

Configuring a Red Hat Linux System For Multiple RAID-1 Devices

Graham Leach

Managing Partner

TIG – The Imperators Group, Inc.

© 2002 TIG

Table Of Contents

Why Do This At All?.....	4
RAID-1 Is A Good Thing.....	4
Setting up Multiple RAID-1 Devices in Red Hat 7.2.....	5
The Partitioning Strategy.....	5
What We're Trying To Achieve.....	6
The Distribution.....	7
The BIOS Setup.....	8
The BIOS Setup Screen.....	9
Getting Into the IDE Auto Detect Screen.....	10
Checking the results of the IDE Auto Detect Screen.....	11
Enable boot from CDROM.....	12
Saving BIOS Changes.....	13
Bootting From The CD-ROM.....	14
Loading the Linux Setup Kernel.....	16
Loading The Linux Kernel From CD-ROM.....	17
Starting The Anaconda Setup Program, part I.....	18
Starting The Anaconda Setup Program, part II.....	19
Seeing X-Windows for the First Time.....	20
The Red Hat Setup Splash Screen.....	21
Language Selection.....	22
Keyboard Configuration.....	23
Mouse Configuration.....	24
Select The System Profile.....	26
Choosing Your Partitioning Strategy.....	27
Using fdisk.....	28
Partitioning hda.....	29
The fdisk Commands.....	29
How To View Partitions With fdisk.....	30
How To Delete A Partition With fdisk.....	31
Creating A 20Mb Boot Partition.....	32
Making A Partition Bootable With fdisk.....	33
Creating A Swap Partition.....	35
How To Create An Extended Partition For Linux.....	36
How To Create A Logical Drive In An Extended Partition.....	37
How To Tag A Partition For RAID.....	38
All Of The RAID Partitions Prepared.....	39
How To Write The Partition Table.....	40
Partitioning hdb.....	41
Make hdb An Exact Duplicate Of hda.....	42
Writing The Partition Table A Second Time.....	43
Disk Druid.....	44
Using Disk Druid To Setup Drives.....	44
How To Configure the /boot Partition.....	45
How To Configure The swap Partition.....	46
Using Disk Druid To Make A RAID Device.....	47
How To Select The Appropriate RAID-1 Partitions for /.....	48
How To Configure the /usr Partition.....	49
How To Configure the /var Partition.....	50
How To Configure The /home Partition.....	51
How To Configure The Duplicate /boot Partition.....	52
How To Verify The RAID Configuration.....	53
How To Confirm Partition Formatting.....	54
The Boot Loader.....	55
How To Configure The System To Use LILO.....	55

Where To Locate The Boot Code.....	56
Networking.....	57
How To Configure Networking.....	57
An Example Network Configuration.....	58
How To Defer Firewalling.....	59
Additional Language Support.....	60
How To Enable Additional Language Support.....	60
Time Zone.....	61
How To Configure Time Zone Support.....	61
Accounts And Passwords.....	62
Setting the root Password.....	62
Enabling MD5 And Shadow Passwords.....	63
Subsystem Installation.....	64
Choosing which Subsystems To Install.....	64
X-Windows.....	65
Skipping The X-Windows Setup.....	65
Initiating The Installation.....	66
Launching the Package Installation.....	66
Watching The Package Installation Process.....	67
Watching The System Prepare The Installation.....	68
Creating the Red Hat Boot Disk.....	69
Concluding The Install.....	70
The First Boot.....	71
The First Post-Installation Boot.....	71
Watching The Kernel Boot For The First Time.....	72
The Red Hat Login Screen.....	73
Verifying the RAID Partitions.....	74
In the above example the mount command was used to display the active partitions.....	74
Viewing RAID Partition Status in /proc.....	75
Acknowledgements.....	Error! Bookmark not defined.

Why Do This At All?

Some people would say that enabling multiple RAID-1 devices under Linux is a non-trivial task, some would even describe it as difficult, poorly documented and not worth the trouble. So why bother?

Here are some compelling facts that support the idea that RAID-1 is a good idea.

RAID-1 Is A Good Thing

- 1) RAID raises the dependability of your system.
- 2) RAID can raise the performance of your website.

When a Redundant Array of Inexpensive Disks (RAID) is deployed on a system, the information on that system is **duplicated** on two physically different disk drives. In the event of a failure of one disk, the other is used – increasing the availability of information by 100%.

Because RAID devices can be deployed on multiple I/O channels, they can offer performance gains over disk systems that only use a single channel.

In other words, providing RAID may enhance your reputation and certainly helps to make your customers feel more secure when using your platform.

Setting up Multiple RAID-1 Devices in Red Hat 7.2

The Partitioning Strategy

The first thing that has to be determined when you are going to go with RAID is how the RAID is to be implemented. In the example used in this document, RAID is being used to mirror partitions as opposed to entire drives.

The boot and swap Partitions

Every Linux system needs somewhere to keep its boot files. On older drive (less than 540Mb) this used to be a non-important decision. Boot files were placed subdirectory of root called /boot and never thought about again, except in the case of a kernel rebuild.

With the advent of truly large drives the entire question of booting has become a bit trickier. BIOS restrictions cause a certain rule: The boot files must reside within the first 1024 cylinders of the drive. To accommodate the rule and guarantee a boot we must define a separate partition to hold the boot file. To make this easy, we define the **first** partition on the drive to be /boot.

The next partition is also **really** important. Linux systems use **virtual memory**, a system where sections of the RAM of the computer are swapped on and off the disk drive according to which program is currently active. This is happening even during installation, so the next task is to create the **swap partition** – in our case we anticipate physical RAM reaching 256Mb.

The rule of thumb is to define a swap space that is **twice** that of physical RAM, so our swap partition should be 512Mb.

Here is the partitions we will need for our “housekeeping” partitions:

Partition	Device	Size	Type
/boot	dev/hda1	20Mb	Linux
<swap>	dev/hda2	512Mb	swap

The third partition is where all of the actual files that compose the Linux system will reside – aside from the kernel itself. But there’s a problem, we want to define the following partitions:

```
/
/home
/usr
/var
```

But the partition table can only hold a maximum of four (4) entries. Summing up the two “housekeeping” and the “working” partitions gives us a total of six (6) partitions. We don’t have enough entries to spare in the partition table to accommodate this scheme!

Fortunately there is a way to get around the four primary partition limit. The **extended** partition was invented for just this reason. Extended partitions contain **logical** partitions inside of them. In this case the **extended** partition is really nothing more than a container for other partitions.

Here is the definition of the extended partition we will need to hold the “working” partitions

Partition	Device	Size	Type
<extended>	dev/hda3	<rest of disk>	extended

The Mirrored Partitions

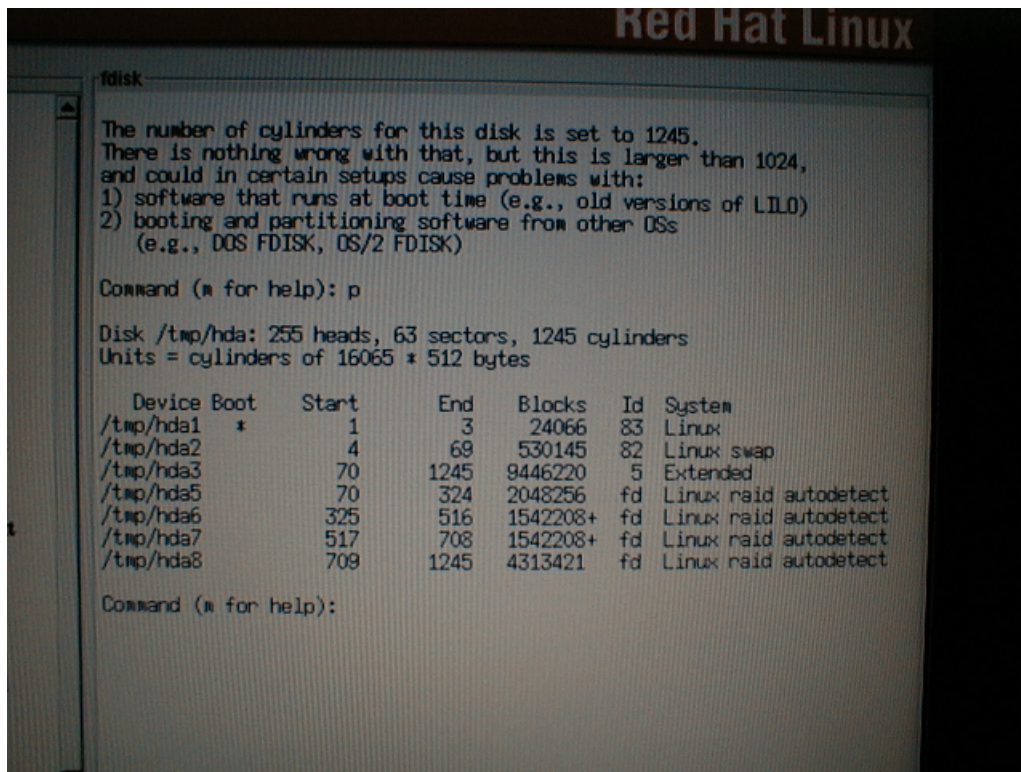
Now that the extended partition has been created, **logical** drives within the partition are defined.

In this scheme we are going to partition the RAID-1 partitions according to the following scheme:

Partition	Device	Size	Type
/	/dev/md5	4000Mb	Linux RAID
/home	/dev/md6	16000Mb	Linux RAID
/usr	/dev/md7	3000Mb	Linux RAID
/var	/dev/md8	5000Mb	Linux RAID

What We're Trying To Achieve

Here's a picture of what we should arrive at when this setup process is complete:



OK, now that we know what we're targeting, let's make it real!

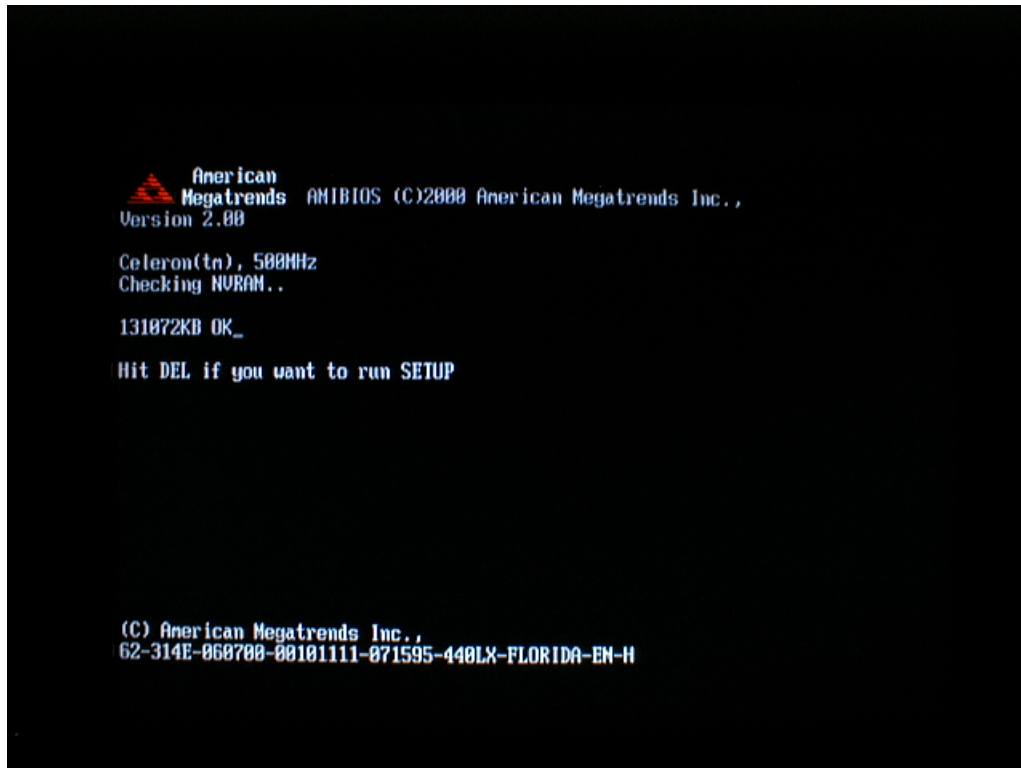
The Distribution



The BIOS Setup

One of the most important steps in making RAID-1 under Linux a success is the proper setup of the hard disks in the BIOS. If they are not set up properly, this entire exercise fails!

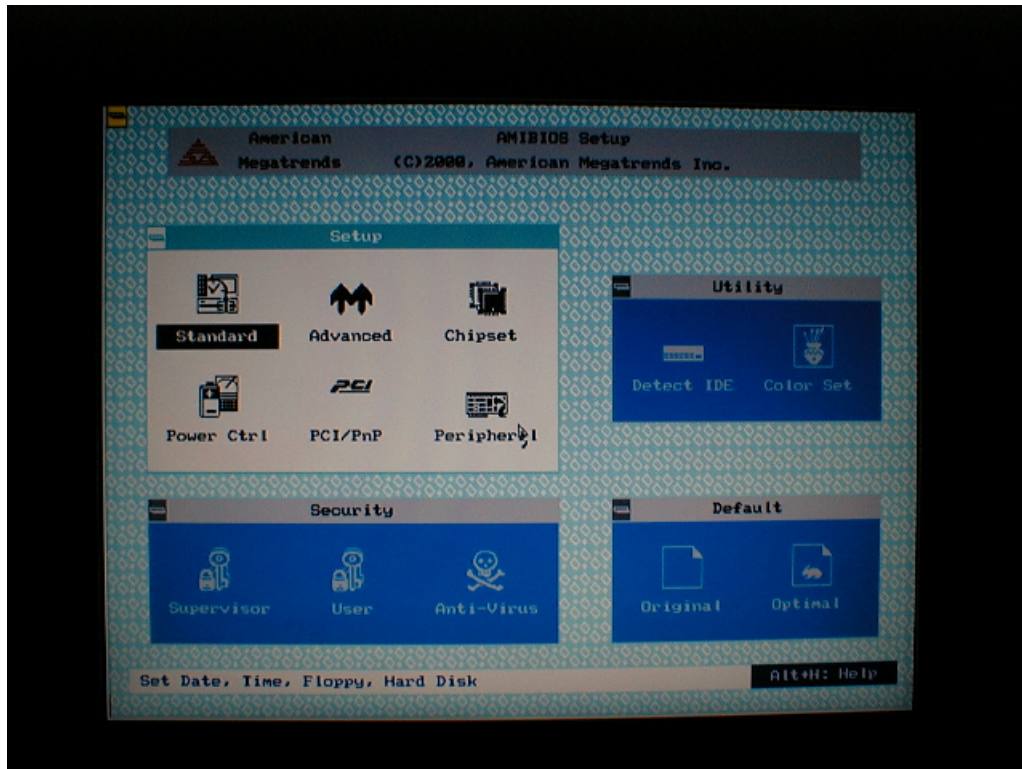
Consider the following:



In the above example the computer is booting and a prompt explains that to enter the BIOS setup the **DEL** key must be pressed.

The BIOS Setup Screen

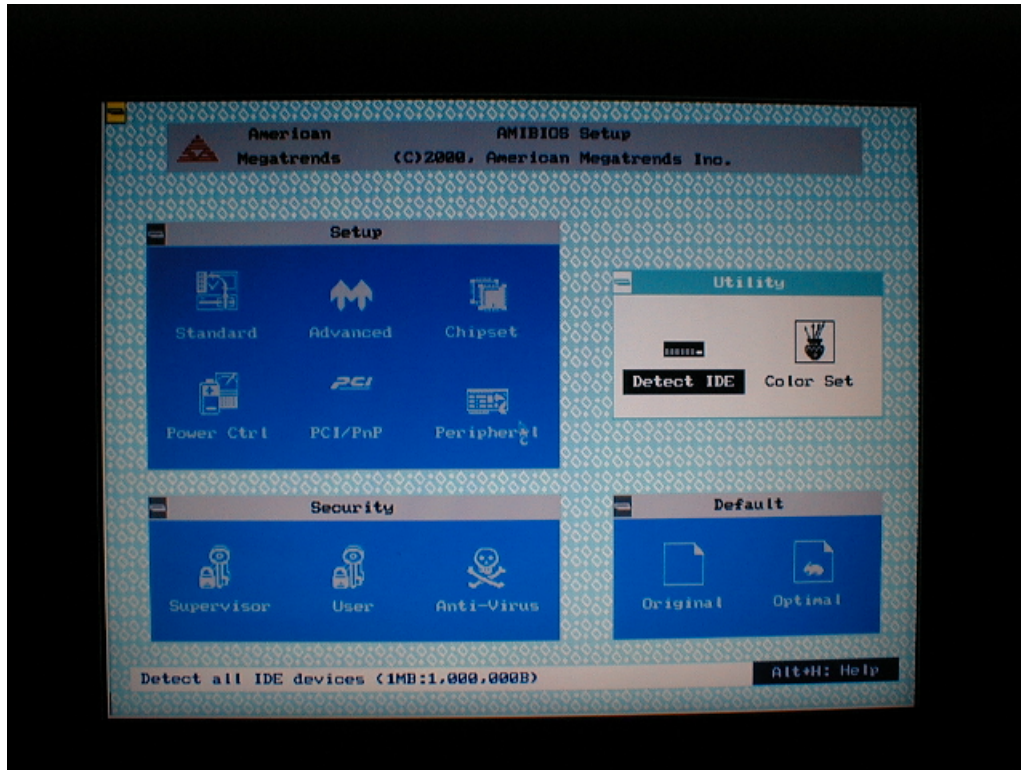
Consider the following:



In the above example, the BIOS has been entered and the initial screen is being displayed. We want to move to the screen to the **right** of this one, so we press <TAB>.

Getting Into the IDE Auto Detect Screen

Consider the following:

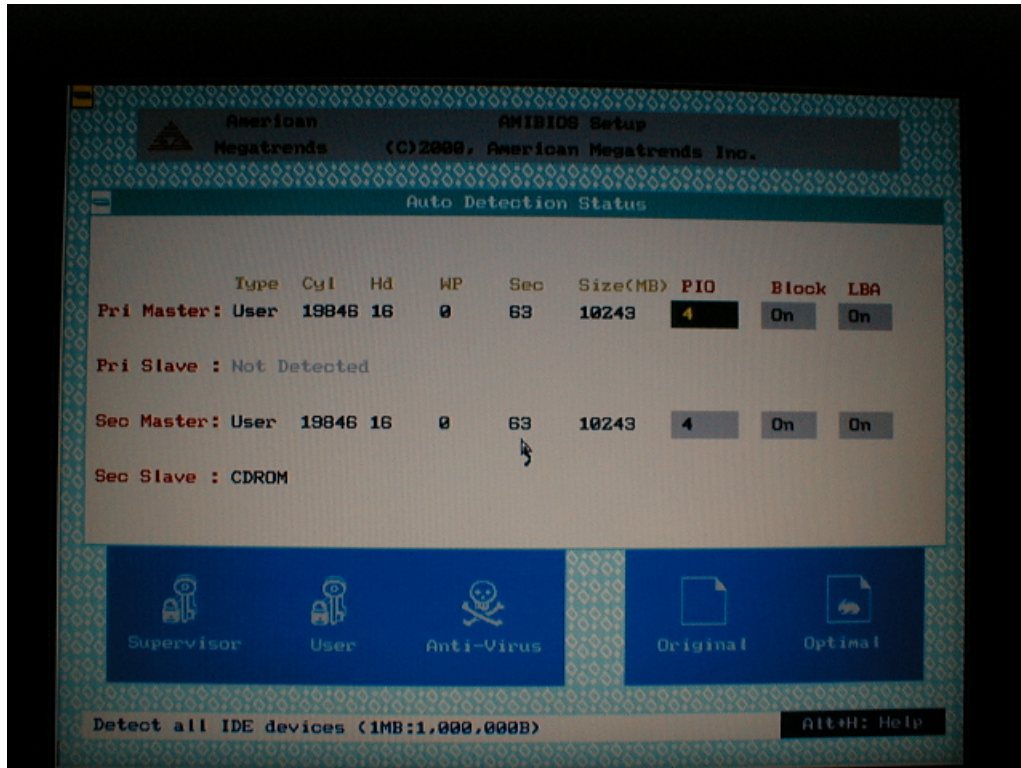


In the above example we are on the upper-right menu and about to invoke the **Detect IDE** utility. This utility will probe the hard drives and obtain their geometry information.

Run the utility by pressing the <ENTER> key.

Checking the results of the IDE Auto Detect Screen

Consider the following:



In the above example the **Detect IDE** utility has been run and the drive geometry for the two drives has been determined and is displayed. Note that each of the drives is **PRIMARY** on their respective IDE bus. I recommend that you make the drives PRIMARY on their own bus and any additional devices (such as CD-ROM) secondary.

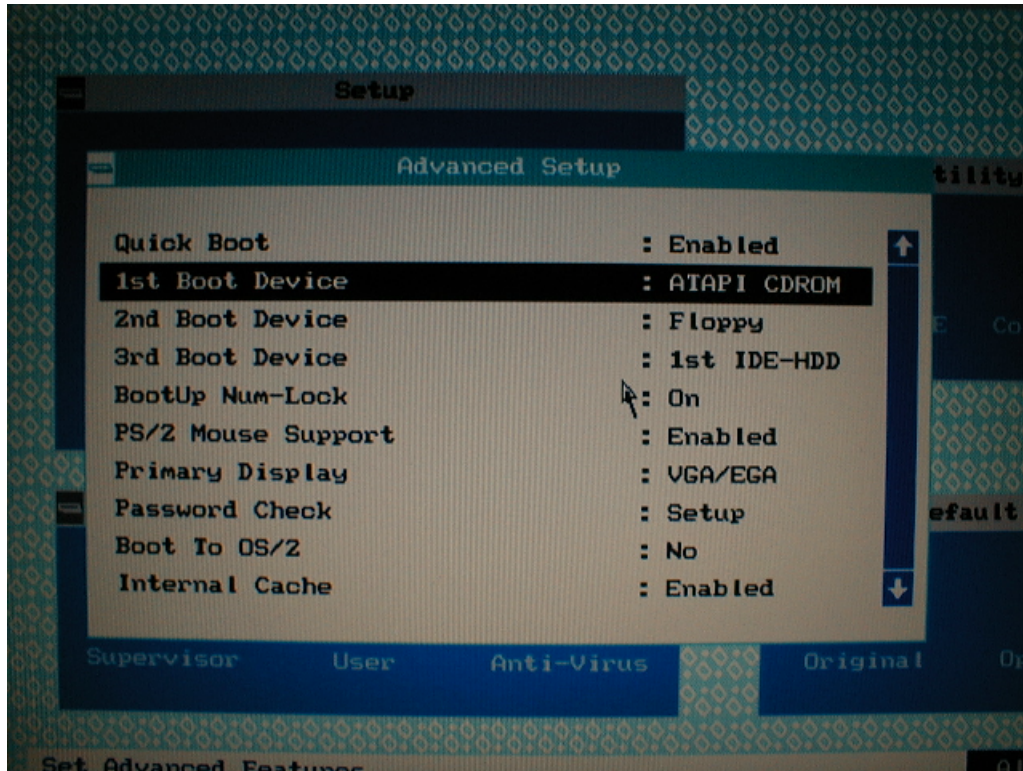
The other thing to verify is the two drives are treated **identically** by the BIOS. Make sure that they have the same translation (LBA or NONE) as well as displaying the same cylinders and heads and so forth. In my first two RAID-1 implementations the translation was AUTO – which caused /dev/had to use one kind of translation and /dev/hdb another – an incredibly frustrating and time-consuming problem that **cannot** be resolved within Linux, no matter what parameters you pass to the kernel on boot!

Do the BIOS setup properly or you will be sorry. You have been warned!

Enable boot from CDROM

In the **advanced** section of most BIOS is the ability to choose the first boot device.

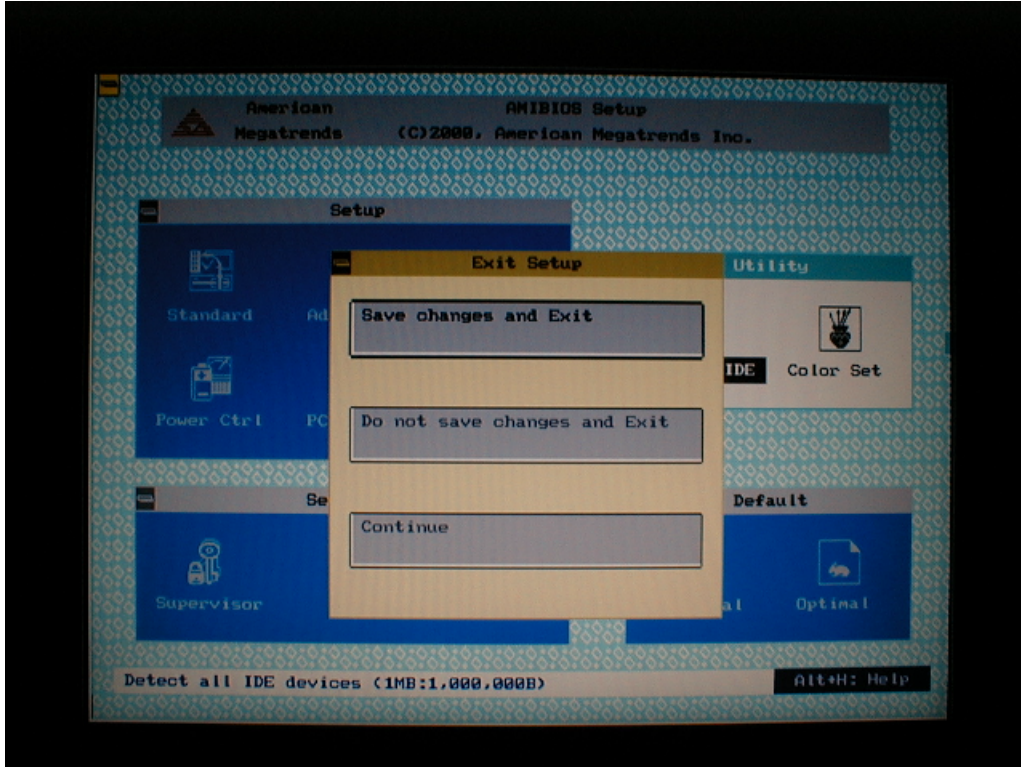
Consider the following:



In the above example the **CD-ROM** has been selected as the first boot device. This will enable the installation of Red Hat from CD.

Saving BIOS Changes

Consider the following:



After the BIOS settings have been verified (and written down) they are saved to NVRAM and the machine rebooted.

Booting From The CD-ROM

Consider the following:

```
Main Processor : Celeron(tm)
Math Processor : Built-In
Floppy Drive A : 1.44 MB 3½"
Floppy Drive B : None
AMIBIOS Date : 06/07/00
Processor Clock : 500MHz
Power Management : APM, SMI

Base Memory Size : 640KB
Ext. Memory Size : 130048KB
Display Type : VGA/EGA
Serial Port(s) : 03F8
Parallel Port(s) : 0378
External Cache : 128KB, Enabled

ATA(P) Device(s) Type Size LBA 32Bit Block PIO UDMA
Mode Mode Mode Mode Mode
Primary Master : Hard Disk 10244MB LBA On 32Sec 4 2
Secondary Master : Hard Disk 10244MB LBA On 32Sec 4 2
Secondary Slave : ATAPI CDROM 4 N/A

PCI Devices:
PCI Onboard PCI Bridge PCI Onboard Bridge Device
PCI Onboard USB Controller, IRQ10 PCI Onboard IDE
PCI Onboard Multimedia Device, IRQ10 PCI Slot 1 Ethernet, IRQ11
PCI Bridge VGA

Searching for Boot Record from CDROM..OK

SVSLINUX 1.52 2001-02-07 Copyright (C) 1994-2001 H. Peter Anvin
```

In the above example the machine is displaying its configuration – notice that the drives display the same **size, translation, mode** – in all respects their logical configuration is **identical**. If you do not have this symmetry your install may not be successful!

Note that the second to last message on the screen indicates that the CD-ROM is being booted, the last message is output from the actual boot loader.

The Red Hat 7.2 Installation Start Screen

Consider the following:

```

Welcome to Red Hat Linux 7.2!

- To install or upgrade Red Hat Linux in graphical mode,
  press the <ENTER> key.

- To install or upgrade Red Hat Linux in text mode, type: text <ENTER>.

- To enable low resolution mode, type: lowres <ENTER>.
  Press <F2> for more information about low resolution mode.

- To disable framebuffer mode, type: nofb <ENTER>.
  Press <F2> for more information about disabling framebuffer mode.

- To enable expert mode, type: expert <ENTER>.
  Press <F3> for more information about expert mode.

- To enable rescue mode, type: linux rescue <ENTER>.
  Press <F5> for more information about rescue mode.

- If you have a driver disk, type: linux dd <ENTER>.

- Use the function keys listed below for more information.

[F1-Main] [F2-General] [F3-Expert] [F4-Kernel] [F5-Rescue]
boot: _
```

In the above example the machine has booted from the CD-ROM and the installation is officially underway! The screen offers a variety of installation options, but we're going to select the default one by pressing **<ENTER>**.

Press **<ENTER>** to continue

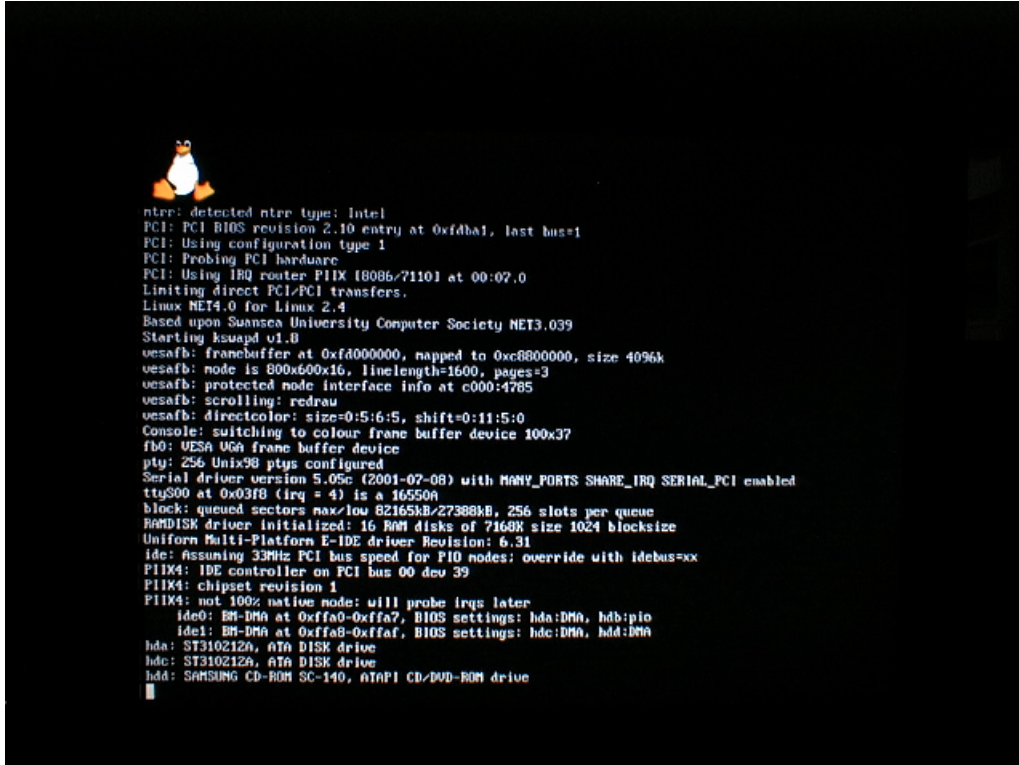
Loading the Linux Setup Kernel

- To install or upgrade Red Hat Linux in graphical mode, press the <ENTER> key.
- To install or upgrade Red Hat Linux in text mode, type: text <ENTER>.
- To enable low resolution mode, type: lowres <ENTER>. Press <F2> for more information about low resolution mode.
- To disable framebuffer mode, type: nofb <ENTER>. Press <F2> for more information about disabling framebuffer mode.
- To enable expert mode, type: expert <ENTER>. Press <F3> for more information about expert mode.
- To enable rescue mode, type: linux rescue <ENTER>. Press <F5> for more information about rescue mode.
- If you have a driver disk, type: linux dd <ENTER>.
- Use the function keys listed below for more information.

[F1-Main] [F2-General] [F3-Expert] [F4-Kernel] [F5-Rescue]
boot:
Loading initrd.img.....

Loading The Linux Kernel From CD-ROM

Consider the following:



```

ntr: detected ntr type: Intel
PCI: PCI BIOS revision 2.10 entry at 0xfd8a1, last bus=1
PCI: Using configuration type 1
PCI: Probing PCI hardware
PCI: Using IRQ router PIIX 1808b/71101 at 00:07.0
Limiting direct PCI/PCI transfers.
Linux NET4.0 for Linux 2.4
Based upon Swansea University Computer Society NET3.039
Starting ksuid 0.1.0
vesafb: framebuffer at 0xf4000000, mapped to 0xc8800000, size 4096k
vesafb: mode is 800x600x16, linelength=1600, pages=3
vesafb: protected mode interface info at c000:4785
vesafb: scrolling: redraw
vesafb: directcolor: size=0:5:6:5, shift=0:11:5:0
Console: switching to colour frame buffer device 100x37
fb0: VESA VGA frame buffer device
pty: 256 Unix98 ptys configured
Serial driver version 5.05c (2001-07-08) with MANY_PORTS SHARE_IRQ SERIAL_PCI enabled
ttyS00 at 0x03f8 (irq = 4) is a 16550A
block: queued sectors max/low 82165kB/27388kB, 256 slots per queue
RAMDISK driver initialized: 16 RAM disks of 7168K size 1024 blocksize
Uniform Multi-Platform E-IDE driver Revision: 6.31
ide: Assuming 33MHz PCI bus speed for PIO modes; override with idebus=xx
PIIX4: IDE controller on PCI bus 00 dev 39
PIIX4: chipset revision 1
PIIX4: not 100% native mode: will probe irqs later
   ide0: BM-DMA at 0xffa0-0xffa7, BIOS settings: hda:DMA, hdb:pio
   ide1: BM-DMA at 0xffa8-0xffaf, BIOS settings: hdc:DMA, hdd:DMA
hda: ST310212A, ATA DISK drive
hdc: ST310212A, ATA DISK drive
hdd: SAMSUNG CD-ROM SC-140, ATAPI CD/DVD-ROM drive

```

In the above example the Red Hat kernel is being loaded – a cute little penguin is displayed on the top of the screen and a bunch of information about system capabilities is output to the screen – in the old days you used to be able to read this, but now it zooms by so quickly that a special utility had to be written to view it! The utility is called **dmesg** and it can be invoked from the command line at any time.

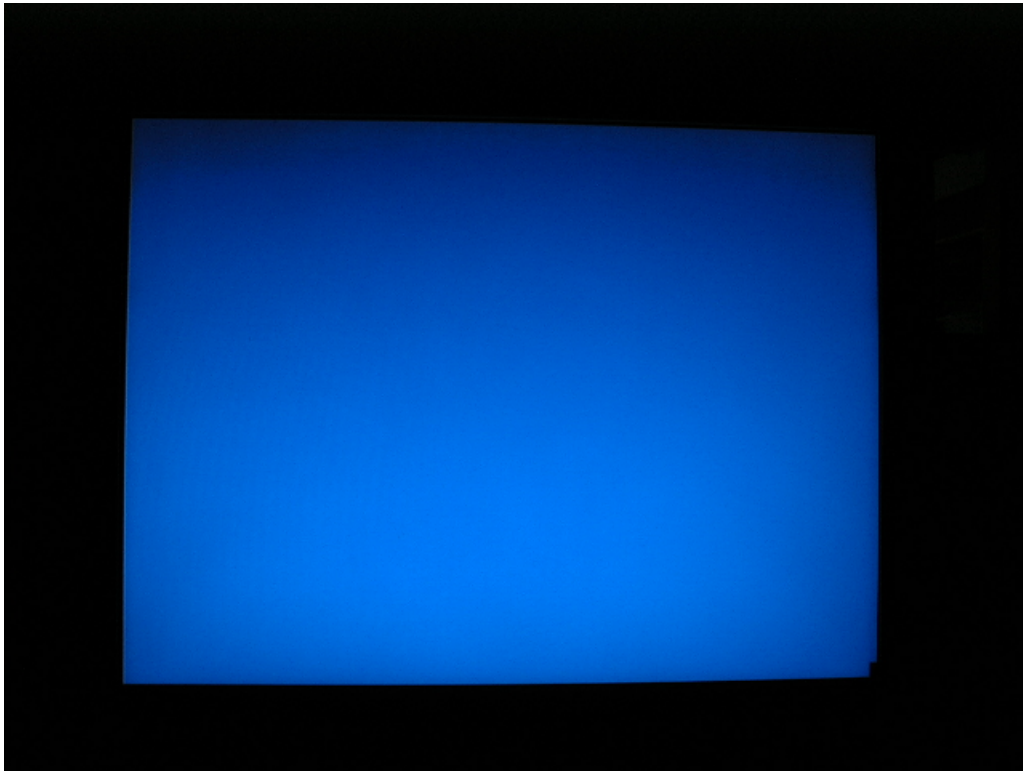
Just sit back and watch the lines fly by – do not be alarmed by anything you see, other than the phrase **kernel panic**. If you see that, there's trouble.

Starting The Anaconda Setup Program, part I

For Microsoft users, the next screen is a bit eerie. Traditionally, a blue screen did not bode well. Fortunately for Red Hat users, this just means that the initial device load is complete and the kernel is now resident in memory. The next thing the install program does after loading the kernel is to load the **anaconda** setup program.

The program is quite large and takes some time to load off of the (comparatively slow) CD-ROM. It is also probing the hardware to determine (as best it can) the monitor, mouse and keyboard attached to the computer

Consider the following:



In the above example the initial kernel load is complete and the **anaconda** program is being loaded from the CD-ROM into memory for execution.

Starting The Anaconda Setup Program, part II

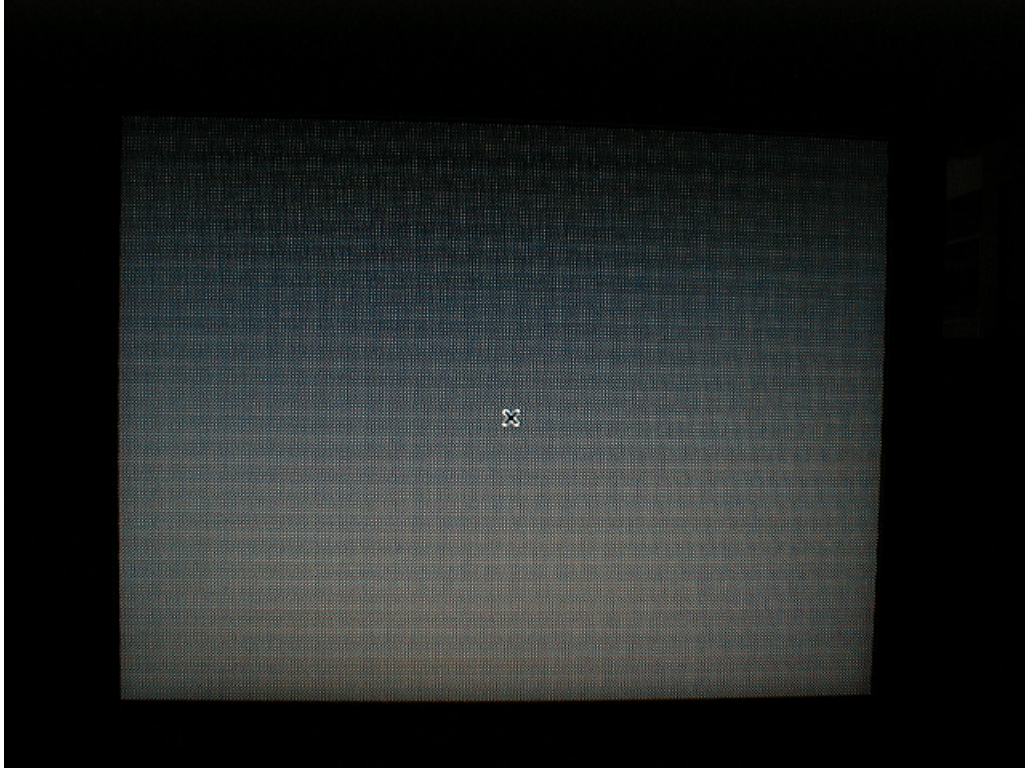
Consider the following:



In the above example,

Seeing X-Windows for the First Time

Consider the following:



In the above example,

The Red Hat Setup Splash Screen

For most Red Hat installers, this is a welcome but somewhat bewildering sight. Anything is better than a blue screen with a single line of text at the bottom, but this splash screen doesn't give **any** information about what's going on – at least with the **anaconda** program there was a promise of progress.

Anyway, after a few moments the splash screen goes away

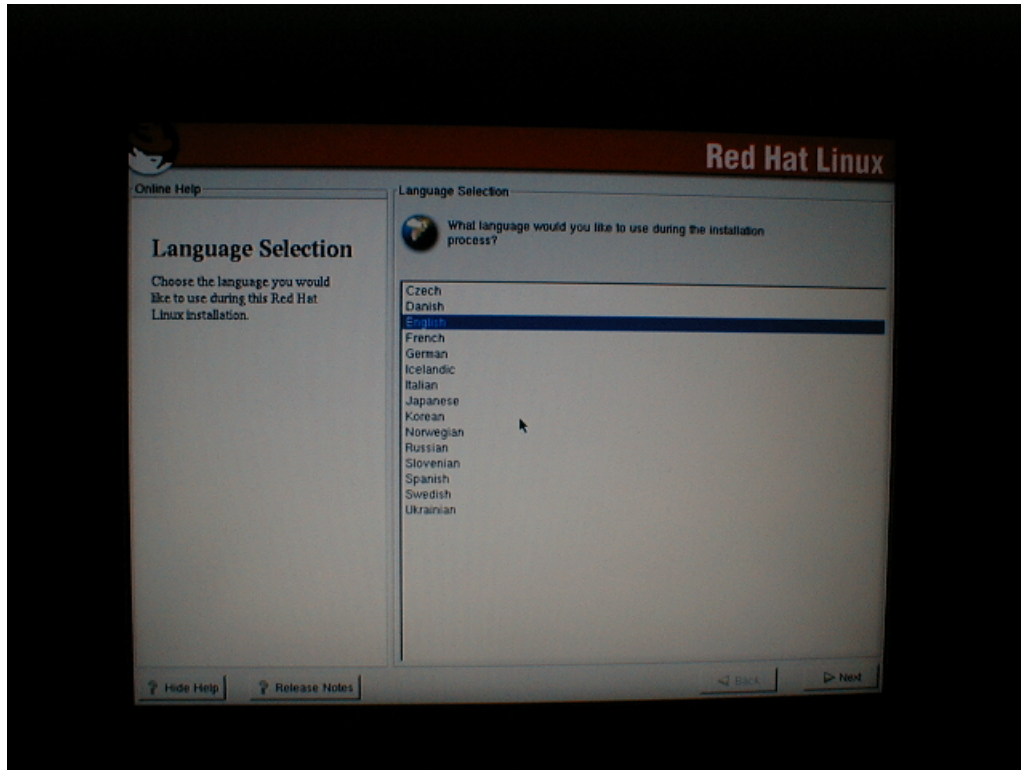
Consider the following:



In the above example the **anaconda** program has been loaded into memory and the first thing it does is display a splash screen while it performs some setup tasks.

Language Selection

Consider the following:

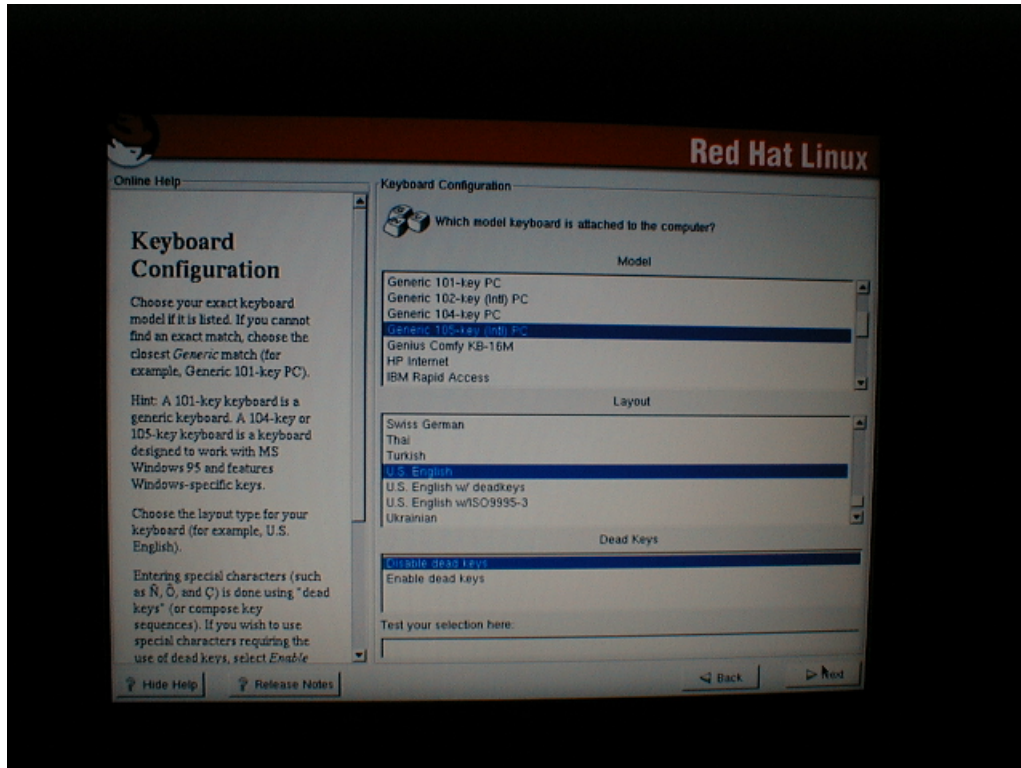


In the above example the **anaconda** program is prompting the user to supply the language they wish to have displayed during the installation.

Select **English** and then click on the **Next** button to continue. If the mouse is not working, press the **<TAB>** key until the **Next** button is highlighted and then press **<ENTER>**.

Keyboard Configuration

Consider the following:

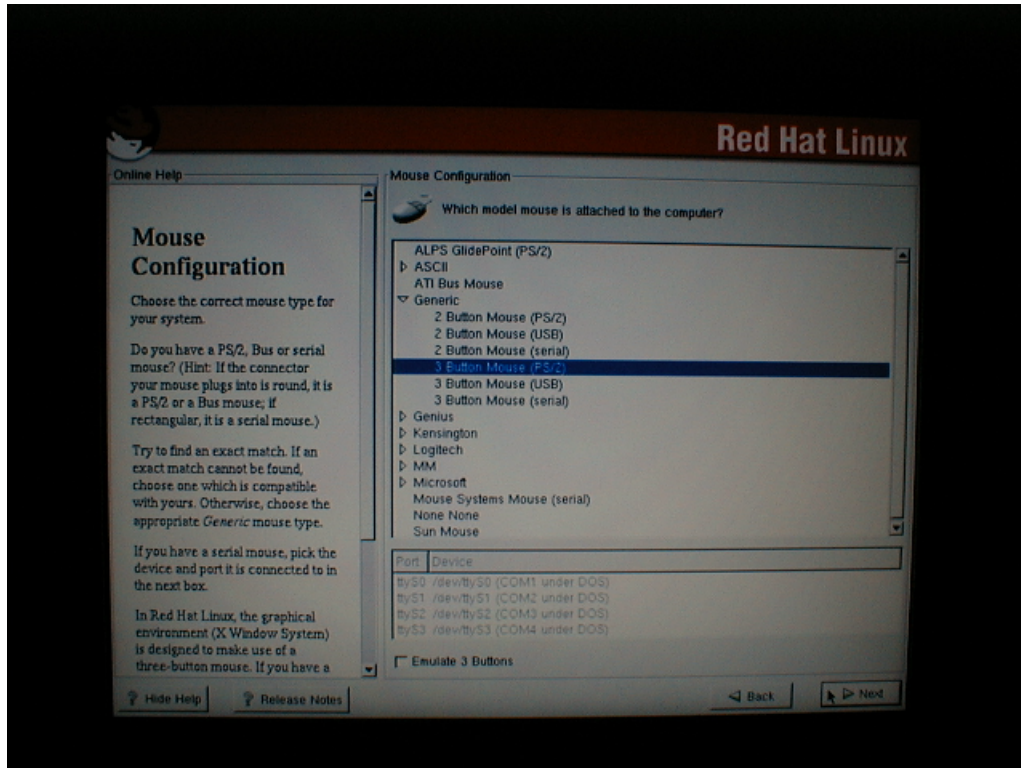


In the above example the installer is being prompted to supply information about their keyboard type and whether they want to enable extended functionality, called **dead keys**, to allow for the generation of special characters.

Pick the appropriate setting, disable **dead keys** and click on **Next** to continue. If the mouse is not working, press the **<TAB>** key until the **Next** button is highlighted and then press **<ENTER>**.

Mouse Configuration

Consider the following:



For some reason the mouse detection in **anaconda** always mis-identifies the mouse type. Most mice now are **two button, PS/2** mice – not three button. If you select a two button mouse, be sure to enable three button functionality with the check box at the bottom of the screen.

Anyway, for your situation pick the appropriate setting and click on **Next** to continue.

If the mouse is not working, press the **<TAB>** key until the **Next** button is highlighted and then press **<ENTER>**.

Commencement of System Setup

Consider the following:

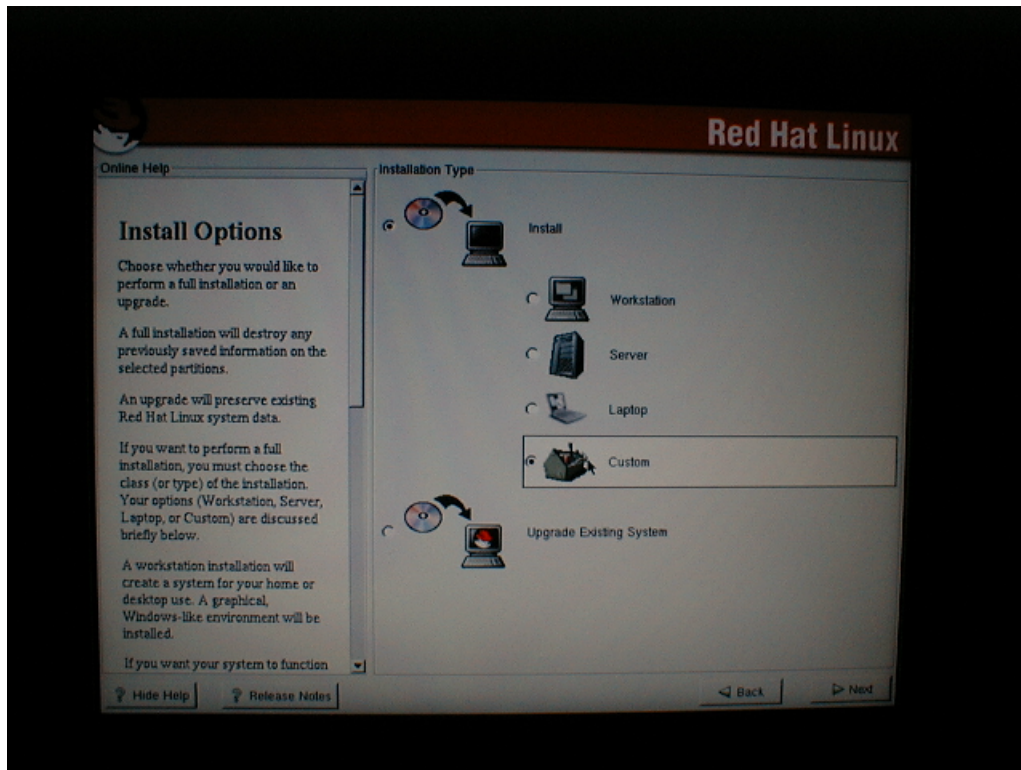


In the above example the Red Hat setup program is offering an information screen describing what is about to happen in the install, along with some other supporting information.

Click on **Next** to continue.

Select The System Profile

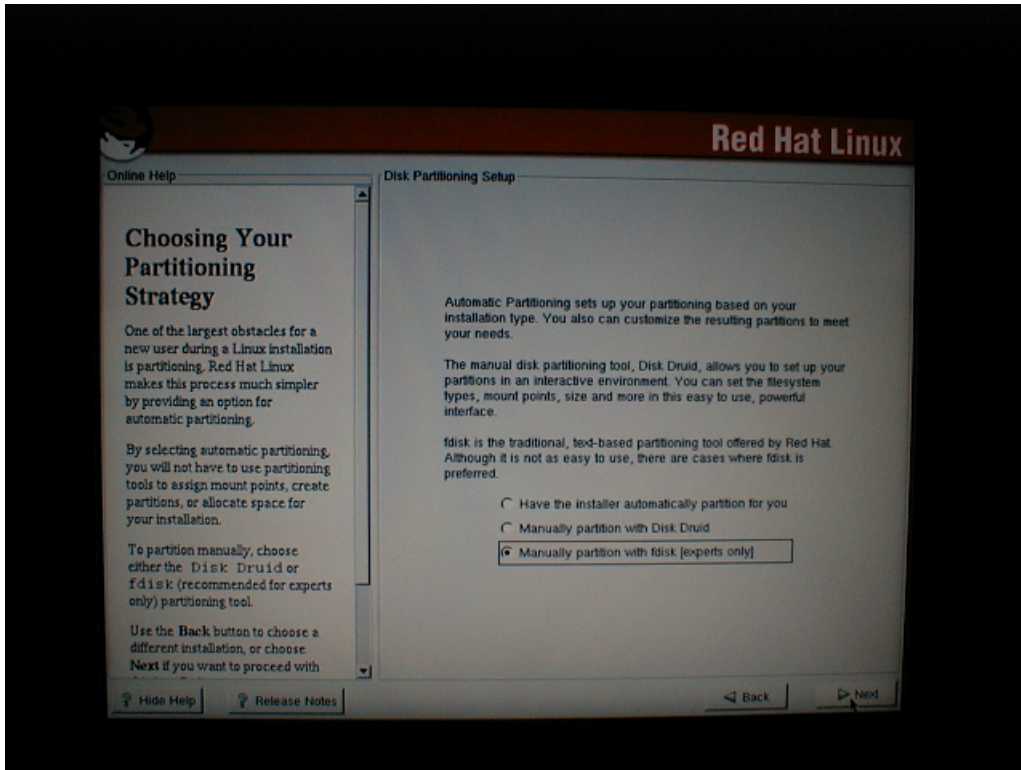
Consider the following:



Select **custom** and click on **Next** to continue.

Choosing Your Partitioning Strategy

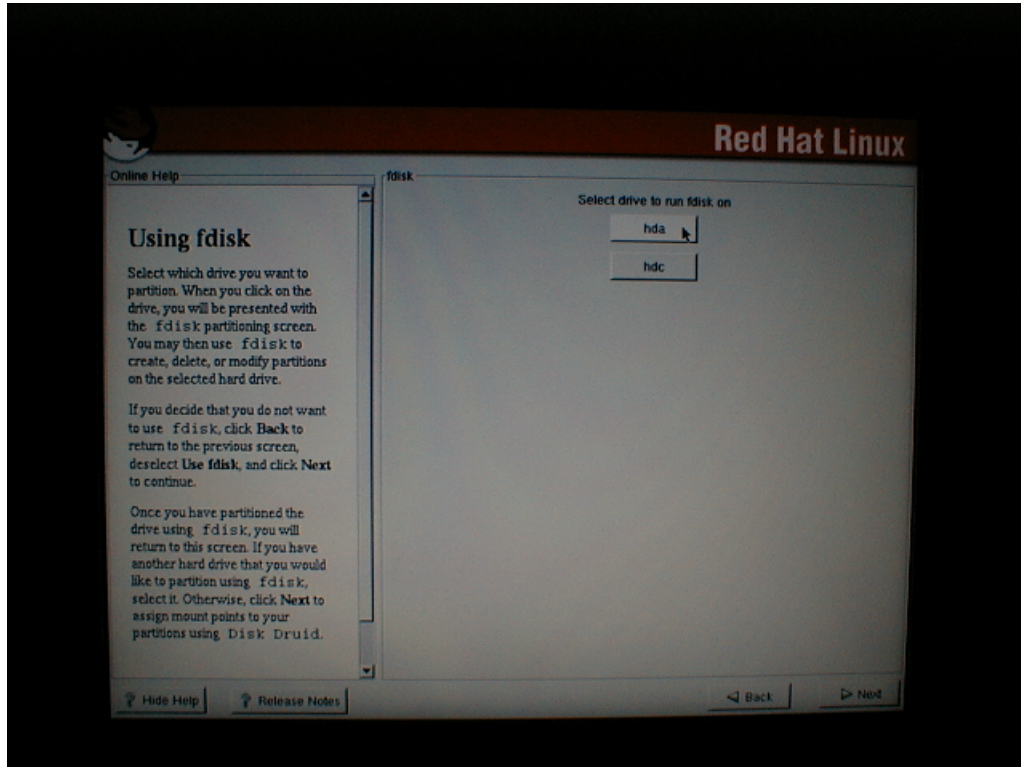
Consider the following:



Select **manually partition using fdisk** and click on **Next** to continue.

Using fdisk

Consider the following:



Select **hda** and click on **Next** to continue.

Partitioning hda

The fdisk Commands

fdisk is not the most friendly of programs, but it is quite powerful once the basic commands are mastered.

Consider the following:

NEED SHOT OF HELP SCREEN HERE

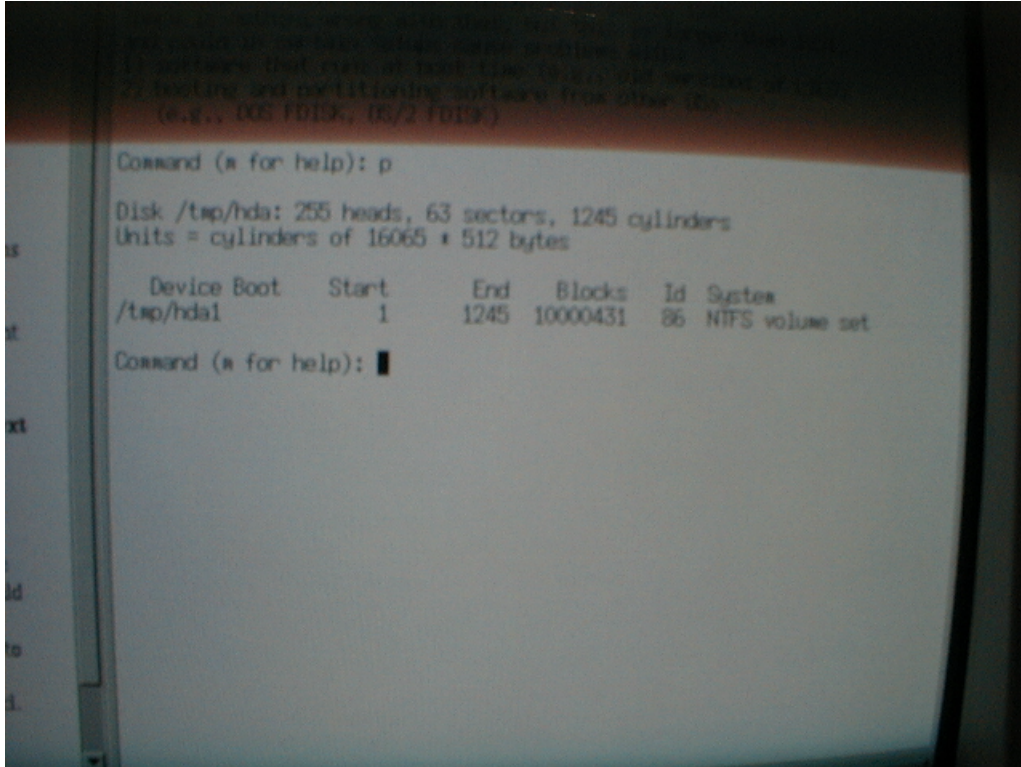
In the above example the **fdisk** program has been invoked and the program is running.

Basic fdisk Commands

Character	Function
?	Display help menu
n	New Partition
p	Print the partition table
a	Tag a partition bootable
t	Tag a partition type
w	Write the partition table and exit fdisk

How To View Partitions With fdisk

Consider the following:



```
Command (m for help): p
Disk /tmp/hda: 255 heads, 63 sectors, 1245 cylinders
Units = cylinders of 16065 * 512 bytes

   Device Boot      Start         End      Blocks   Id  System
  /tmp/hda1                1         1245     10000431   86  NTFS volume set

Command (m for help): █
```

In the above example the partition table has been printed with the **p** command. In this case, a partition exists on the disk of type **NTFS volume set**. This partition spans the entire disk, from cylinder 1 to cylinder 1245.

This partition must be deleted.

How To Delete A Partition With fdisk

Consider the following:

```
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
1) software that runs at boot time (e.g., old versions of LILO)
2) booting and partitioning software from other OSs
   (e.g., DOS FDISK, OS/2 FDISK)

Command (w for help): p

Disk /tmp/hda: 255 heads, 63 sectors, 1245 cylinders
Units = cylinders of 16065 * 512 bytes

   Device Boot   Start     End  Blocks  Id System
 /tmp/hda1             1     1245  10000431  86  NTFS volume set

Command (w for help): d
Partition number (1-4): 1

Command (w for help): p

Disk /tmp/hda: 255 heads, 63 sectors, 1245 cylinders
Units = cylinders of 16065 * 512 bytes

   Device Boot   Start     End  Blocks  Id System

Command (w for help):
```

In the above example the first partition has been deleted from the disk with the **d** command.

Creating A 20Mb Boot Partition

```
Command (m for help):  
Command (m for help):  
Command (m for help):  
Command (m for help):  
Command (m for help):  
Command (m for help):  
Command (m for help):  
Command (m for help):  
Command (m for help):  
Command (m for help):  
Command (m for help): n  
Command action  
  e   extended  
  p   primary partition (1-4)  
p  
Partition number (1-4): 1  
First cylinder (1-1245, default 1): 1  
Last cylinder or +size or +sizeM or +sizeK (1-1245, default 1245): +20M  
  
Command (m for help): p  
  
Disk /tmp/hda: 255 heads, 63 sectors, 1245 cylinders  
Units = cylinders of 16065 * 512 bytes  
  
   Device Boot      Start         End      Blocks   Id  System  
/tmp/hda1          1             3        24066   83  Linux  
  
Command (m for help): █
```

◀ Back ▶ Next

Making A Partition Bootable With fdisk

```
Command (m for help):  
Command (m for help):  
Command (m for help):  
Command (m for help):  
Command (m for help):  
Command (m for help):  
Command (m for help):  
Command (m for help):  
Command (m for help):  
Command (m for help):  
Command (m for help):  
Command (m for help):  
Command (m for help):  
Command (m for help):  
Command (m for help):  
Command (m for help):  
Command (m for help):  
Command (m for help):  
Command (m for help):  
Command (m for help):  
Command (m for help):  
Command (m for help):  
Command (m for help): a  
Partition number (1-4): 1  
  
Command (m for help): p  
  
Disk /tmp/hda: 255 heads, 63 sectors, 1245 cylinders  
Units = cylinders of 16065 * 512 bytes  
  
   Device Boot      Start         End      Blocks   Id  System  
/tmp/hda1  *            1             3        24066   83  Linux  
  
Command (m for help):
```

◀ Back ▶ Next

Tagging A Partition For Linux

```
Command (M for help):  
Command (M for help):  
Command (M for help):  
Command (M for help):  
Command (M for help):  
Command (M for help):  
Command (M for help):  
Command (M for help):  
Command (M for help):  
Command (M for help):  
Command (M for help):  
Command (M for help):  
Command (M for help):  
Command (M for help):  
Command (M for help):  
Command (M for help):  
Command (M for help):  
Command (M for help):  
Command (M for help): t  
Partition number (1-4): 1  
Hex code (type L to list codes): 83  
Changed system type of partition 1 to 83 (Linux)  
  
Command (M for help): p  
  
Disk /tmp/hda: 255 heads, 63 sectors, 1245 cylinders  
Units = cylinders of 16065 * 512 bytes  
  
   Device Boot      Start         End      Blocks   Id  System  
/tmp/hda1  *           1           3        24066   83  Linux  
  
Command (M for help):
```

◀ Back ▶ Next

Creating A Swap Partition

```
Command (m for help):  
Command (m for help):  
Command (m for help):  
Command (m for help):  
Command (m for help): n  
Command action  
  e  extended  
  p  primary partition (1-4)  
p  
Partition number (1-4): 2  
First cylinder (4-1245, default 4):  
Using default value 4  
Last cylinder or +size or +sizeM or +sizeK (4-1245, default 1245): +512M  
  
Command (m for help): t  
Partition number (1-4): 2  
Hex code (type L to list codes): 82  
Changed system type of partition 2 to 82 (Linux swap)  
  
Command (m for help): p  
  
Disk /tmp/hda: 255 heads, 63 sectors, 1245 cylinders  
Units = cylinders of 16065 * 512 bytes  
  
   Device Boot      Start         End      Blocks   Id  System  
/tmp/hda1 *          1           3        24066   83  Linux  
/tmp/hda2             4          68       530145   82  Linux swap  
  
Command (m for help): █
```

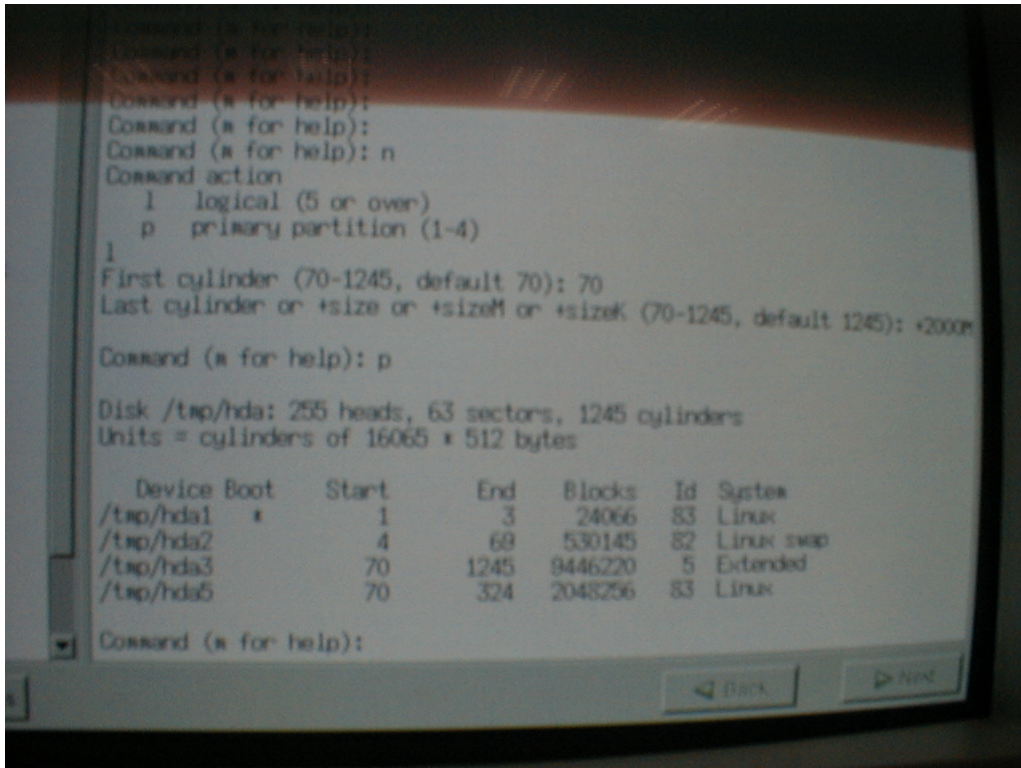
◀ Back ▶ Next

How To Create An Extended Partition For Linux

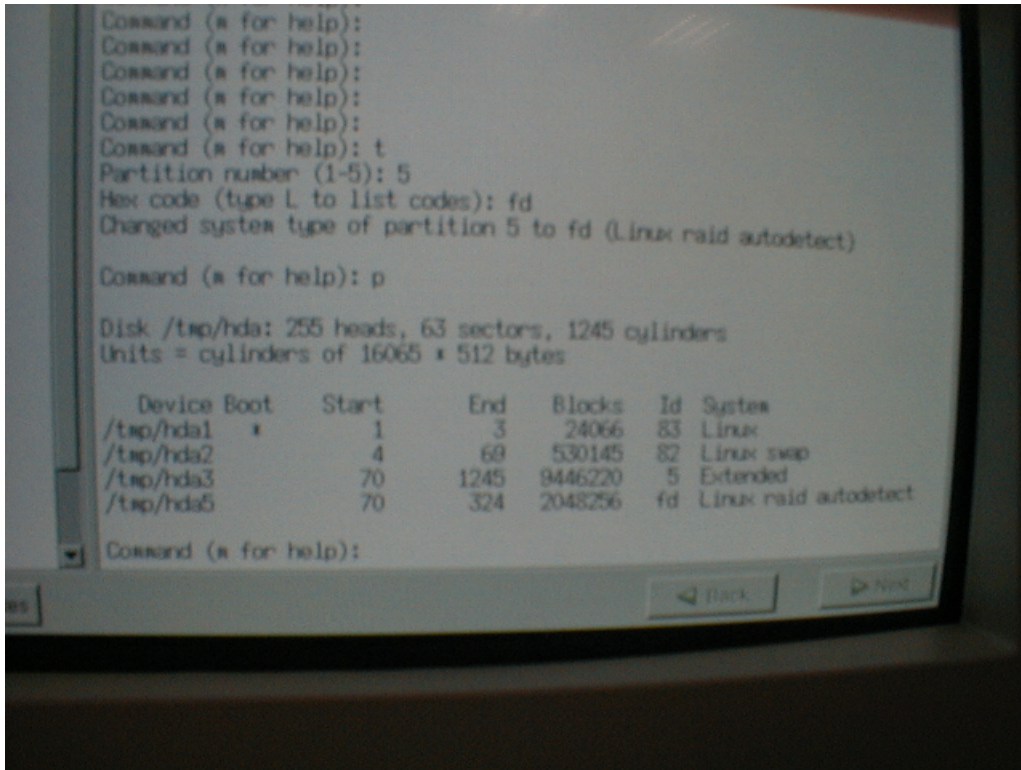
```
Command (m for help):  
Command (m for help):  
Command (m for help):  
Command (m for help):  
Command (m for help):  
Command (m for help):  
Command (m for help):  
Command (m for help): n  
Command action  
  e  extended  
  p  primary partition (1-4)  
e  
Partition number (1-4): 3  
First cylinder (70-1245, default 70):  
Using default value 70  
Last cylinder or +size or +sizeM or +sizeK (70-1245, default 1245):  
Using default value 1245  
  
Command (m for help): p  
  
Disk /tmp/hda: 255 heads, 63 sectors, 1245 cylinders  
Units = cylinders of 16065 * 512 bytes  
  
   Device Boot      Start         End      Blocks   Id  System  
/tmp/hda1  *           1           3        24066   83  Linux  
/tmp/hda2             4          69       530145   82  Linux swap  
/tmp/hda3            70        1245     9446220    5  Extended  
  
Command (m for help):
```

◀ Back ▶ Next

How To Create A Logical Drive In An Extended Partition

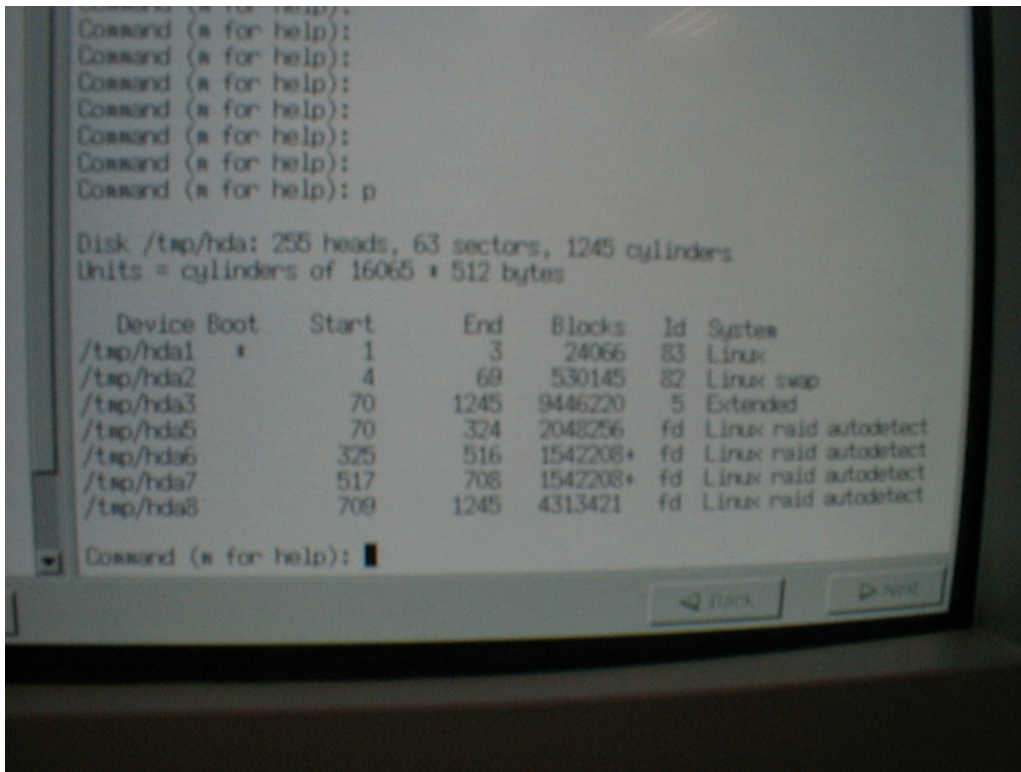


How To Tag A Partition For RAID

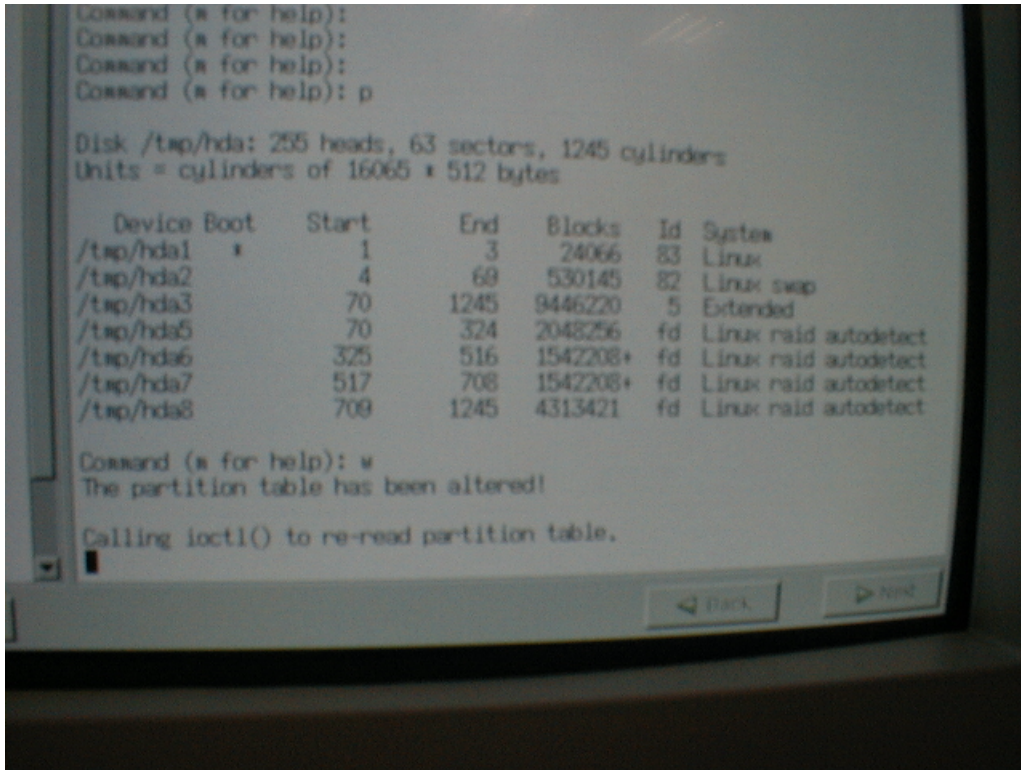


```
Command (* for help):  
Command (* for help):  
Command (* for help):  
Command (* for help):  
Command (* for help):  
Command (* for help): t  
Partition number (1-5): 5  
Hex code (type L to list codes): fd  
Changed system type of partition 5 to fd (Linux raid autodetect)  
  
Command (* for help): p  
  
Disk /tmp/hda: 255 heads, 63 sectors, 1245 cylinders  
Units = cylinders of 16065 * 512 bytes  
  
   Device Boot   Start    End  Blocks  Id System  
/tmp/hda1 *       1       3    24066  83 Linux  
/tmp/hda2         4       69   530145  82 Linux swap  
/tmp/hda3        70     1245  9446220   5 Extended  
/tmp/hda5        70     324   2048256  fd Linux raid autodetect  
  
Command (* for help):
```

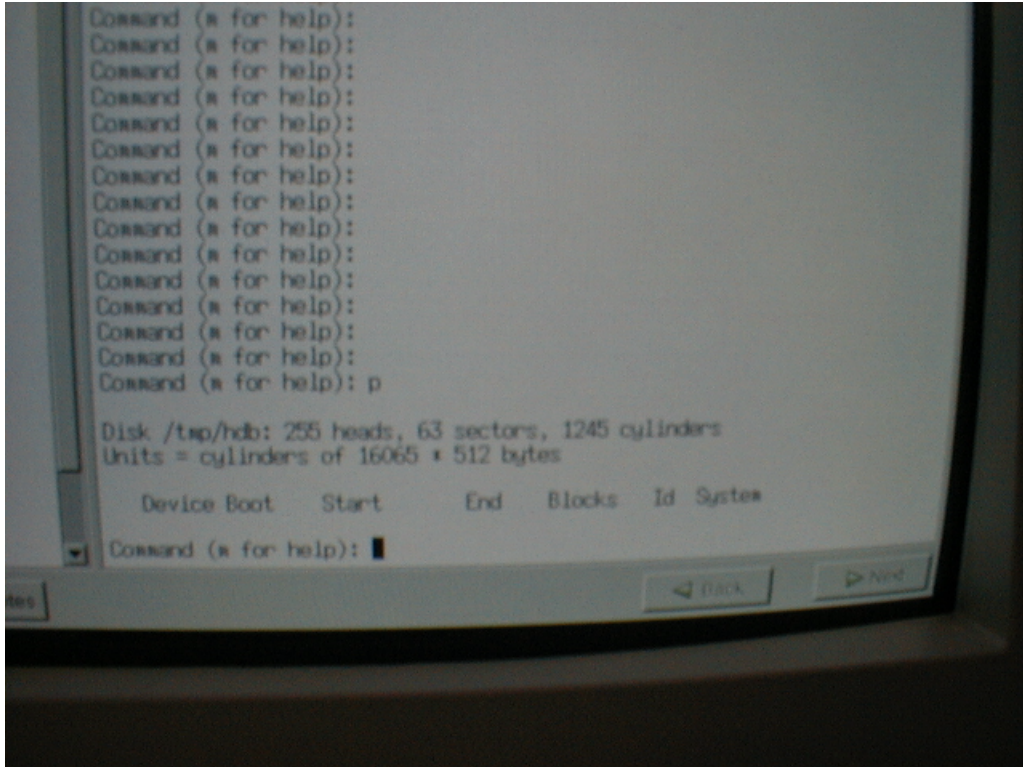

All Of The RAID Partitions Prepared



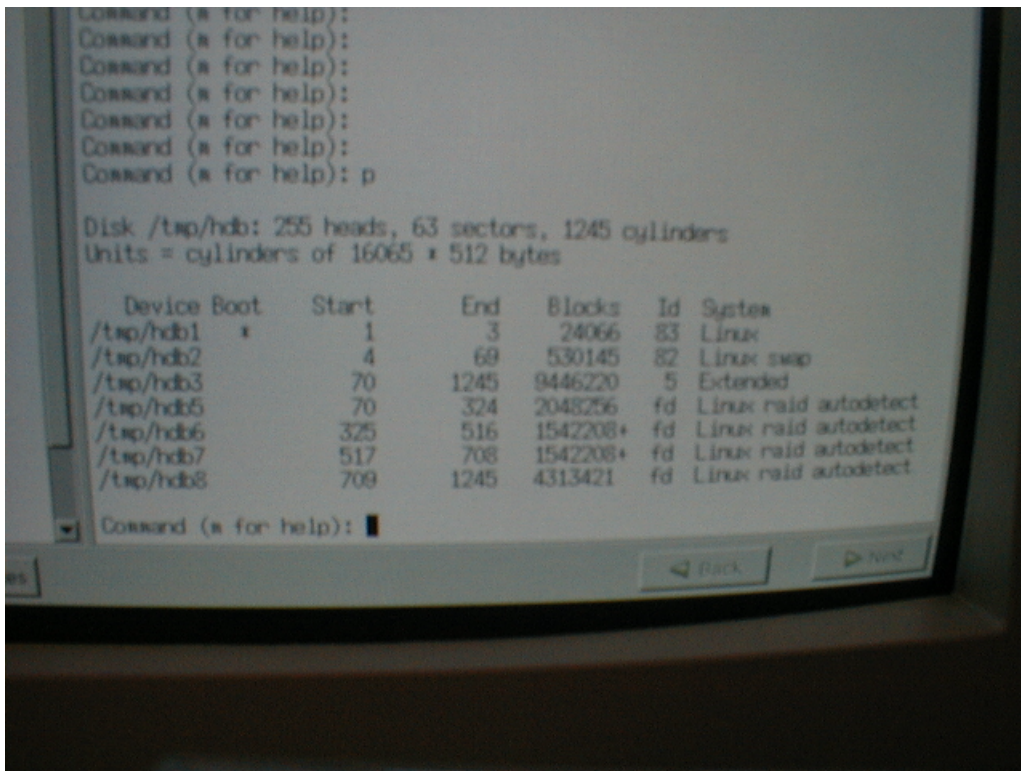
How To Write The Partition Table



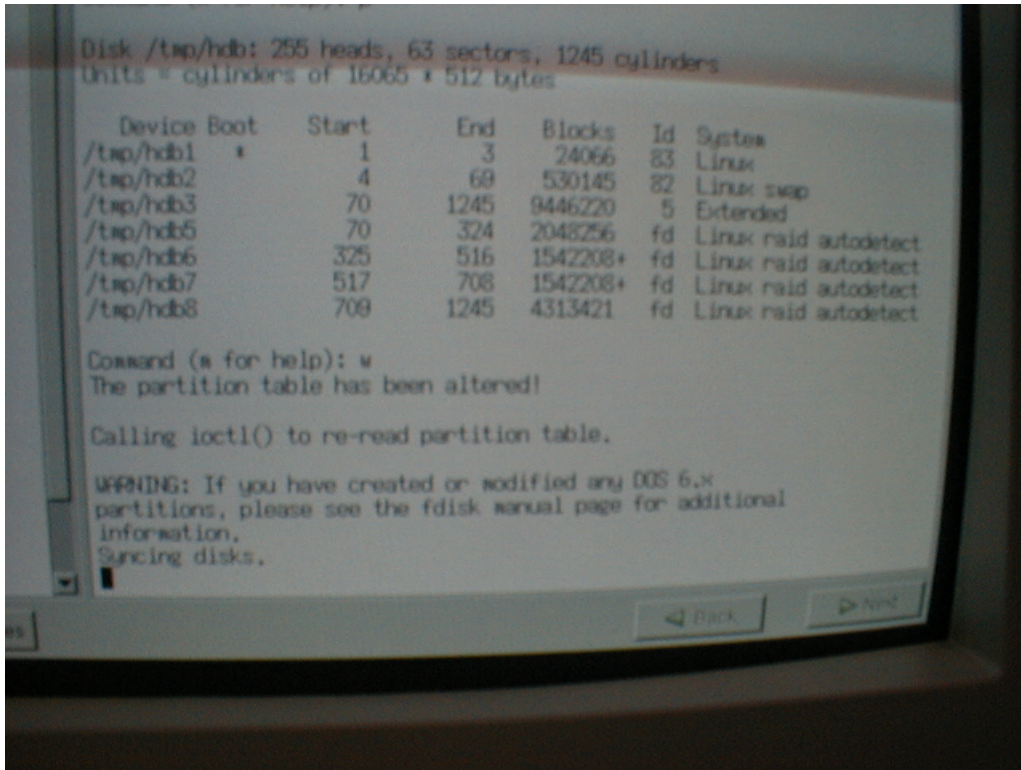
Partitioning hdb



Make hdb An Exact Duplicate Of hda

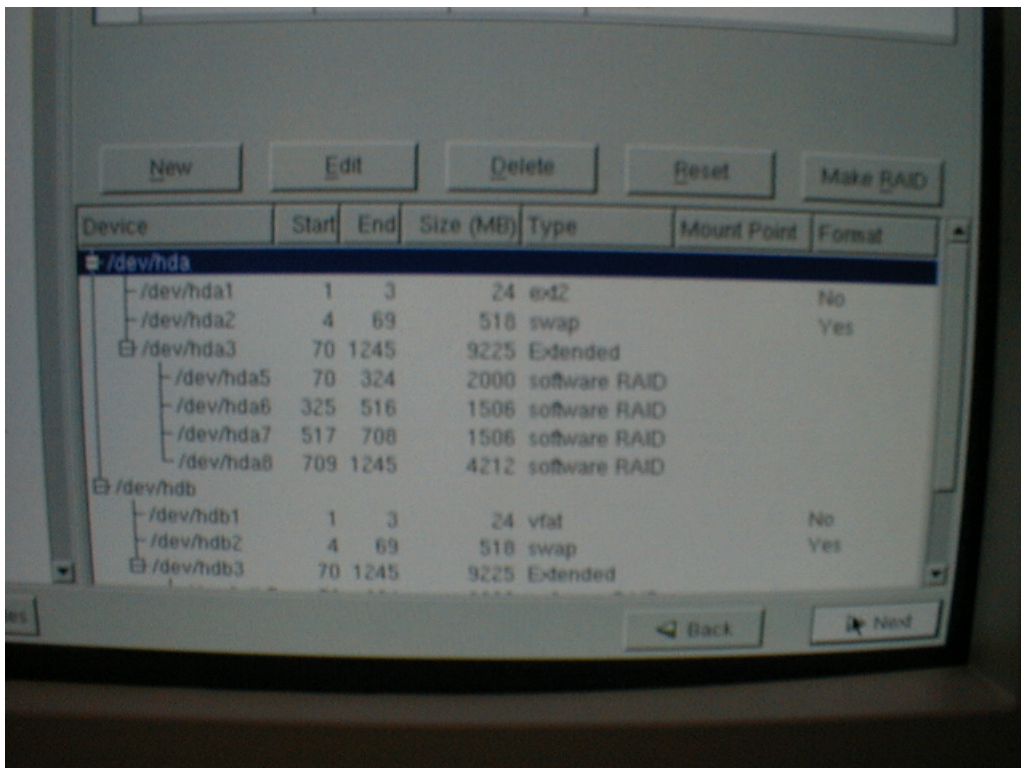
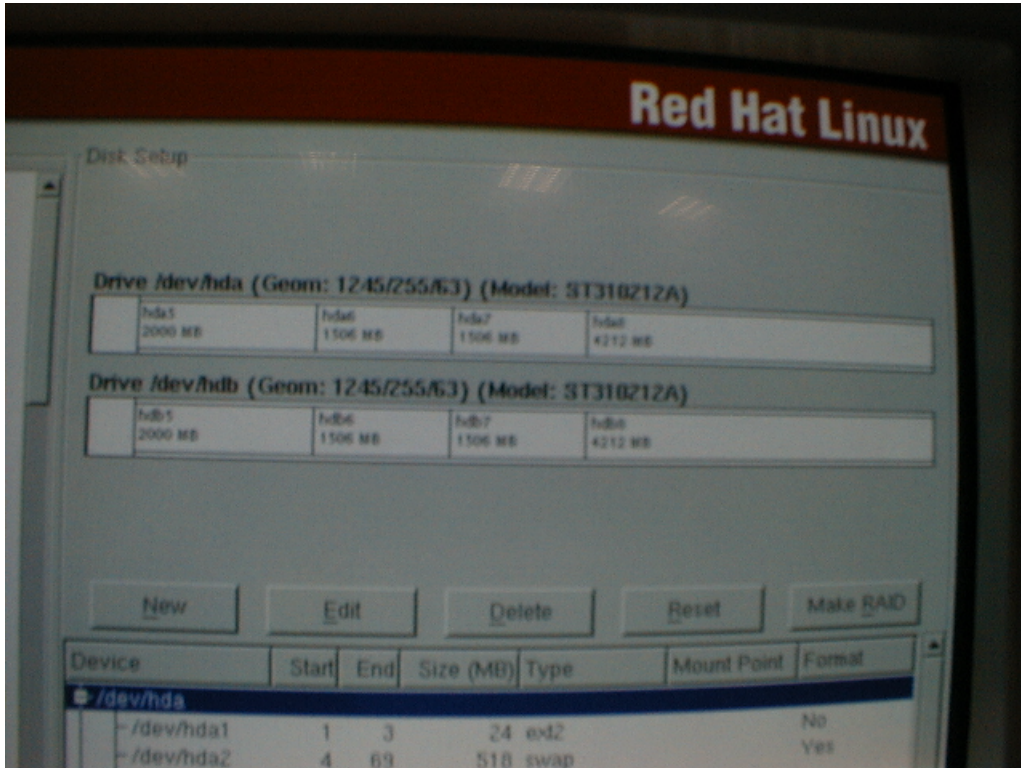


Writing The Partition Table A Second Time

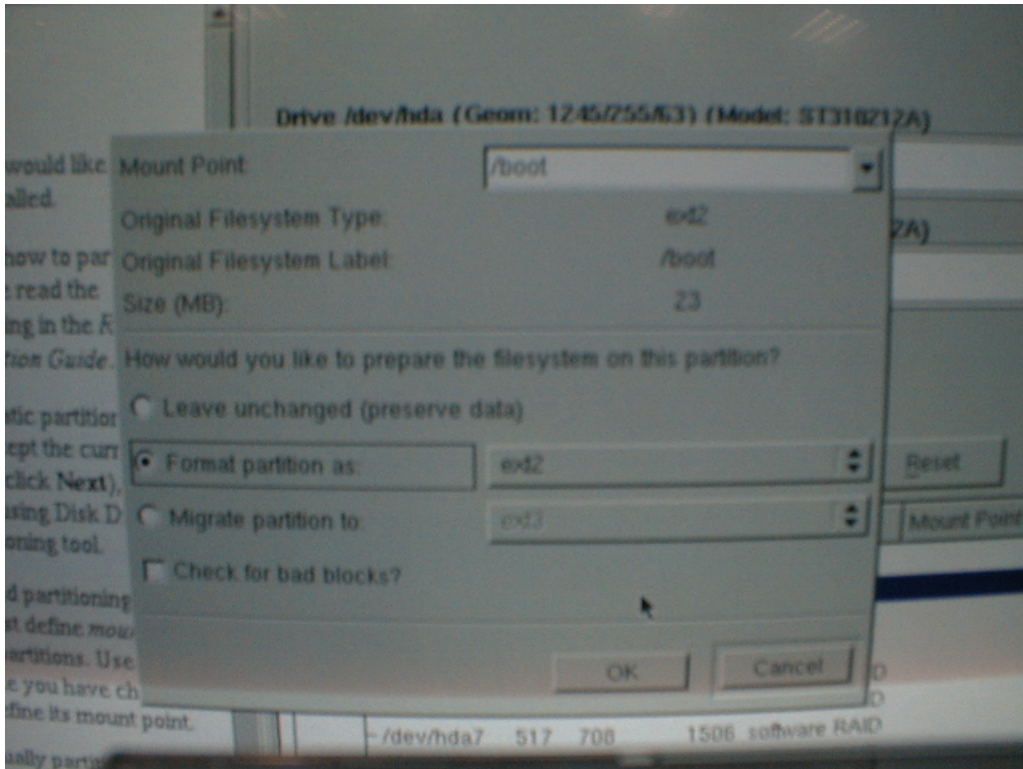


Disk Druid

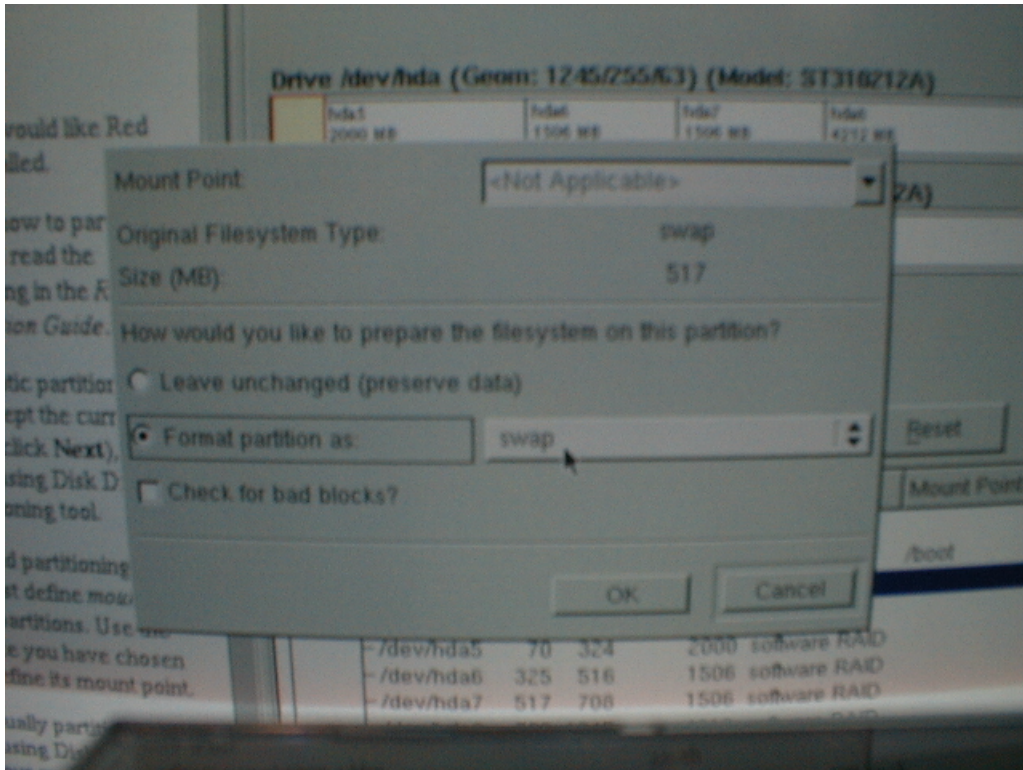
Using Disk Druid To Setup Drives



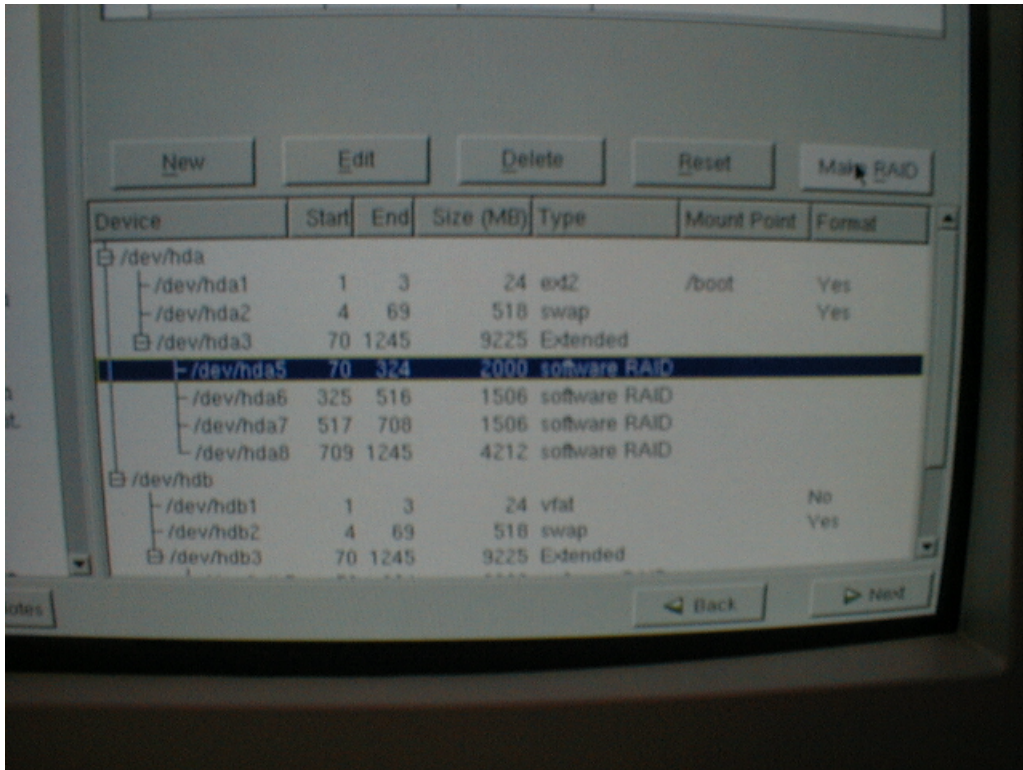
How To Configure the /boot Partition



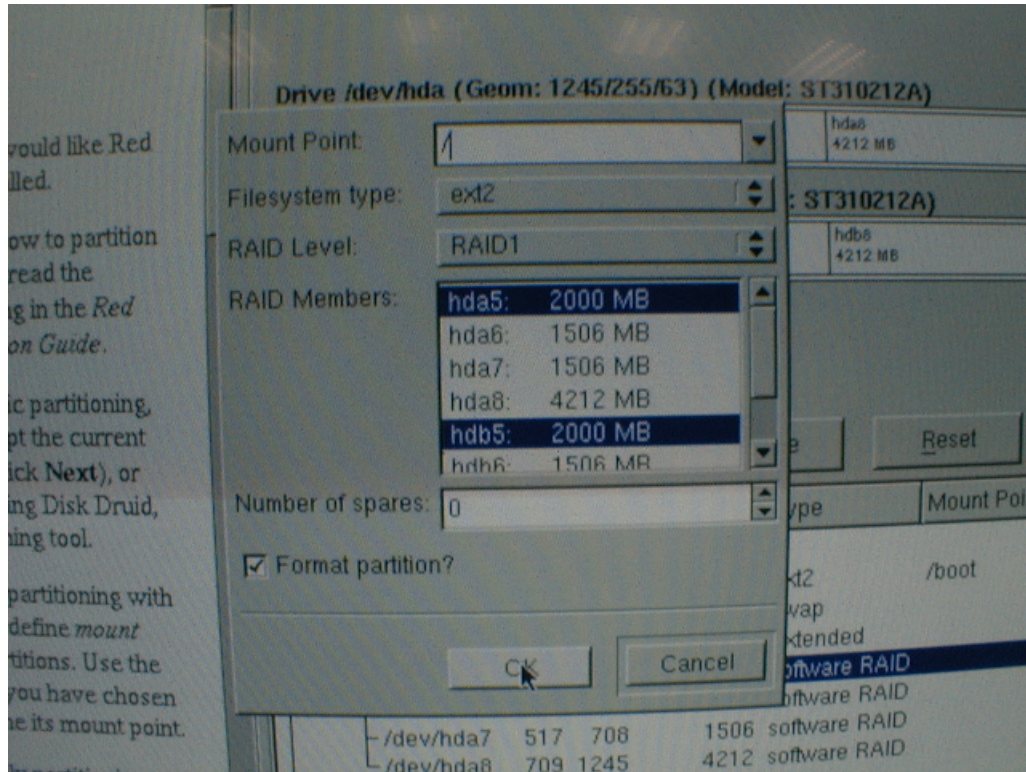
How To Configure The swap Partition



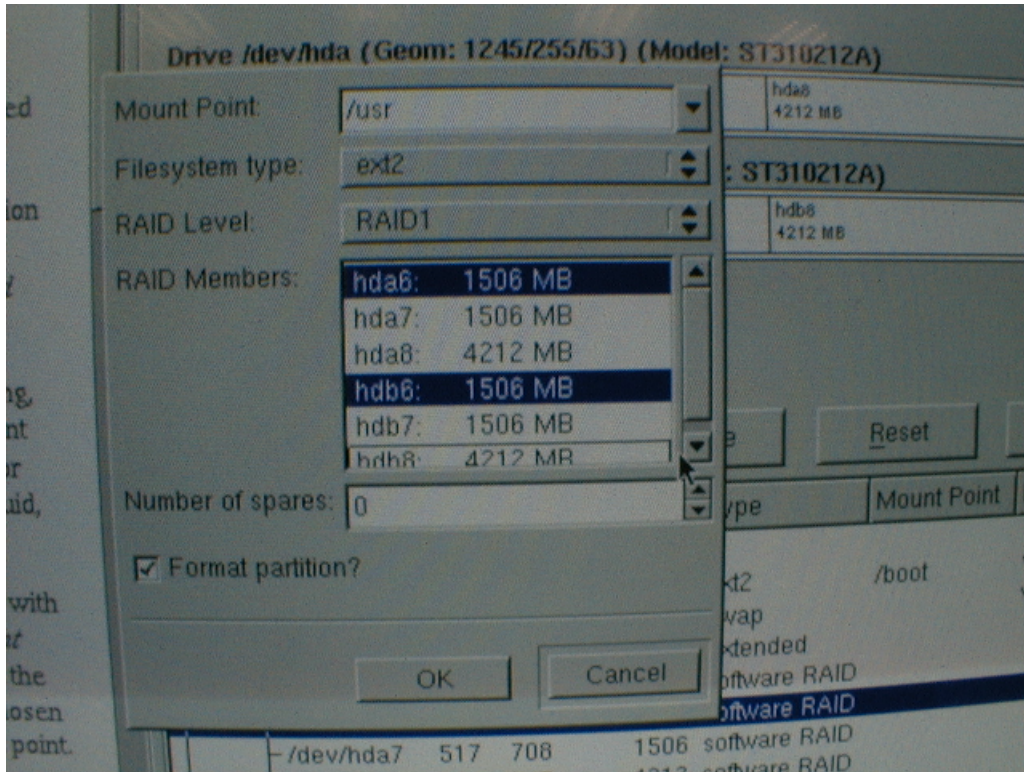
Using Disk Druid To Make A RAID Device



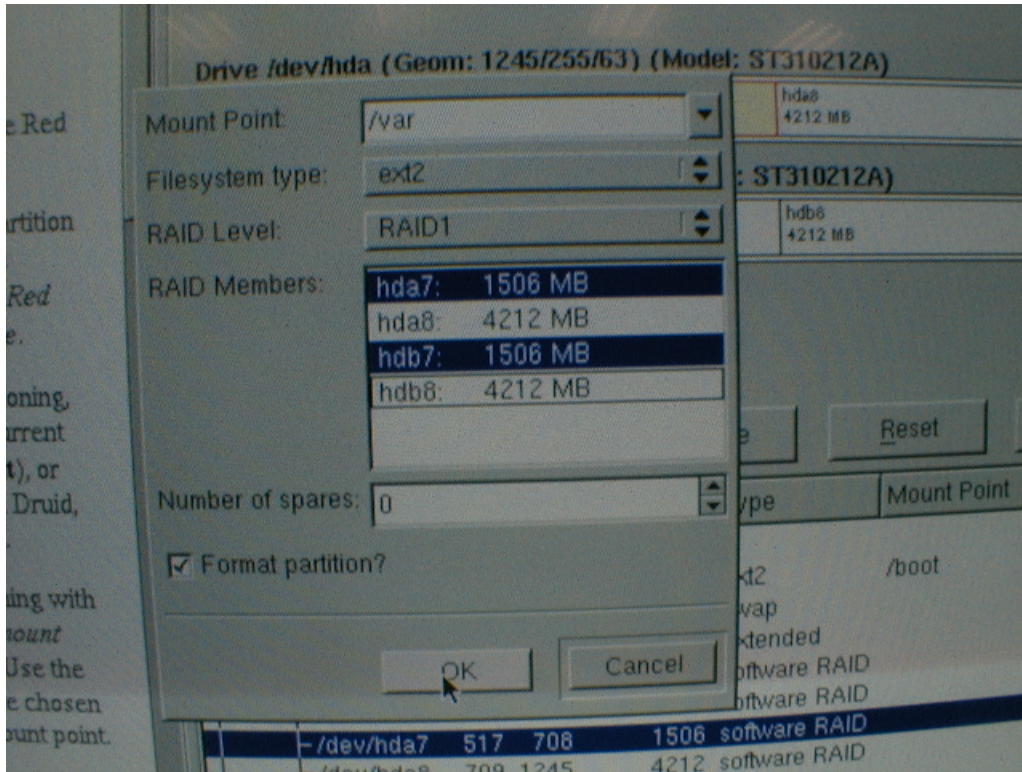
How To Select The Appropriate RAID-1 Partitions for /



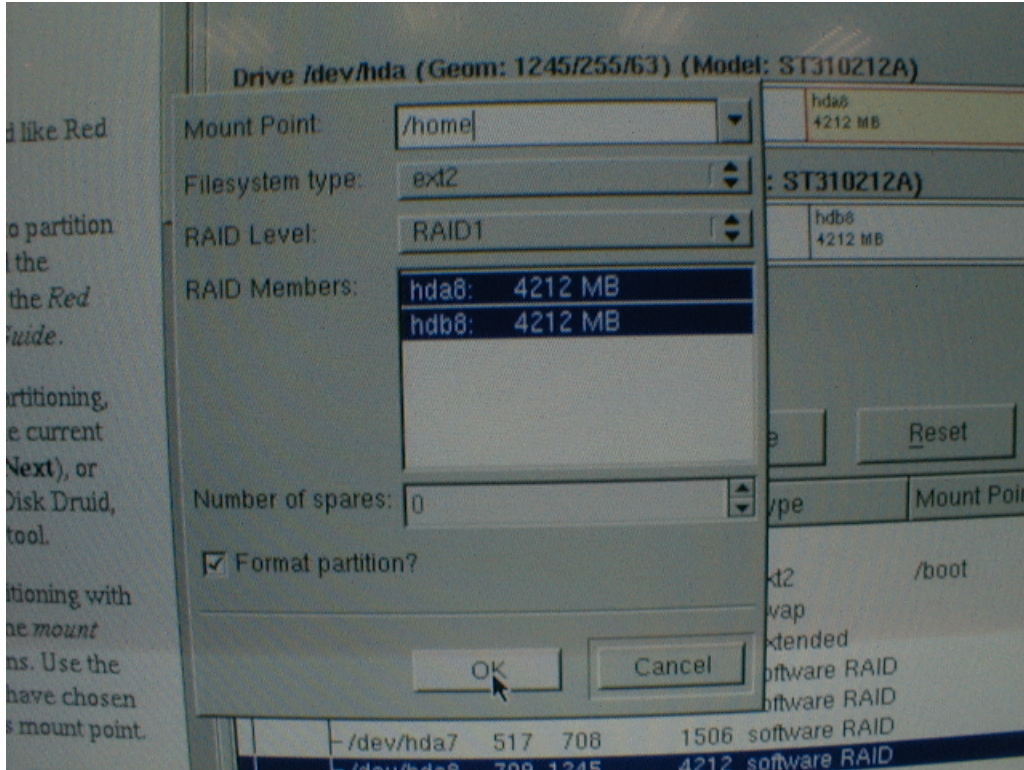
How To Configure the /usr Partition



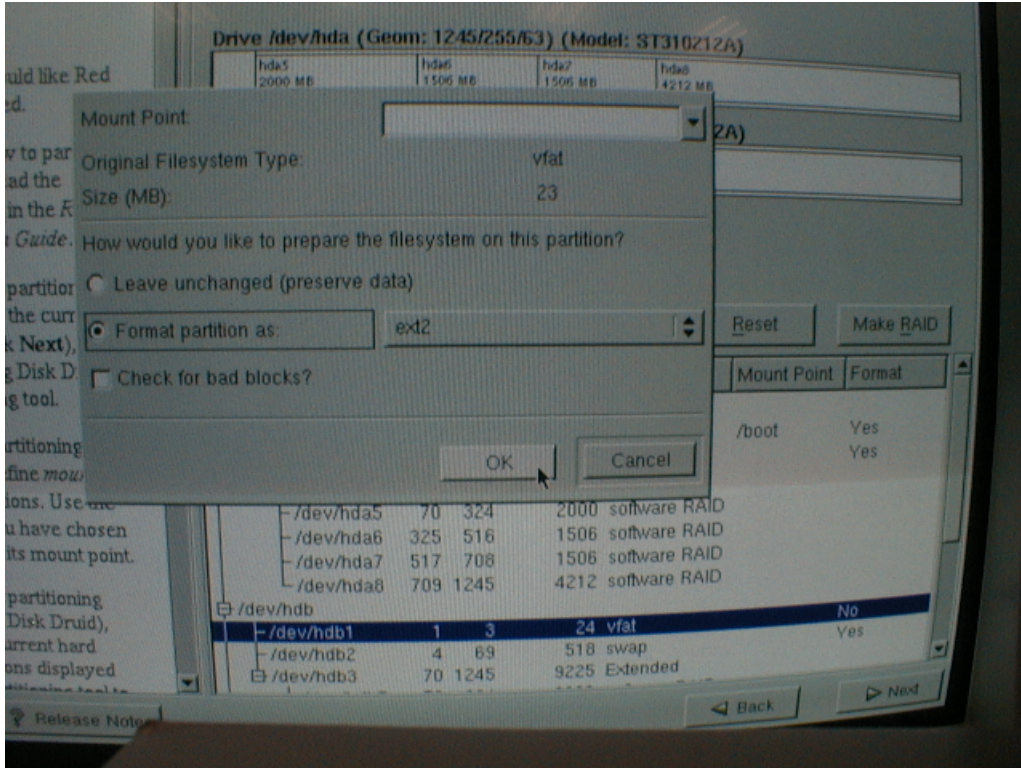
How To Configure the /var Partition



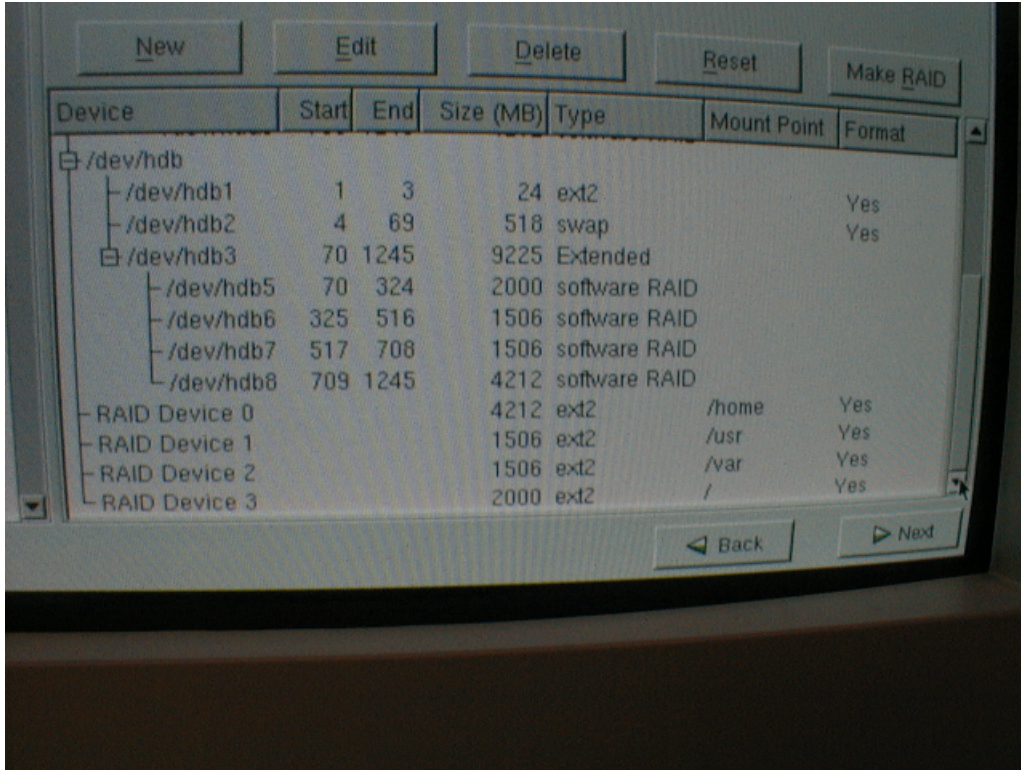
How To Configure The /home Partition



How To Configure The Duplicate /boot Partition



How To Verify The RAID Configuration

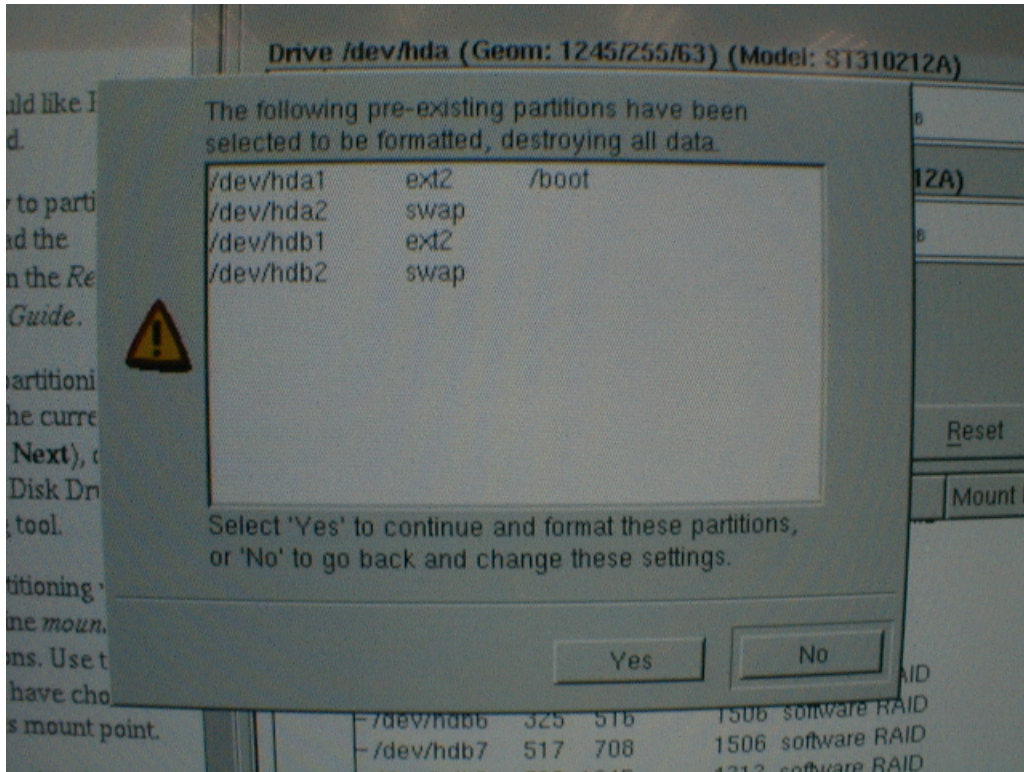


The screenshot shows a RAID configuration utility window with a tree view of the disk layout. At the top, there are buttons for 'New', 'Edit', 'Delete', 'Reset', and 'Make RAID'. The main area is a table with columns for 'Device', 'Start', 'End', 'Size (MB)', 'Type', 'Mount Point', and 'Format'. The tree view shows the following structure:

- /dev/hdb
 - /dev/hdb1 (24 MB, ex12, Format: Yes)
 - /dev/hdb2 (518 MB, swap, Format: Yes)
 - /dev/hdb3 (9225 MB, Extended)
 - /dev/hdb5 (2000 MB, software RAID)
 - /dev/hdb6 (1506 MB, software RAID)
 - /dev/hdb7 (1506 MB, software RAID)
 - /dev/hdb8 (4212 MB, software RAID)
 - RAID Device 0 (4212 MB, ex12, Mount Point: /home, Format: Yes)
 - RAID Device 1 (1506 MB, ex12, Mount Point: /usr, Format: Yes)
 - RAID Device 2 (1506 MB, ex12, Mount Point: /var, Format: Yes)
 - RAID Device 3 (2000 MB, ex12, Mount Point: /, Format: Yes)

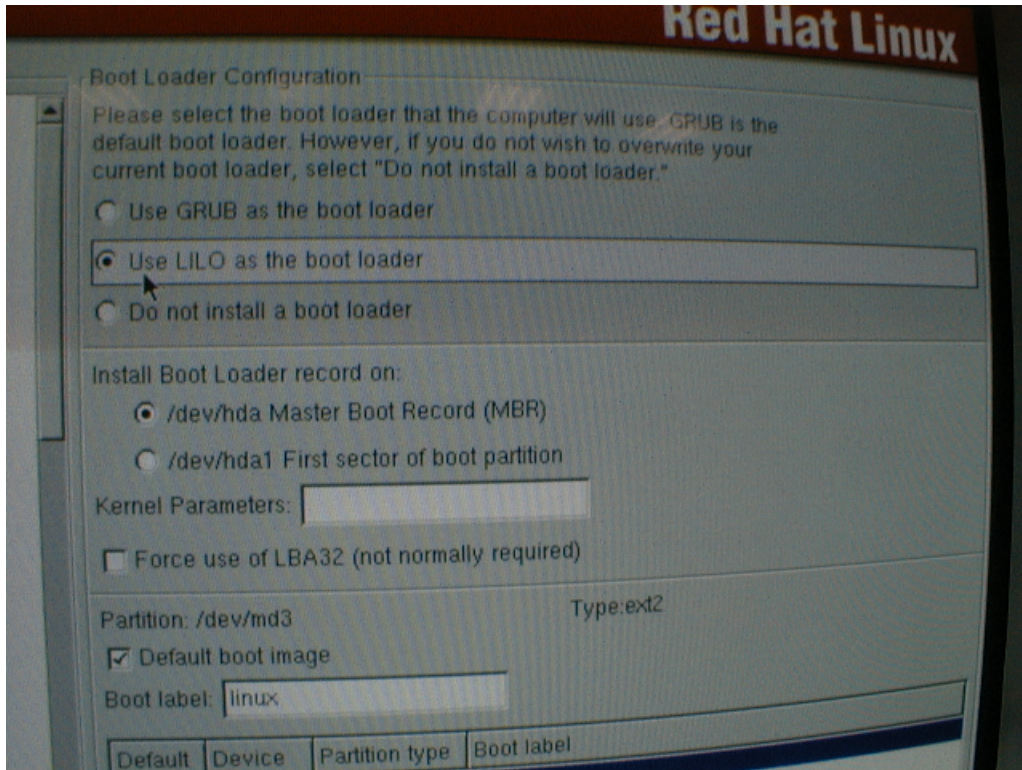
At the bottom of the window, there are 'Back' and 'Next' navigation buttons.

How To Confirm Partition Formatting



The Boot Loader

How To Configure The System To Use LILO



Red Hat Linux

Boot Loader Configuration

Please select the boot loader that the computer will use. GRUB is the default boot loader. However, if you do not wish to overwrite your current boot loader, select "Do not install a boot loader."

Use GRUB as the boot loader

Use LILO as the boot loader

Do not install a boot loader

Install Boot Loader record on:

/dev/hda Master Boot Record (MBR)

/dev/hda1 First sector of boot partition

Kernel Parameters:

Force use of LBA32 (not normally required)

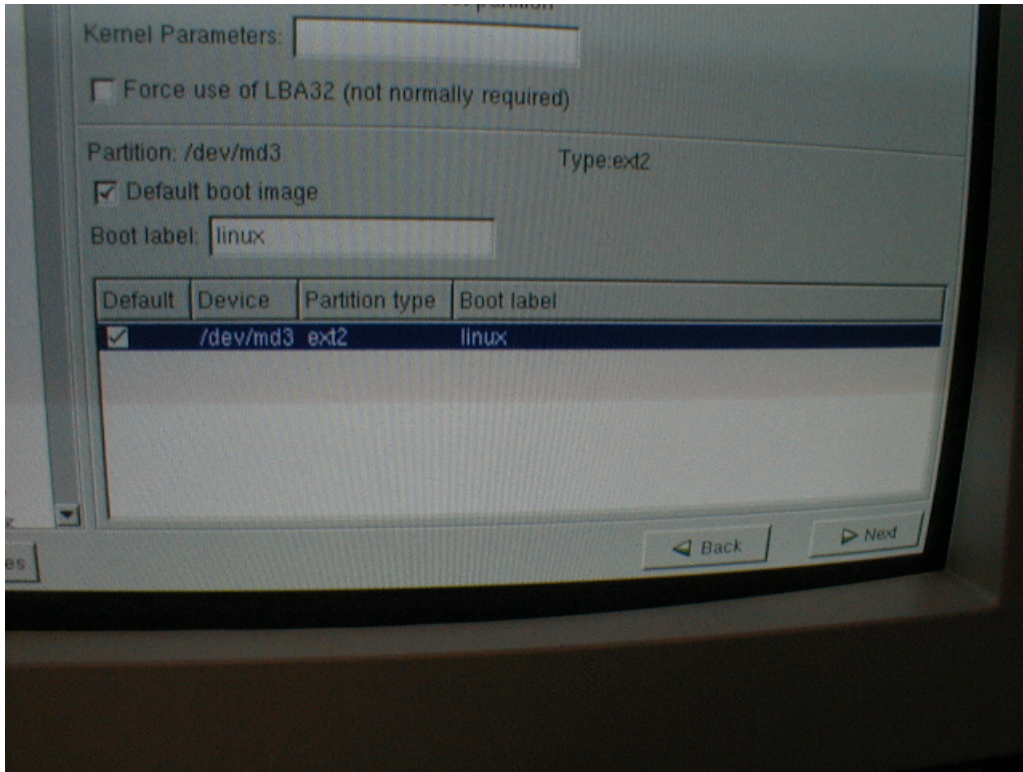
Partition: /dev/md3 Type: ext2

Default boot image

Boot label:

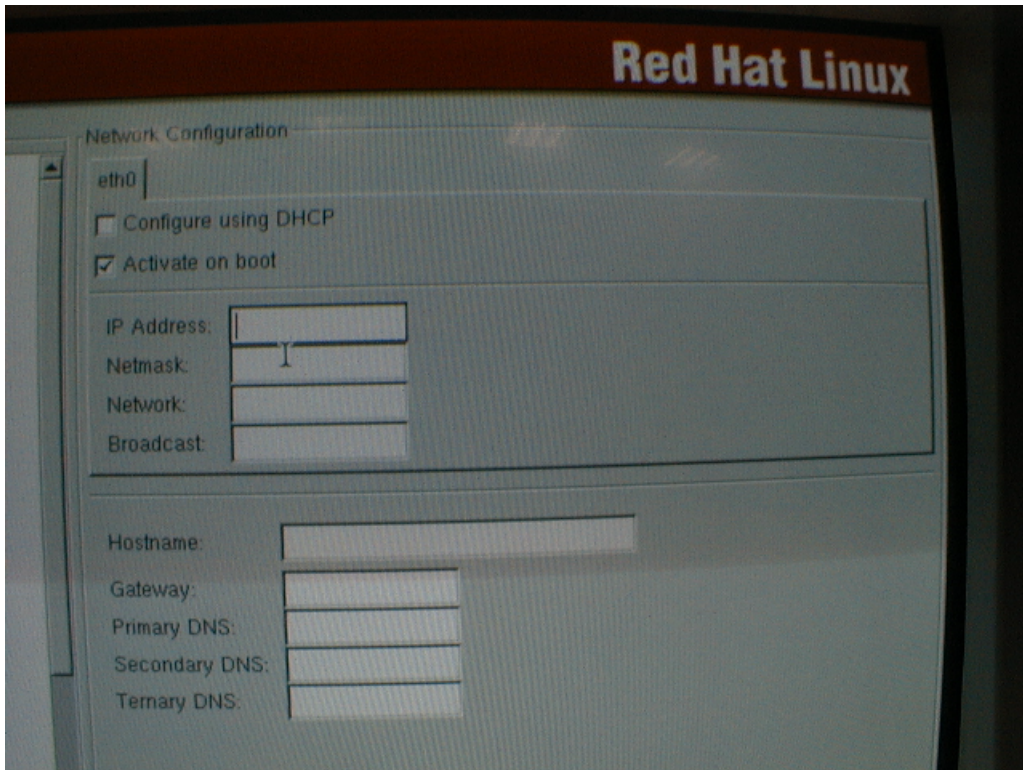
Default	Device	Partition type	Boot label
---------	--------	----------------	------------

Where To Locate The Boot Code

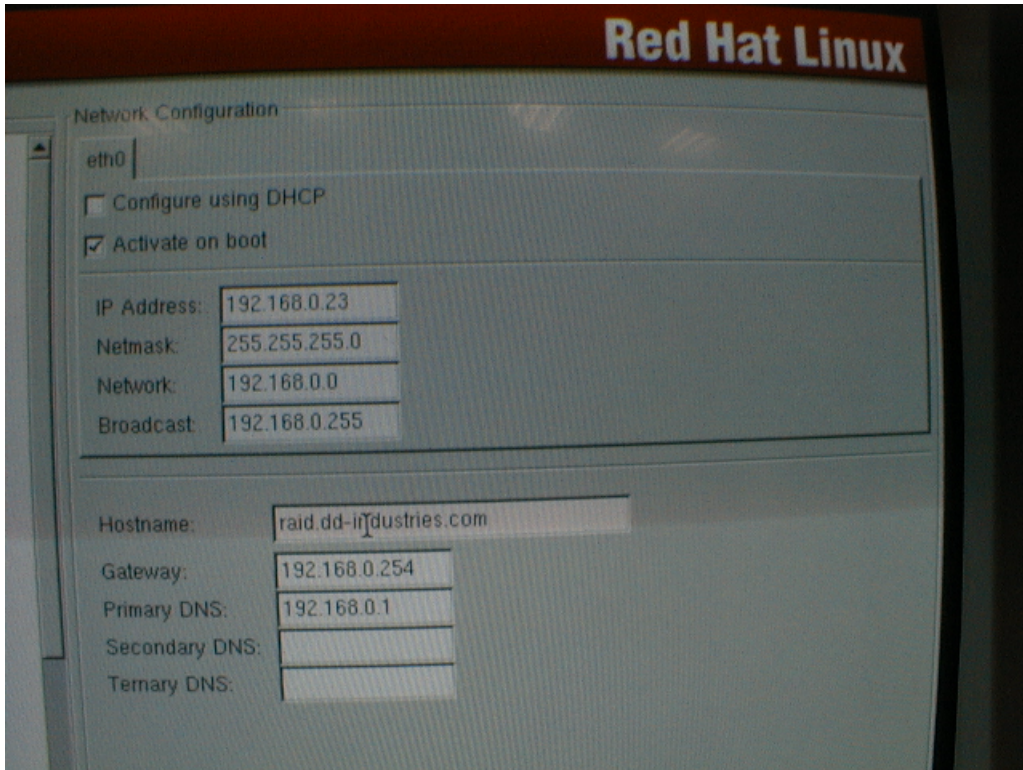


Networking

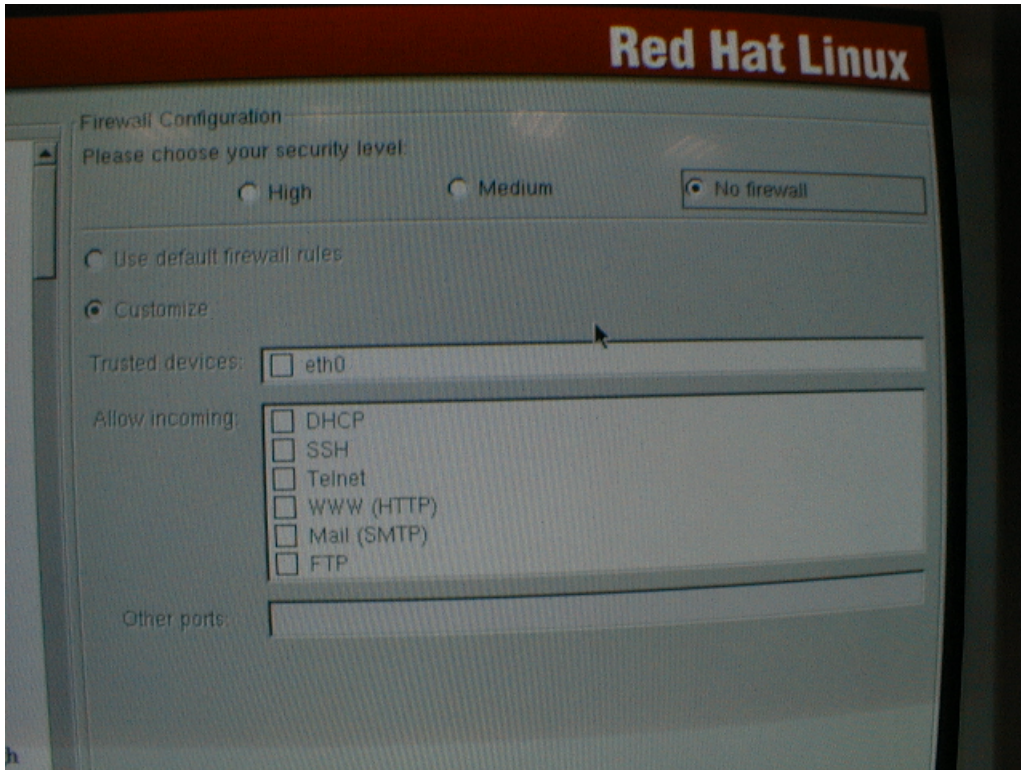
How To Configure Networking



An Example Network Configuration

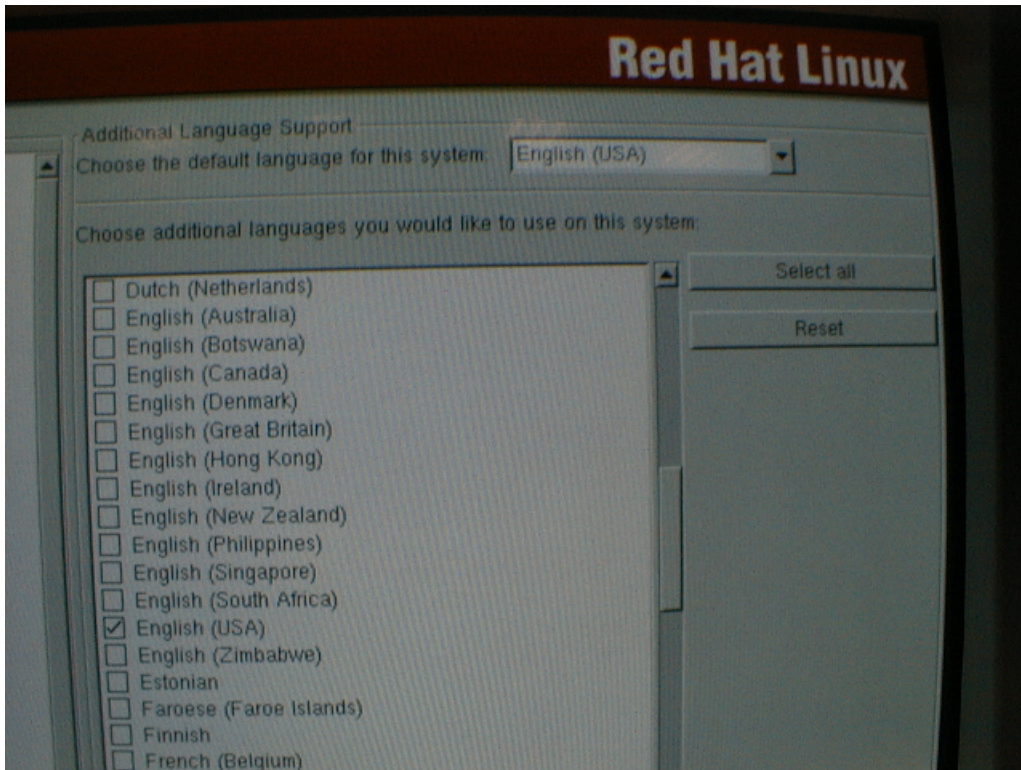


How To Defer Firewalling



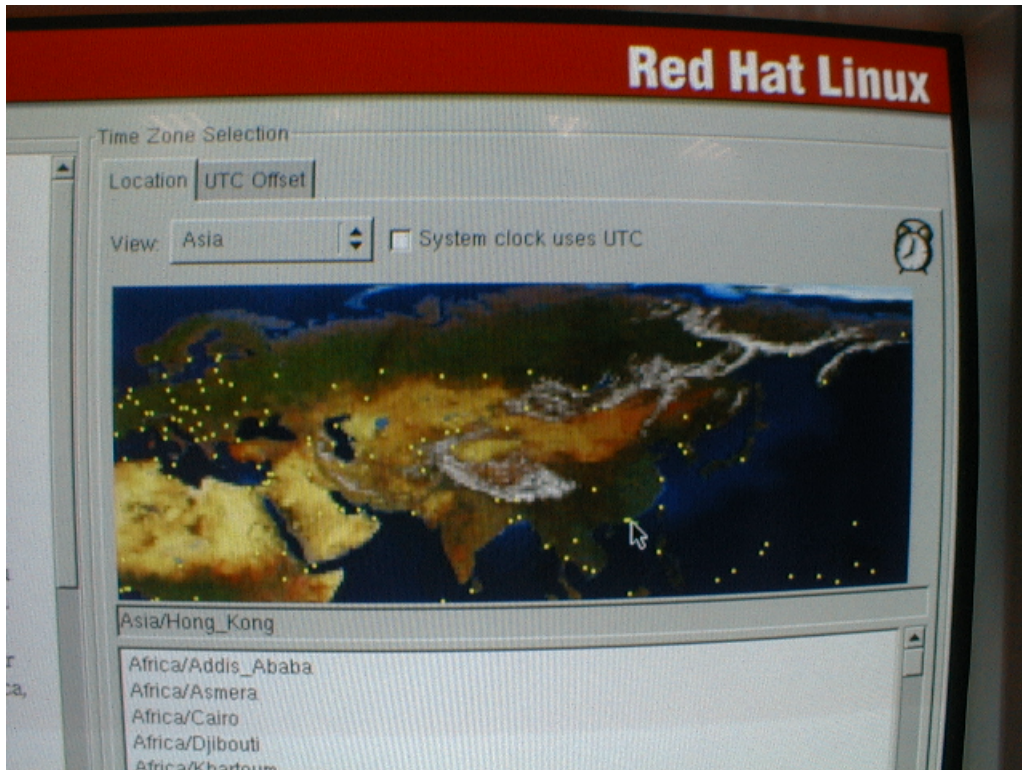
Additional Language Support

How To Enable Additional Language Support



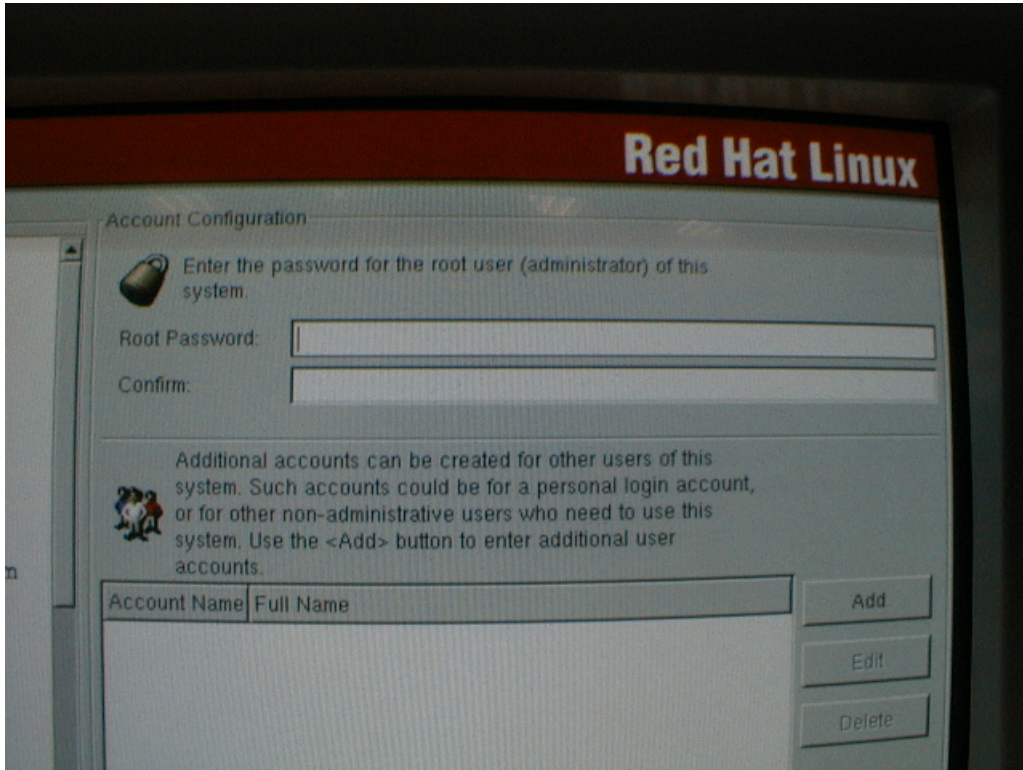
Time Zone

How To Configure Time Zone Support

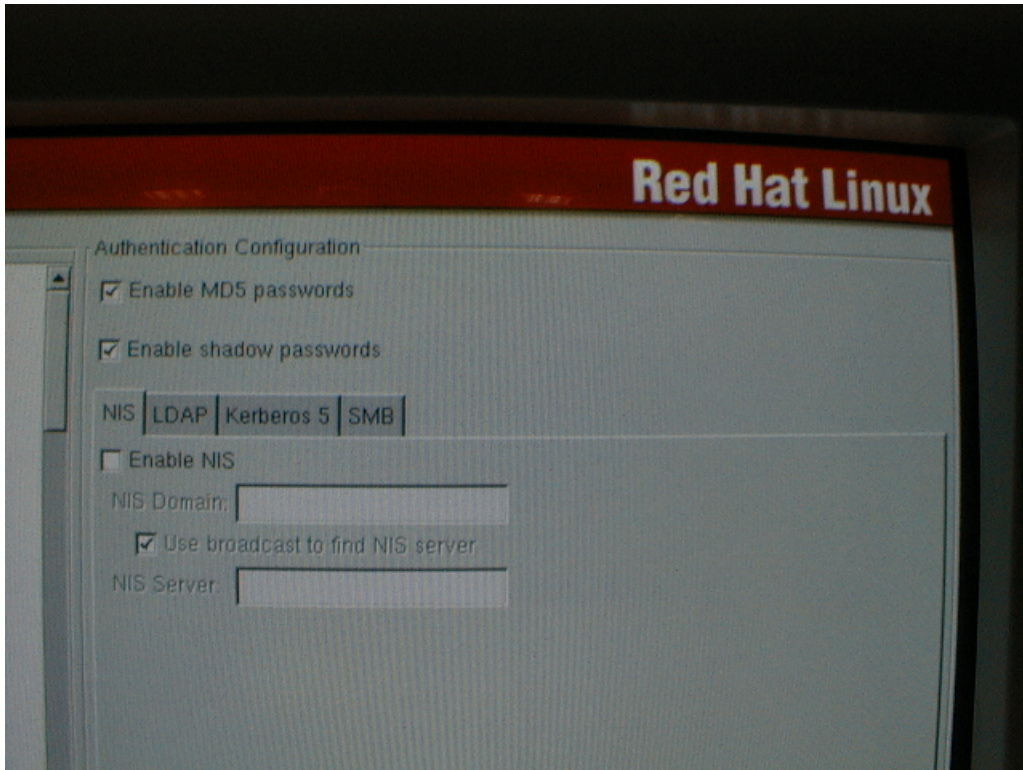


Accounts And Passwords

Setting the root Password

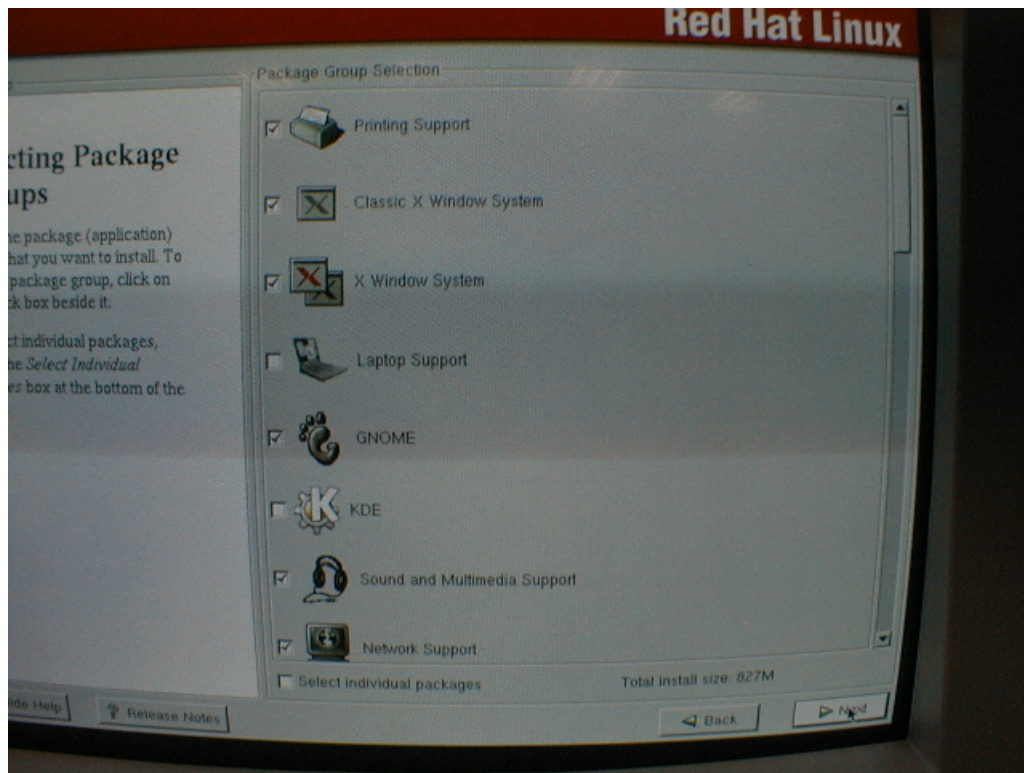


Enabling MD5 And Shadow Passwords



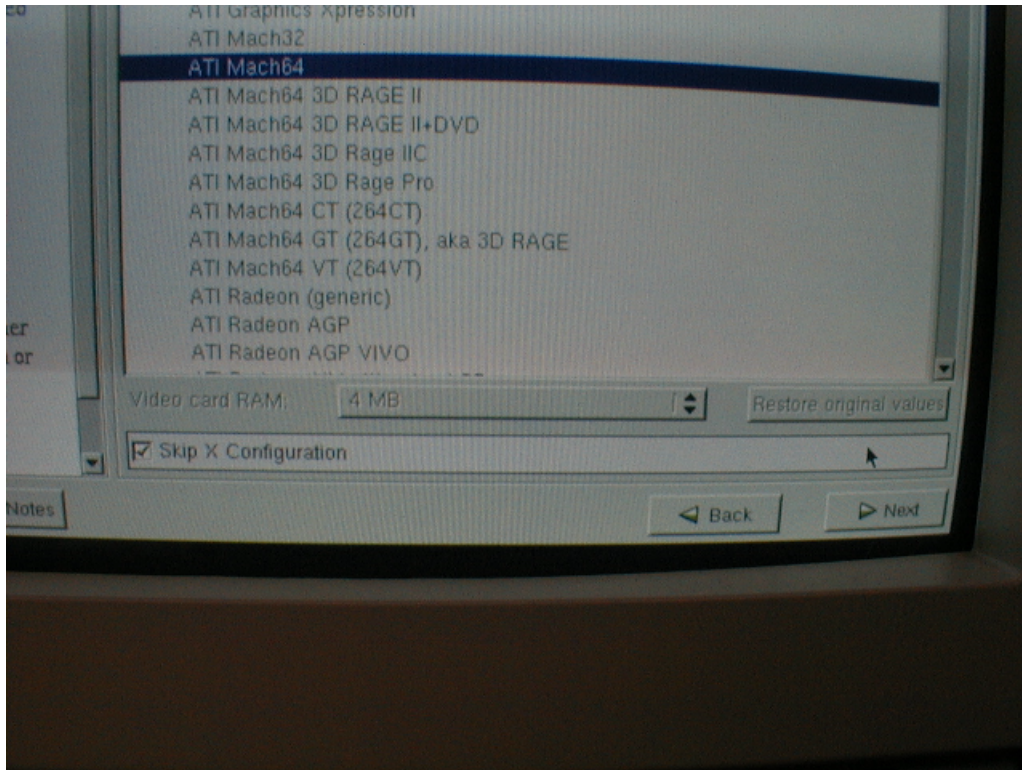
Subsystem Installation

Choosing which Subsystems To Install



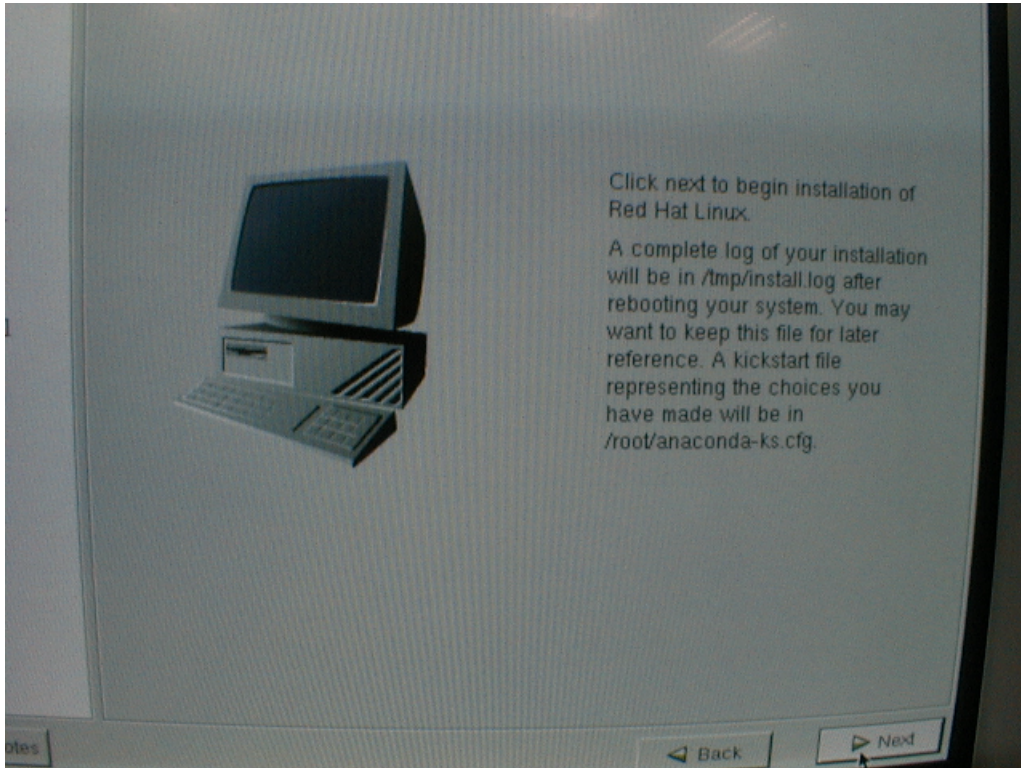
X-Windows

Skipping The X-Windows Setup



Initiating The Installation

Launching the Package Installation

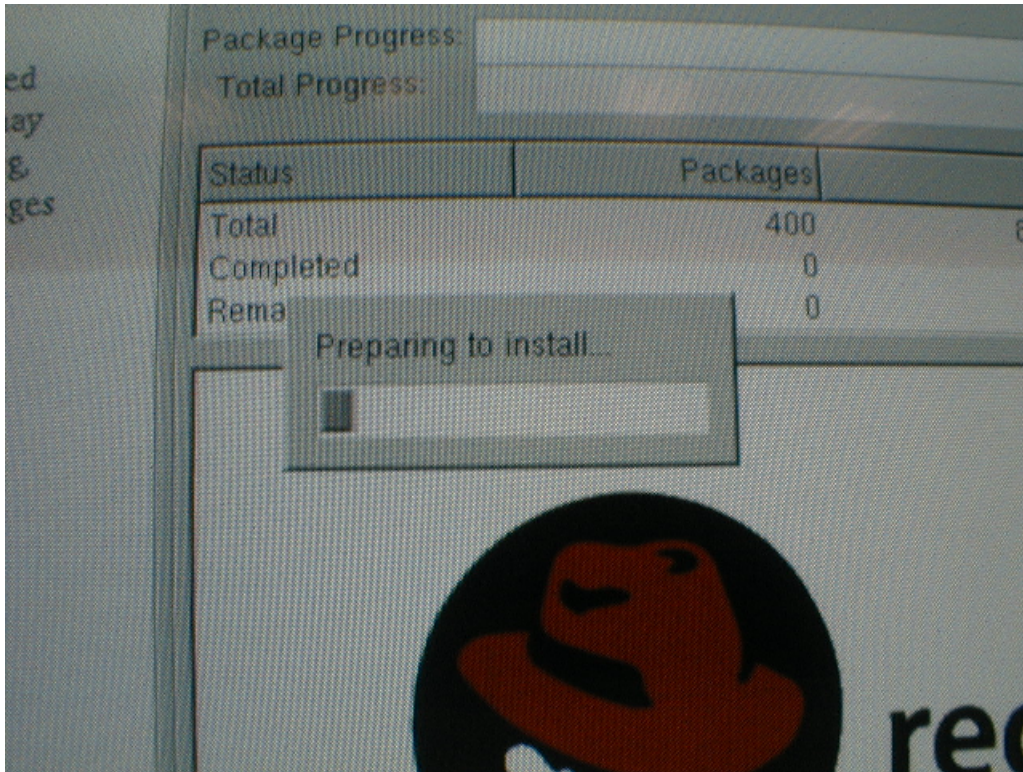


Watching The Package Installation Process

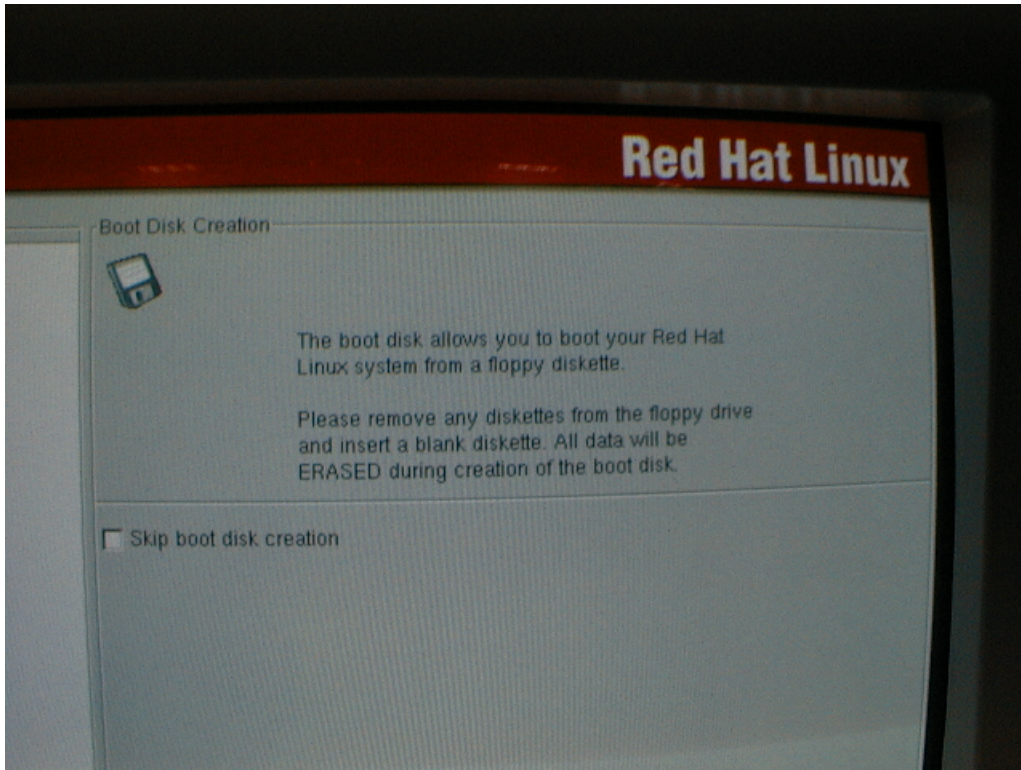
Formatting Partitions



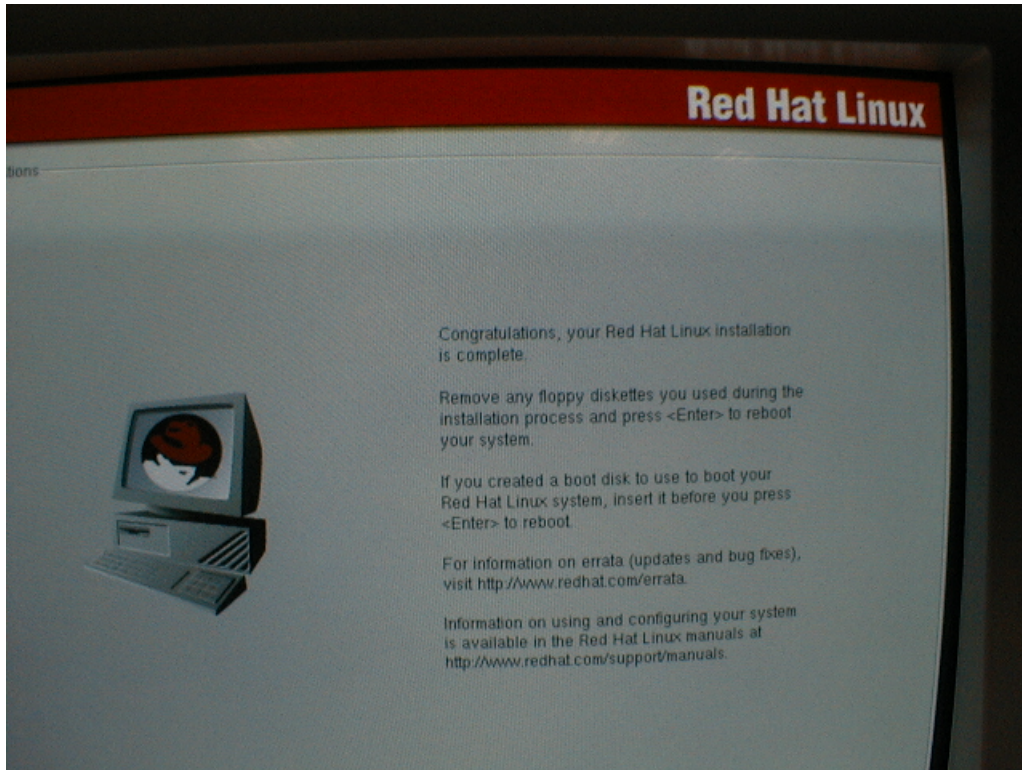
Watching The System Prepare The Installation



Creating the Red Hat Boot Disk

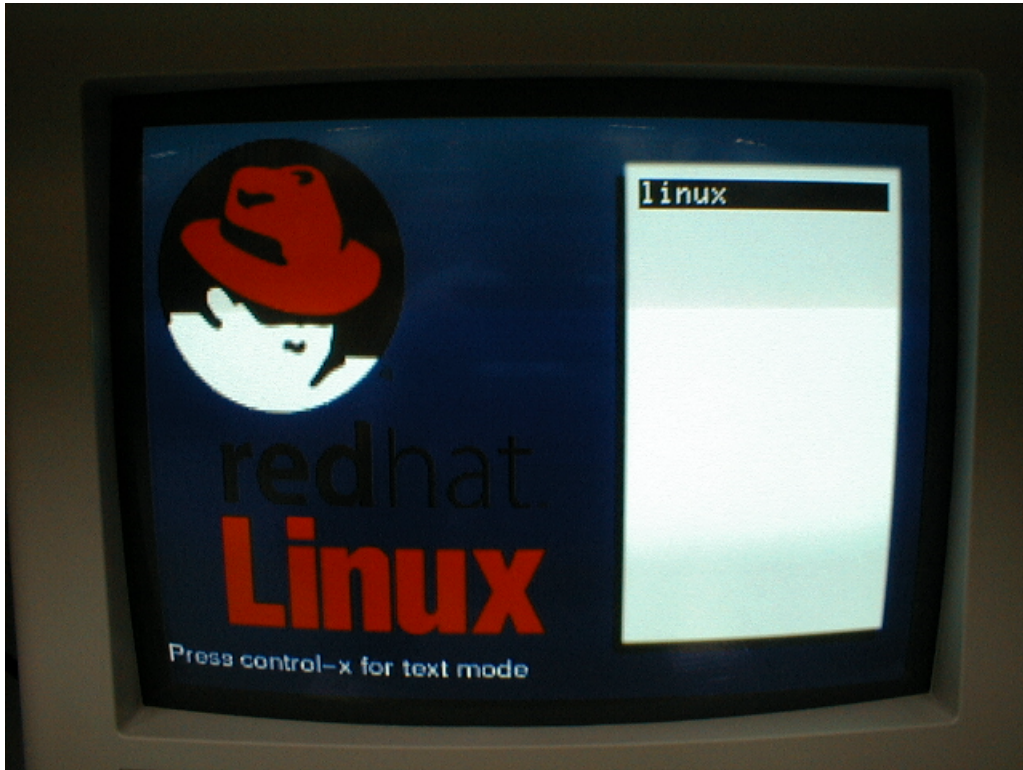


Concluding The Install



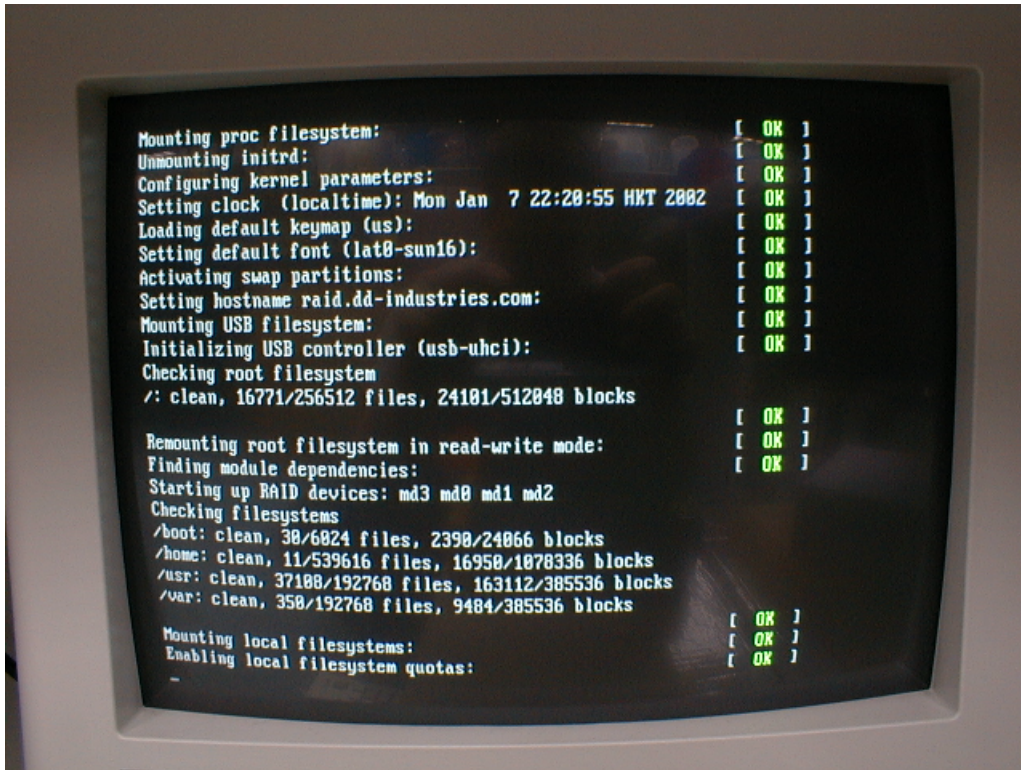
The First Boot

The First Post-Installation Boot



Watching The Kernel Boot For The First Time

Consider the following:



```
Mounting proc filesystem: [ OK ]
Unmounting initrd: [ OK ]
Configuring kernel parameters: [ OK ]
Setting clock (localtime): Mon Jan 7 22:20:55 HKT 2002 [ OK ]
Loading default keymap (us): [ OK ]
Setting default font (lat0-sun16): [ OK ]
Activating swap partitions: [ OK ]
Setting hostname raid.dd-industries.com: [ OK ]
Mounting USB filesystem: [ OK ]
Initializing USB controller (usb-uhci): [ OK ]
Checking root filesystem
/: clean, 16771/256512 files, 24101/512048 blocks [ OK ]

Remounting root filesystem in read-write mode: [ OK ]
Finding module dependencies: [ OK ]
Starting up RAID devices: md3 md0 md1 md2
Checking filesystems
/boot: clean, 30/6024 files, 2390/24066 blocks
/home: clean, 11/539616 files, 16950/1070336 blocks
/usr: clean, 37108/192768 files, 163112/385536 blocks
/var: clean, 358/192768 files, 9484/385536 blocks [ OK ]

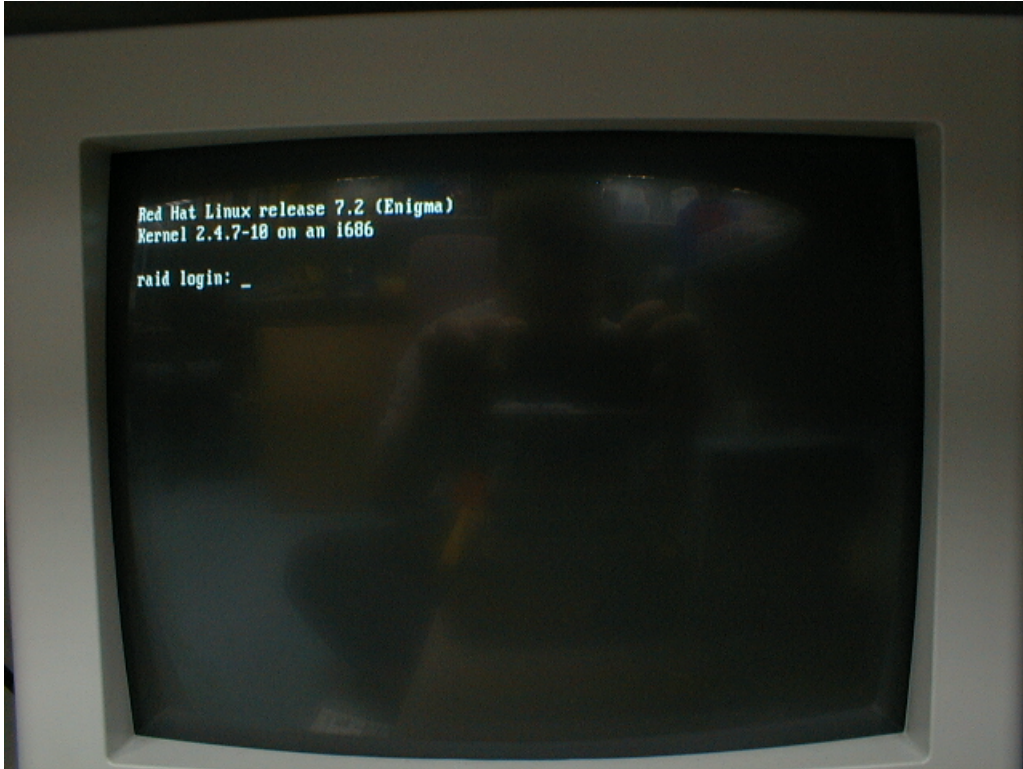
Mounting local filesystems: [ OK ]
Enabling local filesystem quotas: [ OK ]
```

Note the RAID devices md3, md0, md1 and md2 coming up.

Note the partitions coming up clean.

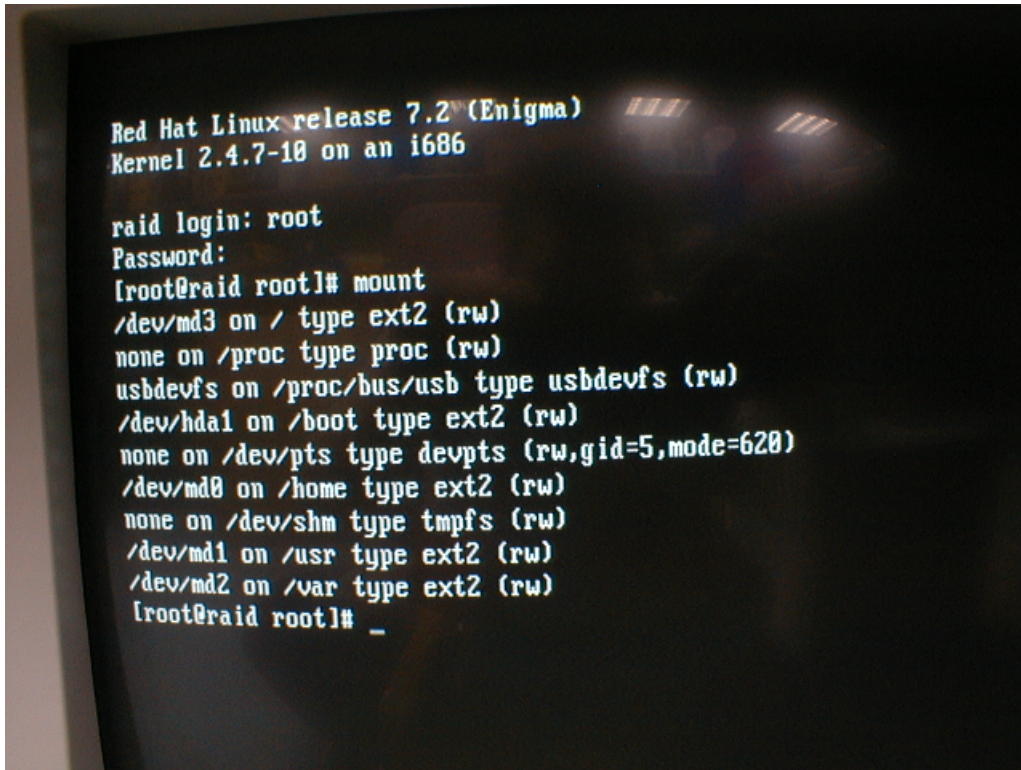
The Red Hat Login Screen

Consider the following:



Verifying the RAID Partitions

Consider the following:

A terminal window showing the output of the 'mount' command in a RAID environment. The text is as follows:

```
Red Hat Linux release 7.2 (Enigma)
Kernel 2.4.7-10 on an i686

raid login: root
Password:
[root@raid root]# mount
/dev/md3 on / type ext2 (rw)
none on /proc type proc (rw)
usbdevfs on /proc/bus/usb type usbdevfs (rw)
/dev/hda1 on /boot type ext2 (rw)
none on /dev/pts type devpts (rw,gid=5,mode=620)
/dev/md0 on /home type ext2 (rw)
none on /dev/shm type tmpfs (rw)
/dev/md1 on /usr type ext2 (rw)
/dev/md2 on /var type ext2 (rw)
[root@raid root]# _
```

In the above example the **mount** command was used to display the active partitions.

Viewing RAID Partition Status in /proc

Consider the following:

```
[root@raid root]# cat /proc/mdstat
Personalities : [raid1]
read_ahead 1024 sectors
md0 : active raid1 hda8[0] hdb8[1]
      4313344 blocks [2/2] [UU]

md2 : active raid1 hda7[0] hdb7[1]
      1542144 blocks [2/2] [UU]

md1 : active raid1 hda6[0] hdb6[1]
      1542144 blocks [2/2] [UU]

md3 : active raid1 hda5[0] hdb5[1]
      2048192 blocks [2/2] [UU]

unused devices: <none>
[root@raid root]# _
```

In the above example, the **cat** command was used to display the contents of the file **/proc/mdstat**, which holds the current status of the raid array.

Everything looks fine - looks like a working system.