About This Guide

This guide describes the ways in which hardware devices are integrated into and controlled from a Silicon Graphics® computer system running the IRIX™ operating system version 6.2 and above.

Three general classes of device-control software exist in an IRIX system: process-level drivers, kernel-level drivers, and STREAMS drivers.

- A process-level driver executes as part of a user-initiated process. Examples include the use of programmed I/O (PIO) to the VME bus, and control of external interrupts in a Challenge™ system.
- A kernel-level driver is loaded as part of the IRIX kernel and executes in the kernel address space, controlling one device in response to calls to its read, write, and ioctl (control) entry points.
- A STREAMS driver is dynamically loaded into the kernel address space to monitor or modify a stream of data passing between a device and a user process.

All three classes are discussed in this guide, although the greatest amount of attention is given to kernel-level drivers.

Note: This edition applies only to IRIX 6.2 and later. If you are working with an earlier release (4.x, 5.2, 5.3, 6.0.x, or 6.1), you should use the version of this manual appropriate to that release.

Audience

In order to write a process-level driver you must be an experienced C programmer with a thorough understanding of the use of IRIX system services and, of course, detailed knowledge of the device to be managed.

In order to write a kernel-level driver or a STREAMS driver you must be an experienced C programmer who knows UNIX® system administration, and expecially IRIX system administration, and who understands the concepts of UNIX device management.

What This Guide Contains

This guide is divided into the following major parts.

Part I, "IRIX Device Integration"	How devices are attached to Silicon Graphics computers, configured to IRIX, and initialized at boot time.
Part II, "Device Control From Process Space"	Details of user-level interrupt handling, programmed I/O of VME and EISA devices, SCSI control using dslib, and external interrupts.
Part III, "Kernel-Level Drivers"	How kernel-level drivers are designed, compiled, loaded, and tested. Survey of kernel services for drivers.
Part IV, "VME Device Drivers"	Kernel-level drivers for the VME bus.
Part V, "SCSI Device Drivers"	Kernel-level drivers for the SCSI bus.
Part VI, "Network Drivers"	Kernel-level drivers for network interfaces.
Part VII, "EISA Drivers"	Kernel-level drivers for the EISA bus.
Part VIII, "GIO Drivers"	Kernel-level drivers for the GIO bus.
Part IX, "STREAMS Drivers"	Design of STREAMS drivers.

In the printed book, you can locate these parts using the part-tabs printed in the margins. Using IRIX Insight™, each part is a top-level division in the clickable table of contents, or you can jump to any part by clicking the blue cross-references in the list above.

Other Sources of Information

Developer Program

Information and support are available through the Silicon Graphics Developer Program. The Developer Toolbox CDROM contains numerous code examples. To join the program, contact the Developer Response Center at (800) 770-3033 or send e-mail to devprogram@sgi.com.

Internet Resources

A great deal of useful material can be found on the internet. Some starting points are in the following list.

SGI patches, examples, and other material.	ftp://ftp.sgi.com
Network of pages of information about Silicon Graphics and MIPS® products	http://www.sgi.com
Text of all Internet RFC documents.	ftp://ds.internic.net/rfc/
Computer graphics pointers at the UCSC Perceptual Science Labororatory.	http://mambo.ucsc.edu/psl/cg.html
Pointers to binaries and sources at The National Research Council of Canada's Institute For Biodiagnostics.	http://zeno.ibd.nrc.ca:80/~sgi/
A Silicon Graphics "meta page" at the Georgia Institute of Technology College of Computing.	http://www.cc.gatech.edu/service/sgimeta.html
Dazzling Silicon Graphics "meta page" at NASA in Huntsville, AL.	http://chernobog.msfc.nasa.gov/SGI/html/SGI.html
Complete SCSI-2 standard in HTML.	http://abekas.com:8080/SCSI2/
IEEE Catalog and worldwide ordering information.	http://stdsbbs.ieee.org:70/0/pub/htmlfiles/stctoc.htm
MIPS processor manuals in HTML form.	http://www.mips.com/

Standards Documents

The following documents are the official standard descriptions of buses:

- *EISA Technical Reference*, available from BCPR Services, Inc., 1400 L Street NW, Washington, DC 20005.
- *Intel 82350D Reference*, Intel™ order number 290377 (EISA reference implementation chip set).
- *ANSI/IEEE standard 1014-1987* (VME Bus), available from IEEE Customer Service, 445 Hoes Lane, PO Box 1331, Piscataway, NJ 08855-1331 (but see also "Internet Resources" on page xxxvii).

Important Reference Pages

The following reference pages contain important details about software tools and practices that you need.

getinvent(3)	The interface to the inventory database
hinv(1)	The use of the inventory display command
intro(7)	The conventions used for special device filenames
MAKEDEV(1)	The use of the program that creates device special files
master(4)	Syntax of files in */var/sysgen/master.d*
prom(1)	Commands of the "miniroot" and other features of the boot PROM, which you use to bring up the system when testing a new device driver
system(4)	Syntax of files in */var/sysgen/system/*.sm*
udmalib(3)	Functions for performing user-level DMA from VME.
uli(3)	Functions for registering and using a user-level interrupt handler.
usrvme(7)	Naming conventions for mappable VME device special files.

Additional Reading

The following books, obtainable from Silicon Graphics, can be helpful when designing or testing a device driver.

- *MIPS Compiling and Performance Tuning Guide*, document number 007-2479-001, tells how to use the C compiler and related tools.
- *MIPSpro Assembly Language Programmer's Guide*, document number 007-2418-001, tells how to compile assembly-language modules.
- *MIPSpro 64-Bit Porting and Transition Guide*, document number 007-2391-001, documents the implications of the 64-bit execution mode for user programs.
- *Topics in IRIX Programming*, document number 008-2478-002, documents some of the sophisticated services offered by the IRIX kernel to user-level programs.
- *MIPS R4000 User's Manual* (2nd ed.) by Joe Heinrich, document number 007-2489-001, gives detailed information on the MIPS instruction set and hardware registers for the processor used in many Silicon Graphics computer systems (also available as HTML on http://www.mips.com/).
- *MIPS R10000 User's Manual* by Joe Heinrich gives detailed information on the MIPS instruction set and hardware registers for the processor used in certain high-end systems. Available only in HTML form from http://www.mips.com/.
- *IRIX Administration: System Configuration and Operation*, document number 007-2859-001, describes the basic adminstrative tools for configuring, operating, and tuning IRIX.
- *IRIX Administration: Disks and File Systems*, document number 007-2825-001, describes the configuration of new disk subsystems and the management of logical volumes and file systems.
- *IRIX Administration: Peripheral Devices*, document number 007-2861-001, describes the adminstration of tapes, printers, and other devices.

The following books, obtainable from bookstores or libraries, can also be helpful.

- Egan, Janet I., and Thomas J. Teixeira. *Writing a UNIX Device Driver.* John Wiley & Sons, 1992.
- Leffler, Samuel J., et alia. *The Design and Implementation of the 4.3BSD UNIX Operating System*. Palo Alto, California: Addison-Wesley Publishing Company, 1989.
- A. Silberschatz, J. Peterson, and P. Galvin. *Operating System Concepts*, Third Edition. Addison Wesley Publishing Company, 1991.
- Heath, Steve. *VMEbus User's Handbook*. CRC Press, Inc, 1989. ISBN 0-8493-7130-9.
- *Device Driver Reference, UNIX SVR4.2*, UNIX Press 1992.
- *UNIX System V Release 4 Programmer's Guide*, UNIX SVR4.2. UNIX Press, 1992.
- *STREAMS Modules and Drivers, UNIX SVR4.2*, UNIX Press 1992. ISBN 0-13-066879.

Conventions Used in This Guide

Special terms and special kinds of words are indicated with the following typographical conventions:

Data structures, variables, function arguments, and macros.	The *dsiovec* structure has members *iov_base* and *iov_len*. Use the *IOVLEN* macro to access them.
Kernel and library functions and functions in examples.	When successful, **v_mapphys**() returns 0.
Driver entry point names that must be completed with a unique prefix string.	The **munmap**() system function calls the *pfx***unmap**() entry point.
Files and directories.	Device special files are in */dev,* and are created using the */dev/MAKEDEV* script.
First use of terms defined in the glossary (see "Glossary" on page 585).	The *inode* of a *device special file* contains the *major device number.*
Literal quotes of code examples.	The SCSI driver's prefix is `scsi_`.