



Linux® OS Version 1.3 Release Notes

May 13, 2003

0898003-1.3a

**READ ME BEFORE
INSTALLING
THIS PRODUCT**



Disclaimer

The information contained in this document is subject to change without notice. Concurrent Computer Corporation has taken efforts to remove errors from this document, however, Concurrent Computer Corporation's only liability regarding errors that may still exist is to correct said errors upon their being made known to Concurrent Computer Corporation.

License

Duplication of this manual without the written consent of Concurrent Computer Corporation is prohibited. Any copy of this manual reproduced with permission must include the Concurrent Computer Corporation copyright notice.

Trademark Acknowledgments

Linux is a registered trademark of Linus Torvalds.

Red Hat is a registered trademark of Red Hat, Inc.

Night Hawk is a registered trademark of Concurrent Computer Corporation.

RedHawk, iHawk, NightStar, NightView, NightSim, NightProbe, NightTune, NightGraphics, MAXAda, NightBench and ARMS are trademarks of Concurrent Computer Corporation.

POSIX is a registered trademark of the Institute of Electronics and Electrical Engineers.

The X Window System is a trademark of The Open Group.

OSF/Motif is a registered trademark of the Open Software Foundation, Inc.

Ethernet is a trademark of the Xerox Corporation.

PowerEdge is a trademark of Dell Computer Corporation.

All other trademarks are the property of their respective owners.

© 2003 Concurrent Computer Corporation – All Rights Reserved

Concurrent Computer Corporation

2881 Gateway Drive

Pompano Beach, FL 33069

Note: Information subject to change without notice.

Contents

1.0. Introduction	1
2.0. Documentation	2
2.1. RedHawk Linux OS Software Documentation ..	2
2.2. Vendor Documentation	2
3.0. Prerequisites	3
3.1. Software	3
3.2. Hardware	3
3.2.1. iHawk Series 860 system	3
3.2.2. iHawk Series 860G system	3
3.2.3. Real-Time Clock and Interrupt Module (RCIM) ..	3
4.0. Release 1.3 Installation Procedures	4
4.1. Installing Red Hat Software	4
4.2. Updating Red Hat Software	7
4.3. Installing RedHawk Software	8
4.4. Download and Install RedHawk Updates	8
4.5. Installing the RCIM	9
4.5.1. Quick Hardware Installation Checklist	9
4.5.2. Check RCIM Operation	10
4.6. Installing Frequency-Based Scheduler (FBS) Software	10
4.7. Download and Install FBS Updates	11
5.0. Upgrading from a Previous Release	12
5.1. Upgrade to Red Hat 8.0	12
5.2. Updating Red Hat Software	13
5.3. Upgrade to RedHawk Linux OS 1.3	14
5.4. Download and Install RedHawk Updates	15
5.5. Upgrade to FBS 1.3	15
5.6. Download and Install FBS Updates	16
6.0. Updating NVIDIA Quadro Drivers	17
7.0. Additional Considerations	20
8.0. Information on Release 1.3	21
8.1. General	21
8.2. RedHawk Linux Kernels	21
8.3. Red Hat Version	22
8.4. Concurrent Real-Time Features in RedHawk Linux	22
8.5. Open Source Real-Time Features	26
8.6. NightStar Development Tools	27
9.0. Changes in this Release	28
9.1. Enhancements	28
9.1.1. Real-Time Hyper-threading	28
9.1.2. POSIX Message Queues	28
9.1.3. usermap(3) Support	28
9.1.4. User-level Spin Locks	29
9.1.5. Additional Real-Time Scheduling Priorities Available	29

- 9.1.6. Daemons for Deferred Interrupt Processing 29
- 9.1.7. ioctl Flag for Big Kernel Lock (BKL) 30
- 9.1.8. XFS File System 30
- 9.1.9. New RAID Controller Support 30
- 9.2. Modifications 31
 - 9.2.1. CPU Identification with Hyper-threading 31
 - 9.2.2. POSIX Timers 32
 - 9.2.3. Write Access to RCIM Device Files 33
 - 9.2.4. IRQ-Blocking Spinlock Services Made
Preemptive 33
 - 9.2.5. High Resolution Process Timing Facility Changes 33
- 10.0. Known Issues 35
- 11.0. Using the Linux Kernel Crash Dump (LKCD) 37
 - 11.1. Documentation 37
 - 11.2. Installation/Configuration Details 37
 - 11.3. Forcing a Crash Dump on a Hung System 38
- 12.0. Useful Information 39
 - 12.1. Man Pages 39
 - 12.2. RCIM Connection Modes 39
 - 12.3. File System Swap Size Guidelines 39
- 13.0. Software Updates and Support 40
 - 13.1. How to Download and Install Updates 40
- 14.0. Direct Software Support 41

1.0. Introduction

This document provides a general overview of RedHawk Linux OS Release Version 1.3. This release is designed to provide support for the Concurrent iHawk™ Series 860 and 860G systems.

The iHawk™ Series 860 is Concurrent Computer Corporation's new high-performance Intel-based multiprocessor platform for real-time data acquisition, simulation, and industrial systems applications. The iHawk 860 features from one to eight Intel® Pentium® Xeon™ processors in a single rackmount or tower enclosure.

The iHawk 860G is a high-performance, AGP/PCI-based platform for real-time imaging applications. It features one or two Intel Pentium Xeon processors and dual-monitor capable AGP 8X port in a tower enclosure.

The iHawk 860 family of systems offer leading-edge integrated circuit and packaging technology. They are true symmetric multiprocessors (SMP) that run a single copy of Concurrent's RedHawk™ Linux® real-time operating system. All CPUs in a system are linked by a high-speed front-side processor bus and have direct, cache-coherent access to all of main memory.

Concurrent's RedHawk Linux operating system, based on the popular Red Hat® Linux distribution, features high I/O throughput, fast response to external events, and optimized interprocess communication. RedHawk is the ideal Linux environment for complex real-time applications. RedHawk is based upon a multithreaded, fully preemptible Linux kernel with low-latency enhancements. RedHawk's true symmetric multiprocessing support includes load-balancing and CPU shielding to maximize determinism and real-time performance in mission-critical solutions.

RedHawk Linux supports Concurrent's powerful set of NightStar™ development tools. Users can schedule, monitor, debug and analyze their real-time applications on iHawk multiprocessor systems or remotely from a desktop PC or laptop. Each tool runs on the iHawk target system non-intrusively, thus preserving the deterministic characteristics of the real-time application.

RedHawk Linux Release Version 1.3 consists of the following CDs:

Label on Disc		Qty
Red Hat 8.0 Update CD for RedHawk Linux	LXUPDATE Version 1.3 WA9086-J	1
RedHawk Linux OS	WA9086-J Version 1.3	1
RedHawk Linux Documentation CD	LXDOKIT Version 1.3 WA9086-J	1

The following optional CDs may also be provided:

Label on Disc		Qty
RedHawk Linux NightStar Tools	WU1220-LDV NightStar Tools Version <i>x.x</i> (where <i>x.x</i> = latest version)	1
RedHawk Linux Frequency-Based Scheduler	WU1210-JL Version 1.3	1

In addition to the above, CDs for the official Red Hat Linux 8.0 distribution are provided. This distribution consists of multiple discs.

Warning

While much of the software included on the RedHawk Linux CDs is licensed under the GNU GPL license, there are some components that are not released under this license. Therefore, these CDs cannot be freely copied.

2.0. Documentation

2.1 RedHawk Linux OS Software Documentation

Table 2-1 lists the documentation provided by Concurrent as pdf files on the RedHawk Linux documentation CD.

Note that printed copies of Concurrent documentation can be ordered by contacting the Concurrent Software Support Center. The toll-free number for calls within the continental United States is 1-800-245-6453. For calls outside the continental United States, the number is 1-954-971-6248. If available, a CD containing the pdf of selected manuals will be provided with the system.

Table 2-1 RedHawk Linux OS Software Documentation

Manual Name	Document Number
<i>RedHawk Linux OS Version 1.3 Release Notes</i>	0898003-1.3a
<i>RedHawk Linux User's Guide</i>	0898004-300
<i>Frequency-Based Scheduler (FBS) User's Guide</i>	0898005-100
<i>Real-Time Clock & Interrupt Module (RCIM) PCI Form Factor User's Guide</i>	0898007-200

2.2 Vendor Documentation

Red Hat Linux documentation is available on the CDs provided with Red Hat.

Commercial off-the-shelf (COTS) documentation applicable to the iHawk Series 860 systems is listed in Table 2-2. Contact your local Dell sales office to purchase additional copies.

Note

Actual manuals shipped, document numbers and revision numbers are subject to change without notice.

Table 2-2 Vendor Documentation

Manual Name	Document Number
System Information	01F780800 Rev A00
Information Update	2K180 Rev A01
User's Reference, Adaptec SCSI Card 2940U2W	DP/N 077TWJ

3.0. Prerequisites

3.1 Software

- Red Hat Linux Release 8.0
- RedHawk Linux OS Release Version 1.3

3.2 Hardware

3.2.1 iHawk Series 860 system

- Tightly-coupled symmetric multiprocessing (SMP) systems
- One to eight Intel Pentium Xeon processors
- Up to 2 MB of cache per CPU
- 128 MB to 4 GB of memory per system
- Tower or rackmount chassis with disk, tape, and CD
- 10/100/1000BaseT Ethernet
- Ultra320 SCSI
- Real-Time Clock & Interrupt Module (RCIM)

3.2.2 iHawk Series 860G system

- AGP 8X symmetric multiprocessing (SMP) systems
- Dual-monitor capable
- One or two Intel Pentium Xeon processors
- 512 KB of cache per CPU
- 128 MB to 4 GB of memory per system
- Tower chassis with disk, tape, and CD
- 10/100/1000BaseT Ethernet
- Ultra160 SCSI
- Real-Time Clock & Interrupt Module (RCIM)

3.2.3 Real-Time Clock and Interrupt Module (RCIM)

The RCIM is a multi-function PCI card that provides functions ideally suited for time-critical applications. The RCIM can be installed on any system with an available PCI slot. Refer to section 4.5 for installation instructions.

4.0. Release 1.3 Installation Procedures

iHawk systems are normally shipped fully installed. The installation procedures provided below need only be used in the event a system must be regenerated. If you have a previous release of RedHawk installed and want to upgrade to Release 1.3, refer to *Upgrading from a Previous Release* on page 12 for detailed instructions.

Review the following points before installing the software:

- It is recommended that the installer first read through the entire installation procedure once before actually performing the installation.
- It is **strongly recommended** that installation and updates of software be performed in single-user mode to guard against the possibility of corrupting a running application.
- The installation media in use is accessed at various times and must be kept in the drive during the installation until you are instructed to install different media.
- If you plan to configure networking using fixed numbers (recommended), you will need to know the machine's IP address, netmask address, gateway address, and primary and secondary DNS addresses to do so. In addition, the system's main memory size needs to be known in order to properly set up the file system swap size. If not sure of main memory size, you can depress the 'Setup' key that is specified during the power up sequence and view the contents of the system's BIOS to ascertain the size of main memory.
- Prior to installation, remove any floppy media that may have been left in the machine.
- Disc install times specified below may vary and are only provided as a guideline.

4.1 Installing Red Hat Software

Complete the following steps to install and configure Red Hat on the system. Refer to Online Help if needed.

1. Power on the system, if needed, to open the CD-ROM drawer.
2. Insert Red Hat 8.0 disc 1 of 3 in the CD-ROM drive and power down the system.
3. Power on the system with the Red Hat 8.0 disc 1 of 3 installed in order to boot from the disc. A full screen of choices is displayed. Only one item can be chosen. At the **boot:** prompt, choose "To Install or upgrade Red Hat Linux in graphical mode," then click ENTER.

You have the option to test the Red Hat media for errors. If you wish to do so, click OK, otherwise, select **Skip** and then click ENTER.

- At **Welcome** sub-menu, click **Next**.
- At **Language Selection** sub-menu:
Choose English language. Click **Next**.
- At **Keyboard** sub-menu:
Accept keyboard defaults. Click **Next**.

- At Mouse Configuration sub-menu:
Select appropriate mouse model. **Note:** click “Emulate 3 buttons” box at bottom of page if using a 2 button mouse. Click **Next**.
- At Installation Type sub-menu:
Choose “Custom” install. Click **Next**.
- At Disk Partition Setup sub-menu:
Choose “Manually Partition with Disk Druid” and click **Next** to partition the disk manually and create the partition sizes as shown in Table 4-1 below. Note that this table shows recommended partition sizes for a 36 GB disk or larger.

Note

In the next step, all partitions must be placed on disk `/dev/sda`. If the system has multiple drives, you must un-select all other drives as shown in the “Allowable Drives:” section of the Add Partition setup menu.

- At Partitioning sub-menu, first delete all existing partitions by highlighting each line and clicking **DELETE**. This must be done one partition at a time. To add partitions, highlight the disk drive and click **NEW**. Select the Force to be primary partition and Check for bad blocks boxes at the bottom of each partition setup page. Click **OK** to continue to set up the next partition.

Table 4-1 Recommended Disk Partition Sizes

Mount Point	Type	Filesystem Type	Partition Size
<code>/boot</code>	primary	ext3	100 MB
<code>/</code>	primary	ext3	16384 MB (16 GB)
<code>swap</code> (See Note below)	primary	swap	Size of system’s main memory plus 5 percent additional. (See Table 12-1 on page 39 for additional information)
<code>/home</code>	primary	ext3	Fill to maximum allowable size
Note: On menu page, select File System Type to open sub-menu for selection of <code>swap</code> file system. Enter recommended size.			

- Click **Next** when finished with partitioning.
- At Boot Loader Configuration sub-menu:
Accept GRUB boot loader defaults. Enter grub password if desired (not recommended). Select “Configure advanced boot loader options,” then click **Next**.
- At Advanced Boot Loader Configuration sub-menu:
At “General kernel parameters,” enter “`ide=nodma`.” Click **Next**.

- At **Network Configuration** sub-menu:
Configure networking as appropriate (fixed network numbers recommended - you will need address information if using fixed numbers). Refer to Online Help for additional instructions. Click **Next** after completion.
 - At **Firewall Configuration** sub-menu:
Choose “No Firewall”, then click **Next**.
 - At **Additional Language Support** sub-menu:
Choose any additional languages as necessary. Click **Next** at completion.
 - At **Time Zone Selection** sub-menu:
Select appropriate time zone, then click **Next**.
 - At **Account Configuration** sub-menu:
Enter root password, then click **Next**.
 - At **Authentication Configuration** sub-menu:
Accept defaults, click **Next**.
 - At **Package Group Selection** sub-menu:
Scroll down to bottom of list and select “Everything.” Click **Next**.
 - At **About to install** page, click **Next**.
At this point in the installation, the disk will format and packages on Red Hat 8.0 disc #1 will be installed. After approximately 20 minutes, you will be prompted to insert disc #2. Remove disc #1 and insert disc #2, then click **OK**. After another 20 minutes elapses (approximately) you will be prompted to insert disc #3. Remove disc #2 and insert disc #3, then click **OK**.
 - After approximately 15 minutes, you should get a pop-up mini-page stating the system is “Performing post installation configuration”.
 - At **Boot Disk Creation** sub-menu:
Choose “Yes, I would like to create a boot disk”. Click **Next**. You will be prompted to insert a diskette. Insert diskette, and click “Make boot disk”. System responds with “Creating boot disk...”.
 - At **Graphical Interface (X) Configuration** sub-menu:
Accept defaults, click **Next**.
 - At **Monitor Configuration** sub-menu:
Accept monitor defaults, click **Next**.
 - At **Customize Graphical Configuration** sub-menu:
Choose “Graphical” login, click **Next**.
4. At the next window prompt, click **Exit** to reboot the system. Verify that Red Hat boots up properly.
 5. Remove any media from respective drives and store.

6. Once the system boots up successfully, proceed as follows:
 - At **Welcome** sub-menu click **Forward** to continue.
 - At **Date and Time Configuration** sub-menu:
Set the appropriate date and time and click **Forward** to continue.
 - At **Red Hat Update Agent** sub-menu:
Select 'No, I do not want to register my system.' Click **Forward** to continue.
 - At **Install Additional Software** sub-menu:
Three choices should appear. Read the instructions before proceeding. If you have any of the CDs referred to, you may install them now.
 - At **Finished Setup!** sub-menu:
Click **Forward** to continue.
7. Installation is now complete. Login window should appear at this point. Continue with the steps in section 4.2 to install Red Hat software updates. Remove any media from respective drives.

4.2 Updating Red Hat Software

To update the installed Red Hat software, perform the following steps.

1. Log in as root and take the system down to single-user mode:
 - a. Right click on the desktop and select **New Terminal**.
 - b. At the system prompt, type **init 1**.
2. Locate the disc labelled "Red Hat 8.0 Update CD for RedHawk Linux" and insert into the CD-ROM drive.
3. If the cdrom device does not mount automatically, execute the following command:

```
mount /mnt/cdrom
```

4. Execute the following commands to install the update software:

```
cd /mnt/cdrom  
./install-updates
```

Follow the on-screen instructions until the installation script completes. This will take some time to complete.

5. Upon completion of software installation, execute the following commands:

```
cd /  
umount /mnt/cdrom  
eject
```

6. Remove disc from the CD-ROM drive and store.

7. **Perform this step ONLY if you are installing on a single processor system.** If installing on a multiprocessor system, skip to the next step.

Execute the following:

```
rpm -e kernel-smp
```

8. Exit single-user mode (^D), reboot and verify that the new kernel boots.

4.3 Installing RedHawk Software

To install the RedHawk Linux software, perform the following steps:

1. Log in as root and take the system down to single-user mode:
 - a. Right click on the desktop and select New Terminal.
 - b. At the system prompt, type **init 1**.
2. Locate the disc labelled “RedHawk Linux OS Version 1.3” and insert into the CD-ROM drive.
3. If the cdrom device does not mount automatically, execute the following command:

```
mount /mnt/cdrom
```

4. Execute the following commands to install RedHawk Linux:

```
cd /mnt/cdrom  
./install-redhawk
```

Follow the on-screen instructions until the installation script completes.

5. Upon completion of software installation, execute the following commands:

```
cd /  
umount /mnt/cdrom  
eject
```

6. Remove the disc from the CD-ROM drive and store. Exit single-user mode (^D). Check for RedHawk updates as outlined in the following section.

4.4 Download and Install RedHawk Updates

If there are any updates to this release available for downloading from Concurrent’s web site, they should be installed now. Follow these steps:

1. In multi-user mode, access the **redhawk.ccur.com** web site.
2. Log onto the site using the login and password provided by Concurrent.
3. A list of packages that you have not yet downloaded will display. Click on the packages to download them and save them in a suitable location (e.g. **/root/updates**).

4. Return to single-user mode as root.
5. To install the newly downloaded packages, change directory to the location of the updates and issue the following command:

```
rpm -Fvh * .rpm
```

The command should complete successfully, though it may take several minutes depending on the number of updates being installed.

6. Exit single-user mode (^D), reboot the system choosing the new RedHawk kernel and verify that the system boots properly.

4.5 Installing the RCIM

The following serves as an installation checklist for installing a PCI RCIM in an iHawk system. See the *Real-Time Clock and Interrupt Module PCI Form Factor User's Guide* for complete details.

4.5.1 Quick Hardware Installation Checklist

1. Before installing the RCIM, decide what mode the RCIM will run in (see *RCIM Connection Modes* on page 39) and if you will be using the RCIM to accept external interrupts.
2. Verify that all pre-requisite software has been installed.
3. Verify that "ccur-rcim rpm" has been installed. To verify, execute the command `rpm -q ccur-rcim`. The system will inform you if the ccur-rcim rpm is not installed.
4. Power down the system and remove all power cords.
5. Open the case and identify a PCI slot for installation. (Refer to the chart on the case for help in identifying card slot types.)

Note

Concurrent Computer Corporation strongly recommends that you use an antistatic wrist strap and a conductive foam pad when installing and removing printed circuit boards.

6. Insert the RCIM into the selected PCI slot and lock it down.
7. Attach data chaining input cable if required (see the RCIM manual for complete details).
8. Close the case and reconnect all power cords.
9. Power up the system and verify the system boots up properly. Refer to the *Real-Time Clock and Interrupt Module PCI Form Factor User's Guide* for configuration options.

4.5.2 Check RCIM Operation

1. To check that the RCIM is operating correctly, perform the following:

```
cat /proc/driver/rcim/status
```

2. You should see an output similar to that shown below:

```
Rcim board revision 1.  
This is a standalone (isolated) rcim.
```

3. The following is an example output after issuing the “**cat /proc/driver/rcim/config**” command:

```
$cat config  
h/Not_Configured, sync, pig0|out0, pig1|out1, pig2|  
out2, pig3|out3, none|di0, none|di1, none|di2, none|  
di3, none|di4, none|di5, none|di6, none|di7, di0/f, i  
1/f, di2/f, di3/f, di4/f, di5/f, di6/f, di7/f, eti0/f,  
eti1/f, eti2/f, eti3/f
```

4.6 Installing Frequency-Based Scheduler (FBS) Software

To install the FBS software, perform the following steps:

1. Log in as root and take the system down to single-user mode:
 - a. Right click on the desktop and select New Terminal.
 - b. At the system prompt, type **init 1**.
2. Locate the disc labelled “RedHawk Linux Frequency-Based Scheduler Version 1.3” and insert into the CD-ROM drive.
3. To mount cdrom device, execute the following command:

```
mount /mnt/cdrom
```

4. To install, execute the following commands:

```
cd /mnt/cdrom  
./install-fbs
```

Follow the on-screen instructions until the installation script completes.

5. Upon completion of software installation, execute the following commands:

```
cd /  
umount /mnt/cdrom  
eject
```

6. Remove the disc from the CD-ROM drive and store. Exit single-user mode (^D). Check for FBS updates as outlined in the following section.

4.7 Download and Install FBS Updates

If there are any updates to this FBS release available for downloading from Concurrent's web site, they should be installed now. Follow these steps:

1. In multi-user mode, access the **redhawk.ccur.com** web site.
2. Log onto the site using the login and password provided by Concurrent.
3. A list of packages that you have not yet downloaded will display. Click on the packages to download them and save them in a suitable location (e.g. **/root/updates**).
4. Return to single-user mode as root.
5. To install the newly downloaded packages, change directory to the location of the updates and issue the following command:

```
rpm -Fvh *.rpm
```

The command should complete successfully, though it may take several minutes depending on the number of updates being installed.

6. Exit single-user mode (^D), reboot the system choosing the new RedHawk kernel and verify that the system boots properly.

5.0. Upgrading from a Previous Release

This section contains the instructions for upgrading to RedHawk Linux Release 1.3 on a system currently running a previous release version of RedHawk Linux.

Review the following points before upgrading the software:

- It is recommended that the installer first read through the entire procedure once before actually performing the upgrade.
- It is **strongly recommended** that installation and updates of software be performed in single-user mode to guard against the possibility of corrupting a running application.
- The media in use is accessed at various times and must be kept in the drive during the installation until you are instructed to install different media.
- Prior to upgrade, remove any floppy media that may have been left in the machine.
- Disc upgrade times specified below may vary and are only provided as a guideline.
- The Red Hat kernel should be running when upgrading to RedHawk Linux Release 1.3.

If you are upgrading from RedHawk Linux Release 1.2, proceed directly to Section 5.2, *Updating Red Hat Software*.

If you are upgrading from RedHawk Linux Release 1.1, follow all directions in this section beginning with Section 5.1 to upgrade to RedHawk Linux 1.3. This is necessary because RedHawk Linux 1.1 systems were built using the Red Hat 8.0 Beta distribution, and it is necessary to first upgrade to the official Red Hat 8.0 distribution before upgrading to RedHawk Linux 1.3. Upgrading Red Hat Linux 8.0 Beta will not delete any existing data (however, back-up is recommended). The installation program updates the modular kernel and all currently installed software packages.

5.1 Upgrade to Red Hat 8.0

Complete the following steps to install and configure Red Hat on the system.

Refer to the following items (in order) for answers to installation questions and configuration options. Also refer to the Online Help for additional instructions.

1. Power on the system, if needed, to open the CD-ROM drawer.
2. Insert Red Hat 8.0 disc 1 of 3 in the CD-ROM drive and power down the system.
3. Power on the system with the disc 1 of 3 installed in order to boot from the disc. A full screen of choices is displayed. Only one item can be chosen. At the **boot:** prompt, choose "To Install or upgrade Red Hat Linux in graphical mode," then press ENTER.

You have the option to test the Red Hat media for errors. If you wish to do so, click OK, otherwise, select Skip and then click ENTER.

- At Welcome sub-menu, click Next.
- At Language Selection sub-menu:
Choose English language. Click Next.

- At **Keyboard** sub-menu:
Accept keyboard defaults. Click **Next**.
 - At **Mouse Configuration** sub-menu:
Select appropriate mouse model. Note, click “Emulate 3 buttons” box at bottom of page if using a 2 button mouse. Click **Next**.
 - At **Installation Type** sub-menu:
Select **Upgrade Existing System** button, then click **Next**.
 - At **Upgrade Examine** sub-menu:
Click **Next** to perform automated upgrade of your packages.
 - At **Upgrade Boot Loader Configuration** sub-menu:
Select “Update boot loader configuration.” click **Next**.
 - At **About to Upgrade** sub-menu:
Click **Next** to continue.
 - At **Installing Packages** sub-menu:
At this point no operator intervention is required. Packages are being upgraded. Time to install depends on how many packages need to be installed. Remove previous disc and insert new disc when requested to do so.
 - At **Boot Disk Creation** sub-menu:
Choose “Yes, I would like to create a boot disk”. Click **Next**.
You will be prompted to insert a diskette. Insert diskette, and click **Next** to continue. System responds with “Creating boot disk...”.
4. Installation is complete. Click **Exit** to reboot the system. Verify that Red Hat boots up properly.
 5. Continue with the steps in section 5.2 to install Red Hat software updates. Remove any media from respective drives.

5.2 Updating Red Hat Software

To update the installed Red Hat software, perform the following steps:

1. If not already running Red Hat, boot the system with the Red Hat kernel.
2. Log in as root and take the system down to single-user mode:
 - a. Right click on the desktop and select **New Terminal**.
 - b. At the system prompt, type **init 1**.
3. Locate the disc labelled “Red Hat 8.0 Update CD for RedHawk Linux Version 1.3” and insert into the CD-ROM drive.
4. If the cdrom device does not mount automatically, execute the following command:

```
mount /mnt/cdrom
```

5. Execute the following commands to install the update software:

```
cd /mnt/cdrom
./install-updates
```

Follow the on-screen instructions until the installation script completes. This will take some time to complete.

6. Upon completion of software installation, execute the following commands:

```
cd /
umount /mnt/cdrom
eject
```

7. Remove the disc from the CD-ROM drive and store.

8. **Perform this step ONLY if you are installing on a single processor system.** If installing on a multiprocessor system, skip to the next step.

Execute the following:

```
rpm -e kernel-smp
```

9. Exit single-user mode (^D). Reboot the system choosing the “Red Hat Linux (2.4.18-27.8.0)” kernel and verify that the new kernel boots.

5.3 Upgrade to RedHawk Linux OS 1.3

To install the RedHawk Linux 1.3 update software, perform the following steps:

1. With the above named Red Hat kernel running on the system, log in as root and take the system down to single-user mode:
 - a. Right click on the desktop and select New Terminal.
 - b. At the system prompt, type **init 1**.

2. Locate the disc labelled “RedHawk Linux OS Version 1.3” and insert into the CD-ROM drive.

3. If the cdrom device does not mount automatically, execute the following command:

```
mount /mnt/cdrom
```

4. Execute the following commands to install RedHawk Linux OS 1.3 upgrade software:

```
cd /mnt/cdrom
./install-redhawk
```

Follow the on-screen instructions until the installation script completes.

5. Upon completion of software installation, execute the following commands:

```
cd /
umount /mnt/cdrom
eject
```

6. Remove the disc from the CD-ROM drive and store. Exit single-user mode (^D). Check for RedHawk updates as outlined in the following section.

5.4 Download and Install RedHawk Updates

If there are any updates to this release available for downloading from Concurrent's web site, they should be installed now. Follow these steps:

1. In multi-user mode, access the **redhawk.ccur.com** web site.
2. Log onto the site using the login and password provided by Concurrent.
3. A list of packages that you have not yet downloaded will display. Click on the packages to download them and save them in a suitable location (e.g. **/root/updates**).
4. Return to single-user mode as root.
5. To install the newly downloaded packages, change directory to the location of the updates and issue the following command:

```
rpm -Fvh *.rpm
```

The command should complete successfully, though it may take several minutes depending on the number of updates being installed.

6. Exit single-user mode (^D), reboot the system choosing the new RedHawk kernel and verify that the system boots properly.

5.5 Upgrade to FBS 1.3

To install the FBS Version 1.3 update software, perform the following steps:

1. Log in as root and take the system down to single-user mode:
 - a. Right click on the desktop and select New Terminal.
 - b. At the system prompt, type **init 1**.
2. Locate the disc labelled "RedHawk Linux Frequency-Based Scheduler Version 1.3" and insert into the CD-ROM drive.
3. To mount cdrom device, execute the following command:

```
mount /mnt/cdrom
```

4. Execute the following commands to install the FBS software:

```
cd /mnt/cdrom  
./install-fbs
```

Follow the on-screen instructions until the installation script completes.

5. Upon completion of software installation, execute the following commands:

```
cd /  
umount /mnt/cdrom  
eject
```

6. Remove the disc from the CD-ROM drive and store. Exit single-user mode (^D). Check for FBS updates as outlined in the following section.

5.6 Download and Install FBS Updates

If there are any updates to this FBS release available for downloading from Concurrent's web site, they should be installed now. Follow these steps:

1. In multi-user mode, access the **redhawk.ccur.com** web site.
2. Log onto the site using the login and password provided by Concurrent.
3. A list of packages that you have not yet downloaded will display. Click on the packages to download them and save them in a suitable location (e.g. **/root/updates**).
4. Return to single-user mode as root.
5. To install the newly downloaded packages, change directory to the location of the updates and issue the following command:

```
rpm -Fvh *.rpm
```

The command should complete successfully, though it may take several minutes depending on the number of updates being installed.

6. Exit single-user mode (^D), reboot the system choosing the new RedHawk kernel and verify that the system boots properly.

6.0. Updating NVIDIA Quadro Drivers

Use the following procedure if you need to update or re-install the NVIDIA Quadro graphics drivers on your Concurrent iHawk 860G system.

Notes

The **.run** installation script now offered by NVIDIA is not compatible with RedHawk Linux; therefore, you must download the tar files from the ftp site as instructed here.

Driver files are updated regularly so the names may be slightly different than those used as examples in this procedure.

1. Login to the system as root.
2. If needed, create a directory to hold the driver; for example, **/usr/local/nvidia**.
3. Using your web browser, go to **ftp://download.nvidia.com** and make the following selections:

```
XFree86
Linux-x86
```

Then select the latest version in the list. The date (when details are displayed) and the **latest.txt** file both indicate which version is the latest offering; e.g., 1.0-4363.

4. From the list of files for the latest version, double-click on the following and save them into your designated nvidia directory.

```
NVIDIA_GLX-<version>.tar.gz
NVIDIA_kernel-<version>.tar.gz
```

The **README.txt** and other available documentation files can be viewed immediately by double-clicking on the item, or you can download them to the desired location by right-clicking and selecting the appropriate copy or save command.

5. Exit the web browser.
6. Make a backup copy of file **/etc/X11/XF86Config**:

```
cp /etc/X11/XF86Config /etc/X11/XF86Config.BAK
```
7. If you are running a custom kernel, skip the following sub-steps and proceed to step 8 instead.

If you are running a kernel supplied by Concurrent, follow the sub-steps outlined below to the prepare the kernel directory for building drivers to match the running kernel. Refer to the *RedHawk Linux User's Guide* for additional information, if needed, on building and configuring the kernel.

- a. Change directory (**cd**) to **/usr/src/<redhawk release directory>**.

- b. Run the `ccur-config` tool as follows:

```
./ccur-config -n -c
```

Wait for on-screen messages to complete. At the prompt, proceed to the next step.

8. To allow a recovery if there is a problem during the installation, you will need to set the system's default run level to enable the system to boot to console instead of starting up X.

- a. Using `vi` to edit `/etc/inittab`, locate the line reading:
`id:5:initdefault`

Change the 5 to 3, save and exit.

- b. Reboot the system with `init 3`.
- c. At the prompt, login as root.

9. Change directory (`cd`) to the directory where you downloaded the driver files (e.g., `/usr/local/nvidia`) and execute the following commands to unzip the tar files:

```
tar xvfz *GLX*.tar.gz
tar xvfz *kernel*.tar.gz
```

10. Change directory (`cd`) to the GLX directory (e.g., `NVIDIA_GLX-1.0-4363`) and execute the following command:

```
make install
```

11. Change directory (`cd`) to the kernel directory (e.g., `NVIDIA_kernel-1.0-4363`).

12. Using `vi`, edit `os-interface.c`. Search for 'jiffies' and immediately below the comment that contains 'jiffies' (but before the C source line that references jiffies), insert the line:

```
#undef jiffies
```

Save and exit the file.

13. Execute the following commands:

```
export CC=gcc296
make
```

14. Executing the command `lsmod` should now show the "NVdriver" driver installed.

15. Examine and modify as necessary your `/etc/X11/XF86Config` file as follows (you may refer to the NVIDIA README for more details):

In the section labeled “Module” comment out the following entries, if present:

```
Load "dri"
Load "GLcore"
```

In the same section, add the following entry (if not already there):

```
Load "glx"
```

In the section labeled “Device” change the entry for “Driver” to read:

```
Driver "nvidia"
```

Save and exit the file.

16. To test the success of the installation, run the following command:

```
startx
```

If successful, you will see the NVIDIA logo and then the desktop appears. Reset the system’s default run level back to run level 5 (X11 startup mode).

- a. Using `vi` to edit `/etc/inittab`, locate the line reading:

```
id:3:initdefault
```

 Change the 3 to 5, save and exit.
- b. Execute the command `init 6`.

If unsuccessful, the problem may be due to a misconfigured `XF86Config` file. Review the `/var/log/XFree86*.log` files along with the NVIDIA README document to debug your `/etc/X11/XF86Config` file.

17. If you want to test the 3D performance of your card, perform the following:

```
glxgears
```

This should bring up three gears rotating at high speed with frames per second displayed in the associated window.

If the setup is incorrect, you may get the following on-screen messages:

```
Xlib: extension "GLX" missing on display ":0.0".
Error: couldn't get an RGB, Double-buffered visual
```

Refer to the NVIDIA README document for troubleshooting tips.

7.0. Additional Considerations

If you plan to build a kernel from source residing on your iHawk system on a different system, that system must be using **gcc-2.96-98** or later.

8.0. Information on Release 1.3

8.1 General

The RedHawk Linux operating system is based on the Red Hat Linux distribution. A standard set of Red Hat Linux CDs is included with each iHawk system. Except for the kernel, all Red Hat components operate in their standard fashion. The unmodified Red Hat components that are included in the RedHawk Linux offering include the Linux utilities, libraries, compilers, tools and installer.

The components in the RedHawk distribution that are unique to RedHawk include a modified Linux kernel, some additional user-level libraries and commands and optionally, the NightStar real-time application development tools. The RedHawk Linux kernel is based on the kernel maintained by Linus Torvalds and utilizes the 2.4.21-pre4 version.

There are three classes of changes that have been applied to the Linux kernel:

- new kernel features added by Concurrent based on features in the real-time operating systems that Concurrent has been deploying for many years
- open source real-time patches that have not been accepted into standard Linux and which supply features or performance benefits for real-time applications
- performance improvements developed by Concurrent to improve worst-case process dispatch latency or to improve determinism in process execution

These kernel enhancements provide the support needed for developing complex real-time applications. The additional user-level libraries in RedHawk provide interfaces to some of the real-time features present in the Red Hat Linux kernel.

Linux conforms to many of the interface standards that are defined by POSIX. However, Linux does not completely conform to these standards. Red Hat Linux has the same level of POSIX conformance as other Linux distributions that are based on the 2.4 series of kernels. Linux on the Intel x86 architecture has defined a binary standard of its own which allows shrink-wrapped applications that are designed to run on the Linux/Intel x86 platform to run on Concurrent's iHawk platform.

8.2 RedHawk Linux Kernels

There are three flavors of RedHawk Linux kernels installed on the iHawk system. The system administrator can select which version of the RedHawk Linux kernel is loaded via the GRUB boot loader. The three flavors of RedHawk kernels available are:

Kernel Name	Kernel Description
vmlinuz-2.4.21-pre4-RedHawk-1.3-trace	Default kernel with trace points but no debug checks
vmlinuz-2.4.21-pre4-RedHawk-1.3-debug	Kernel with both debug checks and kernel trace points
vmlinuz-2.4.21-pre4-RedHawk-1.3	Kernel with no trace points and no debug checks

The default RedHawk Linux trace kernel has been built with kernel trace points enabled. The kernel trace points allow the NightTrace™ tool to trace kernel activity.

The debug kernel has been built with both debugging checks and kernel trace points enabled. The debugging checks are extra sanity checks that allow kernel problems to be detected earlier than they might otherwise be detected. However, these checks do produce extra overhead. If you are measuring performance metrics on your iHawk system, this activity would be best performed using a non-debug version of the kernel.

8.3 Red Hat Version

Most user-level components on RedHawk Linux are provided by the Red Hat Linux 8.0 distribution. The RedHawk Linux kernel replaces the standard Red Hat kernel. While the RedHawk Linux kernel is likely to work with any version of the Red Hat 8.0 distribution, it is not recommended that the user download additional Red Hat updates as these could potentially destabilize the system. With the RedHawk Linux release, there will be Red Hat updates made available to RedHawk Linux users as those updates are validated against the RedHawk Linux kernel.

8.4 Concurrent Real-Time Features in RedHawk Linux

Refer to Table 8-1 (next page) for a description of the Concurrent added real-time features available in the RedHawk Linux kernel.

Table 8-1 Concurrent Added Real-Time Features

Feature	Description
Processor Shielding	Enhancements to the scheduler and the <code>/proc</code> file system allow the user to dynamically shield a set of processors from interrupts and other background activity that can interfere with determinism in an application's execution time. Shielded processors provide the best possible environment for guaranteeing that a real-time application can meet its execution time deadlines. See the <i>RedHawk Linux User's Guide</i> and the <code>shield(5)</code> man page for complete details.
Frequency-Based Scheduler	The Frequency-Based Scheduler (FBS) provides a mechanism for scheduling applications that run according to a predetermined cyclic execution pattern. The FBS also provides a very fast mechanism for waking a process when it is time for that process to execute. In addition, the FBS allows the performance of cyclical applications to be tracked, with various options available to the programmer when deadlines are not being met. The FBS is the kernel mechanism that underlies the NightSim GUI for scheduling cyclical applications. See the <i>NightSim User's Guide</i> and the <i>Frequency-Based Scheduler (FBS) User's Guide</i> for additional information.
User-level Preemption Control	When an application has multiple processes that can run on multiple CPUs and those processes operate on data that is shared between the processes, then access to the shared data must be protected to prevent the data from being corrupted by simultaneous access by more than one process. The most efficient mechanism for protecting shared data is a spin lock. An attempt to lock an unlocked spin lock succeeds with a single test and set instruction. However, spin locks cannot be effectively used by an application if there is a possibility that the application can be preempted while holding the spin lock. To remain effective, the operating system must support a mechanism that allows the application to quickly disable preemption. This is the capability provided by user-level preemption control. See the <i>RedHawk Linux User's Guide</i> and the <code>resched_cntl(2)</code> man page for more information about user-level preemption control.
User-level Spin Locks	Busy-wait mutual exclusion is achieved by associating a synchronizing variable with a shared resource. When a process or thread wishes to gain access to the resource, it locks the synchronizing variable. When it completes its use of the resource, it unlocks the synchronizing variable. If another process or thread attempts to gain access to the resource while the first process or thread has the resource locked, that process or thread must delay by repeatedly testing the state of the lock. This form of synchronization requires that the synchronizing variable be accessible directly from user mode and that the lock and unlock operations have very low overhead. RedHawk Linux busy-wait mutual exclusion tools include a low-overhead busy-wait mutual exclusion variable (a spin lock) and a corresponding set of macros. Refer to the <i>RedHawk Linux User's Guide</i> for details.

Table 8-1 Concurrent Added Real-Time Features (Continued)

Feature	Description
Fast Block/Wake Services	Many real-time applications are composed of multiple cooperating processes. These applications require efficient means for doing inter-process synchronization. The fast block/wake services allow a process to quickly suspend itself awaiting a wakeup notification from another cooperating process. See the <i>RedHawk Linux User's Guide</i> and the <code>postwait(2)</code> and <code>server_block(3)</code> man pages for more details.
Processor Affinity	In a real-time application where multiple processes execute on multiple CPUs it is desirable to have explicit control over the CPU assignments of all processes in the system. This capability is supplied through the <code>mpadvise(3)</code> library routine and the <code>run(1)</code> command. See the <i>RedHawk Linux User's Guide</i> and the man pages for more details.
/proc Modifications	Modifications were made to the process address space support in <code>/proc</code> to allow a privileged process to read or write the values in another process' address space. This is for support of the NightProbe data monitoring tool and for the NightView debugger.
RCIM Driver	<p>A driver was added for support of the Real-Time Clock and Interrupt Module. This multi-purpose PCI card has the following functionality:</p> <ul style="list-style-type: none"> - allows the connection of four external device interrupts - four real time clocks that can interrupt the system - four programmable interrupt generators which allow generation of an interrupt from an application program <p>The above functions can all generate local interrupts on the system where the RCIM card is installed. The RCIM can also connect multiple RedHawk systems together using a ribbon cable, allowing up to eight of the local interrupts to be distributed to other RCIM-connected systems. This allows one timer, or one external interrupt or one application program to interrupt multiple RedHawk systems almost simultaneously to create synchronized actions on multiple RedHawk systems. In addition, the RCIM contains a synchronized high-resolution clock so that multiple systems can share a common time base. See the <i>Real-Time Clock and Interrupt Module (RCIM) PCI Form Factor User's Guide</i> for additional information.</p>
Kernel Trace Facility	Support was added to the kernel to allow kernel activity to be traced. This includes insertion of kernel trace points where a time-stamped trace record is written to a memory buffer when the trace point is executed, mechanisms for enabling kernel trace points, mechanisms for reading trace memory buffers from the kernel, and mechanisms for managing trace buffers. The kernel trace facility is used by the NightTrace tool.

Table 8-1 Concurrent Added Real-Time Features (Continued)

Feature	Description
High Resolution Timing	In the standard Linux kernel, the system accounts for a process' CPU execution times using a very coarse-grained mechanism. This means that the amount of CPU time charged to a particular process can be very inaccurate. This patch provides a mechanism for very accurate CPU execution time accounting. This allows better performance monitoring of applications. See the <i>RedHawk Linux User's Guide</i> for more details.
ptrace Extensions	The ptrace debugging interface has been extended to support the capabilities of the NightView debugger. Features added include: <ul style="list-style-type: none"> - the capability for a debugger process to read and write memory in a process that is not currently in the stopped state - the capability for a debugger to trace only a subset of the signals in a process that is being debugged - the capability for a debugger to efficiently resume execution at a new address within a process being debugged - the capability for a debugger process to automatically attach to all children of a process that is being debugged
usermap (3) and /proc mmap (2)	The usermap (3) library routine, which resides in the libccur_rt library, provides applications with a way to efficiently monitor and modify locations in currently executing programs through the use of simple CPU reads and writes. The /proc filesystem mmap (2) is the underlying kernel support for usermap (3) , which lets a process map portions of another process' address space into its own address space. Thus, monitoring and modifying other executing programs becomes simple CPU reads and writes within the application's own address space, without incurring the overhead of /proc filesystem read (2) and write (2) system service calls. Refer to the <i>RedHawk Linux User's Guide</i> for more information.
Hyper-threading	Hyper-threading is a feature of the Intel Pentium Xeon processor that allows for a single <i>physical</i> processor to appear to the operating system as two <i>logical</i> processors. Two program counters run simultaneously within each CPU chip so that in effect, each chip is a dual-CPU SMP. Hyper-threading is enabled by default on all RedHawk Linux pre-built kernels. For more information about hyper-threading and how to use it for best performance, refer to the <i>RedHawk Linux User's Guide</i> .

8.5 Open Source Real-Time Features

Refer to Table 8-2 for a description of the features that have been applied to the RedHawk Linux kernel using non-standard open source Linux patches.

Table 8-2 Open Source Real-Time Features

Features	Description
Kernel Preemption	This patch adds the ability for a high priority process to preempt a lower priority process that is currently executing inside the kernel. Under standard Linux the lower priority process would continue running until it exited from the kernel, creating longer worst case process dispatch latency. This patch leverages the data structure protection mechanisms built into the kernel to support symmetric multiprocessing.
Real-Time Scheduler	This patch was actually created for the 2.5 Linux development baseline. It provides fixed-length context switch times regardless of how many processes are active in the system. It also provides a true real-time scheduling class that operates on a symmetric multiprocessor. This patch is known by the name of the open source developer who created it - Ingo Molnar.
POSIX Timers (high-resolution)	This patch provides high-resolution support for the POSIX timers interfaces. This includes a version of <code>nanosleep(2)</code> which can be used to put a process to sleep for a very short time quantum. <code>clock_nanosleep(2)</code> also accepts an extra argument that specifies which clock should be used for measuring the duration of the sleep. See the <code>clock_nanosleep(2)</code> man page for a complete description. These patches also support high-resolution timers that allow the process to specify a time in the future when a signal should be asynchronously sent to the calling process. See the <i>RedHawk Linux User's Guide</i> for more information on POSIX timers.
Low Latency Patches	In order to protect shared data structures used by the kernel, the kernel protects code paths that access these shared data structures with spin locks and semaphores. The locking of a spin lock requires that interrupts and preemption be disabled while the spin lock is held. A study was made which identified the worst-case interrupt off times. The low latency patches modify the algorithms in the identified worst-case interrupt off scenarios to provide better interrupt response times.
Kernel Debuggers	Two different kernel debuggers are supported. Each provides a different set of features. The <code>kdb</code> kernel debugger is linked with the kernel and can be run as a native kernel debugger. Using <code>kgdb</code> , the RedHawk Linux kernel can be debugged with <code>gdb</code> as if it were a user application. <code>gdb</code> runs on a separate system, communicating with the kernel being debugged through the console's serial port.

Table 8-2 Open Source Real-Time Features (Continued)

Features	Description
Capabilities Support	This patch provides a mechanism to assign privilege to users. This allows a non-root user to run applications that require privileges that only root would normally be allowed. For example, one privilege allows users to lock pages in memory. See the <i>RedHawk Linux User's Guide</i> and the pam_capability(8) and capability.conf(5) man pages for more information about capabilities support.
Kernel Core Dumps/ Crash Analysis	This patch provides the support for dumping physical memory contents to a file as well as support for utilities that do a post mortem analysis of a kernel core dump. See the lcrash(1) man page and the section entitled <i>Using the Linux Kernel Crash Dump (LKCD)</i> on page 37 of these notes for more information about crash dump analysis. In addition, see the <i>RedHawk Linux User's Guide</i> for additional information on LKCD.
XFS Journaling File System	Support for the XFS journaling file system from SGI is enabled by default in RedHawk Linux. XFS is a multithreaded, 64-bit file system capable of handling large files and large filesystems, extended attributes, variable block sizes, is extent based and makes extensive use of Btrees (directories, extents, free space) to aid both performance and scalability. Both user and group quotas are supported. See the <i>RedHawk Linux User's Guide</i> for more information.
POSIX Message Queues	This patch adds POSIX style IPC message queues to the Linux kernel. It consists of a kernel patch and a C library with appropriate interfaces. All of the functions are independent of the System V IPC message queues already implemented in RedHawk Linux. POSIX message queues are visible as the mqueue filesystem to be mounted in /dev/mqueue . See the <i>RedHawk Linux User's Guide</i> for more information.

8.6 NightStar Development Tools

Users can schedule, monitor, debug and analyze run time behavior of their real-time applications and the RedHawk Linux operating system kernel using the following NightStar tools:

- NightTrace Event Analyzer
- NightSim Periodic Scheduler
- NightProbe Data Monitor
- NightView Debugger (Beta)

NightStar's non-intrusive operation ensures real-time determinism.

Refer to the *RedHawk Linux NightStar Tools Release Notes*, Pubs No. 0898008-*x.x* (where *x.x* = latest version) for additional information.

9.0. Changes in this Release

This section describes the enhancements and other changes included in Release 1.3.

9.1 Enhancements

9.1.1 Real-Time Hyper-threading

RedHawk Linux Release 1.3 has hyper-threading enabled by default. Hyper-threading is a feature of the iHawk's Intel Pentium Xeon processor that allows for a single physical processor to run multiple threads of software applications simultaneously. Each processor has two sets of architecture state while sharing one set of processor execution resources. Each architecture state can be thought of as a logical CPU. With hyper-threading enabled, there are now twice as many logical CPUs in a system. In fact, a uniprocessor system will have two logical CPUs, making it possible to shield one of them from interrupts and background processes.

Refer to section 9.2.1 for a discussion about CPU identification and how it differs in Release 1.3 from previous releases.

Refer to the “Real-Time Performance” chapter of the *RedHawk Linux User's Guide* for detailed information about hyper-threading and how to utilize it for best real-time performance.

9.1.2 POSIX Message Queues

POSIX message queues have been added to RedHawk Linux Release 1.3. This enhancement provides POSIX style IPC message queues to the kernel and is implemented as a filesystem (`/dev/mqueue`). Appropriate library interfaces are included in Concurrent's real-time library, `libccur_rt`. Refer to the “Interprocess Communication” chapter of the *RedHawk Linux User's Guide* for information about POSIX message queues.

9.1.3 usermap(3) Support

Users can now utilize the `usermap(3)` routine to efficiently monitor and modify locations in currently executing programs through the use of simple CPU reads and writes. The underlying kernel support for this routine is the `/proc` filesystem `mmap(2)` system service call, which lets a process map portions of another process' address space into its own address space. This allows variables in the mapped program to be directly accessed via memory references. In previous releases, variables could only be accessed by making a `read(2)` or `write(2)` system call to a `/proc` file. This capability will be utilized by a new version of NightProbe to allow for monitoring variables with less system overhead and therefore less intrusion to the application's behavior. Refer to the “Memory Mapping” chapter in the *RedHawk Linux User's Guide* for more information.

9.1.4 User-level Spin Locks

New RedHawk Linux busy-wait mutual exclusion tools include a low-overhead busy-wait mutual exclusion variable (a spin lock) and a corresponding set of macros that allow you to initialize, lock and unlock spin locks and determine whether or not a particular spin lock is locked. To be effective, user-level spin locks must be used with user-level preemption control. These are explained in the “Interprocess Synchronization” chapter of the *RedHawk Linux User’s Guide*.

9.1.5 Additional Real-Time Scheduling Priorities Available

Processes scheduled under the fixed priority scheduling policies `SCHED_FIFO` and `SCHED_RR` have a user priority in the range 1 to 99 by default. This range can now be modified to include as many as 2000 real-time priority levels. The range can be modified with the `CONFIG_MAX_USER_RT_PRIO` kernel tunable accessible through the General Setup selection of the Linux Kernel Configuration menu.

9.1.6 Daemons for Deferred Interrupt Processing

By default in RedHawk Linux, all deferred interrupt functions now execute in the context of a kernel daemon. The priority and scheduling policy of these kernel daemons can be set via kernel configuration parameters. This allows the system to be configured so that a high-priority real-time task can preempt the activity of deferred functions.

`Softirqs` and tasklets are both run by the `ksoftirqd` daemon; one daemon per CPU. A `softirq` or tasklet runs on the CPU that triggered its execution. The priority of this daemon is determined by the `CONFIG_SOFTIRQ_PRI` kernel tunable, accessible by the Processor Types and Features selection of the Linux Kernel Configuration menu. By default this value is set to zero, which indicates that the `ksoftirqd` daemon will run as under the `SCHED_FIFO` scheduling policy at a priority of one less than the highest real-time priority. Setting this tunable to a positive value specifies the real-time priority value that will be assigned to all `ksoftirqd` daemons.

Bottom halves can be run by two different kernel daemons - `keventd` and `kbottomd`. Unlike the `ksoftirqd`, there is only one instance of `keventd` and one instance of `kbottomd`. These daemons would therefore generally be run on a non-shielded CPU. The priority of these kernel daemons can be set via the kernel tunables `CONFIG_BH_PRI` and `CONFIG_EVENTD_PRI`, respectively. By default the `keventd` kernel daemon runs under the `SCHED_OTHER` scheduling policy, at the highest `SCHED_OTHER` priority. By default, the `kbottomd` kernel daemon is set to the same priority and scheduling policy as the `ksoftirqd` kernel daemon. By setting the value of the appropriate kernel configuration parameter to a positive value, the affected kernel daemon will run as a `SCHED_FIFO` process at the specified priority.

For more information about deferred interrupt processing within device drivers, refer to the “Device Drivers and Real Time” chapter of the *RedHawk Linux User’s Guide*. Refer to the “Configuring and Building the Kernel” chapter of the *RedHawk Linux User’s Guide* for information about configuring the kernel.

9.1.7 ioctl Flag for Big Kernel Lock (BKL)

RedHawk Linux Release 1.3 provides a mechanism for a device driver to specify that the Big Kernel Lock (BKL) should not be locked before calling the `ioctl` routine.

The BKL is a spin lock in the Linux kernel, which is used when a piece of kernel source code has not been fine-grain multi-threaded. While much use of the BKL has been removed by systematically multi-threading the Linux kernel, the BKL is still the most highly contended and longest held lock in the Linux kernel. By default, the Linux kernel will lock the BKL before calling the `ioctl(2)` function that is associated with a device driver. If a device driver is multi-threaded, then it is not necessary to lock the BKL before calling the `ioctl` routine.

When a device is used to support real-time functions or when an application makes calls to a device's `ioctl` routine on a shielded CPU, it is very important that the device driver be modified so that the BKL is not locked. Without this modification, a process could stall spinning on the BKL spin lock for an extended period of time causing jitter to the programs and interrupts that are assigned to the same CPU.

The mechanism for specifying that the BKL should not be locked on entry to a device's `ioctl` routine is to set the `FOPS_IOCTL_NOBKL` flag in the `file_operations` structure in the device driver source code. For more information, refer to the "Device Drivers and Real Time" chapter of the *RedHawk Linux User's Guide*.

9.1.8 XFS File System

The XFS journaling file system from SGI is included in RedHawk Linux Release 1.3 and enabled. XFS combines advanced journaling technology with full 64-bit addressing and scalable structures and algorithms. The journaling technology allows it to restart very quickly after an unexpected interruption, regardless of the number of files it is managing. Refer to the *RedHawk Linux User's Guide* for information.

The XFS file system cannot be used for installation purposes. Only a user partition can be XFS.

The XFS file system is a recent addition to RedHawk Linux Release 1.3 and should be used with caution at this time.

9.1.9 New RAID Controller Support

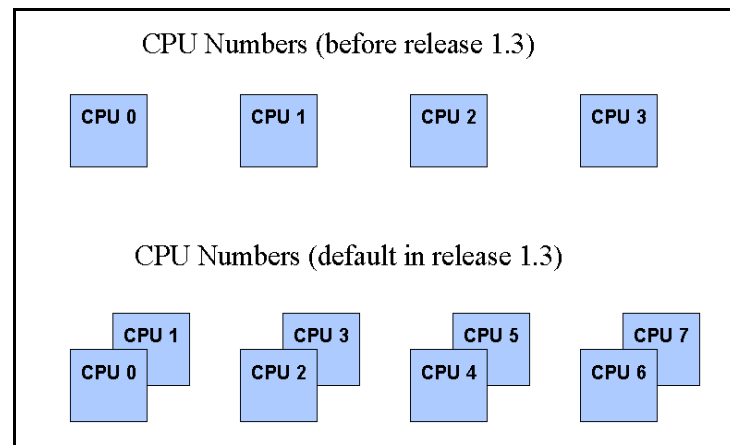
Support has been added for the Dell PERC3 RAID controller. This is an option to the PowerEdge 650 graphic server.

9.2 Modifications

9.2.1 CPU Identification with Hyper-threading

In RedHawk Release 1.3, because hyper-threading has been enabled by default on all physical CPUs, the method used to identify CPUs is different than in previous releases. With hyper-threading enabled, there are now twice as many logical CPUs in a system. Figure 9-1 shows how the CPUs are identified by RedHawk Linux on an iHawk system with four physical CPUs.

Figure 9-1 CPU Identification



The term “sibling CPU” refers to the other CPU in a pair of logical CPUs that reside on the same physical CPU. For example, CPUs 0 and 1 are siblings in a hyper-threaded system, as are CPUs 2 and 3, etc.

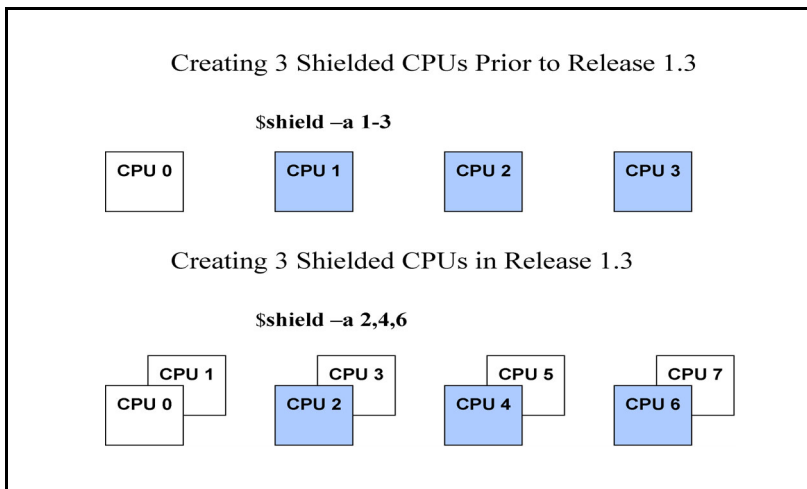
The CPU number is displayed or specified in commands like `top(1)`, `run(1)` and `shield(1)`. System calls like `sched_setaffinity(2)` and library routines like `mpadvise(3)` use bit masks of CPU numbers to specify the CPUs that are affected by these calls. There are also bit masks of CPU numbers that can be written and read for setting the CPU affinity of processes or interrupts which are located in `/proc/pid/affinity` and `/proc/irq/irqnumber/smp_affinity`. Finally there are bit masks of CPU numbers that can be read or written to determine or change CPU shielding attributes in `/proc/shield/*`.

Because of this, the following scenarios may require modifications in order to reference the correct CPU:

- scripts that utilize the commands that specify CPUs
- programs that utilize the application programming interfaces that specify CPUs
- scripts or programs that utilize the `/proc` bit masks that reference CPUs

Figure 9-2 shows an example of how the **shield(1)** command that was used in RedHawk Release 1.2 to create three shielded CPUs might be modified to work in a similar fashion under Release 1.3.

Figure 9-2 Shielding Modification Example



It is possible to configure RedHawk Release 1.3 so that CPU identification works in the same manner as it did in prior releases by disabling hyper-threading on a system-wide basis. The system administrator can completely disable hyper-threading on all CPUs by turning off the `CONFIG_HT` kernel tunable through the Processor Type and Features selection of the Linux Kernel Configuration menu or at boot time by specifying the “noht” option. Note that when hyper-threading is disabled on a system-wide basis, the logical CPU numbers will be equivalent to the physical CPU numbers as was true in RedHawk prior to Release 1.3.

Hyper-threading can be disabled on a per CPU basis using the **cpu(1)** command to mark one of the siblings “down”. This new command can be used to take one or more individual CPUs down, bring downed CPUs up, and view the current active status and shield characteristics of all the CPUs in the system. Note that marking a CPU down does not change the logical CPU numbers assigned by RedHawk. Refer to the **cpu(1)** man page for more details.

It should be noted that hyper-threading introduces new considerations into how a system should be configured in terms of CPU shielding in order to achieve a balance between deterministic application performance and general system throughput. For a better understanding of these issues, please see the “Real-Time Performance” chapter in the *RedHawk Linux User’s Guide*.

9.2.2 POSIX Timers

The hardware timer used to drive the POSIX timers and the POSIX **nanosleep(2)** call were modified in this release. A per CPU timer, the local APIC timer (also known as the local timer interrupt) is now utilized. This change allows the overhead of timer operations to be more closely associated with the processes that initiated the timer. It is possible to shield a CPU from local timer interrupts. In this case the local timer interrupts will still occur on a shielded CPU if a process on that CPU has an active POSIX timer or **nanosleep(2)** function.

If a process with an active POSIX timer or `nanosleep(2)` call is migrated away from a CPU because that CPU has been marked “shielded from processes,” then the timer will also be migrated.

If it is critical to totally eliminate local timer interrupts for optimum performance on a specific shielded CPU, applications utilizing POSIX timers or nanosleep functionality should not be biased to that CPU.

9.2.3 Write Access to RCIM Device Files

In RedHawk Linux Release Version 1.3, during system initialization the RCIM device files, by default, are created writeable by all users (mode=0666). The administrator may wish to restrict the permissions to only certain users. This must be done in such a way as to allow FBS/NightSim users permission to write these files. One way to achieve this is to define a group that provides write access to the RCIM device files and assign FBS users to that group.

For example, to add a group named “realtimer”, do the following:

```
# groupadd -r realtimer
```

Next, give the realtimer group write access to the RCIM device files by editing `/etc/sysconfig/rcim` and setting the following:

```
RCIM_OWN=root
RCIM_GRP=realtimer
RCIM_PERM=0664
```

Finally, add FBS users to the group. The following example adds “joe” to the realtimer group:

```
# adduser -u 50 -g 100 -d /home/joe -G realtimer -s /bin/bash -c "Joe User" joe
```

9.2.4 IRQ-Blocking Spinlock Services Made Preemptive

The irq-blocking spinlock services have been made preemptive while spinning (but before the lock is actually acquired). These services are:

- `spin_lock_irq`
- `spin_write_irq`
- `spin_read_irq`
- `spin_lock_irqsave`
- `spin_write_irqsave`
- `spin_read_irqsave`

The `spin_*_irqsave` services now enforce (via compilation error) that the `flags` argument be unsigned long. This has always been a requirement, however, it was not enforced prior to Release 1.3. Therefore, some improperly written add-on drivers that used to compile may no longer compile.

9.2.5 High Resolution Process Timing Facility Changes

Changes have been made to the way in which the High Resolution Process Timing Facility is configured in the system. This facility can be used for per-process user and system execution time accounting and by the Performance Monitor through the Frequency-Based Scheduler and NightSim.

As an alternative to using the local timer interrupt for CPU accounting, the High Resolution Process Timing Facility can be enabled in the system by setting the CONFIG_HR_PROC_ACCT kernel tunable, accessible through the General Setup selection of the Linux Kernel Configuration menu.

In order to use the Performance Monitor, CONFIG_FBS_PM must be set, accessible through the Frequency-Based Scheduler selection of the Linux Kernel Configuration menu.

If it is desirable to exclude interrupt time from the system and user time during time accounting or performance monitoring, CONFIG_FBS_PM_EXCL_INTRS (under the Frequency-Based Scheduler menu selection) must be set in addition to either of the configuration methods described above.

Refer to the “Configuring and Building the Kernel” chapter of the *RedHawk Linux User’s Guide* for information about configuring the kernel.

10.0. Known Issues

Special consideration should be given to the following areas:

1. There are sporadic problems with password files that are distributed to multiple systems via NIS. This can prevent valid users from logging into the system when their accounts are made known to that system via NIS.
2. NFS mounts of large partitions will sporadically have problems in seeing all of the files that should be mounted in the partition.
3. When running the Linux Test Project test suite LTP-2002(*), the user must remove the **Fork07** test case prior to building the test suite to prevent a system hang on RedHawk systems with greater than 1GB of memory. This is a known bug in the official Linux kernel. For more information, refer to the Linux Test Project web site at <http://ltp.sourceforge.net/>
4. The “Console Redirection” BIOS feature, when enabled, has been observed to interfere with the proper operation of the integrated VGA video and the XFree86 X server with some iHawk platforms, such as the Dell PowerEdge™ 6650.
5. Shielding local timers on all CPUs can make the system hang.
6. Users of `pw_wait(2)`, `pw_post(2)` and `pw_postv(2)` may have to make slight changes to their code, as shown below.

pw_wait(2) changes:

```
old: int pw_wait(struct timespec *, void *);
new: int pw_wait(struct timespec *, struct resched_var *);
```

pw_post(2) changes:

```
old: int pw_post(ukid_t, void *);
new: int pw_post(ukid_t, struct resched_var *);
old: int pw_post(ukid_t, void *);
new: int pw_post(ukid_t, struct resched_var *);
```

pw_postv changes:

```
old: int pw_postv(int, ukid_t *[], int *[], void *);
new: int pw_postv(int, ukid_t[], int [], struct resched_var *);
```

7. Using the NMI button on the processor enters **kdb** on the console. It cannot be used, however, to step repeatedly through an error condition.
8. Note that in order to use the `pam_capability(8)` feature with `ssh`, the `/etc/ssh/sshd_config` file must have the following option set:

```
UsePrivilegeSeparation no
```

9. When a system configuration change is made that affects the number of CPUs (for example when booting with a uniprocessor kernel or with hyper-threading disabled), the **sadc (8)** (sar data collector) program cannot successfully write data in the daily data file **/var/log/sa/sa??** (?? is the day of the month). This results in **crond** sending an email every ten minutes to root@localhost with the following message:

```
Cannot append data to that file
```

To eliminate the email, remove or move the **/var/log/sa/sa??** file for the current date.

11.0. Using the Linux Kernel Crash Dump (LKCD)

LKCD contains kernel and user level code designed to:

- save the kernel memory image when the system dies due to a software failure
- recover the kernel memory image when the system is rebooted
- analyze the memory image to determine what happened when the failure occurred

The memory image is stored into a dump device, which is represented by one of the disk partitions on the system.

11.1 Documentation

After LKCD has been installed on your system, the documents listed in Table 11-1 are available at: `/usr/share/doc/lkcd`. The user should be familiar with the contents of these documents.

Table 11-1 LKCD and Lcrash Documents

File Name	Title of Document
lkcd_tutorial.pdf	LKCD Installation and Configuration
lcrash.pdf and lcrash.htm	Lcrash HOWTO

Additional information about LKCD and the LKCD project can be found at:

<http://lkcd.sourceforge.net/>

11.2 Installation/Configuration Details

The `lkcdutils` rpm is automatically installed as part of the RedHawk Linux installation. The default RedHawk kernel configurations include the LKCD kernel patch. Concurrent has added scripts to automatically patch the `/etc/rc.d/rc.sysinit` file to configure LKCD to take dumps and to save dumps at boot time. LKCD will automatically self-configure to use the swap partition as the dump device. This is done by creating the file `/dev/vmdump` as a symbolic link to the swap partition.

The dump is saved to a disk partition and then copied to the dump directory on reboot. The system administrator must set up the disk partitions so there is enough space in `/var/log/dump` and the swap partition.

The `/etc/sysconfig/dump` configuration file contains configuration details. This file may be edited by the user for example, to select compression types, or to change the directory where dumps are saved. (GZIP compressed dumps are the default.)

11.3 Forcing a Crash Dump on a Hung System

LKCD adds a SysRq option to force a dump. By default, Concurrent has configured the kernel with the Magic System Request option. To use SysRq, the user must enable it through the `/proc` filesystem as follows:

```
echo 1 >/proc/sys/kernel/sysrq
```

The configuration file `/etc/sysctl.conf` sets the default value.

Two methods to force a dump are described below. Use the method applicable to your configuration:

Using PC keyboard:	Use the Ctrl+Alt+SysRq+C
Using serial console:	Break followed by C

An example of how you can send a Break using a serial console is illustrated below:

Using minicom as terminal emulator Ctrl+A+F to send break, followed by C

Refer to the `lcrash` documentation for information on how to run `lcrash`. A dump is a set of files, for example:

```
# cd /var/log/dump
# ls
0 1 bounds

# cd 0
# ls
analysis.0  dump.0  kerntypes.0  lcrash.0  map.0

# lcrash -n 0

Please wait...
  Initializing vmdump access... Done.
  Loading system map... Done.
  Loading type info (Kerntypes)... Done.
  Initializing arch specific data... Done.
  Loading ksyms from dump..... Done.

>> stat | less
```

This should display information about when the dump was taken, the panic string and the kernel messages.

12.0. Useful Information

12.1 Man Pages

Man pages can be found online. The easiest way to access these is by typing `man` followed by a man page name or a command name.

12.2 RCIM Connection Modes

Before installing an RCIM, determine the connection mode. It is easier to connect the synchronization cable to the input connector before the RCIM is installed. An RCIM can be connected in one of the following modes:

Isolated mode	Neither the input nor output connector is used. There is no connection between this RCIM and other RCIMs.
Master mode	The RCIM is connected to one or more other RCIMs, and the RCIM is at the head of the chain. There is an output cable connection from a master RCIM but there is no input cable connection into it. The RCIM master is unique in that it controls the synchronized clocks. See Chapter 3 of the <i>Real-Time Clock and Interrupt Module PCI Form Factor User's Guide</i> for a discussion of synchronized clocks.
Slave mode	The RCIM is connected to one or more other RCIMs, but is not at the head of the chain. There is an input cable connection into a slave RCIM and there may be an output cable connection.

12.3 File System Swap Size Guidelines

The table below is intended to help you determine the partition size of the swap file system. The swap partition size is based on the size of the system's main memory plus adding an additional 5 percent. This swap space allocation should be adequate to accommodate system crash dumps.

Table 12-1 supplies the recommended swap partition sizes for various sizes of main memory.

Table 12-1 Swap Partition Size Guidelines

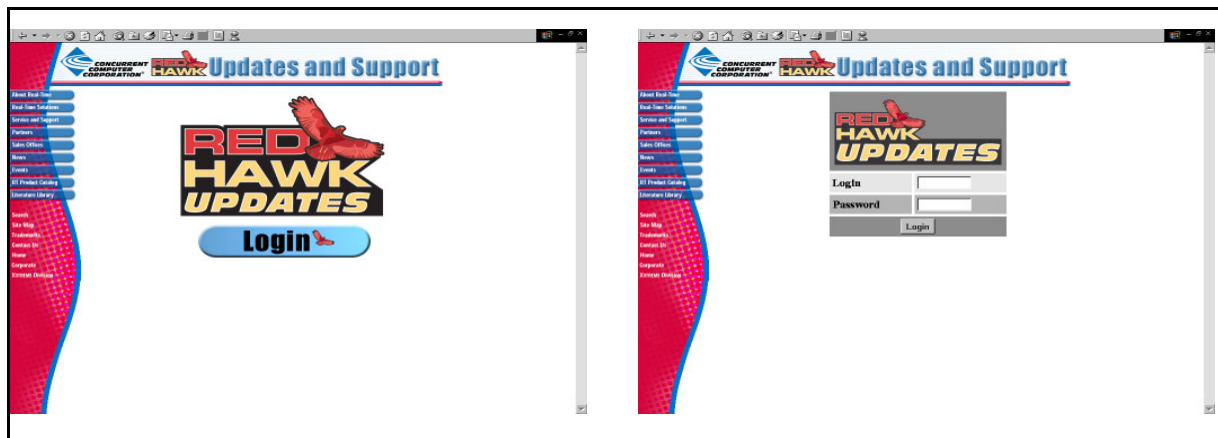
Main Memory Size	Recommended Swap Partition Size
256 MB	269 MB
1 GB	1076 MB
2 GB	2151 MB
4 GB	4301 MB

13.0. Software Updates and Support

Available updates to RedHawk Linux software products can be downloaded from: <http://redhawk.ccur.com>. This site also includes the most recent documentation available.

A view of the site login windows is shown below.

Customer login ID and password can be found in the shipping documents accompanying the system.



13.1 How to Download and Install Updates

Upon login you will by default see a list of new packages (that is, packages that you have not yet downloaded). Click on the packages to download them and save them in a suitable location (e.g. `/root/updates`).

To install the newly downloaded packages, log in as root in single-user mode. Change directory to the location of the updates and issue the following command:

```
rpm -Fvh *.rpm
```

The command should complete successfully, though it may take several minutes depending on the number of updates being installed. When complete, exit single-user mode (^D).

If you have problems, please contact the Concurrent Software Support Center (see *Direct Software Support* on page 41 for further information).

14.0. Direct Software Support

For answers to basic RedHawk Linux questions, check out the FAQ posted on Concurrent's web site: <http://www.ccur.com>. Select Real-Time Division, Service and Support, FAQ.

If you need assistance or information about your system, please contact the Concurrent Software Support Center at our toll free number (1-800-245-6453). Our customers outside the continental United States can contact us directly at 1-954-977-5554. The Software Support Center operates Monday through Friday from 8 a.m. to 7 p.m., Eastern Standard Time.

Calling the Software Support Center gives you immediate access to a broad range of skilled personnel and guarantees you a prompt response from the person most qualified to assist you. If you have a question requiring on-site assistance or consultation, the Software Support Center staff will arrange for a field analyst to return your call and schedule a visit.

Concurrent provides a Software Action Request (SAR) form which our customers can fill out and submit to their local field analyst or the Software Support Center. This procedure ensures that your request is entered into our SAR database for follow-up and action.

To obtain copies of SAR forms, call the Software Support Center and request form number CSD1833B.

