory is available, where Linux will eventually be running from and it must be able to allocate temporary memory for the Linux device drivers. The code maintains a memory map that has entries for permanent and temporary allocated pages. As it boots, MILO uncompresses itself into the correct place in physical memory. When it passes control to the Linux kernel, it reserves memory for the compressed version of itself, the PALcode (which the kernel needs) and some data structures. This leaves `most` of the memory in the system for Linux itself.

The final act of the memory code is to set up and turn on virtual addressing so that the data structures that Linux expects to see are at the correct place in virtual memory.

MILO contains video code that initialises and uses the video device for the system. It will detect and use a VGA device if there is one, otherwise it will try to use a TGA (21030) video device. Failing that, it will assume that there is no graphics device. The BIOS emulation that the standard, pre-built, images include is Digital's own BIOS emulation which supports most, if not all, of the standard graphics devices available.

Linux device drivers live within the kernel and expect certain services from the kernel. Some of these services are provided directly by Linux kernel code, for example the interrupt handling and some is provided by kernel look-alike routines.

MILO's most powerful feature is that you can embed unaltered Linux device drivers into it. This gives it the potential to support every device that Linux does. MILO includes all of the block devices that are configured into the Linux kernel that it is built against as well as a lot of the block device code (for example, ll_rw_blk()).

MILO loads the Linux kernel from real file systems rather than from boot blocks and other strange places. It understands MSDOS, EXT2 and ISO9660 filesystems. Gzip'd files are supported and these are recommended, particularly if you are loading from floppy which is rather slow. MILO recognises these by their *.gz* suffix.

Built into MILO is a simple keyboard driver which, together with an equally simple video driver allows it to have a simple user interface. That interface allows you to list file systems on configured devices, boot Linux or run flash update utilities and set environment variables that control the system's booting. Like LILO, you can pass arguments to the Kernel.

MILO must tell the Linux kernel what sort of system this is, how much memory there is and which of that memory is free. It does this using the HWRPB (Hardware Restart Parameter Block) data structure and associated memory cluster descriptions. These are placed at the appropriate place in virtual memory just before control is passed to the Linux kernel.

D.3 Pre-Built Standard MILO Images.

If you are planning to run Linux on a standard Alpha based system, then there are pre-built "standard" MILO images that you might use. These (along with the sources and other interesting stuff) can be found in `ftp://gatekeeper.dec.com/pub/Digital/Linux-Alpha/Miniloader`.

The `images` subdirectory contains a directory per standard system (eg AlphaPC64) with MILO images having the following naming convention:

1. `MILO` - Miniloader executable image, this image can be loaded in a variety of ways,
2. `fmu.gz` - Flash management utility,

3. `MILO.dd` - Boot block floppy disk image. These should be written using rawrite.exe or `dd` on Linux.

The `test-images`, like the `images` subdirectory contains a directory per standard system. These images are somewhat experimental but tend to contain all the latest features.

D.4 How To Build MILO

You build MILO seperately from the Kernel. As MILO requires parts of the kernel to function (for example interrupt handling) you must first configure and build the kernel that matches with MILO that you want to build. Mostly this means building the kernel with the same version number. So, MILO-2.0.25.tar.gz will build against linux-2.0.25.tar.gz. MILO `may` build against a higher version of the kernel, but there again it may not. Also, now that ELF shared libraries are fully supported, there are two versions of the MILO sources. To build under an ELF system you must first unpack the standard MILO sources and then patch those sources with the same version numbered ELF patch. In the remainder of this discussion, I assume that your kernel sources and object files are stored in the subtree at `/usr/src/linux` and that the linux kernel has been fully built with the command `make boot`

To build MILO, change your working directory to the MILO source directory and invoke `make` with:

```
$    make KSRC=/usr/src/linux config
```

Just like the Linux kernel, you will be asked a series of questions

```
Echo output to the serial port (MINI_SERIAL_ECHO) [y]
```

It's a good idea to echo kernel printk to `/dev/ttyS0` if you can. If you can (and want to), then type "y", otherwise "n". All of the standard, pre-built, MILO images include serial port I/O using COM1.

```
Use Digital's BIOS emulation code (not free) \
                    (MINI_DIGITAL_BIOS_EMU) [y]
```

This code is included as a library which is freely distributable so long as it is used on an Alpha based system. The sources are not available. If you answer `n` then the freeware alternative BIOS emulation will be built. It's sources are included with MILO. Note that you cannot right now build choose Digital's BIOS emulation code in an ELF system (the library is not yet ready) and so you must answer no to this question.

```
          Build PALcode from sources (Warning this is dangerous) \
                             (MINI_BUILD_PALCODE_FROM_SOURCES) [n]
```

You should only do this if you have changed the PALcode sources, otherwise use the standard, pre-built PALcode included with MILO.

You are now all set to build the MILO image itself:

```
     $    make KSRC=/usr/src/linux
```

When the build has successfully completed, the MILO image is in the file called `milo`. There are a lot of images called `milo.*`, these should be ignored.

D.5 How To Load MILO

The most commonly supported method of loading MILO is from the Windows NT ARC firmware as most shipping systems support this. However, there are a wide variety of loading MILO. It may be loaded from:

- a failsafe boot block floppy,
- Windows NT ARC firmware,
- Windows NT AlphaBIOS,
- Digital's SRM console,
- an Alpha Evaluation Board Debug Monitor,
- flash/ROM.

D.5.1 Loading MILO from the Windows NT ARC firmware

Most, if not all, Alpha AXP based systems include the Windows NT ARC firmware and this is the prefered method of booting MILO and thus Linux. Once the Windows NT firmware is running and you have the correct MILO image for your system, this method is completely generic.

The Windows NT ARC firmware is an environment in which programs can run and make callbacks into the firmware to perform actions. The Windows NT OSLoader is a program that does exactly this. Linload.exe is a much simpler program which does just enough to load and execute MILO. It loads the appropriate image file into memory at 0x00000000 and then makes a swap-PAL PALcall to it. MILO, like Linux, uses a different PALcode to Windows NT which is why the swap has to happen. MILO relocates itself to 0x200000 and continues on through the PALcode reset entry point as before.

Before you add a Linux boot option, you will need to copy linload.exe and the appropriate MILO that you wish to load to someplace that the Windows NT ARC firmware can read from. In the following example, I assume that you are booting from a DOS format floppy disk:

1. At the boot menu, select `"Supplementary menu..."`
2. At the `"Supplementary menu"`, select `"Set up the system..."`
3. At the `"Setup menu"`, select `"Manage boot selection menu..."`
4. In the `"Boot selections menu"`, choose `"Add a boot selection"`
5. Choose `"Floppy Disk 0"`
6. Enter `"linload.exe"` as the osloader directory and name
7. Say `"yes"` to the operating system being on the same partition as the osloader
8. Enter `"\"` as the operating system root directory
9. I usually enter `"Linux"` as the name for this boot selection
10. Say `"No"` you do not want to initialise the debugger at boot time

11. You should now be back in the `"Boot selections menu"`, choose the `"Change a boot selection option"` and pick the selection you just created as the one to edit
12. Use the down arrow to get `"OSLOADFILENAME"` up and then type in the name of the MILO image that you wish to use, for example `"noname.arc"` followed by return.
13. Press ESC to get back to the `"Boot Selections menu"`
14. Choose `"Setup Menu"` (or hit ESC again) and choose `"Supplementary menu, and save changes"` option
15. ESC will get you back to the `"Boot menu"` and you can attempt to boot MILO. If you do not want Linux as the first boot option, then you can alter the order of the boot options in the `"Boot selections menu"`.

At the end of all this, you should have a boot selection that looks something like:

```
LOADIDENTIFIER=Linux
SYSTEMPARTITION=multi(0)disk(0)fdisk(0)
OSLOADER=multi(0)disk(0)fdisk(0)\linload.exe
OSLOADPARTITION=multi(0)disk(0)fdisk(0)
OSLOADFILENAME=\noname.arc
OSLOADOPTIONS=
```

You can now boot MILO (and then Linux). You can load linload.exe and MILO directly from a file system that Windows NT understands such as NTFS or DOS on a hard disk.

The contents `OSLOADOPTIONS` are passed to MILO which interprets it as a command. So, in order to boot Linux directly from Windows NT without pausing in MILO, you could pass the following in `OSLOADOPTIONS`:

```
boot sda2:vmlinux.gz root=/dev/sda2
```

See D.6 for more information on the commands available.

Another (rather sneaky) way of loading of loading MILO via the WNT ARC firmware is to put MILO onto an MS-DOS floppy and call it `fwupdate.exe` and then choose the "Upgrade Firmware" option.

D.5.2 Loading MILO from the Windows NT AlphaBIOS

With the introduction of the XLT series, Digital changed the system console interface for its NT systems from ARC to AlphaBIOS. AlphaBIOS is a screen-oriented interface which should be more familiar to PC users. This change in console interface necessitates a change in setup procedure for those who wish to run Linux/Alpha on AlphaBIOS-based systems.

The first thing you must do is to install the latest version of AlphaBIOS on your system. This can be obtained from Digital's "System Software and Driver Updates" Web page, `http://www.windows.digital.com/support/sysoft.htp`. Download the ZIP file, unzip it, and install it as follows:

- Copy the files to a FAT-formatted floppy
- Turn on the system and insert the floppy. At the opening screen, press F2 to go into setup mode
- Select `"Upgrade AlphaBIOS"`
- Follow the directions on the screen

Once your AlphaBIOS is at the latest revision level, you can start bootstrapping your system as follows:

- Create a FAT-formatted floppy with the following files:
  - linload.exe (from this directory)
  - milo (the version appropriate to your system)
- Turn on the system and insert the floppy. At the opening screen, press F2 to go into setup mode
- Select `"Utilities->OS Selection Setup..."`
- Press INSERT to add a new operating system selection
- For `"Boot Name"`, enter something like `"Linux"`. Press TAB to get over to the next field.
- Press down-arrow until the selection for `"Boot File"` is `"A:"`. TAB over to the next field

- Enter `"linload.exe"`. TAB *twice* (i.e. skip the OS Path load device – it's irrelevant)
- Enter `"\"` for the OS Path load file
- Press ENTER to add the selection.

At this point, AlphaBIOS will probably put up a big, unfriendly dialog box labelled "Warning: Operating System Selection not valid!". Ignore this error (it's only a problem for NT) and press ENTER to continue.

- Press F10 to save the changes you just made; press ENTER to confirm the changes.
- Press ESC twice to get back to the opening screen.
- Use the up and down arrows to select the boot selection you just added, and press ENTER to boot it.
- AlphaBIOS will load linload, which will in turn load MILO. When you get to the MILO prompt, proceed as you would for a normal ARC-based system installation.

If you dedicated the first partition of your first disk drive to a small FAT partition for booting (as the installation procedure advises you to), then once Linux is installed you should copy linload.exe and milo to this partition. Once you shut down, you can then modify the Linux menu selection to load MILO from this partition as follows:

- At the opening screen, select F2 to go into setup mode
- Select `"Utilities->OS Selection setup"`
- Highlight the entry for Linux, then press F6 to edit it.
- TAB over to the device portion of the `"Boot File"` line. Use the up and down arrow keys to select the hard-disk partition where linload and milo reside (typically `"Disk 0 Partition 1"` or `"Disk 1 Partition 1"`). Press ENTER to confirm the selection
- If you wish to auto-boot linux after MILO is loaded, then TAB over to the `"OS Options"` line and enter the MILO command to boot the system, e.g. `"boot sda2:vmlinux.gz"` Press ENTER to confirm the selection.

- Press F10 to save the changes. Press ENTER to confirm.

Once you have done this, booting and running Linux on an AlphaBIOS based system should be very similar to doing so on an ARC system.

D.5.3 Loading MILO from the Evaluation Board Debug Monitor

Evaluation boards (and often designs cloned from them) include support for the Alpha Evaluation Board Debug Monitor. Consult your system document before considering this method of booting MILO. The following systems are *known* to include Debug Monitor support:

- AlphaPC64 (Section D.5.7)
- EB64+ (Section D.5.7)
- EB66+ (Section D.5.7)
- EB164 (Section D.5.7)
- PC164 (Section D.5.7)

Before you consider this method, you should note that the early versions of the Evaluation Board Debug Monitor did not include video or keyboard drivers and so you must be prepared to connect another system via the serial port so that you can use the Debug Monitor. Its interface is very simple and typing help shows a whole heap of commands. The ones that are most interesting include the word `boot` or `load` in them.

The Evaluation Board Debug Monitor can load an image either via the network (netboot) or via a floppy (flboot). In either case, set the boot address to 0x200000 (`> bootadr 200000`) before booting the image.

If the image is on floppy (and note that only DOS formatted floppies are supported), then you will need to type the following command:

```
AlphaPC64> flboot <MILO-image-name>
```

D.5.4 Loading MILO from a Failsafe Boot Block Floppy

Only the AxpPCI33 is *known* to include failsafe boot block floppy support (Section D.5.7).

If you do not have a standard pre-built MILO .dd image, then you may need to build an SRM boot block floppy. Once you have built MILO, you need to do the following on Digital Unix box:

```
fddisk -fmt /dev/rfd0a
cat mboot bootm > /dev/rfd0a
disklabel -rw rfd0a 'rx23' mboot bootm
```

Or on a Linux box:

```
cat mboot bootm > /dev/fd0
```

If you have a standard MILO image available (say `MILO.dd`) then you would build a boot block floppy using the following command:

```
dd if=MILO.dd of=/dev/fd0
```

D.5.5 Loading MILO from Flash

There are a number of systems where MILO can be blown into flash and booted directly (instead of via the Windows NT ARC firmware):

- AlphaPC64 (Section D.5.7)
- Noname (Section D.5.7)
- EB66+ (Section D.5.7)
- EB164 (Section D.5.7)
- PC164 (Section D.5.7)

D.5.6 Loading MILO from the SRM Console

The SRM (short for System Reference Manual) Console knows nothing about filesystems or disk-partitions, it simply expects that the secondary bootstrap loader occupies a consecutive range of physical disk sectors starting from a given offset. The information describing the secondary bootstrap loader (its size and offset) is given in the first 512 byte block. To load MILO via the SRM you must generate that structure on a device which the SRM can access (such as a floppy disk). This is what `mboot` and `bootm`, `mboot` is the first block (or boot description) and `mboot` is the `MILO` image rounded up to a 512 byte boundary.

To load MILO from a boot block device, either build `mboot` and `bootm` and push them onto the boot device using the following command:

```
$ cat mboot bootm > /dev/fd0
```

Or, grab the appropriate `MILO.dd` from a web site and write it onto the boot device using either `RAWRITE.EXE` or `dd`.

Once you have done that you can boot the SRM console and use one of its many commands to boot MILO. For example, to boot MILO from a boot block floppy you would use the following command:

```
>>>boot dva0
(boot dva0.0.0.0.1 -flags 0)
block 0 of dva0.0.0.0.1 is a valid boot block
reading 621 blocks from dva0.0.0.0.1
bootstrap code read in
base = 112000, image_start = 0, image_bytes = 4da00
initializing HWRPB at 2000
initializing page table at 104000
initializing machine state
setting affinity to the primary CPU
jumping to bootstrap code
MILO Stub: V1.1
Unzipping MILO into position
Allocating memory for unzip
####...
```

The following systems are *known* to have SRM Console support:

- Noname (Section D.5.7)
- AlphaPC64 (Section D.5.7)
- EB164 (Section D.5.7)
- PC164 (Section D.5.7)

D.5.7 System Specific Information

AxpPCI33 (Noname)

The Noname board can load MILO from the Windows NT ARC firmware (Section D.5.1), from the SRM Console (Section D.5.6). and from a failsafe boot block floppy (Section D.5.4). A flash management utility, runnable from MILO is available so that once MILO is running, it can be blown into flash (Section D.7). However, be warned that once you have done this you will lose the previous image held there as there is only room for one image.

The way that Noname boots is controlled by a set of jumpers on the board, J29 and J28. These look like:

```
                4
J29         2 x x x 6
            1 x x x 5

J28         2 x x x 6
            1 x x x 5
                3
```

The two options that we're interested in are J28, pins 1-3 which boots the console/loader from flash and J29, pins 1-3 which boots the console/loader from a boot block floppy. The second option is the one that you need to first boot MILO on the Noname board.

Once you've selected the boot from floppy option via the jumpers, put the SRM boot block floppy containing MILO into the floppy and reboot. In a few seconds (after the floppy light goes out) you should see the screen blank to white and MILO telling you what's going on.

If you are really interested in technical stuff, the Noname loads images off of the floppy into physical address 0x104000 and images from flash into

0x100000. For this reason, MILO is built with it's PALcode starting at 0x200000. When it is first loaded, it moves itself to the correct location (see relocate.S).

AlphaPC64 (Cabriolet)

The AlphaPC64 includes the Windows NT ARC firmware (Section D.5.1), the SRM Console (Section D.5.6) and the Evaluation Debug Monitor (Section D.5.3). These images are in flash and there is room to add MILO so that you can boot MILO directly from flash. A flash management utility, runnable from MILO is available so that once MILO is running, it can be blown into flash (Section D.7). This system supports MILO environment variables.

You select between the boot options (and MILO when it is been put into flash) using a combination of jumpers and a boot option which is saved in the NVRAM of the TOY clock.

The jumper is J2, SP bits 6 and 7 have the following meanings:

- SP bit 6 should always be out. If this jumper is set then the SROM mini-debugger gets booted,
- SP bit 7 in is boot image selected by the boot option byte in the TOY clock,
- SP bit 7 out is boot first image in flash.

So, with bit 7 out, the Debug Monitor will be booted as it is `always` the first image in flash. With bit 7 in, the image selected by the boot option in the TOY clock will be selected. The Debug Monitor, the Windows NT ARC firmware and MILO all support setting this boot option byte but you must be very careful using it. In particular, you cannot set the boot option so that next time the system boots MILO when you are running the Windows NT ARC firmware, it only allows you to set Debug Monitor or Windows NT ARC as boot options.

To get MILO into flash via the Evaluation Board Debug Monitor, you will need a flashable image. The build proceedures make MILO.rom, but you can also make a rom image using the makerom tool in the Debug Monitor software that comes with the board:

```
> makerom -v -i7 -l200000 MILO -o mini.flash
```

(type makerom to find out what the arguments mean, but 7 is a flash image id used by the srom and -l200000 gives the load address for the image as 0x200000).

Load that image into memory (via the Debug Monitor commands flload, netload, and so on) at 0x200000 and then blow the image into flash:

```
AlphaPC64> flash 200000 8
```

(200000 is where the image to be blown is in memory and 8 is the segment number where you put the image. There are 16 1024*64 byte segments in the flash and the Debug Monitor is at seg 0 and the Windows NT ARC firmware is at seg 4).

Set up the image that the srom will boot by writing the number of the image into the TOY clock.

```
AlphaPC64> bootopt 131
```

(131 means boot the 3rd image, 129 = 1st, 130 = 2nd and so on).

Power off, put jumper 7 on and power on and you should see the MILO burst into life. If you don't then take jumper 7 back off and reboot the Debug Monitor.

EB66+

The EB66+, like all of the Alpha Evaluation Boards built by Digital contains the Evaluation Board Debug Monitor and so this is available to load MILO (Section D.5.3). Quite often (although not always) boards whose design is derived from these include the Debug Monitor also. Usually, these boards include the Windows NT ARC firmware (Section D.5.1). A flash management utility, runnable from MILO is available so that once MILO is running, it can be blown into flash (Section D.7). This system supports MILO environment variables.

These systems have several boot images in flash controlled by jumpers. The two jumper banks are J18 and J16 and are located at the bottom of the board in the middle (if the Alpha chip is at the top). You select between the boot options (and MILO when it is been put into flash) using a combination of jumpers and a boot option which is saved in the NVRAM of the TOY clock.

Jumper 7-8 of J18 in means boot the image described by the boot option. Jumper 7-8 of J18 out means boot the Evaluation Board Debug Monitor.

Blowing an image into flash via the Evaluation Board Debug Monitor is exactly the same proceedure as for the AlphaPC64 (Section D.5.7).

EB64+/Aspen Alpine

This system is quite like the AlphaPC64 except that it does not contain flash which MILO can be loaded from. The EB64+ has two ROMs, one of which contains the Windows NT ARC firmware (Section D.5.1). and the other contains the Evaluation Board Debug Monitor (Section D.5.3).

The Aspen Alpine is a little different in that it only has one ROM; this contains the Windows NT ARC firmware.

Universal Desktop Box (Multia)

This is a very compact pre-packaged 21066 based system that includes a TGA (21030) graphics device. Although you can *just* fit a half height PCI graphics card in the box you are better off waiting for full TGA support in XFree86. It includes the Windows NT ARC firmware and so booting from that is the prefered method (Section D.5.1).

EB164

The EB164, like all of the Alpha Evaluation Boards built by Digital contains the Evaluation Board Debug Monitor and so this is available to load MILO (Section D.5.3). Quite often (although not always) boards whose design is derived from these include the Debug Monitor also. Usually, these boards include the Windows NT ARC firmware (Section D.5.1). The SRM console is also available (Section D.5.6). A flash management utility, runnable from MILO is available so that once MILO is running, it can be blown into flash (Section D.7). This system supports MILO environment variables.

These systems have several boot images in flash controlled by jumpers. The two jumper bank is J1 and is located at the bottom of the board on the left (if the Alpha chip is at the top). You select between the boot options (and MILO when it is been put into flash) using a combination of jumpers and a boot option which is saved in the NVRAM of the TOY clock.

Jumper SP-11 of J1 in means boot the image described by the boot option. Jumper SP-11 of J1 out means boot the Evaluation Board Debug Monitor.

Blowing an image into flash via the Evaluation Board Debug Monitor is exactly the same proceedure as for the AlphaPC64 (Section D.5.7).

PC164

The PC164, like all of the Alpha Evaluation Boards built by Digital contains the Evaluation Board Debug Monitor and so this is available to load MILO (Section D.5.3). Quite often (although not always) boards whose design is derived from these include the Debug Monitor also. Usually, these boards include the Windows NT ARC firmware (Section D.5.1). The SRM console is also available (Section D.5.6). A flash management utility, runnable from MILO is available so that once MILO is running, it can be blown into flash (Section D.7). This system supports MILO environment variables.

These systems have several boot images in flash controlled by jumpers. The main jumper block, J30, contains the system configuration jumpers and jumper CF6 in means that the system will boot the Debug Monitor, the default is out.

Blowing an image into flash via the Evaluation Board Debug Monitor is exactly the same proceedure as for the AlphaPC64 (Section D.5.7).

XL266

The XL266 is one of a family of systems that are known as Avanti. It has a riser card containing the Alpha chip and cache which plugs into the main board at right angles. This board can replace the equivalent Pentium board.

Some of these systems ship with the SRM console but others, notably the XL266 ship with only the Windows NT ARC firmware (Section D.5.1).

Here is my list of compatible systems:

- AlphaStation 400 (Avanti),
- AlphaStation 250,
- AlphaStation 200 (Mustang),
- XL. There are two flavours, XL266 and XL233 with the only difference being in processor speed and cache size.

Note The system that I use to develop and test MILO is an XL266 and so this is the only one that I can guarentee will work. However, technically, all of the above systems are equivalent; they have the same support chipsets and the same interrupt handling mechanisms.

Platform2000

This is a 233Mhz 21066 based system.

D.6 MILO's User Interface

Once you have correctly installed/loaded/run MILO you will see the MILO (for MIniLOader) prompt displayed on your screen. There is a very simple interface that you must use in order to boot a particular Linux kernel image. Typing "help" is a good idea as it gives a useful summary of the commands.

D.6.1 The "help" Command

Probably the most useful command that MILO has:

```
MILO> help
MILO command summary:

ls [-t fs] [dev:[dir]]
                 - List files in directory on device
boot [-t fs] [dev:file] [boot string]
```

```
                 - Boot Linux from the specified device and file
run [-t fs] dev:file
                 - Run the standalone program dev:file
show             - Display all known devices and file systems
set VAR VALUE    - Set the variable VAR to the specified VALUE
unset VAR        - Delete the specified variable
reset            - Delete all variables
print            - Display current variable settings
help [var]       - Print this help text

Devices are specified as: fd0, hda1, hda2, sda1...
Use the '-t filesystem-name' option if you want to use
  anything but the default filesystem  ('ext2').
Use the 'show' command to show known devices and filesystems.
Type 'help var' for a list of variables.
```

Note that the `bootopt` command only appears on AlphaPC64 (and similar) systems. Refer to the board's dcoumentation to find out just what it means.

Devices. Until you use a command that needs to make use of a device, no device inititalisation will take place. The first `show`, `ls`, `boot` or `run` commands all cause the devices within MILO to be initialised. Devices are named in the same way (exactly) that Linux itself will name them. So, the first IDE disk will be called 'hda' and it's first partition will be 'hda1'. Use the `show` command to show what devices are available.

File Systems. MILO supports three file systems, MSDOS, EXT2 and ISO9660. So long as a device is available to it, MILO can `listboot` or `run` an image stored on one of these file systems. MILO's default file system is `EXT2` and so you have tell MILO that the file system is something other than that. All of the commands that use filenames allow you to pass the file system using the `-t [filesystem]`option. So, if you wanted to list the contents of a SCSI CD ROM, you might type the following:

```
MILO> ls -t iso9660 scd0:
```

Variables. MILO contains some settable variables that help the boot process. If you are loading via the Windows NT ARC firmware, then MILO makes use of the boot option environment variables set up by that firmware. For some systems, MILO (for example, the AlphaPC64) maintains its own set of environment variables that do not change from boot to boot. These variables are:

```
MILO> help var
Variables that MILO cares about:
  MEMORY_SIZE      - System memory size in megabytes
  BOOT_DEV         - Specifies the default boot device
  BOOT_FILE        - Specifies the default boot file
  BOOT_STRING      - Specifies the boot string to pass to the kernel
  SCSIn_HOSTID     - Specifies the host id of the n-th SCSI controller.
  PCI_LATENCY      - Specifies the PCI master device latency
  AUTOBOOT         - If set, MILO attempts to boot on powerup
                     and enters command loop only on failure.
  AUTOBOOT_TIMEOUT - Seconds to wait before auto-booting on powerup.
```

BEWARE of setting AUTOBOOT without setting the timeout, you can end up automatically booting after a timemout of 0 seconds which may not be what you want.

PCI_LATENCY is the number of PCI bus cycles that PCI bus mastering device hangs onto the PCI bus for once it has control. The default value for this is 32 and the maximum value is 255. Setting a high value means that each time a device gets control of the PCI bus it can transfer more data. However, setting it high also means that devices may have to wait longer to get control.

D.6.2 Booting Linux

The `boot` command boots a linux kernel from a device. You will need to have a linux kernel image on an EXT2 formated disk (SCSI, IDE or floppy) or an ISO9660 formatted CD available to MILO. The image can be gzip'd and in this case MILO will automatically gunzip it. Early versions of MILO recognised a gzip'd file by the .gz suffix but later MILOs look for magic numbers in the image.

You should note that the version of MILO does not usually have to match the version of the Linux kernel that you are loading. You boot Linux using the following command syntax:

```
MILO> boot [-t file-system] device-name:file-name \
                 [[boot-option] [boot-option] ...]
```

Where `device-name` is the name of the device that you wish to use and `file-name` is the name of the file containing the Linux kernel. All arguments supplied after the file name are passed directly to the Linux kernel.

If you are installing Red Hat, then you will need to specify a root device and so on. So you would use:

```
MILO> boot fd0:vmlinux.gz root=/dev/fd0 load_ramdisk=1
```

MILO will automatically contain the block devices that you configure into your vmlinux. I have tested the floppy driver, the IDE driver and a number of SCSI drivers (for example, the NCR 810), and these work fine. Also, it is important to set the host id of the SCSI controller to a reasonable value. By default, MILO will initialize it to the highest possible value (7) which should normally work just fine. However, if you wish, you can explicitly set the host id of the n-th SCSI controller in the system by setting environment variable `SCSI`n`_HOSTID` to the appropriate value. For example, to set the hostid of the first SCSI controller to 7, you can issue the following command at the MILO prompt:

```
setenv SCSI0_HOSTID 7
```

D.6.3 Rebooting Linux

You may want to reboot a running Linux system using the `shutdown -r now` command. In this case, the Linux kernel returns control to MILO (via the HALT CallPAL entrypoint). MILO leaves a compressed copy of itself in memory for just this reason and detects that the system is being rebooted from information held in the HWRPB (Hardware Restart Parameter Block). In this case it starts to reboot using exactly the same command that was used to boot the Linux kernel the last time. There is a 30 second timeout that allows you to interrupt this process and boot whatever kernel you wish in whatever way you wish.

D.6.4 The "bootopt" command

For flash based systems such as the AlphaPC64, EB164 and the EB66+, there are a number of possible boot options and these are changed using the `bootopt` command. This has one argument, a decimal number which is the type of the image to be booted the next time the system is power cycled or reset:

0 Boot the Evaluation Board Debug Monitor,

1 Boot the Windows NT ARC firmware.

In order to tell the boot code to boot the MILO firmware from flash then you need a boot option that means boot the N'th image. For this, you need to 128 plus N, so if MILO is the third image, you would use the command:

```
MILO> bootopt 131
```

`Note:` Be very careful with this command. A good rule is never to set bootopt to 0 (the Evaluation Board Debug Monitor), but instead use the system's jumpers to achieve the same thing.

D.7 Running the Flash Management Utility

The `run` command is used to run the flash management utility. Before you start you will need a device available to MILO that contains the updateflash program. This (like vmlinux) can be gzip'd. You need to run the flash management utility program from the MILO using the (`run`) command:

```
MILO> run fd0:fmu.gz
```

Once it has loaded and initialised, the flash management utility will tell you some information about the flash device and give you a command prompt. Again the `help` command is most useful.

```
Linux MILO Flash Management Utility V1.0

Flash device is an Intel 28f008SA
  16 segments, each of 0x10000 (65536) bytes
Scanning Flash blocks for usage
Block 12 contains the environment variables
FMU>
```

`Note` that on systems where environment variables may be stored and where there is more than one flash block (for example, the AlphaPC64) the flash management utility will look for a block to hold MILO's environment variables. If such a block already exists, the flash management utility will tell you where it is. Otherwise, you must use the `environment` command to set a block and initialise it. In the above example, flash block 12 contains MILO's environment variables.

D.7.1 The "help" command

```
FMU> help
FMU command summary:

list             - List the contents of flash
program          - program an image into flash
quit             - Quit
environment      - Set which block should contain the environment \
                                                         variables
bootopt num      - Select firmware type to use on next power up
help             - Print this help text
FMU>
```

Note that the `environment` and `bootopt` commands are only available on the EB66+, the AlphaPC64, EB164 and PC164 systems (and their clones).

D.7.2 The "list" command

The "list" command shows the current usage of the flash memory. Where there is more than one flash block, the usage of each flash block is shown. In the example below you can see that Windows NT ARC is using blocks 4:7 and block 15.

```
FMU> list
Flash blocks:  0:DBM  1:DBM  2:DBM  3:WNT  4:WNT  5:WNT  6:WNT
        7:WNT  8:MILO  9:MILO 10:MILO 11:MILO 12:MILO 13:U 14:U
15:WNT
Listing flash Images
  Flash image starting at block 0:
    Firmware Id:  0 (Alpha Evaluation Board Debug Monitor)
    Image size is 191248 bytes (3 blocks)
    Executing at 0x300000
  Flash image starting at block 3:
```

```
    Firmware Id:  1 (Windows NT ARC)
    Image size is 277664 bytes (5 blocks)
    Executing at 0x300000
  Flash image starting at block 8:
    Firmware Id:  7 (MILO/Linux)
    Image size is 217896 bytes (4 blocks)
    Executing at 0x200000
FMU>
```

D.7.3 The "program" command

The flash management utility contains a compressed copy of a flash image of MILO. The "program" command allows you to blow this image into flash. The command allows you to back out, but before you run it you should use the "list" command to see where to put MILO. If MILO is already in flash, then the flash management utility will offer to overwrite it.

```
FMU> program
Image is:
    Firmware Id:  7 (MILO/Linux)
    Image size is 217896 bytes (4 blocks)
    Executing at 0x200000
Found existing image at block 8
Overwrite existing image? (N/y)? y
Do you really want to do this (y/N)? y
Deleting blocks ready to program: 8 9 10 11
Programming image into flash
Scanning Flash blocks for usage
FMU>
```

Wait until it has completed before powering off your system.

`Note:` I cannot emphasise just how careful you must be here not to overwrite an existing flash image that you might need or render your system useless. A very good rule is never to overwrite the Debug Monitor.

D.7.4 The "environment" command

This selects a flash block to contain MILO's environment variables.

D.7.5 The "bootopt" command

This is just the same as MILO's "bootopt" command, see (Section D.6.4).

D.7.6 The "quit" command

This is really pretty meaningless. The only way back to MILO (or anything else) once the flash management utility has run is to reboot the system.

D.8 Restrictions.

Unfortunately this is not a perfect world and there, as always, some restrictions that you should be aware of.

MILO is not meant to load operating systems other than Linux, although it can load images linked to run at the same place in memory as Linux (which is 0xFFFFFC0000310000). This is how the flash management utilities can be run.

The PALcode sources included in `miniboot/palcode/`*blah* are correct, however there are problems when they are built using the latest `gas`. They *do* build if you use the ancient a.out gas that's supplied in the Alpha Evaluation Board toolset (and that's how they were built). I'm trying to get someone to fix the new gas. Meanwhile, as a workaround, I have provided pre-built PALcode for the supported boards and David Mosberger-Tang has a fixed gas on his ftp site.

D.9 Problem Solving.

Here are some common problems that people have seen, together with the solutions.

Reading MS-DOS floppies from the Evaluation Board Debug Monitor.

Some of the older versions of the Evaluation Board Debug Monitor (pre-version 2.0) have a problem with DOS format flopies generated from Linux.

Usually, the Debug Monitor can load the first few sectors all right, but then goes into an endless loop complaining about "bad sectors." Apparently, there is an incompatibility between the DOS file system as expected by the Debug Monitor and the Linux implementation of DOSFS. To make the long story short: if you run into this problem, try using DOS to write the floppy disk. For example, if loading the file `MILO.cab` doesn't work, use a DOS machine, insert the floppy and then do:

```
copy a:MILO.cab c:
copy c:MILO.cab a:
del c:MILO.cab
```

Then try booting from that floppy again. This normally solves the problem.

MILO displays a long sequence of `0>` and does not accept input.

This usually happens when MILO was built to use COM1 as a secondary console device. In such a case, MILO echo output to COM1 and accepts input from there also. This is great for debugging but not so great if you have a device other than a terminal connected. If this happens, disconnect the device or power it down until the Linux kernel has booted. Once Linux is up and running, everything will work as expected.

MILO complains that the kernel image has the wrong magic number

Older versions of MILO did not support the ELF object file format and so could not recognise an ELF image and this might be your problem. If this is reported, upgrade to the latest MILO that you can find. All 2.0.20 and beyond MILOs support ELF. On the other hand it could be that the image is indeed damaged. You should also note that MILO does not yet automatically distinquish between GZIP'd and non-GZIP'd images; you need to add the ".gz" suffix to the file name.

MILO prints "...turning on virtual addressing and jumping to the Linux Kernel" and nothing else happens

One obvious problem is that the kernel image is wrongly built or is built for another Alpha system altogether. Another is that the video board is a TGA (Zlxp) device and the kernel has been built for a VGA device (or vice versa). It is worth building the kernel to echo to COM1 and then connecting a terminal to that serial port or retrying the kernel that came with the Linux distribution that you installed.

MILO does not recognise the SCSI device

The standard MILO images include as many device drivers as are known to be stable for Alpha (as of now that includes the NCR 810, QLOGIC ISP, Buslogic and Adaptec 2940s and 3940 cards). If your card is not included, it may be that the driver is not stable enough on an Alpha system yet. Again, the latest MILO images are worth trying. You can tell which SCSI devices a MILO image has built into it by using the "show" command.

D.10 Acknowledgements.

I would like to thank:

- Eric Rasmussen and Eilleen Samberg the authors of the PALcode,
- Jim Paradis for the keyboard driver, the original MILO interface and the AlphaBIOS work,
- Jay Estabrook for his help and bugfixes,
- David Mosberger-Tang for the freeware BIOS emulation and his support and encouragement,
- Last (and `not` least) Linus Torvalds for the timer code and his kernel.

There are a number of things that still need doing to MILO, if you want to add something yourself, then do let me know `mailto:david.rusling@reo.mts.dec.com` so that we do not duplicate our efforts.

Finally, a big thank you to Digital for producing such a wonderful chip (and paying me to do this).

Appendix E

Red Hat Linux Frequently Asked Questions

E.1 Introduction

This is the Official Red Hat Linux FAQ. It is maintained by `faq-maintainer@redhat.com`; all comments or suggestions for this FAQ should be sent to that address. A recent version of this FAQ is available from `http://www.redhat.com`, or by email from `faq@redhat.com`.

E.2 General Questions

E.2.1 Where can I get information on Red Hat Linux?

- Via the Web: `http://www.redhat.com`
- Via FTP: `ftp://ftp.redhat.com`

E.2.2 How Can I Get Red Hat Linux?

Red Hat Linux is available on CD from Red Hat Software or various Red Hat Linux resellers; it is also available via FTP from `ftp://ftp.redhat.com` and its many mirror sites.

E.2.3 The Red Hat FTP site is *slow*. Are there mirrors?

There are several mirrors of the Red Hat FTP site:

```
FTP Site                Directory
========                =========

UK
--

sunsite.doc.ic.ac.uk    /packages/linux/redhat
nfs-uxsup.csx.cam.ac.uk /pub/redhat/
ftp.ox.ac.uk            /pub/linux/redhat

Europe
------

ftp.osb.hu              /pub/mirrors/linux/redhat
sunsite.icm.edu.pl      /pub/Linux/redhat
ftp.vma.bme.hu          /pub/linux/redhat
ftp.sfos.ro             /mirror/RedHat
ftp.u-bordeaux.fr       /pub/Linux/redhat
ftp.tu-chemnitz.de      /pub/linux/redhat-mirror/
ftp.wcfauna.ee          /pub/Linux/RedHat/
ftp.uni-bayreuth.de     /pub/linux/ftp.redhat.com/
ftp.pk.edu.pl           /pub/linux/redhat
sunsite.uio.no          /pub/unix/linux/redhat/
ftp.etud.insa-tlse.fr   /pub/redhat
ftp.arch.pwr.wroc.pl    /mirror/linux/redhat/
ftp.ras.ru              /pub/Linux/redhat
ftp.u-bordeaux.fr       /pub/Linux/redhat
ftp.ibp.fr              /pub/linux/distributions/redhat
redhat.netvision.net.il /pub/mirrors/redhat
ftp.sorosis.ro          /pub/mirrors/ftp.redhat.com
ftp.mpi-sb.mpg.de       /pub/linux/mirror/ftp.redhat.com
```

```
ftp.jate.u-szeged.hu    /pub/linux/redhat
ftp.ibp.fr              /pub/linux/distributions/redhat
ftp.gwdg.de             /pub/linux/install/redhat
ftp.sgg.ru              /mirror/redhat
sunsite.mff.cuni.cz     /OS/Linux/Distributions/Redhat
ftp.ton.tut.fi          /pub/Linux/RedHat
garbo.uwasa.fi          /pub/mirror/RedHat
ftp.funet.fi            /pub/Linux/images/RedHat
ftp.arch.pwr.proc.pl    /mirror/linux/redhat
ftp.rhi.hi.is           /pub/linux/RedHat
ftp.nvg.ntnu.no         /pub/linux/redhat
ftp.pk.edu.pl           /pub/linux/redhat
ftp.nluug.nl            /pub/os/Linux/distr/RedHat
dutepp0.et.tudelft.nl   /pub/Unix/Linux/Distributions/redhat
ftp.iol.ie              /pub/Unix/Linux/distributions/RedHat
sunsite.auc.dk          /pub/os/linux/redhat
ftp.sorosis.ro          /pub/mirrors/ftp.redhat.com

Africa
------

ftp.is.co.za            /linux/distributions/redhat
ftp.leg.uct.ac.za       /pub/linux/distributions/redhat/
ftp.sun.ac.za           /pub/linux/redhat/

Australia
---------

dopey.gist.net.au       /pub/redhat
ftp.cse.unsw.EDU.AU     /pub/linux/redhat
www.unimelb.edu.au      /pub/unix/linux/redhat
sunsite.anu.edu.au      /pub/linux/distributions/redhat/

Asia
----

ftp.jaist.ac.jp         /pub/os/linux/redhat
NCTUCCCA.edu.tw         /OS/Linux/distributions/RedHat
ftp.dacom.co.kr         /pub/Linux/redhat
ftp.lab.kdd.co.jp       /OS/Linux/packages/redhat
sunsite.ust.hk          /pub/Linux/distributions/redhat
ftp.tku.edu.tw          /Unix/Linux/RedHat
ftp.wownet.net          /LINUX/redhat
ftp.kreonet.re.kr       /pub/Linux/redhat
```

```
ftp.iss.nus.sg          /pub/redhat

South America
-------------

ftp.sunsite.dcc.uchile.cl  /pub/OS/linux/redhat

North America
-------------

ftp.kernel.org          /pub/mirrors/redhat
sunsite.unc.edu         /pub/Linux/distributions/redhat
tsx-11.mit.edu          /pub/linux/distributions/redhat/
ftp.redhat.com          /pub/linux/distributions/RedHat/
peyote-asesino.nuclecu.unam.mx   /linux/redhat-4.1
ftp.epix.net            /pub/Linux/distributions/redhat
linux.eecs.umich.edu    /pub/linux/redhat
redhat.aiss.uiuc.edu    /pub/redhat
ftp.varesearch.com      /pub/mirrors/redhat
ftp.xtn.net             /pub/linux/redhat
ftp-nog.rutgers.edu     /pub/linux/distributions/redhat
ftp.interpath.net       /pub/linux/redhat
schlitz.cae.wisc.edu    /pub/Linux/RedHat
ftp.engr.uark.edu       /pub/linux/redhat
ftp.infomagic.com       /pub/mirrors/linux/RedHat
ftp.wgs.com             /pub/linux/redhat
ftp.hkstar.com          /pub/Linux/redhat
ftp.pht.com             /pub/linux/redhat
linux.ucs.indiana.edu   /pub/linux/redhat
ftp.ou.edu              /linux/redhat
ftp.cc.gatech.edu       /pub/linux/distributions/redhat
uiarchive.uiuc.edu      /pub/systems/linux/redhat
ftp.caldera.com         /pub/mirrors/redhat
ftp.cms.uncwil.edu      /linux/redhat
ftp.wilmington.net      /linux/redhat
gatekeeper.dec.com      /pub/linux/redhat
ftp.rge.com             /pub/systems/linux/redhat
linuxwww.db.erau.edu    /pub/linux/distrib/redhat
ftp.eit.com             /pub/mirrors/redhat
ftp.real-time.com       /pub/redhat
ftp.priority.net        /pub/redhat
ftp.miranda.org         /pub/mirrors/redhat
ftp.siriuscc.com        /pub/Linux/Redhat
gandalf.ncssm.edu       /pub/redhat
```

The rest of this document has path references to files on these mirrors. Just concatenate that file name to any of the above mirror locations to find the file.

E.2.4 How do I order from Red Hat Software?

- Via the Web: `http://www.redhat.com`
- Via email: `sales@redhat.com`
- Via fax: +1-919-572-6726
- Via telephone (toll-free): +1-888-RED-HAT1
- Via telephone (toll call): +1-919-572-6500

E.2.5 How do I contact Red Hat Software?

```
Red Hat Software
3203 Yorktown Ave., Suite 123
Durham, NC  27713
USA
tel: +1-919-572-6500
fax: +1-919-572-6726
http://www.redhat.com
ftp://ftp.redhat.com
mailto:redhat@redhat.com
```

E.2.6 How do I get new updates to Red Hat as they happen?

Keep your eye on:

`http://www.redhat.com/errata`

for new information we've made available since the current CD was pressed. Also check the `contrib` directory on our FTP mirrors for packages that users have contributed. We also make periodic announcements to the `redhat-announce-list` with updates.

E.2.7 What do I do if I have media (manual or CD) problems?

If you are experiencing problems with the product you received (e.g., your manual is missing pages or your CD is scratched), please contact Red Hat Software at `sales@redhat.com` or by phone using the above contact information. The problem will be rectified promptly. Please do not report these problems on the `redhat-list` or to Red Hat Support, as they are likely isolated and your product will simply need to be replaced.

(Note: This is not a common problem, but it does happen from time to time.)

E.2.8 What's the quickest and best way to get support for Red Hat?

- The Red Hat Software WWW site (`http://www.redhat.com`)
- The Red Hat Linux Users' Mailing Lists; see `http://www.redhat.com/mailing-lists/`.

E.2.9 How do I report a bug?

Send email to `bugs@redhat.com`.

E.2.10 What versions of Red Hat Linux are there?

There have been several now. Here's a rundown:

- Preview (or Beta) - Summer 1994
- Halloween - Fall 1994
- Mother's Day 1.0 - Summer 1995
- Mother's Day 1.1 - Late Summer 1995
- Red Hat Linux 2.0beta - Late Late Summer 1995
- Red Hat Linux 2.0 - Early Fall 1995

- Red Hat Linux 2.1 - Late Fall 1995
- Red Hat Linux/AXP 2.1-Beta - Late Fall 1995
- Red Hat Linux/AXP 2.1 - January 1996
- Red Hat Linux 3.0.3 (Picasso) - March 1996
- Red Hat Linux Rembrandt (beta) - August 1996
- Red Hat Linux (Intel/Alpha/SPARC) 4.0 - October 1996
- Red Hat Linux (Intel/Alpha/SPARC) 4.1 - January 1997
- Red Hat Linux (Intel/Alpha/SPARC) 4.2 - May 1997

E.2.11 What hardware does Red Hat Linux support?

For a current list of supported hardware, please see:

`http://www.redhat.com/hardware`

E.2.12 Why won't my Adaptec (274x, 284x) work?

The Adaptec 274x and 284x are supported but may need to be reconfigured to work properly. It has been reported that the wide negotiation on these cards is not quite working, so it should be disabled.

E.2.13 What applications are there for Red Hat Linux?

There are many applications for Linux available. Please see `http://www.redhat.com/linux-info` and go to the "Linux Applications and Utilities" section. This section is maintained by a private individual that is not related to Red Hat in any way. You will find information on nearly every product available for Linux on his pages.

E.2.14 Does Red Hat Linux include source code?

Yes. We include the exact source that was used to build the distribution. From release 2.0 on, Red Hat Linux is built with a packaging system called

RPM that ONLY uses pristine source (the same as what you'd find at the author's site) and possibly a patch by Red Hat. Complete information on RPM is available on `http://www.redhat.com/`.

E.2.15 Does Red Hat Linux support UMSDOS installs?

No. It is not a good performance idea.

E.3 Installation Questions

E.3.1 How do I get my IDE CD-ROM on my SB16 to work?

The Creative Labs kit comes with the IDE Interface set to the third IDE port by default (IDE2, I think). This won't work under Linux as it only recognizes two IDE ports. What you need to do is move a jumper on the card to configure it for IDE1 (the secondary IDE port). I'm not sure if you have to make changes under DOS for this configuration to work, but your manuals should help with this (it *will* still work, but you may have to edit a file or two).

You may also have to specify command line parameters like:

```
boot:  linux hdX=cdrom
```

If you have one IDE drive, X is most likely "b" in this case (ie "linux hdb=cdrom"). If you have two, then it will be "c".

E.3.2 How do I run the install program again?

Reboot your machine using the disk that you made for the installation. You will most likely need to completely reinstall your system.

If you just want to reconfigure networking or install more packages, try using the control-panel tools.

E.3.3 Why am I missing some programs?

Try doing a 'df' at the shell prompt. If you have "/" or "/usr" showing up as 100% used, chances are your partition was too small for the entire install. Unfortunately, this does not result in any type of error messages unless you are watching closely (which most people don't when the thing tells you it is going to take 2 hours to install :-). You will most likely need to reinstall and do one of two things: a) make your root or "/usr" partition larger or b) install fewer packages.

Also, if none of the partitions you made for Linux is full, you are probably just missing some packages. You will need to use `glint` or `rpm` to install those.

E.3.4 How do I mount my CD-ROM?

If you installed from CD-ROM, you should be able to do:

```
mount /mnt/cdrom
```

If you get an error that says the mount point does not exist, do:

```
mkdir /mnt/cdrom
```

If you did an install via NFS: You need to know what type of CD-ROM you have. The best way is to watch the boot messages as you reboot, or look at /var/adm/messages (`more /var/adm/messages`). You most likely have an IDE CD-ROM that will show up as /dev/hdc or /dev/hdd, or you have a Sound Blaster type that is /dev/sbpcd, or you have a SCSI that is /dev/scd0.

In any case, you can do the following:

```
mount -t iso9660 /dev/xxxx /mnt/cdrom
```

and substitute xxxx for YOUR device. You can then: `cd /mnt/cdrom` and then: `ls` and you will be looking at the contents of the CD-ROM.

E.3.5 Why does Linux only see part of my RAM?

There are a couple things that could be wrong. On *some* 386's, you need to compile your kernel with "Limit memory to 16M?" enabled. Some AMI BIOS motherboards have a memory option to split memory into two segments. I'm not sure what the option is called, but it has something to do with "linear". You'll find it in your BIOS setup. You may need to specify the amount of memory to the kernel at boot time from lilo. To try this, type: `linux mem=32M` at the lilo prompt. If that works, you can add that line to the append in your /etc/lilo.conf and rerun lilo. The example from above would be:

```
append = "mem=32M"
```

Don't forget to run 'lilo' after editing the file.

E.3.6 How do I install to my DOS partition?

You can't. None of the current Red Hat releases support UMSDOS installs. If you have one drive that has one big DOS partition with free space you want to use, you have two options. One is to backup all your DOS data somehow and then repartition the drive into smaller partitions and then restore the DOS data back to a DOS partition. You then have a free partition for linux. The second choice is to use fips. fips is a program that will let you change your partition size of a DOS partition without harming the data on it. It is *STRONGLY* suggested that you backup your data before using fips. Red Hat Software makes no warranties or claims as to the suitability of using fips for this purpose nor can we guarantee that data will remain unharmed. I haven't heard of anyone having a bad experience with fips, but I won't guarantee it either.

E.3.7 How do I use the NYS/NIS system that comes with RedHat?

Please see the NYS-Tips. You can find it at `http://www.redhat.com/`.

E.3.8 Why can't I get to Unixware now that I've installed Linux?

You need to boot Unixware off of a boot floppy and run 'fdisk /mbr'. That should allow you to get back into Unixware, but will not let you access Linux.

E.3.9 Why does my Diamond video card do odd things under X Windows?

If you experience problems with XFree86 and your Diamond video card, you may try putting:

```
"Option Diamond"
```

In your /etc/X11/XF86Config (or in /etc/XF86Config if you use that).

E.3.10 Where can I find an list of known problems?

To find solutions to known problems, please see:

```
http://www.redhat.com/errata
```

and look for both a "General Errata" page and a platform-specific Errata page for your version of Red Hat Linux.

E.3.11 I installed Metro-X, now how do I use it?

To get the `startx` command to use Metro-X instead of XFree86 you must make `/etc/X11/X` a symbolic link to `/usr/X11R6/bin/Xmetro`:

```
rm /etc/X11/X
ln -s ../../usr/X11R6/bin/Xmetro /etc/X11/X
```

E.4 Using Red Hat Linux

E.4.1 Why doesn't the xmh shipped with Red Hat work properly?

We only ship xmh because it is part of the standard X11 distribution. xmh is basicaly shipped as-is from the X11 distribution, and is most likely broken. We've found that there are several better alternatives. If you want to use mh mail, install the mh RPM and the exmh RPM. exmh is a much better GUI mail reader than xmh.

E.5 Copyright Notice

Appendix F

Red Hat and Freely Distributable Software.

F.0.1 Red Hat History

Linux, like Unix itself, is a very modular operating system. The skills required to select, compile, link, and install, the various components that are needed for a complete Linux OS are beyond the experience of most people who might want to use Linux. The various Linux distributions go a long way towards solving this for the average Linux user, but most do not address the problem of how to upgrade your Linux system once you get it successfully installed. Most users found it easier to delete their whole Linux system and reinstall from scratch when they needed to upgrade.

The Red Hat distribution makes Linux easier to install and maintain by providing the user with advanced package management, graphical (point and click!) system installation and control, and system administration tools.

F.1 The Free Software Model

Probably the best feature of Linux and the GNU utilities in general and Red Hat Linux in particular is that it is distributable under the terms of the GNU Public License (GPL). This feature has allowed research institutions, universities, commercial enterprises, and hackers, to develop and use Red Hat Linux and related technologies co-operatively without fear that their work would someday be controlled and restricted by a commercial vendor.

In short, the GPL changes the model of software development and distribution to one much like the model our Legal system and its industry uses. If a lawyer designs an argument that wins his case in front of the supreme court his reward is not only the fees his client pays him but also the additional clients that his achievement attracts to his practice. The "argument" he used becomes available for any other lawyer to use without restriction, and in fact becomes part of our collective legal heritage.

This new model is already a new industry with companies like Cygnus Support (see `http://www.cygnus.com`) becoming multi-million dollar enterprises on the basis of providing support and services for large commercial users of GPL'd and other "freely distributable" software. Red Hat Software, Inc.'s rapid growth is based, similarly, on our development and support of the Red Hat Linux distribution, a product that we distribute worldwide on CD and over the Internet under the terms of the GPL.

F.2 The Evolution of "Free Software"

The Unix operating system was also developed on a cooperative basis by AT&T with the active support of many universities and research establishments. Many of the early Unix contributors were disappointed when AT&T began in the late 1970's and 1980's, to enforce their copyright terms over the use and distribution of Unix. This eventually resulted in the long and bitter legal battle between AT&T (and subsequently Novell who bought the Unix copyrights from AT&T in 1991) and the University of California at Berkeley. In a partial response to this problem of individual control of a public software development project, The Massachusetts Institute of Technology (MIT) in Cambridge, MA has, since 1984, supported Richard Stallman's efforts to launch a movement in the software development community to support the development of free software. This resulted in the establishment of The Free Software Foundation (the FSF), and its GNU project. The name GNU follows in an early Unix tradition of using recursive acronyms as program names, and stands for Gnu's Not Unix. The FSF continues to be head-quartered at MIT, and has enjoyed amazing success over the last decade with products like GNU C, GNU C++, Emacs, and many other products, many of which form substantial and key building blocks of the Linux operating system. Richard Stallman insists, with some justification, that Linux should correctly be known as a GNU system.

There are many other "free software" licenses that some of the code in Red Hat Linux has been licensed under. The most popular of these are the University of California at Berkeley BSD License whereby the copyright holder remains the registered owner but where no restrictions on further distribution are imposed on the user of the code.

Also coming from MIT, but not related to the FSF or the GPL, is the X Consortium, which is a Unix industry sponsored effort to build and maintain the X Window System. In turn, a group called the XFree86 Project has taken on the responsibility to port the X Window System to the Intel platform. We have included copies of each of these licenses in the appendix of this manual for your convenience. Linux is continuing to progress very rapidly because of the contributions of this world wide development effort, to which you and your company or school are encouraged to contribute.

F.3 The GNU Public License

Red Hat Software, Inc. has chosen to license its code under the GPL for two reasons. One is that this is the easiest thing to do since the Linux kernel is licensed this way and our code interacts with other GPL'd code sufficiently to make any other Licensing scheme difficult. The other is that for the purpose of cooperative development purposes we genuinely feel that the GPL is in fact the most effective license format.

The purpose of the GPL is much like any standard copyright license, to define what you can and cannot use the product for. However it differs substantially from commercial licenses because instead of restricting your use of the software its purpose is to *guarantee* your use of the software.

It in no way restricts the use of standard commercially licensed software with Linux or other GPL'd code. Only the changes that you make to the code that you "borrowed" from someone else that was licensed under the GPL must be made available to others under the same unrestricted rules defined by the GPL. Any product that simply uses or operates on top of Linux can be licensed under any commercial rules you chose.

There are many leading commercial software developers who are actively supporting their commercial products on Linux for which you must purchase a license, such as Motif from the OSF, Empress RDBMS from Empress Software, WordPerfect, and recently Applixware - the Office Suite from Applix Inc. and Red Hat Software. You will find a complete copy of the GPL in section G.

F.4 Will Linux Replace Commercial OSes?

The huge development effort and wide distribution of the Linux OS will ensure that it takes its place as a real, viable, and significant alternative to commercially restricted operating systems. The open development model, availability of source, and lack of license restrictions are “features” of the Linux OS that commercial OS developers simply cannot offer. Software development groups who need this model include groups from government affiliated research organizations, to academic research and teaching projects, to commercial software application developers.

The recent rapid increase in new applications becoming available for Linux and the rapidly growing user base of these technologies are causing even the largest computer industry organizations to take Linux seriously. Even Datapro (a McGraw-Hill Company) in their recent 1996 Survey of the UNIX industry concluded that: "Programmers are taking a hard look at the viability of Linux on production platforms now that Linux costs less than Microsoft and has the added benefits of UNIX, such as great performance, inherent power tool sets, and communication capabilities".

The new distribution and development model that Linux, the FSF's GNU code, and other “freely distributable” software represent is a new model that is already successfully competing with the older commercially-restricted model. There is an exciting future for these technologies that we are pleased that you are now a part of.

F.5 Red Hat Software, Inc.'s Role in Linux Development

The Red Hat Linux distribution is also issued under the GPL. We see our role as making the Linux OS easy to install, configure, and maintain for professional developers, system administrators, and students. Red Hat Linux is available at no cost and few restrictions (see the GPL) on many ftp sites on the Internet. It is also available on CD from Red Hat Software `http://www.redhat.com` or (919) 572-6500 and from most of the Linux archive CD vendors.

Red Hat Software, Inc. provides support for users of the Linux OS, from individuals running a single workstation to large organizations running hundreds of networked machines, in a variety of programs.

We gratefully acknowledge the support that we have received from the Linux development community including Linus Torvalds, everyone on the Linux contributors list, and the thousands of other developers who helped make Linux the success it is today. We would also like to acknowledge the growing contribution of commercial developers to our efforts including Digital Equipment Corporation, O'Reilly & Associates, Metro Linux Inc, and many others. You are encouraged to use Red Hat's Linux distribution, to develop and add features, or simply package programs that you need or that you think others would appreciate. Please do not hesitate to contact us for any help you require.

You can contact Red Hat Software directly at: `info@redhat.com`.

F.6 Further Reading

There are Internet based listings of both the freely distributable, and commercial software available for Linux in documents such as the Linux Journal, the Commercial HOWTO, the Linux Software Map, and from many Linux vendors, including Red Hat Software's retail division. Many third party and commercial products can be purchased through Red Hat's web site, including Motif, database development systems, editors, and more. Info can obtained by sending a fax request to (919) 572-6726, and the web site is located at `http://www.redhat.com`.

For more information on Linux please see:

- *Running Linux*, M. Welsh, O'Reilly & Associates, 1995.
- *Linux: Network Administrators Guide*, O'Reilly & Associates, 1995.

- Red Hat Software (`http://www.redhat.com` and `ftp://ftp.redhat.com/pub`)
- The Free Software Foundation (`ftp://prep.ai.mit.edu/pub/gnu` or `gnu@prep.ai.mit.edu`)
- The X Consortium (`http://www.x.org`) and XFree86 (`http://www.xfree86.org`)
- For general Unix questions, please see the full line of O'Reilly & Associates books. The are perhaps the best reference material for all things Unix related. See the O'Reilly web site (`http://www.ora.com`) for more information.

Appendix G

Licenses

Nearly all of the software components on the Red Hat Linux CD-ROM are freely redistributable. A few of them require special permission to redistribute, and Red Hat Software, Inc. has obtained these permissions. The majority of the software components are distributable under the terms of one of the three licenses in this chapter. Please see each software component for precise distribution terms.

All the software on the CD-ROM produced by Red Hat Software is Copyright (C) 1995 by Red Hat Software, Inc. Unless otherwise noted, all such software is freely redistributable under the terms of the GNU Public License (GPL).

The terms Red Hat, rpm, and glint are trademarks of Red Hat Software, Inc.

G.1 The BSD Copyright

Copyright ©1991, 1992, 1993, 1994 The Regents of the University of California. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

G.2 X Copyright

G.3 Gnu Public License

GNU GENERAL PUBLIC LICENSE Version 2, June 1991

G.3.1 Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software– to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

G.3.2 GNU GENERAL PUBLIC LICENSE

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0

This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1

You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2

You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a. You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b. You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c. If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this

License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3

You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a. Accompany it with the complete corresponding machine readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b. Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c. Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4

You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5

You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6

Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7

If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8

If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9

The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10

If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

11

NO WARRANTY

BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

G.3.3 Appendix

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
<one line to give the program's name and a brief idea of what
it does.> Copyright (C) 19yy <name of author>

This program is free software; you can redistribute it and/or
modify it under the terms of the GNU General Public License
as published by the Free Software Foundation; either version
2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty
of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See
the GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.
```

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright ©19yy name of author Gnomovision
comes with ABSOLUTELY NO WARRANTY; for details type 'show w'.
This is free software, and you are welcome to redistribute it
under certain conditions; type 'show c' for details.
```

The hypothetical commands 'show w' and 'show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than 'show w' and 'show c'; they could even be mouse-clicks or menu items–whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright interest in the
program 'Gnomovision' (which makes passes at compilers) written
by James Hacker.

<signature of Ty Coon>, 1 April 1989 Ty Coon, President of
Vice
```

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

Index