Getting Started (Installation)

The goal of this session is to familiarize you with the process of installing Linux on a computer from scratch, taking a computer from the step of initially booting from CD or diskette to the point where Linux is fully installed (and minimally configured) on the hard disk. You may not know what to do with it once you get it installed... but we'll show you how the installation works.

In this session, we'll cover the following:

A1.1 Overview of the Installation Process

Overview....

Installing Linux is a relatively simple operation, once you have done it a few times. There are a few questions that one may not know the answers to, and a few things one could do to confuse the matter. Setting up Linux as a server rather than as a workstation cuts back on many of the confusing issues, though for security reasons you do want to be as careful as possible. This section will walk you through the steps needed to install and configure the basic computer, preparing the system for further configuration and use.

Security During Installation

It is important to note that you do not want to be connected to the Internet while you are installing Linux on your computer unless you are behind a good firewall. It is not at all unheardof for hackers to break into a computer during the relatively vulnerable installation and configuration period, to the point that one is unable to access one's own computer by the end of the installation. So, pull out that ethernet cable that is going to the cable modem or DSL connection!

Basic Installation Process

The following is the basic procedure for installing Linux. The details of each of these elements will be explained at length later in this chapter.

Installation Step...	***Details...***
Planning for the Install	1. Define how the server will be used, both in the near term and in the long term. 2. Select an appropriate Linux distribution for the need, and obtain a reasonably recent copy of it. 3. Gather information about your computer's hardware configuration, and do any needed hardware pre−configuration that is required. 4. Will a boot diskette be needed? 5. Plan how your hard disk will be used. 6. Plan how the network connection will be setup. 7. Plan for security controls and features.
Setting Up the Hard Disk	1. Start the installer and answer a few preliminary questions. 2. Set up the hard disk partitions and mount points. 3. Set up the boot loader.
Configure Networking and Security Choices	1. Set up TCP/IP configuration (IP Address, etc.). 2. Set up the installer's remedial firewall. 3. Set passwords. 4. Choose the user authentication method.
Install the Packages	1. Choose what components should be installed. 2. Wait for the install to finish!

Installation Step...	*Details...*
Post−Installation Configuration and Hardening	1. Log in to the newly−installed system and check current disk and memory usage. 2. Shutdown and disable unnecessary services. 3. Install available security updates and patches.

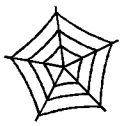
That's it! Once you've completed these steps, you've successfully installed a Linux server in "custom" mode. It is possible to install a system and let the RedHat installer make these decisions for you, but since you're going to be administrating the system, we wanted you to have as firm a grasp on the configuration as possible.

A1.2 Planning for the Installation

Choosing a Linux Distribution....

At LightSys, our Linux focus is primarily on Red Hat®[1] Linux® and related/derived linux distributions. In the fall of 2003, Red Hat (for business reasons) terminated its Red Hat Linux product line in favor of the more expensive Red Hat Enterprise Linux® product. The Red Hat Linux product continues as the Red Hat−sponsored Fedora™ Project[2]. All indications are that Fedora and Red Hat Enterprise Linux will share many technological features, so a knowledge of one will likely transfer to the other.

Unfortunately, Fedora has a very short release and support life cycle, making it far less desirable to use on a server than, say, RedHat Linux 7.3 was. Offsetting this, however, is the Fedora Legacy project which will continue to make patches available for Fedora even beyond RedHat's 8−to−9 month support window (see link below). Fedora Legacy is currently providing errata support for Red Hat Linux 7.2, 7.3, and 8.0 as well.



http://www.fedoralegacy.org/

From the Fedora Legacy website: *"The goal of The Fedora Legacy Project is to work with the Linux community to provide security and critical bug fix errata packages for select End of Life Red Hat Linux and Fedora Core distributions. This will allow for a longer effective life for those releases."*

Other common distribution choices are Debian GNU/Linux, Mandrake Linux, and SuSE Linux (which was bought by Novell in late 2003). Mandrake and SuSE are the most similar to Red Hat Linux, and share the "RPM" package manager. Debian GNU/Linux is a "purist" open source Linux distribution with a terrific package manager called "APT".

Obtaining a Linux Distribution....

The easiest way to get a copy of Linux is by ordering it from a website like www.linuxmall.com, www.lsl.com, or a similar web−site. You can order a copy of Linux quite inexpensively, plus shipping and handling (which will comprise the bulk of the price...), although the official "boxed copies" that you find in stores are not necessarily cheap. You can also download the contents of many Linux distributions for free from ftp and mirror sites, though this could take a day or so of download time,

1 Red Hat, Red Hat Linux, and Red Hat Enterprise Linux are all trademarks of Red Hat, Inc. Linux is a trademark of Linus Torvalds. All other trademarks and service marks are property of their respective owners.

2 As of early 2004, the trademark on the term "Fedora" is the subject of a legal dispute between Red Hat, Inc. and another open−source project developed by the University of Virginia Library at Cornell. Both entities have asserted trademark rights to the term "Fedora".

depending on your bandwidth. For a 384kbit network connection the download might take much of a day for the 1.8GB of a typical minimal CD set. It is usually worth the purchase cost of the CDs.

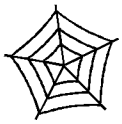
A copy of a current RedHat/Fedora distribution, or a customized distribution based on RedHat/Fedora, is included with this workbook. We've also included a "workshop companion CDROM", which contains some additional tools and utilities as well as copies of this workbook and the lecture presentations.

Gathering the Information for the Install....

The first step in setting up a Linux server is not installation, but rather planning for an installation. Even if you have a large disk−drive that could contain every Linux package known to man, you will not want to install all the packages. Not only could you overwhelm your poor CPU and memory, but you may end up with all of the security holes known to man. The rule of thumb is to only install the packages you need at any given time so you always keep a minimal number of "doors" open, hopefully just enough that you can keep your eyes on all of them at once.

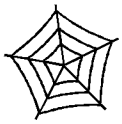
Determine the purpose of your Linux system, and be sure to think ahead to a reasonable degree. How long with the system be in operation before the next install or upgrade, and during that time frame, what will be required of the system?

You will need to know if your computer is compatible with Linux. One of the reasons for Linux's stability is the vast number of device drivers included with Linux itself; those device drivers are of a good quality since they are to some degree reviewed by the kernel development team. However, that also means that third−party device drivers can sometimes be difficult to install. If you use "standard" hardware then you can usually be confidant that Linux will work. See the following two web referencees for more information about hardware compability.



http://www.tldp.org/HOWTO/Hardware−HOWTO

If you have the least question in your mind about compatibility, you can read the Linux Hardware Compatibility HOWTO. A copy of "The Linux Documentation Project" is included on the workshop companion CD, and includes this Hardware Compatibility HOWTO.



http://www.linux−laptop.net/

If you are installing Linux on a laptop system, visit this site for some information on how you might have to customize your system to make everything work. It contains various entires by different people who have attempted, with varying degrees of success, to install Linux on a myiad of different laptops. If you are considering purchasing a laptop for Linux use, this can be a good resource to help you make a good decision.

The next step, then, is to determine what you wish to do with your computer so you will know what needs to be installed. Normally, the "simple" installer options, such as "server", "workstation", or "laptop", are not appropriate, since they often install far more than you need as well as frequently leave out needed components. You will want a trimmed−down custom install, which you can manage as you see fit. For the purposes of the workshop, see the "A1 − Installation" lab worksheet for more information on what components should be installed.

Hardware Pre−Configuration....

There are a few issues, beyond those that are normally considered in setting up PC hardware, that you will need to pay attention to regarding the computer itself before you actually install Linux.

- **Compatibility−Mode vs. Native−Mode.** Some hardware is only compatible with Linux when operating in "compatibility mode". Often, the performance of the device will be lower in "compatibility mode". See the Hardware HOWTO or the device's vendor for

more information. If you must set the hardware (via a switch, the BIOS, etc.) to "compatibility mode", do it before the installation begins.

- **System Clock − GMT vs. Local Time.** Linux allows for the computer's system clock (set via the BIOS) to be set to GMT instead of local time. For servers, this is usually what is desired. So, be sure you set the clock to GMT before doing the installation. If the system is a dual−boot setup with some version of Microsoft Windows®[3] installed on the other partition(s), then the best option is to leave the system clock set to local time, since Windows will attempt to change the system clock every time there is a daylight savings time change. Note that setting the system clock to GMT on a Linux server is largely transparent − you still see the time displayed (on file timestamps, etc.) in local time.
- **Boot Device Order.** Be sure that your computer will boot from the CDROM when you power it on; you'll need to enter the BIOS configuration to verify this. If the machine is older and incapable of booting from CDROM, see the next section on "Boot Diskettes".

Boot Diskettes....

On some systems, a "boot diskette" may be needed for the installation if the computer is not able to boot directly off of the CDROM. This was a common problem when the first revision of this workbook was produced, but has since become very rare and mainly an issue on older hardware and some laptops. Newer laptops often don't have a diskette drive at all!

If your computer cannot boot a CDROM, you will need to make a boot diskette. The following are the steps for creating such a diskette from the files on the CDROM; it can be done from a running Linux system or from a DOS/Windows system. If you intend to install Linux on a laptop, use the "pcmcia.img" boot image instead of "boot.img". Also, if you intend to install Linux from a NFS−shared CDROM on the network, use the "bootnet.img" boot image.

To create a boot diskette from DOS/Windows:

- Insert the Red Hat Linux / Fedora CDROM
- Go to a DOS prompt (Start −> Run −> command.com or Start −> Run −> cmd.exe)
- Go to the directory "\dosutils" in the CDROM.
- If the disk (in A:) is not already formatted, run: `FORMAT A:`
- Run: `RAWRITE −f \images\bootdisk.img`

To create a boot diskette from Linux:

- Insert the Red Hat Linux / Fedora CDROM
- Mount the cdrom: `mount /mnt/cdrom`
- Go to the 'images' directory: `cd /mnt/cdrom/images`
- Format the diskette, if necessary: `fdformat /dev/fd0H1440`
- Copy the boot image to the diskette: `cat bootdisk.img >/dev/fd0`
- Unmount the cdrom: cd /; umount /mnt/cdrom

(note: on earlier versions of Red Hat Linux, the boot diskette image name is "boot.img" instead of "bootdisk.img").

Planning the Hard Disk Usage....

You will need to decide on a partitioning structure for your Linux system. The following guidelines work best on systems with a disk larger than about 4GB.

3 Microsoft and Windows are trademarks of Microsoft Corporation.

Partition	*Sizing*	*Description and Notes*
/boot	75MB	This partition is optional, but it saves many headaches for LILO and GRUB, the Linux boot loaders. Linux systems sometimes need a location to place boot−up information that is *guaranteed* to be in the first 1024 cylinders of the disk. Creating this partition, and making it the first one on the disk, provides this guarantee.
swap	Min. 64MB Rec. 127MB (see notes)	Linux uses a "swap partition" instead of a "swap file" like Windows uses. The largest possible swap partition size used to be 127MB on Red Hat Linux 7.0 and earlier (but you could have more than one). On newer systems, you can have more swap space, but if your server is swapping that much, you need to add memory or act to conserve memory use (see chapter A3). A few early versions of the 2.4 series kernel required the swap to be larger than the amount of physical RAM to be effective; that "mis−feature" was very quickly done away with.
/home	Min. 64MB Rec. 512MB+	Put most of your "extra" hard disk space here; this is where your users will store their files. If your server will not have user accounts (e.g., a web server), then you may consider putting some "extra" space in /var (see below).
/usr	Min. 2.5GB	Plan on more space if you anticipate installing or building many custom packages. This is where most of your installed programs will reside.
/var	Min. 256MB Rec. 384MB+	On Red Hat 7.x and newer, unlike the 6.x series, your web documents are stored in /var, so if you will be having a large number of web documents, adjust this size accordingly. Red Hat 6.x and earlier stored web documents in /home/httpd/html. Lots of system temporary files and "spool files" are created in /var, so having a separate /var partition keeps the / partition from filling up, and localizes any fragmentation to the /var partition. Lastly, your email inboxes are stored in /var − so leave more room here if you plan on having lots of (or large) inboxes.
/tmp	Min. 128MB	Put more space here if your users will be creating a lot of temporary files. Having a separate /tmp partition prevents a user from filling up / by writing lots of files into /tmp.
/	Min. 256MB	At least 256MB, if you create partitions as above. If you do not create some of the above partitions, then you must add the minimum space requirement for those into the space requirement for the / partition. For example, if you don't have a /tmp partition, then make your / partition at least 384MB.

WARNING: Once again, all the previous filesytems, if you do not set aside a specific partition for them, will end up as a part of the root (/) filesystem. Therefore, if you choose not to create a partition listed above, take into account that the root (/) filesystem will need to be larger. Otherwise the install could fail!!! Make sure you pay careful attention to sizing your partitions, since you will have to live with them for a long time. The problem making more room for a partition up can be difficult to address on Linux systems (see chapter A3 for more information), so it is best to deal with the issue up−front and plan your disk usage well.

On disks closer to 2GB, the following partitioning scheme might be useful. Be sure to create a /boot partition (18MB this time) also if the disk has more than 1023 cylinders.

swap − 64MB
/var − 128MB
/home − 256MB
/ − remainder of disk.

On a disk smaller than 1GB or so, use the following partitioning scheme. Be sure to create a /boot partition (18MB this time) also if the disk has more than 1023 cylinders. Note that on newer version of Red Hat Linux / Fedora, a 1GB disk will probably not be large enough.

swap − 64MB
/ − remainder of disk.

WARNING: this partitioning scheme gives you the greatest flexibility for a small disk, but it has been known to be dangerous in some cases when you upgrade versions of Linux. Furthermore, for a number of reasons it isn't best to have only one filesystem partition.

Planning for Network Connectivity....

Be sure you get appropriate network connectivity information from your network administrator. In the workshop, this information is provided on your installation planning form. The most important thing here is to never use DHCP or BOOTP to configure a server, primarily because it makes the server's proper functioning dependent on the other server that is running BOOTP or DHCP services. Here are some notes about the TCP/IP network settings:

- **IP Address:** The numeric address for your computer (such as 192.168.1.1)
- **Netmask:** The subnetwork mask for your network (this should be the same for all computers on your LAN). Example − 255.255.255.0.
- **Gateway:** The default gateway for your network. If the Linux server is your only server on the network (and you do not have an internet connection), or if the Linux system will be handling all network traffic passing to and from your LAN (e.g., over a connection to the Internet), do not specify a default gateway unless you specifically know what it is supposed to be.
- **Nameserver:** The existing DNS server on your network. If the machine is to be your network's DNS server, type the machine's own IP address here.

More About TCP/IP and IP Addresses

Would you like to learn more about TCP/IP, IP addresses, subnet masks, etc.? If so, be sure to check out our course on "Internetworking and Information Security". Previous versions of our Linux Workshop workbook included a chapter on how TCP/IP works, but we've moved that chapter to our other workshop since we believe that it fits better there. Feel free to contact us at Information@LightSys.org for more information!

If you are planning to set up a machine with two or more network interfaces (e.g., a firewall, router, or server connected to a production and administrative LAN simultaneously), be sure to gather information about all network interfaces although if you are installing in "text mode", setting up the second (and third) interfaces may require configuration after the installation is complete. See chapter A4 for more information.

Planning for Basic Security....

First, you need to chose a "root password". The "root" account on a Linux system is the administrative account that has (on most Linux and UNIX®[4] systems) full rights to every file and directory on the entire system, so you will need to guard that password carefully.

Passwords should always have two non−standard characters (capitals, numeric, or punctuation), and should never be a real word. Variations on non−English words (especially in obscure languages) or the first letters of your favorite Bible verse or hymn, *if mixed around a bit*, work very well. It should be eight characters in length at a minimum. Here are some examples (but don't use these exact ones!!!):

"3j4Gsltw" −− John 3:16, "For God so loved the world"
"Hhst0mwig" −− "He has shown thee O man, what is good"

You need to decide how users will be authenticated. That is, what machine or service on your network has the main knowledge of users' passwords and is to be trusted with verifying those passwords? On most Linux systems, nothing special has to be done, but if you have a Windows NT/2K system that will authenticate users, or if you have a Kerberos/LDAP, or NIS server, you need

4 UNIX is a registered trademark of The Open Group.

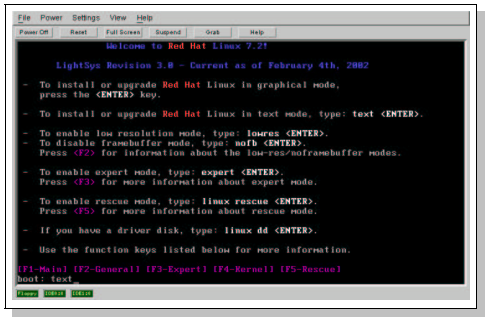
to know that in advance of the installation.

Finally, the installation will want to set up a simple rudimentary firewall on the system to filter out rouge packets coming in. It is especially important to know how the system will be used at this point; for example if you will be running a web server, you will need to tell the installer that during the setup of the firewall. If you want to do more advanced firewalling, see our Internetworking and Information Security workshop for further details.

A1.3 Starting the Install

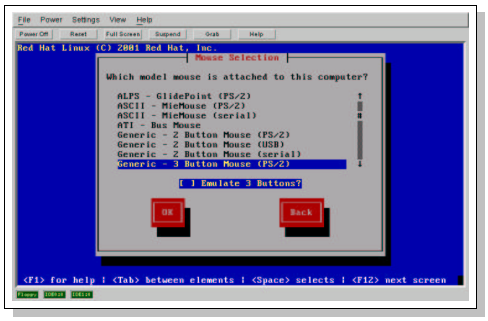
Starting the Installer....

You know how to do this one! Just put the CDROM (and boot diskette, if you had to make one) in the drive and power the machine on. If the CDROM or diskette boots correctly, you will be presented with an initial installer screen with a "Boot:" prompt at the bottom. At this point you normally just press [ENTER] to begin the process. However, there are some special cases that you might come across from time to time:



- **Text Mode.** In some cases you may want to start the installer in "text" mode by typing "text" and pressing [ENTER]. This is normally used if your video card is unusual or otherwise not supported by Linux. You can also use this if your computer has no mouse attached. However, modern Red Hat Linux and Fedora installers sometimes omit features from the text−mode installation, so we recommend the default graphical installer as it also is much easier to use.
- **Memory.** If, shortly after the installer launches, you get a lot of error messages like "Oops" and "Panic", and things appear to crash, then Linux may not have detected the amount of memory in the machine correctly. We see this happen infrequently, but often enough that it is worth mentioning (even the two servers running the LightSys.org domain have this problem!). In this case, try entering the amount of memory in your machine at the Boot: prompt. If your machine has 128MB of memory, type "linux mem=128M" or "text mem=128M" at the prompt, depending on whether you want to use the graphical or text mode installer. Make sure you get the amount of memory correct if you do this! A machine with 128MB of memory will normally display "131072KB" during the BIOS memory check. That does not mean it has 131MB of memory − you have to divide the 131072KB by 1024, not 1000, to get the correct MB, although some BIOSes may display a number slightly smaller than the real amount of memory in the machine. As a rule of thumb, if you get an odd number (such as 127) after dividing by 1024, round it up to the next even number (in this case, 128). Also, some computers use a "UMA" memory setup, where part of the main memory is used by the video card. So, even if you put 128MB in the machine, if the BIOS has the video card set to use 8MB, you actually only have 120MB available for Linux. In this case the computer will crash if you enter "linux mem=128M" at the boot prompt!!!
- **Rescue Mode.** You won't use this during installation, but it is good to note that you can type "linux rescue" at the boot prompt to enter "rescue mode", where the installer attempts to find an existing Linux installation on the hard disk,

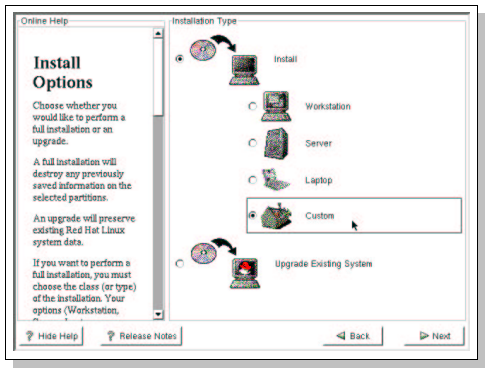
Language, Keyboard, and Mouse....



After the installer has begun, it will ask you a few questions about your language, keyboard, and type of mouse. The only item here which can be difficult, if you have never used Linux much, is the mouse issue. Linux and UNIX systems make full use of a three−button mouse (mice with scroll wheels are three−button mice), and if your mouse only has two buttons, you need to check the box marked "emulate three buttons". In that case, clicking the left and right buttons simultaneously will do the same thing as clicking the (nonexistent) middle button.

Install Type and Upgrade vs. Re−install....

After the initial questions regarding the keyboard, language, and mouse, you will be asked what kind of installation you will be doing. On Red Hat 7.x and earlier, we recommend that you choose the "custom install" option that allows you to select individual groups of packages instead of going with the "automatic" server or workstation installations. In this session we will be showing you what packages and components you need to select, depending on the purpose of the computer. On newer versions, such as Fedora, some options are omitted (!) if you use "Custom Install", and so we recommend selecting "Server".



However, if you are upgrading an existing system, choose the "upgrade" option on this screen (it may be on a previous screen for some versions). Note that we do not recommend upgrading a system using this method more than once − there are small issues that tend to accumulate over the course of many system upgrades, and it is often better to do a complete reinstallation, restoring the needed data files once the reinstall has completed. It is more work to do a reinstall and restore of data files, but our experience is that you are often better off if you do it that way.

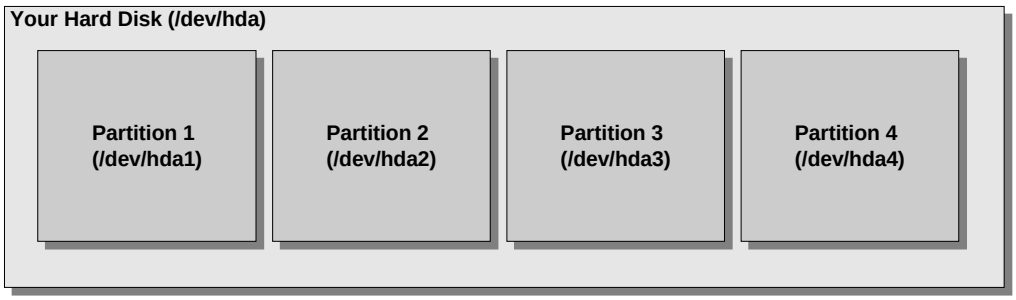
When you upgrade, pay particular attention to the default locations of the various data and configuration files − when you restore your data files, the server may be looking for them in a different place. For example, on Red Hat 6.2, the normal location for the WWW documents was in /home/httpd/html. On Red Hat 7.0 and above, the location changed to /var/www/html. Not only would the files not work in /home if restored there, but when you move them to /var, it will require more room on your /var partition than you needed before. It is often good to read the release notes for the version of Linux you will be installing.

N1.4 Setting up your Hard Disk

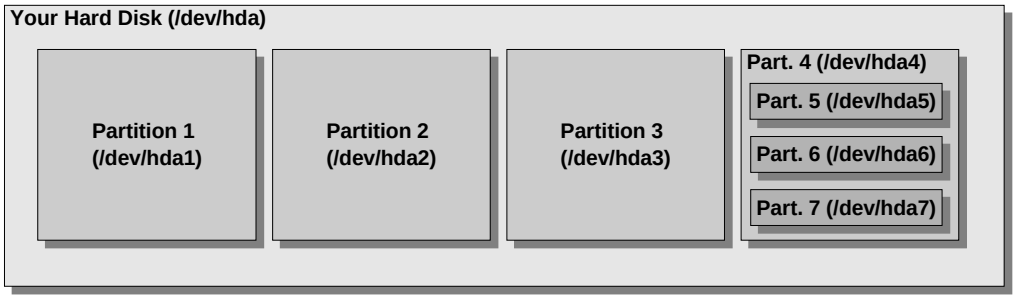
The Partition Table (Primary vs. Extended vs. Logical)....

Before we get into the actual process of partitioning your hard drive, let's discuss the way that the partition table on a normal PC hard disk works. Remember from your installation planning that Linux systems, unlike Windows systems, work best when you use more than just one partition.

PC hard disk partition tables can have up to four main entries, called "primary partitions". The below diagram illustrates this:



Unfortunately, this turned out to be far too restrictive, and room for more than four partitions was needed. In order to avoid "messing with" the main partition table and doing the simple thing by making it larger, the notion of an "extended" partition was created. The extended partition would work in place of one of the primary partitions, and could contain sub−partitions, called "logical partitions". You wouldn't format the extended partition itself, but rather format the logical partitions within the extended partition:



You may have as many partitions as you wish in your extended partition, up to 32 total partitions (including the primary and extended ones). In the above example, you have seven total partitions with three primary (1, 2, and 3), one extended (4), and three logical (5, 6, and 7) in the extended. This gives you six total partitions that you can put data on (3 primary + 3 logical). Usually, five of them will be filesystems and the sixth will be a swap partition.

On DOS/Windows systems, you could only have one primary partition; all other data partitions were created as "logical" partitions in an extended partition. On Linux, you may have between one and four primary partitions, but if you use all four primary partitions, there will not be room for an extended partition.

Here is an example of what a disk partitioned with extended partitions and logical partitions might look like for /dev/hda, the first IDE disk in a machine (this is a sample partitioning for a 4GB disk):

```
/dev/hda          -     entire disk
/dev/hda1         -     /boot        (24MB)      (primary)
/dev/hda2         -     swap         (127MB)     (primary)
/dev/hda3         -     /            (256MB)     (primary)
/dev/hda4         -                  (3.5GB)     (extended)
/dev/hda5         -     /usr         (2GB)       (logical)
/dev/hda6         -     /var         (256MB)     (logical)
/dev/hda7         -     /tmp         (256MB)     (logical)
/dev/hda8         -     /home        (1GB)       (logical)
```

Does something look strange in the above listing? Oh − you were saying that the sizes don't add up to 4GB? The reason for this is that the extended partition, /dev/hda4, "holds" all of the logical partitions (5 through 8, in this case). So, indeed 24MB + 127MB + 256MB + 3.5GB adds up to about 4GB. The 3.5GB, in turn, is comprised of 2GB + 256MB + 256MB + 1GB.

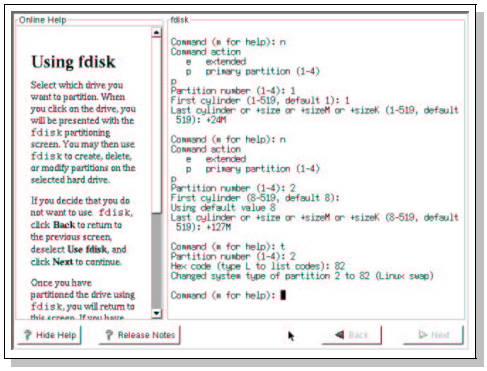
To FDISK or not to FDISK....

You have a choice of what tool you want to use to partition your hard disk. The installer will generally give you three options: automatic, disk druid, or fdisk. Since you already (in the install planning phase) decided on how much space you wanted to put on each of your partitions, we recommend against the 'automatic' partitioning approach − you know much more about how your server is to be used than the installer does at this point.

On newer versions, such as Fedora, the installer no longer displays the fdisk option. You can still use it, however. At the "automatic vs. disk druid" screen, press ALT−CTL−F2, and type "fdisk" followed by the name of your disk device. For the first SCSI hard disk, this would be /dev/sda, and /dev/sdb for the second SCSI hard disk, and so forth. For IDE devices, /dev/hda is your primary master, /dev/hdb the primary slave, /dev/hdc the secondary master, and /dev/hdd the secondary slave device. Then, once you finish using fdisk and are back at the shell prompt, press ALT−CTL−F7 to return to the installer, and select "disk druid" to choose your filesystem types and mount points.

We recommend that you use the **fdisk** program for partitioning. It is a little bit more difficult than using disk druid, but there are a couple of advantages to using fdisk (remember that this workshop is about learning Linux, rather than about figuring out the easiest possible way to install Linux!):

- **You Learn FDISK.** It is a good idea to learn how to use fdisk in the event that you have to use it in an emergency or if you add a new hard drive to your system and need to partition it without the assistance of the installer.
- **Avoids Potential Problems.** Sometimes the "disk druid" partitioner (the one otherwise used by the Red Hat installer) will create the partitions with some anomalies. The system will appear to work fine, but some recovery software might not handle the partition table correctly. We've seen this happen, and you definitely want your recovery software to work!!!



If you intend to use software RAID, then set up all of the disks in the array identically using fdisk, and make a note of what devices (such as /dev/hda5 and /dev/hdb5) are used by what partitions (such as /usr).

FDisk is a command−line program; below is a table of the most common fdisk commands that you might use:

Command	*Description of Command* [5]
A	Make a partition active (bootable)
M	Display help (menu of command options) − this option is your friend!!!
P	Print (display on−screen) the partition table
D	Delete a partition
N	Add a new partition
T	Change partition type code (what the partition will be used for). Use "L" to get a list of them.
L	List partition types. Important ones are 82 (linux swap) and 83 (linux filesystem)
Q	Quit without saving changes
W	Write changes to the hard drive's partition table and then exit.

Here are a few notes about using fdisk:

- **Not On a Running System!** WARNING − fdisk should not be used on an installed and running system unless you really know what you are doing. If you make changes to the partition table with fdisk, it *does NOT* preserve the data on those partitions! You could lose data! See session A3 (general maintenance) for some advice on resizing partitions.
- **On a New System...** To use fdisk when installing a completely new system on a server (which should only have one operating system), begin by deleting all the previous partitions. Then add each partition, in the order you want them. This will destroy any data previously stored on the hard disk.
- **Type of Partition with 'N' command:** When you create a new partition (with 'n'), you will need to immediately thereafter give it a type (using 't'). The type code for a Linux swap partition is 82, and for a Linux filesystem it is 83.
- **Make One Active.** Make sure one partition is marked active (bootable), with the 'a' command. This will be your /boot partition, if you have one, or your / (root) partition if you have no /boot.
- **What is this "/tmp/hda" business?** During installation, the partitions may be listed as something like "/tmp/hda" instead of "/dev/hda". This is normal.
- **Hold Your Horses...** In fdisk, you do not specify *where* partitions are mounted, but rather just the *sizes* and *order*. You specify where they are mounted later. Just make sure you have your partition planning written down, and note the device (such as hda5) next to the partition name. See the "Mount points" section next for more information.
- **"Free Hog" Partition Goes Last.** If you have one partition listed as using "the rest of the disk", make it the last partition you create − on the last one you typically tell it to use the remainder of the disk instead of typing in a specific size, anyhow.

Next, let's take a look at a real−life example of fdisk in use; here we will delete a partition, create a new partition, and set the type of that partition to be a swap partition.

A Sample FDISK Session....

```
Disk /dev/hda: 128 heads, 63 sectors, 620 cylinders
Units = cylinders of 8064 * 512 bytes

Device    Boot Begin  Start    End   Blocks   Id  System
/dev/hda1   *      1      1    204   822496+   6  DOS 16-bit >=32M
/dev/hda2        205    205    237   133056   82  Linux swap
/dev/hda3        238    238    254    68544   83  Linux native
/dev/hda4        255    255    620  1475712    5  Extended
/dev/hda5   *    255    255    287   133024+  83  Linux native
/dev/hda6        288    288    320   133024+  83  Linux native
/dev/hda7        321    321    620  1209568+  83  Linux native
```

5 Source: based in part on the output of fdisk's "m" command. FDisk is Copyright (C) 1992 A. V. Le Blanc (LeBlanc@mcc.ac.uk), and is redistributable under the GNU GPL.

A Sample FDISK Session....

```
Command (m for help): d                          (Delete a partition)
Partition number (1-7): 2
Command (m for help): n                          (Create a new partition)
Command action
   l   logical (5 or over)
   p   primary partition (1-4)
p                                                (specify a "primary" partition)
First cylinder (205-237, default 205): 205
Last cylinder or +size or +sizeM or +sizeK (205-237, def. 205): +127M
                                                 (specify size "+" in Megabytes "M")
Command (m for help): t
Partition number (1-7): 2
Hex code (type L to list codes): 82              (set a partition as being swap)
```

Where to Mount the Partitions....

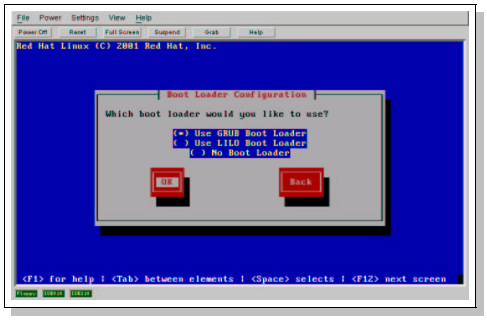
First, if you are considering the use of software RAID, see section A1.7 on "advanced installation topics" for more information.

In addition to partitioning the disk with fdisk, you will need to specify where each partition will be mounted. Once you are finished with fdisk, move on to the next screen and, for each partition except for the "swap" partition, select "edit" & type the mount point name such as "/", "/var", or "/usr".

At this point you will also be able to choose filesystem types. On most modern Linux systems, it is appropriate to chose "ext3", which is a "journaling" filesystem which helps to protect your data in the event of an unclean shutdown of the system. The "ext2" filesystem is the traditional Linux filesystem and is still in use in some circumstances.

The Boot Loader, LILO or GRUB....

LILO is the program which handles the booting of the computer. It works by pointing to the physical location of the boot information (such as the kernel) on the hard disk, so that when the computer boots it doesn't have to go searching through the complicated structure of the filesystem in order to find the boot information. Whenever you change this (by adding a new kernel, moving any of the boot files, etc...) you need to re-run LILO, since the physical location of the boot information on the disk may have changed. Re-running LILO is as simple as typing "lilo" at the command prompt (assuming you have logged in as "root").



The basic limitation of LILO is that in some cases it can only "reach" the first part (actually, 1024 cylinders) of the disk. By making the boot partition the first partition, all the boot information is close to the beginning and LILO will never have this problem.

The newer bootloader that comes with Red Hat Linux 7.3 and above, GRUB, presents a nice menu for booting more than one operating system, and allows for password protection when booting, to prevent someone from arbitrarily providing the booting kernel with potentially dangerous options. GRUB does not require you to re-run any program when you change its configuration, and thus can be less complicated to use in some cases. However, if you choose to use software RAID, do not use GRUB − while it may appear to work at first it does not always handle the boot situation properly

should one drive fail, and cannot be re−installed properly on the MBR due to GRUB limitations. Use LILO instead.

However, the Fedora installer provides no way to automatically use LILO. To do so, you must configure that after the system reboots (see post−Install configuration on page 24).

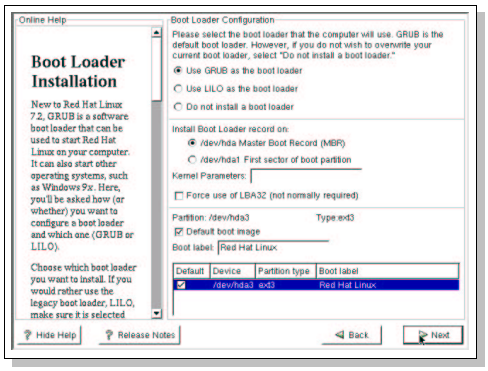
Booting

When we think of "booting" a computer, we often remember the last time that a computer malfunctioned so severely that we were sorely tempted to "give it the boot"! But where did the term "boot" really come from?

The term "boot" is an ancient one in the computer world, derived from the old expression "to pull yourself up by your own bootstraps". It describes the classic catch− 22 scenario where Linux is needed to read a Linux partition, and the startup information is on the partition, but Linux isn't loaded yet! So how does the computer get Linux loaded?

Answer: LILO or GRUB.

Install the boot loader on the MBR where at all possible. The only exception to this is if you are using a different boot loader on your hard drive, such as the Win2K boot loader or something like BootMagic® [6]. However, note that you will have to inform your other boot loader that Linux is installed on your hard disk, otherwise you won't be able to boot your Linux installation!



If you perchance had to specify a "mem=" argument at the Boot: prompt when starting the installer, make sure that it is included in the "arguments to pass to the kernel" option.

A1.5 Initial Networking and Security Choices

Overview....

In recent years, operating systems have gotten a lot better about making the installation more secure out−of−the−box. However, to gain the most from these features, while still being able to effectively use the computer, you need to know how to properly answer some of the questions during the installation process. In the few pages that follow, we'll discuss some of these issues in more detail.

6 BootMagic is a registered trademark of Symantec Corporation (formerly of PowerQuest Corporation).

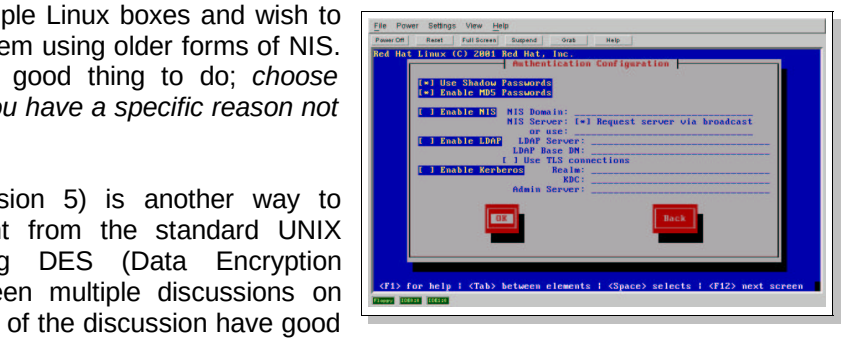
Users and Passwords....

During the installation planning, you chose a root password. You will be entering that during the install process, as well as optionally creating a few user accounts. If you have a large number of user accounts to create, then it is probably better to add them once the install process is complete.

Authentication Configuration...

Note: on newer versions such as Fedora Core 1, the authentication configuration process must be performed after the install has completed. See "Post−Installation Procedures" later in this session for more information.

Shadow passwords provide you with another level of security by preventing ordinary users from even being able to view the encrypted form of other users' passwords. Using shadow passwords isn't helpful if you have multiple Linux boxes and wish to share passwords between them using older forms of NIS. Otherwise, it is definitely a good thing to do; *choose shadow passwords unless you have a specific reason not to do so.*



MD5 (Message Digest version 5) is another way to encrypt passwords, different from the standard UNIX password encryption using DES (Data Encryption Standard). There have been multiple discussions on which is best, and both sides of the discussion have good arguments. DES is good for backwards compatibility with older programs; however, using MD5 is more secure and allows passwords *much* longer than 8 characters to be used (up to 256 characters are possible − that's a long password!). *Choose MD5 unless you specifically know a reason why you must use DES.*

What in the World is MD5?

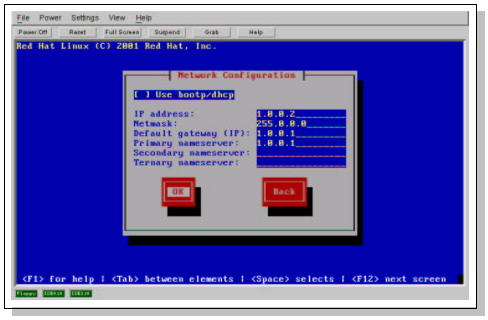
MD5 is what is called a cryptographic hash function. It takes input data of any length and performs a complicated series of calculations on that data that results in a 128−bit (16 byte) value called the "message digest", or MD5 checksum. It is normally used for verifying the integrity of data, but is also used in password encryption.

The process of "authentication" is the computer's way of making sure a user is who he or she claims to be. You wouldn't want just anyone logging into your server and gaining access just because he or she claimed to be "root"! So, your Linux system requires a password, which is something that only the user in question should know. However, you are offered several methods that Linux can use to check that password.

- The **default** is to simply use the Linux system's own user/password information files. Choose this if your computer will be operating in a standalone fashion or have very few users.
- **SMB** uses a Windows NT/2K domain controller or a Samba server to authenticate the users.
- **NIS** (Network Information Service) is a service which allows multiple Linux systems to share passwords. If you do not have multiple Linux systems, and don't anticipate having multiple Linux systems in the future, do not use NIS. If you do, NIS is a good option.

- **LDAP**, or Lightweight Directory Access Protocol, is a means by which a centralized directory can be accessed. Linux can validate password and login information using a centralized LDAP−compatible directory.
- **Kerberos** is a cryptographic single−sign−on authentication system. For a more advanced configuration with multiple Linux systems which support Kerberos v5, this option is definitely something to consider, although Kerberos is conceptually more complicated to setup and use. Note that Kerberos and LDAP are the technologies that products like NDS and Active Directory are based on.

TCP/IP Networking Configuration....



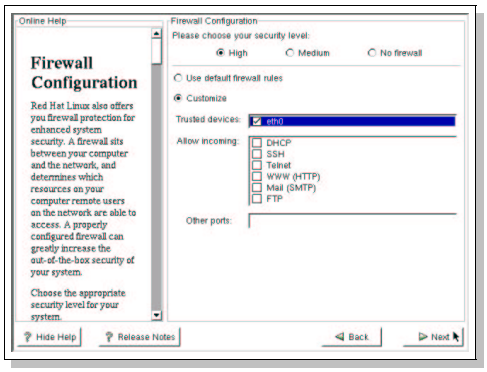
In the networking setup part of the installation, you will be asked for the information you decided upon during your installation planning step (see page 13 in this section). Note that the installer may try to automatically fill−in the default gateway, DNS server, and subnet mask. Be sure that you correct any incorrect guesses that the installer makes.

You will also be asked for your computer's "hostname". You should type the full host and domain name for the machine. For example, don't just type "server", but rather "server.some.domain.tld".

If you are installing in "graphical mode" instead of "text mode", then you have the option of setting up all of your network interfaces if you have more than one. The text mode installer in at least some versions of Red Hat Linux seems to be missing that option. In that case, you will need to configure the other network interfaces after installation is complete.

Firewall....

The installer will offer to set up some simple firewalling rules for you. Unless you plan to immediately construct a more sophisticated set of firewalling rules before hooking the machine up to a network, it is a good idea to go along with the installer's idea of a firewall. Here are our recommendations on the installer's firewall (on later versions, there is no "Medium" option − either the firewall is on or off. Due to improvements in the technology, it provides approximately the same security as the old "High Security" option while allowing the flexibility of the old "Medium Security" option).



- **No Firewall.** Choose this option if you are going to be setting up a more sophisticated firewall on the machine before it is connected to a network, or if the machine will never be connected to a network.
- **Medium Security.** If the machine will be used as a workstation, usually Medium Security is best. The highest security level can block some things that an end−user might want.
- **High Security.** If the machine will be used as a server, choose High Security (unless you plan to develop more sophisticated rules, as we mentioned earlier). Even if you plan to develop a more sophisticated firewall, often it is a good idea to enable the installer's High Security firewall as an interim solution so you can connect to your LAN and download packages to the machine before the full firewall

is in place.

You can also specify additional ports ("holes") to open up. Look at your installation planning materials, and open up ports for those services that the machine will be offering to an untrusted network.

The "trusted network" option can be confusing. If your computer will only have one network interface, such as a server on a DMZ, or a server or workstation on an internal network, then you should not mark any interface as trusted, and configure the firewall with respect to what is allowed and disallowed from the network the machine is attached to. However, if the machine will be used as a firewall, router, or other system having more than one network card, then you might mark the "inside" network interface as "trusted" and configure the firewall with respect to what is allowed and disallowed from the "outside" (have you noticed that the installer's firewall is not really very flexible?).

You can always re−configure the installer's firewall at any time after installation by running the "lokkit" command on the Red Hat 7.x Linux versions (we wonder − is that "Lock It" or "Lock Kit"?).

A1.6 Selecting Packages to Install

Overview....

Keep in mind that you do not usually want to install every available package that comes on the install CD's. Not only will it use up much more of your hard disk space, but as we mentioned earlier, each package you install comes with its own set of probable bugs and potential security holes! Refer back to your installation planning materials, and decide what packages need to be installed.

Problematic Packages....

Occasionally, Linux distributions are overly aggressive about adding new functionality to a distribution before that functionality is really "ready" for widespread use. In these cases, you may have to manually de−select such packages since the installer may want to put them in for you.

Here are a few examples that we can think of:

- **Early up2date Versions.** When the "up2date" system, which can help automate the patch−installation process and thus help with security, was first released, it had a lot of problems and caused a lot of trouble. It got much better with time, but today's challenge is Red Hat's lack of substantial support for Red Hat Linux and Fedora. Fedora uses the "yum" package for managing updates.
- **Early Foomatic Releases.** "Foomatic" is an advanced print spooler management system which helps to automate much of the process of configuring printers and selecting the best drivers. Unfortunately, it was extremely troublesome when first released, so we recommended that Red Hat 7.1 and 7.2 users install the "printtool" and "rhs−printfilters" packages from Red Hat 6.2 or Red Hat 7.0 instead, which provided the much more traditional UNIX−style "printcap" printer configuration and a tool to set it up.

A1.7 Advanced Installation Topics

Software RAID....

The Red Hat installer now makes setting up a system using Linux's builtin software RAID fairly easy. However, know that managing that software RAID, should a device actually fail, is something that is normally done at the command line (and which you will need to learn *before* a hard disk actually fails). We generally recommend hardware RAID controllers, where feasible, if RAID is needed.

RAID

RAID stands for "Redundant Array of Inexpensive (or Independent) Disks". It is a technology that groups several disks together in such a way that if one disk fails, the computer is able to keep running as if nothing happened. RAID does "waste" some space to provide that redundancy, but is a standard technology on servers today. "RAID 1" or "mirroring" requires two disks. "RAID 5" requires three or more disks and is more efficient in terms of space. While explaining how RAID works is beyond the scope of this workshop, now you know who to call if you see a cockroach contentedly gnawing away at one of your hard drives!

To set up software RAID, you will need to create the RAID devices before you specify where to mount each of the partitions. First, go through and mark all of the partitions as being "software RAID". Next, create the RAID devices. To do so, simply click "Make RAID". For each RAID partition, you will want to specify the partitions that comprise it. For example, your /usr RAID partition may be made up of the /dev/hda5 and /dev/hdb5 partitions.

WARNING − Never, ever, include two partitions from the same disk in a software RAID partition. If that disk fails, the entire software RAID partition will fail, which entirely defeats the purpose of software RAID! Furthermore, the performance impact of making this mistake can be very negative. When setting up software RAID, partition all of your drives identically, and include one partition from each drive in each of the RAID partitions.

A1.8 Post−Installation Procedures

Hardening....

WARNING − it is absolutely necessary to do some post−installation "hardening" to your system once the install process is complete. If you do not do these steps, you are opening your machine up to attack before you've really even gotten it to the point of being useful! While newer versions of Red Hat Linux are not as vulnerable out−of−the−box as older versions (such as 6.2) were, nevertheless these steps are needed. Here are the basics:

1. **Shut down and disable unneeded services.** See session A3 on the topic of managing your system to find out more about how to enable and disable system services. You will want to look into services that start and stop via the SYSV init scripts (/etc/rc.d/init.d) as well as via xinetd (or inetd, on older systems). Very rarely, a service might run from /etc/inittab. You can use the "netstat" command (see session A4) to find out what

services are actively listening for network connections. Remember that "shutting down" and "permanently disabling" are two different steps in most cases! Common services that you will usually end up disabling include "sgi_fam", "portmap", and "nfslock".

2. **Set up firewalling.** If you did not go with the installer's firewall, you need to set that up immediately after installation.
3. **Boot Disk.** If you didn't make an emergency boot disk during the installation process, make it now by using the command "mkbootdisk".
4. **Install cryptographic key(s).** Install the RPM signing key off of the distribution media. In the case of Red Hat systems, this is called "RPM−GPG−KEY" and is on each of the CDROMs. Use the "gpg −−import" command to install the key on your system.
5. **Update the system.** You must apply all needed security patches to the system. Immediately after installation, we recommend simply applying all available security and bug−fix packages. If you are going to be using an update manager (like yum or up2date), do this now as well. However, if any of the updates are for the package manager (RPM in our case), update manager (up2date or yum), or gnupg, you will want to install those updates manually before anything else, and verify them using "md5sum packagename.rpm" (comparing with the checksums on the distribution's errata website) and then verify using "rpm −−checksig packagename.rpm". This procedure protects against vulnerabilities and bugs in the package management system itself. After updating these packages, make sure your update manager will verify the signatures on the packages; if it does not, download the packages and verify the signatures before updating (from the newly downloaded packages).
6. **Reboot.** Some updates, like glibc and the kernel, won't take full effect unless you reboot the computer.

Note that this is *not* an exhaustive list of everything that you can (or even should) do to your computer to harden it. Rather, it is the minimum you must do after installing a new system. You should consider the installation incomplete if you have not done this step.

When initially updating your system, for the best security, it is best to not connect the machine to a network before the updates have been applied. However, to do that, you need to download the updates on a separate machine, verify their signatures (and md5sums first, in some cases, see above) and burn them to a CDROM.

Also note that when shutting down and disabling unneeded services, you will likely need to do this with the command−line "service" and "chkconfig" commands even if you eventually plan to use linuxconf, webmin, or gnome for management, since the more advanced tools won't likely be configured or available yet.

Configuring LILO....

If you are using software RAID and are using a newer release of Red Hat Linux or Fedora, you will want to use LILO as the boot loader instead of GRUB since GRUB does not make the second hard disk bootable (should the first one fail) whereas LILO does. However, the installer provides you with the correct configuration file for LILO, in the file /etc/lilo.conf.anaconda.

To set up LILO, simply copy /etc/lilo.conf.anaconda to /etc/lilo.conf and type "lilo" at the prompt.

Note however that the use of LILO is officially deprecated with the release of Fedora Core 1.

Post−Install Configuration....

On newer versions, you will have the opportunity to further configure your system after the installation process completes. If you installed the graphical environment (the X Window System), this post−install configuration will automatically begin when you reboot.

On a server, however, you may have omitted the X Window System entirely, and if so, you may want to configure the authentication mechanism by running the "authconfig" program as root from the command prompt.

Getting Acquainted....

After performing an installation, familiarize yourself with the setup, particularly if it is the first time you have installed a particular version of Linux. You will want to investigate things such as how full your hard disk partitions are, how much memory is in use, and what processes are running on the system. See sessions A2 and A3 for more information.

Knowing your system well can help you recognize and troubleshoot problems (including potential security compromises) more effectively.