intel ®

RAID Software Suite for Linux

Installation Guide Rev 1.001

June 2000

intel ®

intel®

intel ®

Contents

1 RAID Software Suite Installation for Linux

Figures

intel®

RAID Software Suite Installation for Linux 1

1.1 Summary of Contents

- Install the RAID Software Suite onto an Existing Linux Server
  — Rebuild the Kernel with I_2O Support
  — Activate the I_2O Subsystem
- RAID Software Overview
  — RAID Configuration Services
  — RAID Configuration Services Administration
  — RAID Storage Console
  — IRVIEW
- Using the RAID Software Suite
  — Using RAID Configuration Services Administration
  — Configure a RAID Volume in a Linux System
  — Monitor the Status of the RAID Subsystem
  — Uninstalling RAID Software Suite
- Booting to Red Hat Linux on an I_2O RAID Volume
  — Method 1 - MBR on Primary IDE
  — Method 2 - MBR on I_2O RAID Volume
- Technical Notes
  — Linux Dynamic Block Device Limitations
- Miscellaneous Notes

1.2 Software Requirements

- Linux RAID Software Suite CD-ROM
  — Linux Kernel I_2O Support (open source)*
  — RAID Storage Console / RAID Configuration Services
  — Utility tools (Command Line tool: irview)
- Red Hat Linux 6.x (installed on your machine)
- *Root* access on the Linux server

- Hard disk space: Minimum 20 MB

*The I$_2$O kernel modules were developed as an Open Source project. These modules were developed for two kernel versions: 2.2.x (current stable versions) and 2.3.x (development version leading to 2.4 version). The modules can be obtained along with the latest kernel sources from ftp://ftp.kernel.org/pub/linux/kernel/ . We suggest you obtain the very latest modules. *This CD-ROM contains the 2.2.x version of the I$_2$O modules.*

1.3 Install the RAID Software Suite onto an Existing Linux Server

1.3.1 Installation

1. Place the Linux RAID Software Suite CD-ROM in the CD-ROM Drive. If the installation does not automatically begin, then mount the CD-ROM:

   ```
   $ mount /dev/cdrom /mnt/cdrom
   ```

   and run the install script:

   ```
   $/mnt/cdrom/autorun
   ```

   If an X Windows is not available, then run the install script:

   ```
   $/mnt/cdrom/install
   ```

2. Install the software. Select *Yes* to continue the installation. Select *No* to cancel the installation. See Figure 1-1.

Note: Ignore warning messages printed out on the console during installation. If the install script detects a fatal failure, it will automatically exit the program.

Figure 1-1. Installation Confirmation

3. Install the kernel sources with I$_2$O support (supplied with the CD-ROM). Note: check Linux kernel source sites for the latest Kernel source code (later than Linux 2.2.15). See Figure 1-2.

  a. If you select *Yes*, the screen in Figure 1-3 appears. When you install the kernel sources supplied by the installation program, then the sources are unpacked in the `/usr/src` directory and appropriately linked to the `/usr/src/Linux` directory.

  b. If you wish to install source code later than Linux 2.2.15, select *No*. The screen in Figure 1-4 appears. The source code on the CD-ROM is not installed and the remaining components of the software suite are installed.

Figure 1-2. Install the Kernel Sources

Figure 1-3. Install Kernel Sources from Install Program
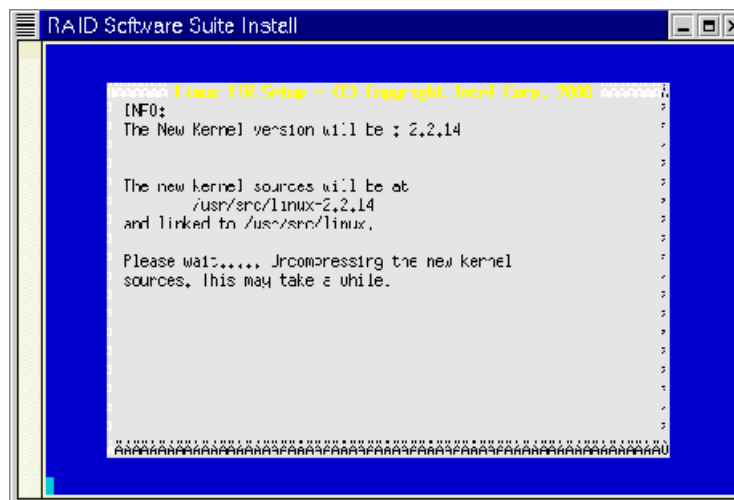
Figure 1-4. Bypass Installation of Kernel Source Code
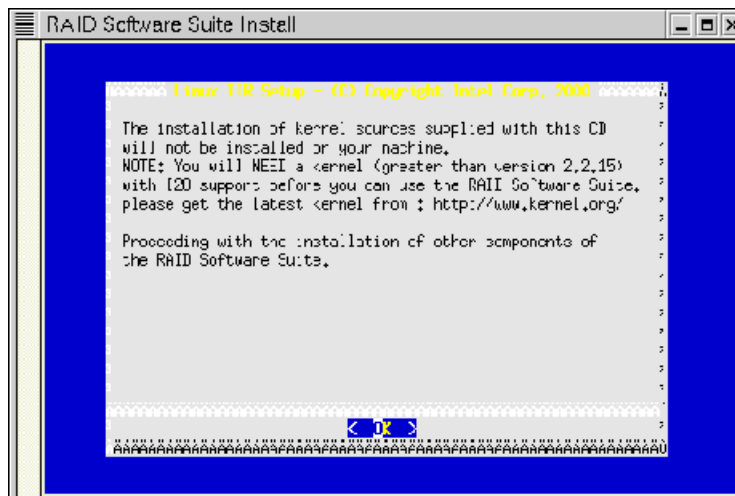
4. Install version 1.0.5 or a later version of Kaffe Java. Check the version of Kaffe installed on your machine, with the command "rpm -q kaffe". If your version of Kaffe Java is later than 1.0.5, then select *No*, otherwise select *Yes*. See Figure 1-5. Note: to use the RAID Configuration Service, you must have "Java Runtime Environment" installed on your machine. Kaffe is an open-source Java initiative, which is also supplied with Red Hat Linux.

Figure 1-5. Install Java
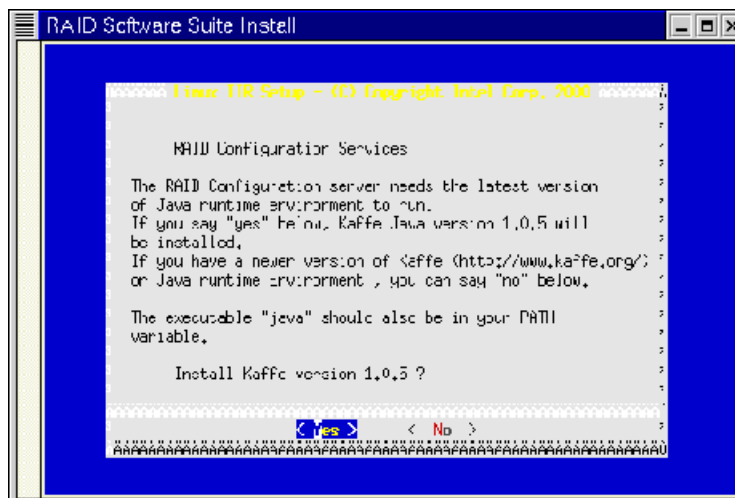
5. Complete the installation by clicking on *OK* in the screens.

6. Rebuild the kernel. See Section 1.3.2.

7. Complete the installation by starting RAID Configuration Services (see Section 1.4.1.) and configuring RAID volumes using RAID Storage Console (see Section 1.4.3.).

1.3.2   Rebuild the Kernel with I$_2$O Support

The Linux RAID Software Suite installation program will install the latest kernel source with I$_2$O support in the `/usr/src/linux` directory. You must rebuild the kernel with I$_2$O support. I$_2$O support can be either statically linked with the kernel or it can be built as kernel modules. The kernel can be built with many options and should be customized for your particular machine.

1. Type the following at the command line:

   ```
   $ cd /usr/src/linux
   ```

   then

   ```
   $ make xconfig
   ```

The screen in Figure 1-6 appears.

Figure 1-6. Linux Kernal Configuration
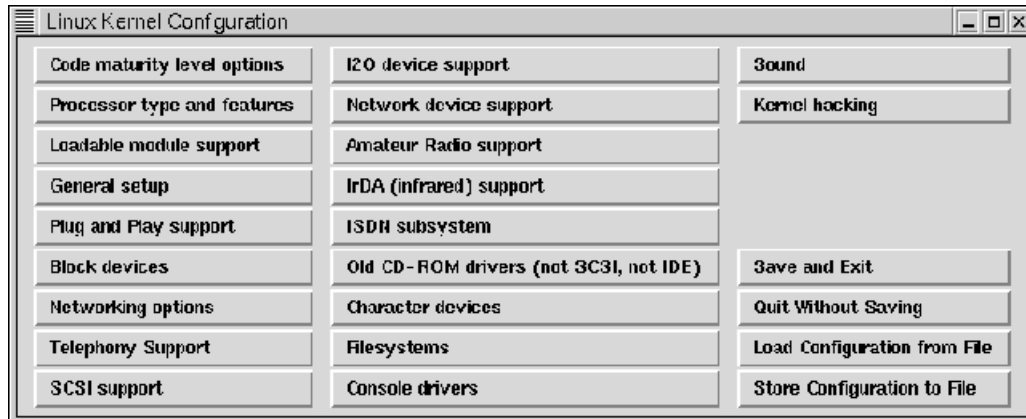
2. Configure I$_2$O support. Click on the I$_2$O support button on the Kernel Configuration screen. See Figure 1-6.

   a. Enable I$_2$O support in the kernel. Select one of the following methods:

      —*Statically* build I$_2$O support into the new kernel. Select "y" to the relevant I$_2$O choices. See Figure 1-7.

      —Build the I$_2$O support as kernel modules. Select "m" to the relevant I$_2$O choices. See Figure 1-8.

   To be able to use the RAID Software Suite, two components of I$_2$O, I$_2$O PCI support and I$_2$O Block OSM, are required. Other components are optional.

Figure 1-7. Configuring I₂O Support in the Kernel

Figure 1-8. Configuring I₂O Support as Kernel Modules
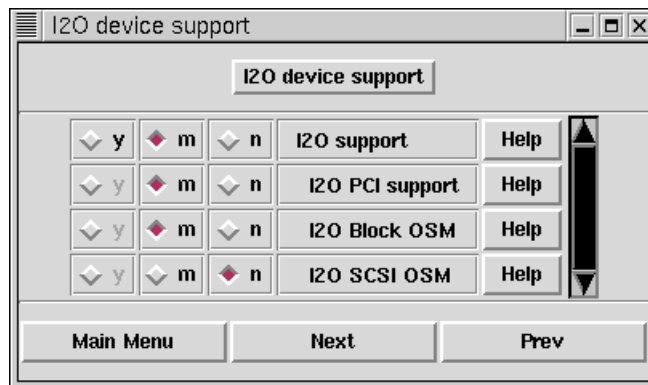
3. Once you have completely configured the kernel, save your configuration and build the kernel and modules as appropriate.

```
$ make dep
$ make clean
$ make bzImage
$ cp /usr/src/linux/boot/arch/i386/bzImage /boot/vmlinuz-
yourlabel
$ make modules
$ make modules_install
```

Additional reading for configuring and rebuilding the Linux kernel:

/usr/src/linux/README

Linux Kernel Configuration HOWTO at http://www.linuxdoc.org/HOWTO/Kernel-HOWTO.html

4. Make relevant changes to /etc/lilo.conf per the following example:

Before example:

```
image=/boot/vmlinuz-2.2.12-20
```

```
label=linux
initrd=/boot/initrd-2.2.12-20.img
read-only
root=/dev/hda2
```

After example:

```
image=/boot/vmlinuz-2.x.x-label
label=mylabel
initrd=/boot/initrd-2.2.12-20.img
read-only
root=/dev/hda2
```

5. Run `/sbin/lilo`. See Linux Loader (LILO) HOWTO at http://www.linuxdoc.org/ HOWTO/mini/LILO.html for more information.

6. Reboot your machine and load the new kernel.

7. If you build I$_2$O support with kernel modules, you must activate the I$_2$O subsystem as described in Section 1.3.3.

1.3.3   Activate the I$_2$O Subsystem

If you built the I$_2$O support statically into the kernel, then the RAID controller card will be automatically detected at bootup.

If you built the kernel with I$_2$O support as kernel modules, you must load the kernel modules.

1. Run the following commands:

    ```
    $ modprobe i2o_block
    ```

    then

    ```
    $ modprobe i2o_config
    ```

2. Confirm that the modules were loaded properly using the `lsmod` command.

    Type the following on the command line: `$ lsmod`

    The following output appears:

    ```
    Module Size Used by
    i2o_config 6576 0
    i2o_block 23476 1
    i2o_core 36016 0 [i2o_config i2o_block]
    i2o_pci 2368 1 [i2o_core]
    ```

For more information on Linux Kernel modules read /usr/src/linux/Documentation/ modules.txt

int_el®

1.4 RAID Software Overview

1.4.1 RAID Configuration Services

The RAID Configuration Services allow you to configure your RAID volumes, disks and arrays. You must start RAID Configuration Services before configuring your RAID volume. Type the following at the command line:

```
$ /opt/iir/bin/iird start | stop | restart
```

Access the RAID Configuration Services Administration through the shortcut located off of the system branch of your GNOME/KDE start menu.

This program may be called from any of the `init` scripts at boot-up and shutdown time. However, the RAID Configuration Services are only required to run when you are configuring the RAID subsystem with RAID Storage Console. Once the RAID subsystem configuration is completed, you may wish to stop RAID Configuration Services to free system resources.

1.4.2 RAID Configuration Services Administration

The RAID Configuration Services Administration utility is used to configure parameters of the RAID Configuration Services. This X Windows program provides the ability to manage user access, TCP/IP port settings, and remote access of the RAID Configuration Services.

An alternative to using the RAID Configuration Services Administration utility is to manually edit the `/opt/iir/iirserver/etc/config.iir` text file and restart the RAID Configuration Services.

1.4.3 RAID Storage Console

The RAID Storage Console is an HTML-based configuration utility that is used to configure the RAID subsystem. The RAID Storage Console can be accessed locally or remotely.

Local

Access the local RAID Storage Console through the shortcut located off of the *system* branch of your GNOME/KDE start menu.

Remote

The RAID Storage Console can also be accessed remotely. If you want to access it remotely, use a web-browser and point it to:

```
http://hostname:960/
```

intel.

The RAID storage console is under restricted access. For authorized access, use the following initial username/password pair: `root/root`

Warning: Change the default password and /or username immediately after the installation. Failing to do so is a security risk. Refer to Section 1.5.1 for instructions.

See the RAID Controller User's Manual for a complete description of the RAID Storage Console utility and its usage.

1.4.4   IRVIEW

IRVIEW is a command line utility that provides real-time status of the RAID subsystem.

1.5   Using the RAID Software Suite

1.5.1   Using RAID Configuration Services Administration

1. Start the RAID Configuration Services Administration program by running:

   ```
   $ /opt/iir/bin/iradmin
   ```
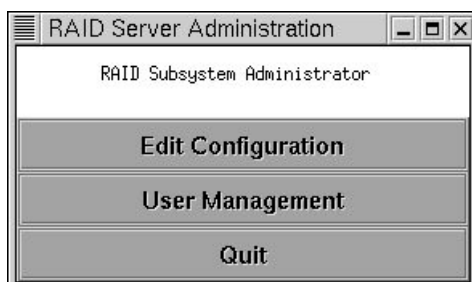
   or

   Access the RAID Configuration Services Administration through the shortcut located off of the system branch of your GNOME/KDE start menu.

   The screen in Figure 1-9 appears.

Figure 1-9. RAID Configuration Services Administration

2. Modify the port and remote access settings of the RAID Configuration Services. Click the button labeled *Edit Configuration.* See Figure 1-10.

Figure 1-10. RAID Configuration Services Administration - Edit Configuration
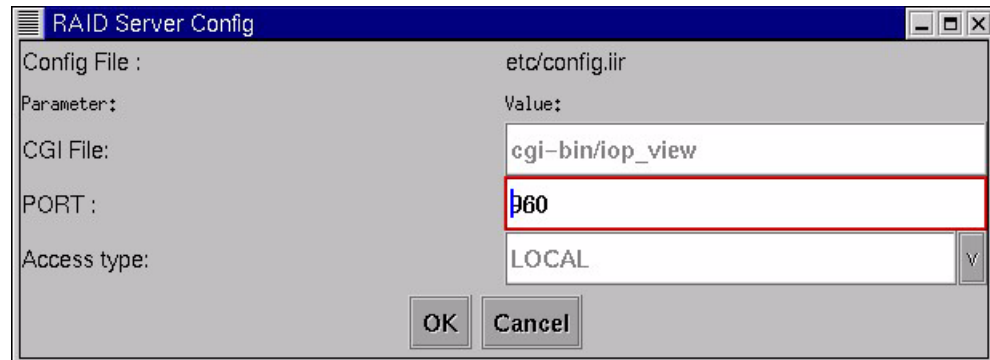
The following parameters are available:

CGI File: This parameter indicates the interface to the I/O processor and should not be changed.

PORT: This is the TCP/IP port used to access the CGI file. Default value is 960.

Access type: This parameter indicates whether or not RAID Storage Console can be accessed from a remote server. The default value of LOCAL. For remote access, change the value to REMOTE.

After any change to the RAID Configuration Server, restart the server with the following command:

```
/opt/iir/bin/iird restart
```

3. To manage user access of the RAID Configuration Services, click on the *User Management* button on the screen in Figure 1-9. The screen in Figure 1-11 appears.

Figure 1-11. RAID Configuration Services Administration - User Management
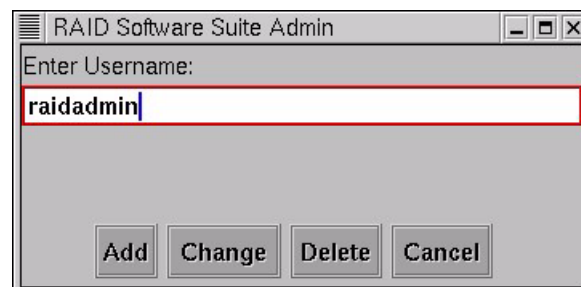
To add a new user, enter a new user name in the text field and click the button labeled *Add*. Enter a new password at the prompt. See Figure 1-12.

intel.

Figure 1-12. User Management Add User Screen

To change the password of an existing user, enter an existing username and click on *Change*. Complete the password prompts as directed. See Figure 1-13.

Figure 1-13. User Management Change Password

User Management changes do not require a restart of the RAID Configuration Services to take effect.

1.5.2   Configure a RAID Volume in a Linux System

Warning: See Section 1.7.1, "Linux Dynamic Block Device Limitations" on page 1-20. Understanding the information in this section is critical to completing the following procedures.

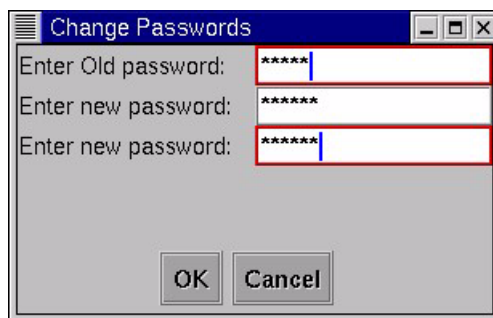1. Use a web browser and point to: http://localhost:960/ to create RAID volumes. Or click on the Raid Storage Console icon on your desktop.

   Use fdisk to create partitions on the RAID volumes using the following command:

   `$ fdisk /dev/i2o/hd` [a : first RAID volume - `p` : sixteenth volume created]

Note: All I$_2$O devices created are represented as "device files" on a Linux system. These device files are: `/dev/i2o/hda` to `/dev/i2o/hdp` for a total of 16 volumes that can be created. Individual partitions can be accesses (after creation using fdisk) as `/dev/i2o/hda1,/dev/i2o/hda2 … /dev/i2o/hdp4`

2. Make Linux native filesystem (ext2) using the following command:

> $ mke2fs /dev/i2o/hd [a-p][1-15: the partition number]

Note: Use `mkfs` to create any type of filesystem that is supported by your Linux system/ kernel.

3. Mount the partition on a directory (create one if necessary):

   $ mkdir /mnt/i2o

   $ mount /dev/i2o/hd [a-f][1-15] /mnt/i2o

   Check the RAID volume as a mounted directory with the following command:

   $ df -k

   The following output appears:

```
Filesystem      1k-blocks   Used      Available  Use%   Mounted on
/dev/hda2       2912980     2356260   408748     85%    /
/dev/i2o/hda1   13116500    20        12450188   0%     /mnt/i2o
[root@localhost /root]#
```
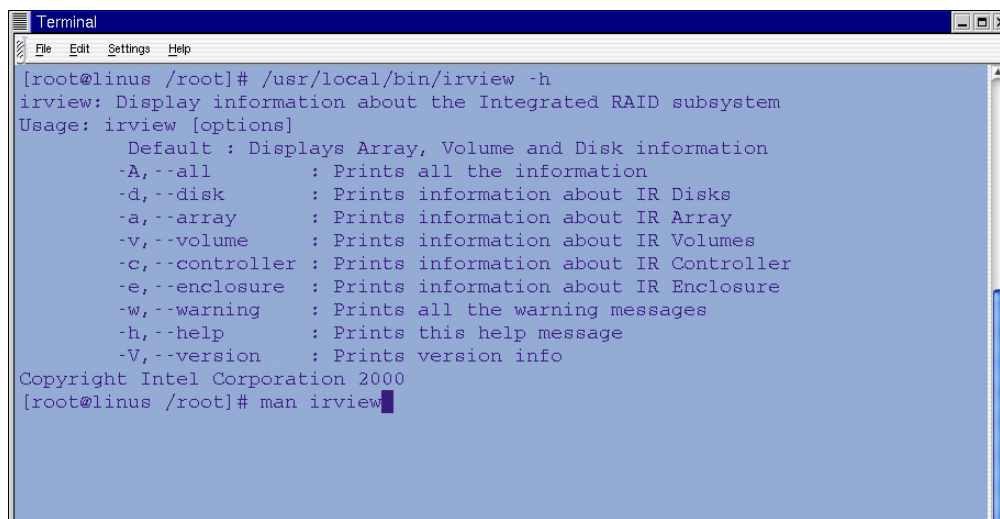
1.5.3   Monitor the Status of the RAID Subsystem

Monitor the status of the RAID subsystem by using the command line utility called `irview`. To invoke this utility, type `/usr/local/bin/irview` on the command line. To get more information on irview, invoke the corresponding man page (`man irview`).

A sample irview screen is shown in Figure 1-14.

Figure 1-14. irview Screen

1.5.4   Uninstalling RAID Software Suite

To uninstall the RAID Software Suite run the following:

`/opt/iir/bin/uninstall`

Note: The uninstall program will not remove the kernel sources or installed modules.

1.6   Booting to Red Hat Linux on an I$_2$O RAID Volume

There are two methods available for booting Red Hat Linux from an I$_2$O RAID Controller. The first method installs Red Hat Linux on an IDE or SCSI hard drive and loads LILO (the Linux Loader) from the Master Boot Record (MBR) on that original drive (`/dev/hda`), which then loads the Linux kernel which is located on a disk or volume connected to the I$_2$O RAID Controller. The second method installs LILO onto the MBR of the I$_2$O boot device (`/dev/i2o/hda`) and allows the system to boot directly from a disk or volume connected to the I$_2$O RAID Controller.

Warning: See Section 1.7.1, "Linux Dynamic Block Device Limitations" on page 1-20. Understanding the information in this section is critical to completing the following procedures.

1.6.1   Method 1 - MBR on Primary IDE

Method 1 assumes that you have installed Red Hat Linux 6.x on the primary IDE drive (`/dev/hda`). Although an IDE drive is assumed for these instructions, Method 1 could be used when Red Hat Linux is installed on a SCSI hard drive. For example, the boot device file might change depending on the installation.

Once you have completed the Red Hat installation, install the I$_2$O kernel as described in Section 1.3.2, "Rebuild the Kernel with I$_2$O Support" on page 1-9. After the RAID Software Suite installation is completed and the system is rebooted with I$_2$O support, follow the steps below to boot your system from the I$_2$O drive.

1. Create a primary I$_2$O volume that is large enough to accommodate your Red Hat installation.

2. Create a `/dev/i2o/hda1` partition on the primary I$_2$O volume. Ensure that the partition is equal to or greater than the partition `/dev/hda1` (it is assumed that your Redhat System is installed on `/dev/hda1`). If your volume is > 8 GB, you must ensure that the partition fits within the first 1024 cylinders of the disk.

3. A modified version of LILO is provided on the RAID Software Suite CD-ROM at `/mnt/cdrom/iir_install/lilo/lilo-i2o`. Copy it as `/sbin/lilo-i2o` and make it executable.

4. Copy the contents of the system drive to the new I$_2$O partition:

   `$ dd if=/dev/hda1 of=/dev/i2o/hda1`

5. Mount the I$_2$O partition on a file system mount point:

```
$ mount /dev/i2o/hda1 /mnt/i2o -t ext2
```

6. Edit `/etc/lilo.conf` and add the following boot entry:

```
image=/mnt/i2o/boot/vmlinuz-i2o-version # Put your i2o kernel
version here
label=boot-i2o
root=/dev/i2o/hda1
read-only
```

7. Edit `/mnt/i2o/etc/fstab`. Change the following line:

```
/dev/hda1 / ext2 defaults 1 1
```

to read:

```
/dev/i2o/hda1 / ext2 defaults 1 1
```

The above line will cause the kernel to mount `/dev/i2o/hda1` as the root file system.

8. Run LILO with I₂O support using this command:

```
$ /sbin/lilo-i2o
```

9. Reboot

1.6.2   Method 2 - MBR on I₂O RAID Volume

Method 2 assumes that you have installed Red Hat Linux 6.x on the primary IDE drive (`/dev/hda`). Copy the Linux kernal, LILO, and the MBR to a RAID volume that is connected to the I₂O RAID controller. To boot from the RAID volume, a modified version of the LILO binary is required. This modified version ignores the existence of IDE drives on the system.

Note: This boot method differs from the way that Linux and LILO are designed to function. The LILO boot loader provided in the 2.2.x kernel has a limitation that does not allow the system to boot from a non-IDE drive if an IDE drive exists. The modified LILO bypasses this requirement by ignoring the fact that there maybe devices other than the I₂O device on the system. However if you have other IDE devices on your system, you may experience problems with those devices when using method 2.

Once you have completed the Red Hat installation, install the I₂O kernel as described in Section 1.3.2, "Rebuild the Kernel with I₂O Support" on page 1-9. After RAID Software Suite installation is completed and the system is rebooted with I₂O support, follow the steps below to boot your system from the I₂O drive.

1. Create a primary I₂O volume that is large enough to accommodate your Red Hat installation.

2. Create a `/dev/i2o/hda1` partition on the primary I₂O volume. Ensure that the partition is equal to or greater than the partition `/dev/hda1` (it is assumed that your Redhat System is installed on `/dev/hda1`). If your volume is > 8 GB, you must ensure that the partition fits within the first 1024 cylinders of the disk.

3. A modified version of LILO is provided on the RAID Software Suite CD-ROM at `/mnt/cdrom/iir_install/lilo/lilo-i2o-hack`. Copy it as `/sbin/lilo-i2o-hack` and make it executable.

4. Copy the contents of the system drive to the new I$_2$O partition:

```
$ dd if=/dev/hda1 of=/dev/i2o/hda1
```

5. Mount the I$_2$O partition on a file system mount point:

```
$ mount /dev/i2o/hda1 /mnt/i2o -t ext2
```

6. Edit `/mnt/i2o/etc/lilo.conf` to read as follows:

```
boot=/dev/i2o/hda
map=/mnt/i2o/boot/map
install=/mnt/i2o/boot/boot.b
prompt
timeout=20
image=/mnt/i2o/boot/vmlinuz-i2o-version # Your kernel w/I2O
support
label=linux
root=/dev/i2o/hda1
read-only
```

7. Edit `/mnt/i2o/etc/fstab.` Change the following line:

```
/dev/hda1 / ext2 defaults 1 1
```

to read:

```
/dev/i2o/hda1 / ext2 defaults 1 1
```

The above line will cause the kernel to mount `/dev/i2o/hda1` as the root file system.

8. Change your swap drive from the IDE drive partition (assumed to be `/dev/hda5`) to the I$_2$O RAID volume by changing the following line in `/mnt/i2o/etc/fstab`:

```
/dev/hda5 swap swap defaults 0 0
```

to read:

```
/dev/i2o/hda2 swap swap defaults 0 0
```

9. Run the modified LILO binary with the LILO configuration on the I$_2$O RAID volume:

```
$ /sbin/lilo-i2o-hack -C .mnt/i2o/etc/lilo.conf
```

10. Shutdown the system:

```
$ shutdown -h now
```

11. Remove the IDE drive and restart the system.

12. Once the system has powered up, clean up `/etc/lilo.conf` by changing the map point reference as indicated below. This will make future updates of the kernel on the I$_2$O device easier.

```
boot=/dev/i2o/hda
map=/mnt/i2o/boot/map
install=/mnt/i2o/boot/boot.b
```

```
prompt
timeout=20
image=/mnt/i2o/boot/vmlinuz-i2o-version # Your kernel w/I2O
support
label=linux
root=/dev/i2o/hda1
read-only
```

In the future, if you edit `lilo.conf`, run LILO w/I_2O support (lilo-i2o) instead of the modified LILO (i.e lilo-i2o-hack).

1.7 Technical Notes

1.7.1 Linux Dynamic Block Device Limitations

The RAID volumes created using the I_2O Linux drivers are seen by Linux as "Block Devices". The RAID Software Suite gives a powerful new feature to create new block devices while a Linux system is up and running via the use of the Storage Console. The new RAID volume created is detected as /dev/i2o/hd* and can be immediately used as a normal block storage device. This block device was dynamically created while the system was up and running and is associated to a particular device file (/dev/i2o/hd*) and a Major/Minor number, however, Linux does not associate a physical storage device with the device file (or rather the Major/Minor) number across reboots.

The implication of this limitation is that when a reboot occurs after a user has dynamically created a dynamic block device, the device files associated with the physical storage device *changes* once the system comes back online. If there were entries in the /etc/fstab and /etc/mtab associated with these dynamc block devices, then they will no longer be valid and the system will either incorrectly mount these devices or it may not mount them at all. Your system may not boot if the root file system was also on a dynamic block device.

Workarounds

- Create all the RAID volumes you require and reboot before you designate persistent mount points for them. Take this approach especially if you are planning to mount the root file system on a RAID volume. Avoid creating additional RAID volumes after designating one of the existing volumes as the root file system.

- Remove any existing entries in /etc/fstab pertaining to I_2O devices. Create a filesystem on the new RAID volume and reboot. When the system has rebooted, examine the I_2O block devices to determine the mapping of the physical I_2O block devices to /dev/i2o/hd* device files. Mount the I_2O block devices onto the desired mount points and enter the corresponding entries in the /etc/fstab config file. If you built the I_2O support as kernel modules, then a reboot is not necessary. Simply unload and reload the I_2O modules.

intel

1.8 Miscellaneous Notes

- The `/opt/iir` directory requires a minimum of read, write, and execute privileges for `root`.