

# **SGI™ Linux® Environment 1.1 with Red Hat® 6.0 Start Here**

Document Number 007-4062-003

© 1999, Silicon Graphics, Inc.— All Rights Reserved

The contents of this document may not be copied or duplicated in any form, in whole or in part, without the prior written permission of Silicon Graphics, Inc.

#### LIMITED AND RESTRICTED RIGHTS LEGEND

Use, duplication, or disclosure by the Government is subject to restrictions as set forth in the Rights in Data clause at FAR 52.227-14 and/or in similar or successor clauses in the FAR, or in the DOD, DOE or NASA FAR Supplements. Unpublished rights reserved under the Copyright Laws of the United States. Contractor/manufacturer is SGI, 1600 Amphitheatre Pkwy., Mountain View, CA 94043-1351.

Silicon Graphics is a registered trademark and SGI is a trademark of Silicon Graphics, Inc. Intel is a trademark of Intel Corporation. Linux is a trademark of Linus Torvalds. PhoenixBIOS is a trademark of Phoenix Technologies, Ltd. Oracle is a trademark of Oracle Corporation. Red Hat is a registered trademark of Red Hat, Inc. UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company, Ltd.

---

# Contents

**List of Tables** v

**About This Guide** vii

- 1. Software Overview** 1
  - Feature Overview 2
    - Performance Improvements 3
    - Raw I/O Path Changes 3
    - Large Physical Memory Support 5
    - Fast Synchronization Mechanism 5
    - Fast *gettimeofday* Library Function 6
    - POSIX Asynchronous I/O 6
    - NFS 7
    - Kernel Spinlock Metering 7
    - Crash Functionality 8
    - Patches, Configuration Options, Commands, and Libraries 9
    - Qualified Drivers 9
  - Changes and Additions 10
- 2. Quick Configuration and Installation Instructions** 13
  - Configuring Linux 13
  - Configuring a Default Router 16
  - Completing the Configuration 16
  - Changing Linux Kernels 17
  - Troubleshooting 18
  - Installing the SGI Linux Environment from a CD 18
  - Upgrading the SGI Linux Environment 19



---

## List of Tables

**Table 1-1** Changed Red Hat Applications 11



---

## About This Guide

This document provides information about the SGI Linux Environment 1.1 release. It is divided into two chapters:

- Chapter 1, “Software Overview,” describes the major features of the release and the differences between this release and the Red Hat 6.0 release and the SGI Linux Environment 1.1 release.
- Chapter 2, “Quick Configuration and Installation Instructions,” describes the procedures for doing the initial configuration of your preinstalled SGI Linux Environment software, as well as describing how to configure a default router, install the operating system from a CD, and upgrade from release 1.0 to release 1.1.

The *Red Hat Linux 6.0 Installation Guide* and the *Red Hat Linux 6.0 Getting Started Guide* are also shipped with your SGI machine. These guides provide essential information for your SGI Linux Environment software. Your release package also includes the manuals for your hardware platform.



---

## Software Overview

This chapter provides a short introduction to the SGI Linux Environment and describes its features.

The errata that describes issues which arose too late in the release cycle to be included in this document and provides help with all known problems is available at the following URL:

<http://support.sgi.com/linux/>

Two full-length manuals are included in the package: the *Red Hat Linux 6.0 Installation Guide* and the *Red Hat Linux 6.0 Getting Started Guide*. Use these manuals as a reference for information not included in this guide.

The SGI documentation for Linux, including man pages, HOWTO guides, and other relevant documentation from the Linux Documentation Project, is available from the following web site:

<http://techpubs.sgi.com>

SGI maintains a web site for open source information that has information about projects that are related to its open source efforts:

<http://oss.sgi.com>

**Note:** Packaged with the SGI Linux Environment software is a separate sheet that contains the Software License Agreement. This software is provided to you solely under the terms and conditions of the Software License Agreement. Please take a few moments to review the Agreement.

## Feature Overview

The SGI Linux Environment is a full featured UNIX-like operating system. It adds functionality to the Red Hat Linux distribution that is specific to SGI hardware platforms. The 1.1 release of the SGI Linux Environment is based on the 6.0 release of the Red Hat Linux distribution.

Some of the most significant features that Linux provides are listed below:

- An extensible UNIX-like kernel, supporting symmetric multiprocessing
- Typical commands you would expect to see on a UNIX-like system
- Typical configuration files you would expect to see on a UNIX-like system, along with an optional graphical frontend
- Development tools such as compilers, debuggers, and libraries
- Internet applications such as web servers and browsers, news servers, network utilities, e-mail servers, and clients
- Everything needed for network file sharing with a wide variety of clients
- Desktop environments and graphical applications

The SGI Linux Environment 1.1 release runs on the Silicon Graphics 1400L Server. Other hardware platforms also may be supported by the 1.1 release, depending on the timing of their releases.

The SGI Linux Environment has been optimized to provide enhanced performance on database workloads. SGI has added a number of features to the operating system kernel and certain packages to provide increased performance and manageability for database workloads (such as Oracle 8i).

The performance enhancements include a kernel-level implementation of POSIX 1003.1-1996 asynchronous I/O, a low-overhead interprocess synchronization mechanism, low overhead and high-volume raw disk I/O, a fast *gettimeofday*(3) library function, and support for large amounts of physical memory.

The manageability and supportability improvements include kernel spinlock metering (for performance bottleneck analysis), kernel profiling enhancements, kernel memory dump capability with analysis tools, kernel *gdb* hooks, and a built-in kernel debugger (*kdb*).

The manageability of the release has been improved by integrating a number of publicly available kernel patches, such as:

- Stephen Tweedie's Raw I/O patch, which forms the basis for the SGI raw disk I/O enhancements. This patch is described in "Raw I/O Path Changes" on page 3.
- The Device File System (CONFIG\_DEVFS\_FS) patch from Richard Gooch. This patch provides a more consistent naming scheme for hardware and software devices. Sites that expect to connect a large number of devices may find DEVFS very useful in helping to manage them. DEVFS can also provide the traditional Linux names for devices, for backward compatibility, and is otherwise very compatible with the rest of the Linux system.

For more information, look for DEVFS at the following URL:

<http://oss.sgi.com/projects/>

- The *sard* utility and associated kernel metrics patch for disk traffic analysis. This patch provides additional disk I/O statistics, useful for tuning database layouts and queries.
- The Alan Cox patch for Linux 2.2.10, which contains numerous bugfixes and driver updates.

## Performance Improvements

The performance enhancements, enumerated in the preceding section, accelerate the performance of I/O intensive applications by streamlining the kernel code and data paths for disk I/O as well as providing larger shared memory segments and a low overhead interprocess synchronization mechanism.

## Raw I/O Path Changes

Current file-system-based disk I/O requires fixed size I/O operations (typically 1024 bytes) into kernel buffers, then the data is moved from the kernel buffer to the user program address space. While this allows the file system to cache frequently accessed data, it also consumes excess system bus bandwidth when copying the data from the kernel buffer(s) into the user address space. Both the small size of the I/O (2 sectors) and the copy operation greatly reduce the I/O subsystem throughput for database operations, where transactions and full-table scan operations operate more quickly with no operating system data intervention.

To help alleviate this problem, Stephen Tweedie of Red Hat developed a mechanism that allows disk I/O directly to a buffer in the application address space (historically known as raw (or unprocessed) I/O). This mechanism will lock the required pages of memory to prevent them from being paged out or swapped during the I/O operation.

Applications required to perform this type of disk I/O would open the character special device `/dev/raw` and bind the disk device to a special raw device using an `ioctl(2)` system call.

This mechanism, however, is cumbersome to use and suffers from some deficiencies. The primary deficiency with the mechanism comes from its continued use of the file-system buffer-header data structures and associated device queueing routines. While use of the buffer headers is a straightforward mechanism, it implies that I/O operations will still need to be fragmented into 1024-bytes per operation, increasing the kernel overhead significantly. The binding mechanism used to bind an existing block device to a new raw device is also somewhat cumbersome and counterintuitive to Unix system administrators, who expect to find a relationship in the device namespace between a block device and its corresponding raw device.

To address these concerns, SGI has added additional capabilities to Stephen Tweedie's raw I/O patch that allow large I/O operations directly to the user address space and bypasses the bulk of the kernel I/O queueing code for SCSI and FiberChannel devices.

You can download a `dd` command that is capable of using the raw device features from the following FTP location:

<ftp://oss.sgi.com/projects/rawio/download/dd.raw>

This feature is off by default, but you can turn it on by setting the `CONFIG_RAW` kernel configuration parameter.

More information about raw I/O is available from the following URL:

<http://oss.sgi.com/projects/rawio/faq.html>

## Large Physical Memory Support

Large amounts of physical memory coupled with the ability to create large (multi-gigabyte) shared memory segments provide a boost in performance to database workloads. With this release, SGI has included a configuration parameter that allows the kernel to support more than 2 GB of physical memory (to approximately 3.8 GB) while still providing a 3 GB application virtual (user) address space.

See “Changing Linux Kernels” on page 17 for information about building a kernel that uses this feature.

## Fast Synchronization Mechanism

While the UNIX System V IPC semaphore facility does provide exceptional capability, its performance leaves much to be desired. Many Unix vendors have released a low-overhead inter-application synchronization primitive known as “post /wait.”

SGI has included in this release a kernel level implementation of post/wait along with the library containing application API's. The post allows for a process to “wait” for an event. This event can either be a timeout or a “post” from another process. A group of cooperating processes can use these “post” and “wait” facilities to synchronize among themselves.

In order to use post/wait, the kernel must be compiled with the **CONFIG\_PW** configuration variable, and you may optionally set an additional configuration variable, **CONFIG\_PW\_VMAX**. These variables are described in the configuration help. For a user program to use the post/wait facilities, it must link against *libdba.so*.

For more information on post/wait, please refer to the *postwait(3)* man page.

## Fast *gettimeofday* Library Function

In order to correctly order operations in a database, the database must timestamp data and log entries frequently. Traditional *gettimeofday(3)* library functions are implemented as system calls, thus causing an address space switch (from user to kernel mode) each time a timestamp is required.

With this release, SGI has included a device driver that allows a read-only page of kernel memory, containing only the time value, to be mapped by an application (or on its behalf by a library function). *gettimeofday* is then implemented as a library function that simply reads the time value from the exported page in memory, thus avoiding a system call to obtain the time value.

In order to use the Fast *gettimeofday(3)* Library Function, the kernel must be compiled with **CONFIG\_SYSDAT** and the user program must link against *libdba* in order to override the *libc* version of *gettimeofday(3)*.

## POSIX Asynchronous I/O

The ability to overlap I/O and processing activities has always been important to high-performance applications. To allow this type of overlap in single-threaded applications, SGI has included a kernel-level implementation of POSIX asynchronous I/O and the associated API library.

This facility is turned on by setting the **CONFIG\_AIO** kernel option. User code can get access to the facility by linking with *libdba.so*. Further information can be found in the *lib/libdba/README* file.

## NFS

The following NFS functionality has been added to the 1.1 release:

- NFS version 3 client and server support. SGI Linux Environment 1.0 had NFS server version 2 implemented in user space without any support for locking.
- Network Lock Manager (NLM) version 4 client and server support.
- Kernel level NFS and NLM implementation.

NFS and NFSD are configured as modules by default, but they can be configured to compile as part of the kernel by setting the `CONFIG_NFS_FS` and `CONFIG_NFSD` configuration parameters. The `CONFIG_NFS_V3` parameter is set by default and can be turned off if the user wants to use NFS version 2 only. The `CONFIG_NFSD` parameter needs to be configured for `LOCKD` to work, so if `CONFIG_LOCKD` is set, `CONFIG_NFSD` should be set also.

## Kernel Spinlock Metering

The SGI Linux Environment 1.1 release includes a feature that allows developers to gather statistical information about the SMP kernel's use of spinlocks and mrlocks (multiple-reader single-writer spinlocks). This functionality is called *spinlock metering*, or *lockmetering*.

Spinlock metering is built into the kernel using the `CONFIG_LOCKMETER` configuration option (in the Kernel Hacking section of `make xconfig`). A kernel built with lockmetering will exhibit a small (roughly 1%) performance degradation relative to a kernel that is not configured for lockmetering. See the following URL for additional information:

<http://oss.sgi.com/projects/lockmeter>

## Crash Functionality

The following changes have been made to the Linux crash utility, which are explained briefly below. General information about *lcrash* can be found in the *cmd/lcrash/README* file.

- Linux kernel crash dump enhancements. SGI Linux 1.1 provides a new configuration option to allow kernel crash dumps to be available. This option is configured to be on by default, and the default dump space is the first swap partition found when booting. If you are building a new kernel, you can specify *Support kernel crash dump capabilities* in the Kernel Hacking section of `make xconfig`.

The crash dump capabilities in the kernel allow the system to create a crash dump when a failure occurs due to a *panic()* call or an exception. For more details on the dump method, compression used, and so on, please read the LKCD FAQ at the following URL:

<http://oss.sgi.com/projects/lkcd/faq.html>.

Information about LKCD is also available in the file *cmd/lcrash/README.lkcd*.

- Boot up process changes. As the system boots up, the */sbin/vmdump* script will be run out of */etc/rc.d/rc.sysinit*. This script saves crash dumps and reads *sysconfig* variables to open the dump device and configure the system for crash dumps.
- Crash dump configuration options. There are a number of configurable options to save system crash dumps. Please read */etc/sysconfig/vmdump* for more details on the options available. The following list describes what the options allow you to do:
  - Determine if you want to implement crash dumps in the kernel
  - Choose whether to save crash dumps to disk or not
  - Change the location to which the crash dumps are saved
  - Specify any block dump device you want
  - Compress (or not compress) the crash dumps
  - Configure the system to reset (or not reset) after a failure
- The *lcrash* utility now uses the new *librl* library for command line input.

## Patches, Configuration Options, Commands, and Libraries

The following list describes patches implemented and enhancements to configuration options, commands, and libraries:

- New *librl* library. This new library supplies command line editing and command history functionality. See the *cmd/lcrash/lib/librl/README* file for information on how to use this library. The *lcrash* command uses this library.
- Remote debugging over a serial line. There is a new configuration option, **CONFIG\_GDB**, which is used to enable *gdb* debugging. To force a kernel that has been compiled with **CONFIG\_GDB** to pause during the boot process and wait for a connection from *gdb*, the parameter **gdb** should be passed to the kernel. This can be done by typing **gdb** after the name of the kernel on the LILO command line. The patch defaults to use `ttyS1` at a baud rate of 38400. These parameters can be changed by using `gdbttyS=port number` and `gdbbaud=baud rate` on the command line.
- *rlimits* patch. In the Linux 2.2.10 kernel, faulty *rlimit* checking will not allow a process to have more than 2 GB total address space, stack size or locked memory. This release has fixed the *rlimit* checking, so (subject to other accounting limitations), the kernel honors **RLIM\_INFINITY** settings on these resources.
- SMP PTE patch. In stock Linux, the page stealing code that is used under high memory load has a bug that might cause it to steal a page from a process without writing out the contents to swap if the page has been modified by the process. This bug is only present in a multiprocessor machine. The 1.1 release provides a fix for this bug.

## Qualified Drivers

QLogic 1080/1280 and 2100 drivers have been supplied by QLogic Corporation, and include updated firmware and better error handling. The Alteon Gigabit Ethernet driver has been modified to recognize and drive the SGI Gigabit Ethernet card.

The following drivers are qualified for the Silicon Graphics 1400L Server:

- NCR 53c8xx SCSI
- Intel Ethernet Express Pro 10/100 mb Ethernet

- QLogic 1080/1280 SCSI
- QLogic 2100 Fibre Channel
- Alteon and SGI Gigabit Ethernet

The SGI Linux Environment works with other drivers. You can check the following URL for a complete list:

<http://support.sgi.com/linux/>

## Changes and Additions

This section provides an overview of the differences between Red Hat 6.0 and the SGI Linux Environment 1.1:

- During installation, SGI Linux automatically selects the **linear** and **compact** options to LILO. It is recommended that these options be preserved.
- During installation, at the X Configurator screen, you will now find many SGI monitors listed.
- All kernel RPMs have been replaced with 2.2.10 based kernels with SGI modifications.
- The following packages were modified by SGI:
  - Xconfigurator 4.2.3 (adds support for SGI monitors)
  - initscripts 4.16. (adds SGI logos)
  - lilo 0.21. (fixes bug with large memory systems)
  - mount 2.90. (adds support for NFS V3)
  - ncurses 4.2. (adds support for IRIX *winterm*)
  - ncurses-devel 4.2. (adds support for IRIX *winterm*)
- The following packages were added by SGI:
  - kernprof 1.0 (kernel profiling tool to identify performance bottlenecks)
  - ktrace 1.0 (kernel trace tool)
  - lcrash 1.0 (post-mortem crash file analysis)
  - libdba.so 1.0 (APIs to enhance database performance)

- lockstat 1.0 (spinlock metering analysis)
- sard 0.2 (disk activity statistics/analysis)
- sgi-logos 1.0. (SGI Logos)
- sgi-extra 1.0. (various SGI modifications)
- sgi-release 1.0. (SGI release identification)
- SGI has integrated many updates from Red Hat for Red Hat 6.0. Table 1-1 lists the affected applications:

**Table 1-1** Changed Red Hat Applications

ORBit 0.4.3	kdenetwork 1.1.1
ORBit-devel 0.4.3	kdesupport 1.1.1
XFree86 3.3.3.1	kdetoys 1.1.1
am-utils 6.0.1s11	kdeutils 1.1.1
apmd 3.0beta5	korganizer 1.1.1
control-center 1.0.5	kpilot 3.1b9
control-center-devel 1.0.5	libtermcap 2.0.8
dev 2.7.7	libtermcap-devel 2.0.8
enlightenment 0.15.5	linuxconf 1.16r1.3
esound 0.2.12	linuxconf-devel 1.16r1.3
esound-devel 0.2.12	mc-4.5.36
gdm 1.0.0	mcserv-4.5.36
gedit 0.5.4	mod_php3 3.0.9
gedit-devel 0.5.4a	mod_php3-imap 3.0.9
gftp 2.0.3	mod_php3-manual 3.0.9
glib 1.2.3	mod_php3-pgsql 3.0.9
glib-devel 1.2.3	net-tools 1.52
gmc 4.5.36	netkit-base 0.10

**Table 1-1 (continued)** Changed Red Hat Applications

---

gnome-core 1.0.7	netscape-common 4.6
gnome-core-devel 1.0.7	netscape-communicator 4.6
gnome-games 1.0.2	netscape-navigator 4.6
gnome-games-devel 1.0.2	pump 0.7.0
gnome-libs 1.0.10	rpm 3.0.2
gnome-libs-devel 1.0.10	rpm-devel 3.0.2
gnotepad+ 1.1.4	rxvt 2.6.0
gnumeric 0.27	rxvt 2.6.0
gqview 0.7.0	samba 2.0.5a
gtk+ 1.2.3	samba-client 2.0.5a
gtk+-devel 1.2.3	screen 3.7.6
imlib 1.9.5	squid 2.2.STABLE4
imlib-cfgeditor 1.9.5	talk 0.11
imlib-devel 1.9.5	telnet 0.10
inews 2.2.1	timetool 2.6
inn 2.2.1	traceroute 1.4a5
inn-devel 2.2.1	utempter 0.5
kdeadmin 1.1.1	vixie-cron 3.0.1
kdebase 1.1.1	wu-ftpd 2.5.0
kdegames 1.1.1	xchat 1.0.0
kdegraphics 1.1.1	xinitrc 2.4.1
kdelibs 1.1.1	xscreensaver 3.09
kdemultimedia 1.1.1	

---

---

## Quick Configuration and Installation Instructions

This chapter provides instructions for configuring the SGI Linux operating system that comes preconfigured on your SGI machine. It also describes how to configure a default router, install the operating system from a CD, and upgrade from release 1.0 to release 1.1.

Before you install or configure your system, please read Chapter 1, “Software Overview,” so that you may have an idea about the features in release 1.1 and their ramifications.

The *Red Hat Linux 6.0 Installation Guide* and the *Red Hat Linux 6.0 Getting Started Guide* are also shipped with your SGI machine. Refer to them for any questions you might have after following the instructions that apply to you in this chapter.

**Note:** SGI Linux 1.1 ships from the factory with a multiprocessor kernel that supports up to 960MB. If you have a single processor, or have more than 960MB, you may want to boot using a different kernel. This process is described in “Changing Linux Kernels” on page 17.

### Configuring Linux

Your SGI machine comes with Linux preinstalled. This section contains instructions for beginning to configure Linux so your machine will run on a network.

After pressing the power button to power on your machine, several screens of information pass before the login prompt appears.

**Note:** Linux can use automated power management (APM) on single-CPU systems but not on SMP systems. During the boot process, you may see the following warning message:

```
Starting APM daemon: [ FAILED ]
```

The word FAILED is displayed in red. You can ignore the message.

See “Changing Linux Kernels” on page 17 for information on building a single processor kernel.

Use the following procedure to boot and configure your system:

1. After the login prompt appears, type `root`. Press **Enter**.
2. When prompted for a root password type `sgisgi`. Press **Enter**.

**Caution:** For security reasons Linux requires a root password for login. If you leave the password field blank, you cannot log in and are unable to use the machine.

3. When prompted for a new UNIX password, type your new root password. Press **Enter**.
4. When prompted to retype the new UNIX password, do so. Press **Enter**.

You are now in the linuxconf program and are presented with its Welcome.help screen.

**Note:** Documentation for linuxconf information can also be found in the *Red Hat Linux 6.0 Installation Guide*, Chapter 8.

The rest of this procedure describes how to configure Linux so your SGI machine will run on a network.

You will need to know information that you can obtain from your system or network administrator, such as the host name of your machine, its primary name and its domain, its aliases (optional information), and its IP address. You will also need to know if your network uses Domain Name Service (DNS), and if it does, you will need the IP address of your name server and your default domain name for DNS. After you have obtained this information, you may choose to write it next to the corresponding steps in this document for future reference.

5. When you are ready, press the **Tab** key and press **Enter**.

**Note:** Many seconds pass before the screen changes. Do not repeatedly press the Tab or Enter keys, or you will skip forward several screens, bypassing the next screen in the configuration sequence.

Use the down/up arrow keys to make selections in the configuration screen.

6. Select “Basic Host Information.” Press **Enter**.
7. Select “Host name.”
  - Delete the default host name using the **Delete** or **Backspace** key.
  - Type the name of your machine.
8. Select “Primary Name and Domain.”
  - Delete the default primary name using the **Delete** or **Backspace** key.
  - Type the name of your machine and its domain.
9. If your machine has aliases, select “Aliases (opt).”
  - Delete the default alias name using the **Delete** or **Backspace** key.
  - Type any aliases for your machine.
10. Select “IP Address.”
  - Delete the default IP address using the **Delete** or **Backspace** key.
  - Type the IP address of your machine.
11. Press the **Tab** key to select the *Accept* button, and press **Enter**.

You see the following:

```
The DNS has been updated.
```

Select *OK* by pressing **Enter**.
12. Using the down arrow key, go to “Name Server Specification.” Press **Enter**.

The Resolver Confirmation screen appears.

  - If you want to use DNS, keep the default DNS selection: “[X] DNS is required for normal operation.”
  - If you do not want to use DNS, remove the “X: [ ] DNS is required for normal operation.” Go to step 16.
13. To use DNS, using the down arrow key, move to “default domain.”
  - Enter your default domain.

14. Using the down arrow key, move to “Name Server 1.”
  - Delete the default name using the **Delete** or **Backspace** key.
  - Type the IP address of the name server.
15. Select “search Domain 1” and change the default name to your default domain.  
If you want additional domains searched, enter these domains in the following “search domain” fields.
16. Using the **Tab** key, select the *Accept* button, and press **Enter** to accept.  
If you want to configure a default router, go to the next section. If you want to configure mail, other servers, and so on, or exit Linux, go to “Completing the Configuration” on page 16.

## Configuring a Default Router

If you want to configure a default router, follow the instructions in this section.

1. Using the down arrow key, go to Defaults. Press **Enter**.
2. Delete the default IP address.
3. Enter the IP address of your default router. Do not choose “Enable routing” unless you want this machine to be a router.
4. Use the **Tab** key to select *Accept*. Press **Enter**.  
You are returned to the linuxconf menu.

## Completing the Configuration

At this point you may configure mail, the Apache Web server, the Samba file server, and so on. To quit and exit the configuration, follow these instructions:

1. When you are finished with the configuration, press **Tab** to select *Quit*. Press **Enter**.  
You go to the Status of the System dialog box.
2. To activate the changes, press **Enter**.

A few seconds pass before the next prompt appears:

If you wish to perform further configuration of this machine, type the command `/bin/linuxconf` at the command prompt.

The machine will now reboot to implement your configuration changes.

You are finished configuring the SGI Linux operating system. Refer to the Red Hat manuals that came with your system for further information.

## Changing Linux Kernels

To change the kernel that you are running at boot time, you may specify the desired kernel at the LILO prompt. During power-on or reset, the system pauses briefly at this prompt. A list of kernels can be obtained by typing a question mark at this prompt.

```
LILO> ?  
  
linux linux-up linux-bigmem
```

On systems shipped from the factory (ones you have not modified), the first entry is the default. If you press **Return**, this kernel will be booted. If you install or upgrade using a CD, the kernel matching your machine configuration is automatically chosen -- so if your machine has more than 960Mb of RAM, *linux-smp-bigmem* is selected. UP or SMP is selected based on whether you have one or more than one processors.

If you want to change to another kernel, simply enter the name of the kernel as listed, and press **Return**.

As the system comes from SGI, the following kernels are supplied:

- *linux* is the default kernel that supports SMP and up to 960MB of memory (additional memory is ignored).
- *linux-up* is the single processor kernel that supports up to 960MB.
- *linux-bigmem* is the large memory kernel that supports SMP and up to 3.8GB of memory.

To change your selection permanently, become superuser, and edit the `/etc/lilo.conf` file. Each of these boot selections is listed in this file, and is arranged in a “paragraph” fashion. There is optionally a “default” line that specifies the default kernel. If this does

not exist, the first paragraph is the default boot kernel. To change the default, either change the “default” line to point to the desired kernel, or move the intended kernel to the first position and save the file. Next, run `/sbin/lilo`. This should install the kernels, giving you a list of the boot options provided. If you encounter syntax errors, refer to the man page `lilo.conf(5)` for additional detail on the LILO configuration file.

## Troubleshooting

Your machine comes with XFree86 preconfigured for an SGI monitor. If this X configuration does not work for you, or you want to change this, run the command `XF86Setup`. See the Red Hat documentation for information on starting and running X.

Errata information for the SGI Linux environment is located at the following URL:

<http://support.sgi.com/linux>

The *README* file for SGI Linux, which contains the information in Chapter 1, “Software Overview,” is located online in `/usr/doc/README.sgi`

## Installing the SGI Linux Environment from a CD

If your machine is configured to boot from a CD, insert the CD and follow any instructions. The *Red Hat Linux 6.0 Installation Guide* provides information on all of the questions you are prompted to answer.

You will need to know information that you can obtain from your system or network administrator, such as the host name of your machine, its primary name and its domain, its aliases (optional information), and its IP address. You will also need to know if your network uses Domain Name Service (DNS), and if it does, you will need the IP address of your name server and your default domain name for DNS. After you have obtained this information, you may choose to write it next to the corresponding steps in this document for future reference.

If your machine is not configured to boot from a CD-ROM, follow these steps to enable the operating system installation from a CD:

Before powering on the machine, locate the F2 key. Within a few seconds of powering on, you are prompted to press the F2 key. If you do not immediately press the F2 key, the system continues to boot from the hard disk. If you miss the F2 key prompt, power off the machine and begin again.

1. Power on the system.
2. The F2 key prompt displays on the lower half of the screen. Press the F2 key when prompted. The machine performs some setup functions, and then takes you to the PhoenixBIOS Setup Utility Screen.
3. Using the right arrow key, select the Boot Menu (top line of screen.)
4. Using the down arrow, select "Boot Device Priority." Press **Enter**.
5. Using the down arrow key, select ATAPI CD-ROM drive.
6. If the ATAPI CD-ROM drive has a lower boot priority than the hard drive, you must move it. With ATAPI CD-ROM selected, using the plus key (+), move the ATAPI CD-ROM line to a position above the Hard drive line on the screen.
7. Place the operating system CD in the drive.
8. Press the F10 key to save and exit.
9. When you see the prompt "Save configuration changes and Exit?," press **Enter**.

Several seconds elapse before the system starts booting from the CD. The normal Red Hat procedure occurs. Follow any Red Hat instructions, referring to the Red Hat documentation if needed.

## Upgrading the SGI Linux Environment

The following instructions explain how to upgrade from an earlier version of the SGI Linux Environment:

1. Power on the system or type `init 0` to reboot the system.
2. Press F2 to enter Setup.
  - From the BIOS Setup Utility, use the right arrow key to select the *Boot* option.
  - Use the down arrow key to select "Boot Device Priority" and press **Enter**.
  - Highlight "[ATAPI CD-ROM Drive]" and press the plus key (+) to move the CD-ROM to first priority.

- Press **F2** for Setup Configuration.
  - Press **Enter** to invoke “Save Configuration Changes and Exit.”
  - Insert the CD into the drive. The system will reboot.
  - Press **Enter** when you see the message, “Welcome to SGI Linux.”
3. You are asked to choose a first a language and then a keyboard type. Highlight your selection by using the arrow keys and use the Tab key to select *OK*. Press **Enter**.
  4. You are asked to select an installation method. Select CD-ROM and use the Tab key to select *OK* and press **Enter**.
  5. When you are asked to select an installation path, use the Tab key to highlight *Upgrade* and press **Enter**.
  6. When you are asked to select an installation class, highlight *Custom* and press **Enter**. Installation classes are described in section 5.3 of the *Red Hat Linux 6.0 Installation Guide*. Information about building a custom kernel can be found in the section 11.8 of the *Red Hat Linux 6.0 Installation Guide*. I
  7. The system will scan for PCI SCSI bus. When you are asked if you have any more SCSI adapters on your system, use the Tab key to select *No* and press **Enter**.
  8. The system will scan for packages. When you are asked if you would like to customize the set of packages that will be upgraded, use the Tab key to select *Yes* and press **Enter**.
  9. You can upgrade one or more packages by highlighting them and pressing the minus and plus keys (- and +) to select or deselect packages. When you have selected your choices, use the **tab** key to select *Done* and press **Enter**.
  10. When you are asked if you want to upgrade the log, use the **tab** key to select *OK*. Press **Enter**.
  11. The system will rebuild the RPM database, process changes, and find overlapping files. Press *OK* to create the boot disk.
  12. The Install Status window shows the upgrade process as it occurs. To install LILO, highlight “/dev/sda Master Boot Record.” Use the **tab** key to select *OK*. Press **Enter**.
  13. Press **Enter** when the installation is complete.
  14. Press **Enter** and remove the CD, which will reboot the system.



---

## Tell Us About This Manual

As a user of Silicon Graphics products, you can help us to better understand your needs and to improve the quality of our documentation.

Any information that you provide will be useful. Here is a list of suggested topics:

- General impression of the document
- Omission of material that you expected to find
- Technical errors
- Relevance of the material to the job you had to do
- Quality of the printing and binding

Please send the title and part number of the document with your comments. The part number for this document is 007-4062-003.

Thank you!

## Three Ways to Reach Us

- To send your comments by **electronic mail**, use either of these addresses:
  - On the Internet: [techpubs@sgi.com](mailto:techpubs@sgi.com)
  - For UUCP mail (through any backbone site): *[your\_site]!sgi!techpubs*
- To **fax** your comments (or annotated copies of manual pages), use this fax number: 650-932-0801
- To send your comments by **traditional mail**, use this address:

Technical Publications  
Silicon Graphics, Inc.  
1600 Amphitheatre Parkway, M/S 535  
Mountain View, California 94043-1351