

Customer Order Number 424510771-410  
NSC Publication Number 424510771-410A  
December 1986

Series 32000®

---

GENIX V.3™ Utilities Notes

© 1986 National Semiconductor Corporation  
2900 Semiconductor Drive  
P.O. Box 58090  
Santa Clara, California 95052-8090



## REVISION RECORD

REVISION	RELEASE DATE	SUMMARY OF CHANGES
A	12/86	First Release. <i>Series 32000</i> ® GENIX V.3™ Utilities Notes NSC Publication Number 424510771-410A.



## PREFACE

This manual provides information about the GENIX V.3<sup>TM</sup> system release. The manual describes new features of this release, how to upgrade from earlier System V releases, and how to change from a single-disk to a dual-disk system. The manual also contains software notes, additional upgrading and installation information and compatibility notes.

For a complete description of the procedures used in the administration of a *Series 32000*<sup>®</sup> computer running GENIX V.3, see the *GENIX V.3 Administrator's Guide*.

The *GENIX V.3 Product Overview* lists all the available software and documentation for the GENIX V.3 System release.

The information contained in this manual is for reference only and is subject to change without notice.

No part of this document may be reproduced in any form or by any means without the prior written consent of National Semiconductor Corporation.

---

GENIX V.3, ISE, ISE16, ISE32, and SYS32 are trademarks of National Semiconductor Corporation.

*Series 32000* is a registered trademark of National Semiconductor Corporation.

The GENIX V.3 Operating System is derived from AT&T's UNIX System V.3 Operating System. Portions of the documentation for the GENIX V.3 Operating System are derived from AT&T copyrighted UNIX V.3 Operating System documentation and reproduced under License from AT&T.



(

(

(

(



## CONTENTS

Chapter 1	UTILITIES NOTES . . . . .	1-1
1.1	INTRODUCTION . . . . .	1-1
1.1.1	Conventions Used in These Utilities Notes . . . . .	1-1
1.1.2	Upgrading to GENIX V.3. . . . .	1-2
1.2	FEATURES OF GENIX V.3 . . . . .	1-2
1.2.1	Remote File Sharing . . . . .	1-2
1.2.2	Networking Support Utilities . . . . .	1-3
1.2.3	Enhanced Basic Networking Commands . . . . .	1-3
1.2.4	Shared Libraries . . . . .	1-4
1.2.5	Command Syntax Standard . . . . .	1-4
1.2.6	Signal Mechanism Enhancements . . . . .	1-4
1.2.7	Improved Facilities for Supporting Terminals . . . . .	1-4
1.2.8	Additional Features (help, crash, encryption) . . . . .	1-5
1.3	UPGRADE PROCEDURES . . . . .	1-6
1.3.1	Upgrade Procedure 1: Release Upgrade . . . . .	1-7
1.3.2	Upgrade Procedure 2: Dual-Disk Upgrade . . . . .	1-13
1.4	ADDITIONAL INSTALLATION INFORMATION . . . . .	1-15
1.4.1	Misleading Help Message During Release Upgrade . . . . .	1-15
1.4.2	Not Having a Console . . . . .	1-15
1.4.3	Upgrading the Basic Networking Utilities . . . . .	1-15
1.4.4	Insufficient Help Message During Full Restore . . . . .	1-16
1.4.5	Messages About Partition Sizes May Be Wrong . . . . .	1-16
1.4.6	Rebooting the System Takes Time . . . . .	1-16
1.4.7	Ignorable Error Messages During Upgrade . . . . .	1-16
1.4.8	Unclear Error Message During Installation of Lp Spooler . . . . .	1-17
1.4.9	Existing Data Not Preserved When Installing Lp Spooler . . . . .	1-17
1.4.10	Printer Uses Wrong Filter . . . . .	1-17
1.4.11	TTY Management Menu . . . . .	1-17
1.4.12	Installing the Terminal Information Utilities . . . . .	1-17
1.4.13	Adding Printers to a Parallel Port . . . . .	1-18
Chapter 2	SOFTWARE NOTES . . . . .	2-1
2.1	INTRODUCTION . . . . .	2-1
2.2	USER COMMANDS . . . . .	2-1
2.2.1	Bc . . . . .	2-1
2.2.2	Cdc . . . . .	2-2
2.2.3	Cpio . . . . .	2-2
2.2.4	Ct . . . . .	2-2
2.2.5	Cu . . . . .	2-2
2.2.6	File . . . . .	2-3



2.2.7	Help . . . . .	2-3
2.2.8	Ipcs . . . . .	2-3
2.2.9	Login . . . . .	2-4
2.2.10	Lp . . . . .	2-4
2.2.11	Mailx . . . . .	2-4
2.2.12	Od . . . . .	2-4
2.2.13	Passwd . . . . .	2-4
2.2.14	Sar . . . . .	2-5
2.2.15	Sdiff . . . . .	2-5
2.2.16	Sh . . . . .	2-5
2.2.17	Shl . . . . .	2-5
2.2.18	Tee . . . . .	2-6
2.2.19	Tput . . . . .	2-7
2.2.20	Uname . . . . .	2-7
2.2.21	Uucp . . . . .	2-7
2.2.22	Uulog . . . . .	2-7
2.2.23	Uuto . . . . .	2-7
2.2.24	Vi . . . . .	2-8
2.2.25	/usr/news . . . . .	2-9
2.2.26	PATH . . . . .	2-9
2.3	PROGRAMMER COMMANDS, SYSTEM CALLS . . . . .	2-9
2.3.1	Ctime . . . . .	2-9
2.3.2	Dial, Undial . . . . .	2-9
2.3.3	Fork . . . . .	2-9
2.3.4	Terminfo . . . . .	2-9
2.3.5	Unlink . . . . .	2-10
2.4	SYSTEM ADMINISTRATOR COMMANDS . . . . .	2-10
2.4.1	Chroot . . . . .	2-10
2.4.2	Crash . . . . .	2-10
2.4.3	Cron . . . . .	2-10
2.4.4	Ctinfo . . . . .	2-11
2.4.5	Dcopy . . . . .	2-11
2.4.6	Dd . . . . .	2-11
2.4.7	Errdump . . . . .	2-11
2.4.8	Ff . . . . .	2-12
2.4.9	Finc . . . . .	2-12
2.4.10	Fmtflop . . . . .	2-12
2.4.11	Fsck . . . . .	2-12
2.4.12	Init . . . . .	2-13
2.4.13	Labelit . . . . .	2-14
2.4.14	Mkfs . . . . .	2-14
2.4.15	Mountfsys . . . . .	2-14
2.4.16	Prtconf . . . . .	2-14
2.4.17	Shutdown . . . . .	2-15
2.4.18	Sysadm . . . . .	2-15
2.4.19	Sysadm autold . . . . .	2-16
2.4.20	Sysadm backup . . . . .	2-16
2.4.21	Sysadm firmware . . . . .	2-16



	2.4.22	Sysadm portmgmt . . . . .	2-16
	2.4.23	Sysadm portmgmt delete . . . . .	2-16
	2.4.24	Sysadm portmgmt modify . . . . .	2-17
	2.4.25	Sysadm uucpmgmt . . . . .	2-17
	2.4.26	Sysdef . . . . .	2-17
	2.4.27	Swap . . . . .	2-18
	2.4.28	Umount . . . . .	2-18
	2.4.29	Uucheck . . . . .	2-18
	2.4.30	Uucheck, Uucleanup, Uutry . . . . .	2-18
	2.4.31	Uucico . . . . .	2-18
	2.4.32	Uutry . . . . .	2-19
	2.4.33	Volcopy . . . . .	2-19
	2.4.34	Path for Superuser Inconsistent . . . . .	2-19
2.5		MISCELLANY . . . . .	2-19
	2.5.1	Console . . . . .	2-19
	2.5.2	Kernel . . . . .	2-20
	2.5.3	Manual Page for Fs Format Is Incorrect . . . . .	2-21
	2.5.4	/etc/ports Does Not Insert Gettys in /etc/inittab . . . . .	2-22
	2.5.5	Tunable Parameters . . . . .	2-22
	2.5.6	Possibility of Data Corruption After a Power Outage . . . . .	2-22
2.6		FILE SYSTEM RESTORE . . . . .	2-23
	2.6.1	Error Messages During Complete Restore . . . . .	2-23
	2.6.2	Turn Off User Terminals During a Full System Restore or an Upgrade . . . . .	2-23
	2.6.3	Change Access Permission After Full Restore on Two- Disk System . . . . .	2-23
	2.6.4	Reinstall Utilities With Software Drivers After a Partial Restore . . . . .	2-24
	2.6.5	Saving Device Files When Backing Up Root File System . . . . .	2-24
	2.6.6	Contents of Sys Crontab Is Wrong . . . . .	2-24
	2.6.7	f450 Printer Needs Unsupplied Filters . . . . .	2-25
	2.6.8	Read/Write Permissions for Basic Networking Do Not Work . . . . .	2-25
	2.6.9	Primary Server Panics . . . . .	2-25
	2.6.10	Converting to Getopts by Hand . . . . .	2-26
Chapter 3		COMPATIBILITY NOTES . . . . .	3-1
	3.1	INTRODUCTION . . . . .	3-1
	3.2	CHAPTER OVERVIEW . . . . .	3-1
	3.3	CHANGES IN COMMANDS . . . . .	3-1
	3.3.1	Cc(1) 2.1 . . . . .	3-3
	3.3.2	Crash(1M) V.3 . . . . .	3-3
	3.3.3	Crypt(1) V.3 . . . . .	3-3
	3.3.4	Dirname(1) V.3 . . . . .	3-3
	3.3.5	Df(1M) V.3 . . . . .	3-4
	3.3.6	Ed(1) V.3 . . . . .	3-4
	3.3.7	Ex(1), Vi(1) V.3 . . . . .	3-4



3.3.8	File(1) V.3 . . . . .	3-5
3.3.9	Fuser(1M) V.3 . . . . .	3-5
3.3.10	Mail(1) V.3 . . . . .	3-5
3.3.11	Pr(1) V.3 . . . . .	3-5
3.3.12	Ps(1) V.3 . . . . .	3-6
3.3.13	Sh(1) V.3 . . . . .	3-6
3.3.14	Who(1) V.3 . . . . .	3-7
3.4	CHANGES IN SYSTEM CALLS . . . . .	3-7
3.4.1	Acct(2) V.3 . . . . .	3-7
3.4.2	Brk(2) 2.1 . . . . .	3-9
3.4.3	Exec(2) 2.1 . . . . .	3-10
3.4.4	Fcntl(2) 2.1 . . . . .	3-10
3.4.5	Fork(2) 2.1, V.3 . . . . .	3-11
3.4.6	Mount(2) 2.1 . . . . .	3-11
3.4.7	Plock(2) 2.1 . . . . .	3-11
3.4.8	Ptrace(2) V.3 . . . . .	3-11
3.4.9	Shmop(2) 2.1 . . . . .	3-11
3.4.10	Shmctl(2) 2.1 . . . . .	3-12
3.4.11	Signal(2) V.3 . . . . .	3-12
3.4.12	Umount(2) V.3 . . . . .	3-13
3.4.13	Ustat(2) V.3 . . . . .	3-14
3.5	CHANGES IN LIBRARY ROUTINES . . . . .	3-14
3.5.1	Abort(3C) V.3 . . . . .	3-14
3.5.2	Ctime(3C) V.3 . . . . .	3-14
3.5.3	Puts(3S) V.3 . . . . .	3-15
3.5.4	Fread(3S) V.3 . . . . .	3-16
3.5.5	Setbuf(3S) V.3 . . . . .	3-16
3.5.6	String(3C) V.3 . . . . .	3-17
3.5.7	Changes in the C Compilation System . . . . .	3-17
3.5.8	#ident 2.1 . . . . .	3-17
3.6	NEW ERROR CODES . . . . .	3-17
3.7	CHANGES IN HEADER FILES 2.1, V.3 . . . . .	3-19
3.7.1	Core.h 2.1 . . . . .	3-19
3.7.2	Errno.h 2.1, V.3 . . . . .	3-19
3.7.3	Fcntl.h 2.1, V.3 . . . . .	3-19
3.7.4	Ldfcn.h 2.1 . . . . .	3-21
3.7.5	Signal.h V.3 . . . . .	3-21
3.7.6	Stdio.h 2.1 . . . . .	3-21
3.7.7	String.h V.3 . . . . .	3-21
3.7.8	Sys/buf.h V.3 . . . . .	3-22
3.7.9	Sys/dir.h V.3 . . . . .	3-22
3.7.10	Sys/fblk.h V.3 . . . . .	3-23
3.7.11	sys/fcntl.h V.3 . . . . .	3-23
3.7.12	Sys/filsys.h V.3 . . . . .	3-23
3.7.13	Sys/inline.h V.3 . . . . .	3-23
3.7.14	Sys/inode.h 2.1, V.3 . . . . .	3-23
3.7.15	Sys/ipc.h 2.1 . . . . .	3-24



3.7.16	Sys/lo.h V.3 . . . . .	3-24
3.7.17	Sys/map.h 2.1 . . . . .	3-24
3.7.18	Sys/open.h 2.1 . . . . .	3-24
3.7.19	Sys/param.h 2.1, V.3 . . . . .	3-24
3.7.20	Sys/pcb.h 2.1 . . . . .	3-25
3.7.21	Sys/proc.h 2.1, V.3 . . . . .	3-25
3.7.22	Sys/psw.h 2.1 . . . . .	3-25
3.7.23	Sys/sbd.h 2.1 . . . . .	3-25
3.7.24	Sys/shm.h 2.1 . . . . .	3-25
3.7.25	Sys/stat.h V.3 . . . . .	3-26
3.7.26	Sys/sysmacros.h V.3 . . . . .	3-26
3.7.27	Sys/systm.h 2.1, V.3 . . . . .	3-26
3.7.28	Sys/tty.h 2.1 . . . . .	3-26
3.7.29	Sys/types.h 2.1 . . . . .	3-27
3.7.30	Sys/user.h 2.1, V.3 . . . . .	3-27
3.7.31	Sys/fs/s5dir.h V.3 . . . . .	3-27
3.7.32	Sys/fs/s5filsys.h V.3 . . . . .	3-27
3.7.33	Sys/fs/s5param.h V.3 . . . . .	3-27
3.7.34	Sys/fs/s5inode.h V.3 . . . . .	3-27
3.7.35	Sys/fs/s5fblk.h V.3 . . . . .	3-28
3.7.36	NSC Miscellaneous Changes . . . . .	3-28
3.7.37	Brc Entry in /etc/inittab V.3 . . . . .	3-28
3.7.38	Longest Allowed Pathnames V.3 . . . . .	3-28
3.8	CHANGES UNDER SHARED LIBRARIES . . . . .	3-28
3.8.1	Chroot(1) V.3 . . . . .	3-29
3.8.2	Exec(2) V.3 . . . . .	3-29
3.8.3	Mount(2) V.3 . . . . .	3-29
3.9	DOCUMENTATION . . . . .	3-30

## TABLES

TABLE 1-1.	GENIX V.3 UTILITIES INSTALLATION CHART . . . . .	1-14
TABLE 3-1.	SUMMARY OF CHANGED COMMANDS . . . . .	3-2
TABLE 3-2.	SUMMARY OF CHANGED SYSTEM CALLS . . . . .	3-8
TABLE 3-3.	DEADLOCK DETECTING SYSTEM CALLS . . . . .	3-9
TABLE 3-4.	SUMMARY OF CHANGED LIBRARY ROUTINES . . . . .	3-15
TABLE 3-5.	ERROR CODES . . . . .	3-18
TABLE 3-6.	CHANGED, DROPPED, OR ADDED HEADER FILES . . . . .	3-20

## INDEX



(

(

(

(



# Chapter 1

## UTILITIES NOTES

### 1.1 INTRODUCTION

These *Utilities Notes* contain important information about the GENIX V.3™ System release. First, they briefly describe the new features of this release. Next, they tell you how to upgrade from earlier releases of GENIX™ and how to change from a single-disk to a dual-disk system. Software notes, additional information about upgrading and installation and compatibility notes are also provided. Finally, these *Utilities Notes* list the documents that pertain to GENIX V.3.

For a comprehensive description of the software and documentation available for the GENIX V.3 release, see the *Product Overview*. For a complete description of the procedures used in the administration of a Series 32000® Computer running GENIX V.3, see the *Administrator's Guide*.

#### 1.1.1 Conventions Used in These Utilities Notes

In this document, as in all GENIX V.3 System documentation, certain typesetting conventions are followed when command names, command line format, files, and directory names are described. There are also conventions for displays of terminal input and output.

- You must type words that are in **bold** font as they appear.
- *Italic* words are variables; you substitute the appropriate values. These values may be file names or they may be data values, as applicable.
- Characters or words in square brackets, [ ], are optional. (Do not type the brackets.)

A command name followed by a number, for example, **ed**(1), refers to that command's manual page, where the number refers to the section of the manual. Manual pages from Section (1) appear in the *User's Reference Manual*, unless otherwise noted. Manual pages from Sections (3) and (4) appear in the *Programmer's Reference Manual*. Manual pages from section (1M) appear in the *Administrator's Reference Manual*.

Examples in these *Utilities Notes* show the default system prompt for GENIX V.3, the dollar sign (\$). They also show the default prompt when you login as the superuser, the pound sign (#).

These *Utilities Notes* refer to packages and to products. A package is a group of programs that do related things. For example, the Editing Utilities package contains GENIX V.3's text editors and their associated files. The GENIX V.3 release comprises many packages. A product is something that you purchase independently of GENIX V.3.



### 1.1.2 Upgrading to GENIX V.3.

In contrast to a Full Restore, which erases everything on the hard disk, an upgrade changes specific data (for example, kernel parameters) while preserving user files. To upgrade a system running System V, Release 2.0, Release 2.0.4, or Release 2.1 to this release, please follow Section 1.3.1. To change your *Series 32000* Computer from a single-disk to a dual-disk system, first upgrade your system to GENIX V.3, and then follow Section 1.3.2. To do a Partial or Full Restore of your system, refer to the *Administrator's Guide*.

The System Administration menus help you set up the system. The **sysadm** command executes these menus during the upgrade of the utilities packages that come with the GENIX V.3 system.

## 1.2 FEATURES OF GENIX V.3

GENIX V.3 provides the following new features. For a more detailed description of them, see the *Product Overview*.

- Remote File Sharing (optional product)
- Networking Support Utilities (optional product)
- Enhanced Basic Networking Utilities
- Shared Libraries
- Command Syntax Standard
- Signal Mechanism Enhancements
- Improved Terminal Support Facilities
  - Terminal Information Utilities Enhancements
- Additional Features
  - **help** Facility Extensions
  - **crash(1M)** Command Changes
  - New System Header Files
  - Encryption Mechanisms Repackaged

### 1.2.1 Remote File Sharing

**NOTE:** To take advantage of Remote File Sharing, you must have Remote File Sharing Utilities, the Networking Support Utilities, and a transport provider. (See Section 1.2.2 for a description of a transport provider.) These optional products require at least 2 megabytes of main memory.

Remote File Sharing lets you share files, directories, devices, and named pipes transparently among computers that are linked by a network. Files are shared transparently by mounting a remote directory as one would mount a file system. Each computer on the network controls which local resources are available to other computers and which remote resources local users



may access. For example, with Remote File Sharing you may share a directory among several departments of your business, or a letter-quality printer or typesetter that no one department could support by itself. For more information, see the *Remote File Sharing Utilities Notes* and the *Administrator's Guide*.

### 1.2.2 Networking Support Utilities

NOTE: The Networking Support Utilities is an optional product that requires at least 2 megabytes of main memory.

The Networking Support Utilities provide STREAMS tools, the AT&T Transport Interface, and the Listener. STREAMS is a set of tools for development of communication and networking services within the GENIX V.3 system; the Transport Interface is based on the Transport Service Definition (Level 4) of the International Organization for Standardization (ISO) Open Systems Interconnection (OSI) reference model, and defines how a user accesses the services of a transport protocol; the Listener receives requests for network services from another system, interprets which network service is needed, and starts a process that has been named to provide the requested network service. The Listener then drops out of the communications path and continues to listen for new service requests.

For more information about the Networking Support Utilities, see the *Networking Support Utilities Utilities Notes*; for more information about the Listener, see these utilities notes and the *Administrator's Guide*.

### 1.2.3 Enhanced Basic Networking Commands

NOTE: Media-independent basic networking commands are provided in the Basic Networking Utilities package, but they require the optional Networking Support Utilities product to be used.

Basic networking commands (for example, `uucp(1C)` and `uux(1C)`) have been enhanced to conform to the Transport Interface (see the *Networking Support Utilities Notes* for details). These utilities can communicate using any Transport Provider that conforms to the Transport Interface. The operation of the Basic Networking Utilities commands is the same regardless of whether you install the Networking Support Utilities product.

You can show which, if any, Transport Providers are available with `nlsadmin -x`. If you install additional Transport Providers, no changes are needed to the software to accommodate the underlying media or protocols; you need only register Basic Networking Utilities services with the network Listener for those Transport Providers, and register the Transport Providers in the administrative files for the Basic Networking Utilities (see `nlsadmin(1M)` in the *Administrator's Reference Manual*).

Even with these enhancements, the syntax of the basic networking commands is no different from before. See the *Administrator's Guide* for details about how to manage this facility.



### 1.2.4 Shared Libraries

The Advanced Programming Utilities product and the C Programming Language Utilities product allow the programmer to build a shared library of routines accessed at run-time, rather than having those routines combined with an application program at load time.

On GENIX V.3 systems, two shared libraries are available: the most commonly-used routines from the C Library and the Networking Services Library. Most of the GENIX V.3 system commands use these two libraries, as do any applications that were built with them. For more information about generating Shared Libraries, see the chapter devoted to this topic in the *Programmer's Guide*.

### 1.2.5 Command Syntax Standard

**Getopt(1)** allows shell procedures to parse command lines to check for legal options and to process option arguments. A new command, **getopts(1)**, is an enhanced version of the **getopt(1)** command. **Getopts** is consistent with and supports rules 3-10 of the GENIX V.3 system command syntax standard. (The standard is described on the **intro(1)** manual page.)

NOTE: You may use **getopt** in shell scripts with the GENIX V.3 System. However, you should use **getopts** instead of **getopt**; beginning with the next major GENIX V.3 System release, **getopt** will no longer be supported.

To assist in the conversion of shell scripts that are affected by a change from **getopt** to **getopts**, a conversion command, **/usr/lib/getoptcvt**, is provided. (See the **getopts** manual page for details.)

### 1.2.6 Signal Mechanism Enhancements

A new set of system calls (see the **sigset(2)** manual page in the *Programmer's Reference Manual*) provides a mechanism to catch and hold signals without losing them during later processing, and to guarantee that a process reaches the signal handler before it is interrupted by another signal. Some additional signal-handling features, provided by other popular operating systems, are also available.

### 1.2.7 Improved Facilities for Supporting Terminals

Support for terminals is improved with new features of the Terminal Information Utilities.



### 1.2.7.1 Terminal Information Utilities

The “Terminal Information Utilities” package (often called **curses/terminfo**) has the following new features:

- expanded support for terminal filters, soft labels, and new AT&T terminals
- new commands: **captoinfo**(1M) converts **termcap** entries to **terminfo** entries; **infocmp**(1M) compares two **terminfo** entries or prints entries in several formats. (Section (1M) is in the *Administrator's Reference Manual*.)
- new options to the **tput**(1) command to initialize, reset, and learn the “long name” of a terminal
- an improved version of the **terminfo** compiler, **tic**
- new documentation on the manual pages and in the “**curses and terminfo**” chapter of the *Programmer's Guide* (Chapter 11)

These new **curses** features are only available with the V.3 version of **curses** on the GENIX V.3 System. (See also **curses**(3X) in the *Programmer's Reference Manual*.) All programs that ran with System V, Release 2 **curses** will run with GENIX V.3 You may link applications with object files based on the Release 2 **curses/terminfo** with the GENIX V.3 **libcurses.a** library. You may link applications with object files based on the GENIX V.3 **curses/terminfo** with the GENIX **libcurses.a** library, as long as the application does not use the new features in the GENIX V.3 **curses/terminfo**.

### 1.2.8 Additional Features (help, crash, encryption)

GENIX V.3 provides additional information in the **help** facility, improvements to the **crash**(1M) command, and repackaging of encryption mechanisms.

#### 1.2.8.1 Help Facility Extensions

Descriptions and examples of many additional commands, terms, and symbols have been added. See **help**(1), **glossary**(1), **starter**(1), and **usage**(1).

#### 1.2.8.2 Crash Command Changes

In addition to providing debugging support for the new operating system features included in GENIX V.3, **crash** has been changed extensively to make it easier to use. The syntax of all the functions has been standardized so that similar functions share similar syntax. There is a **help** function within **crash**, a number base converter, a memory search function, and a disassembler capability. The **crash** (1M) manual page in the *Administrator's Reference Manual* describes the details of this new **crash** command.



### 1.2.8.3 New System Header Files

New header files were added to `/usr/include`: **unistd.h** (definitions for symbolic constants introduced and used throughout the `/usr/group` Standards document, see **unistd(4)** in the *Programmer's Reference Manual*) and **limits.h** (definitions for commonly used values that vary from implementation to implementation, see **limits(4)** in the *Programmer's Reference Manual*). Several new definitions were added to the header file `/usr/include/sys/stat.h` to make it easier for programmers to write portable code.

### 1.2.8.4 Encryption Mechanisms Repackaged

This release introduces several changes in the implementation of the encryption mechanisms. In particular, the text editors that are part of the "Editing Utilities" package are no longer duplicated in the "Security Administration Utilities" package.

NOTE: The Security Administration Utilities package has restricted distribution and is provided only with *Series 32000* Computers sold within the United States.

## 1.3 UPGRADE PROCEDURES

There are two upgrade procedures available for the GENIX V.3 System release:

1. The first, a Release Upgrade procedure, allows you to upgrade an earlier GENIX V.3 release or to restore a system if its Shared Libraries become damaged without loss of any user files.
2. The second, a Dual-Disk Upgrade procedure, allows you, after Release Upgrade, to convert your *Series 32000* Computer from a single-disk to a dual-disk system without loss of any user files.

Two other important procedures are described in the *Administrator's Guide*: one lets you partially restore the system to bring it to a usable state or to remove a forgotten **root** password, and the other lets you fully restore the system if there is a new system or disk, or if you need to increase the size of the swap, `/` or `/usr` partition by repartitioning the disk. (See "Partial System Restore" and "Full System Restore" in Procedure 3.9: Reload the Operating System.)

Six core system diskettes labeled "Essential Utilities Disk — 1" through "Essential Utilities Disk — 6" contain the basic operating software for GENIX V.3. (The terms "core system" diskettes and "Essential Utilities" diskettes are synonymous.) Each of the upgrade procedures transfers files from diskette to the hard disk. The file transfer is automated by menu-driven programs. Adhere to the following instructions carefully to guarantee a smooth upgrade.



### 1.3.1 Upgrade Procedure 1: Release Upgrade

<b>Purpose</b>	To upgrade a system running System V, Release 2.0 to GENIX V.3, or to restore a system if its Shared Libraries become damaged.
<b>Machine</b>	<i>Series 32000</i> Computers.
<b>Current Release</b>	Any version of System V, Release 2.
<b>Media</b>	The six <i>Series 32000</i> Essential Utilities diskettes.
<b>Time</b>	1-2 hours.

The Release Upgrade procedure allows you to upgrade your current release to GENIX V.3. Here are the results of the procedure:

- The new core system is installed from the six diskettes to the hard disk.
- Certain system and user configuration files are moved temporarily to `/usr/old`. When the upgrade is finished, most of these files are moved back to where they were. See Step 7 below for a list of files that you must check after the upgrade.
- The boot programs are updated.
- The labels for `root (/)` and `/usr` are changed to reflect GENIX V.3 release.

#### 1.3.1.1 Before You Upgrade

#### CAUTION

Do not attempt this procedure when sudden loss of power is likely (for example, during electrical storms). Loss of power during certain parts of the upgrade may cause loss of user data.



## CAUTION

It is highly recommended that you backup the hard disk(s) on floppy or cartridge tape before the Release Upgrade. (See Procedure 5.4, "File System Backup and Restore," in the *Administrator's Guide*.)

You should backup individual login directories and save the following files (if they exist on your system) on floppies or cartridge tape before any core package installation:

```
/dgn/edt_data
/etc/fstab
/etc/gettydefs
/etc/group
/etc/inittab
/etc/master.d/kernel
/etc/master.d/msg
/etc/master.d/sem
/etc/master.d/shm
/etc/passwd
/etc/profile
/etc/system
/usr/lib/uucp/Systems
/usr/lib/uucp/Devices
/usr/lib/uucp/Poll
/usr/lib/uucp/Permissions
```

Once the automated Release Upgrade procedure from Essential Utilities Disk 1 begins, do not interrupt the process.

- If you are installing files from the first core floppy and the upgrade is interrupted for any reason, restart the upgrade from the beginning.
- If you are installing files from the second, third, fourth, or fifth core floppy and the upgrade is interrupted for any reason, you should be able to boot from the hard disk and continue where you left off.
- When you are installing files from the sixth core floppy, if the upgrade is interrupted for any reason after you have received the following message

You may now remove the last *Series 32000* Core System floppy

but before you receive the following message

Installation is now complete

you must restart from the beginning.

Any time that the upgrade fails, do a Full System Restore (see Procedure 3.9: Reload the Operating System in the *Administrator's Guide*). With a Full System Restore, all user files are lost unless you have backed them up.



### 1.3.1.2 Upgrading Your Current Release

Here are the steps to upgrade your current release to GENIX V.3:

- Step 1 Log in as **root**. Check the available space in **/** and **/usr**, for example, with **sysadm diskuse** or with **df -t(1M)**. (Section (1M) is in the *Administrator's Reference Manual*.) You must have at least 800 blocks free in **/** and 600 blocks free in **/usr** to do a Release Upgrade. You need more space than this to reinstall all the other packages and products available.

Enter **sysadm firmware** to the shell prompt to bring the system down to firmware mode, and then go to firmware in express mode (answering **y** to the appropriate prompt, as below). (This example shows upgrading from System V, Release 2.1.)

```
Console Login: root <CR>
Password: <password> <CR>
GENIX V.3 System V Release 2.1.0 AT&T Series 32000 Version 2
unix
Copyright (c) 1984 AT&T
All Rights Reserved
# df -t <CR>
.
.
# sysadm firmware <CR>
```

Running subcommand 'firmware' from menu 'machinemgmt'.  
MACHINE MANAGEMENT

Once started, this procedure CANNOT BE STOPPED.  
Do you want to go to firmware "express"? [y, n, q, ?] **y** <CR>

Shutdown started. Thurs Jul 3 12:49:31 EDT 1986

Broadcast Message from root (console) Thurs Jul 3 12:49:34 ...  
THE SYSTEM IS BEING SHUT DOWN NOW !!!  
Log off now or risk your files being damaged.

INIT: New run level: 5  
The system is coming down. Please wait.  
System services are now being stopped.

The system is down.

SELF-CHECK

FIRMWARE MODE



Step 2 Insert diskette 1 of the Essential Utilities Disks into the diskette drive.

### CAUTION

Make sure that diskette 1 is not write protected.

Step 3 Enter the firmware password (**mcp**) or new firmware password (if you have changed it).

*password*

Step 4 Enter **unix** in response to the following prompt:

Enter the name of program to execute [ ] **unix** <CR>

Step 5 Enter **0** in response to the following prompt:

Enter Load Device Option Number [1, (HD30)] **0** <CR>

(The default device displayed may not be "HD30," but you may ignore this discrepancy safely.)

Step 6 After copyright information is displayed, you are asked to enter one of the following options:

1. Full Restore
2. Partial Restore
3. Dual-Disk Upgrade
4. Release Upgrade

Since you are performing a release upgrade to GENIX V.3, enter option 4 in response to the following prompt:

Selection? [ 1 2 3 4 quit help ] **4** <CR>

Step 7 From this point, the installation program is interactive; it prompts you through the release upgrade procedure. Install the diskettes in sequence. Leave diskettes 2 through 6 write protected. You can enter **help** when you want additional information, or you can enter **quit** to stop the upgrade procedure. If you enter **quit** and later reboot the system, the system will prompt you to pick up where you left off in the upgrade.

The Release Upgrade procedure puts the following configuration files in **/usr/old**, and then does one of three things with them:

1. upgrades them and then moves them back
2. keeps them there, replacing them with new versions
3. puts new versions of them in different locations



Here are files that may be upgraded by this procedure, depending on what release you upgrade from.

- /etc/gettydefs
- /etc/inittab
- /etc/profile

Here are files that are put into **/usr/old** and kept there; the Release Upgrade puts new versions of the files where they had been:

- /bin/ed
- /bin/red
- /etc/system
- /etc/master.d/hdelog
- /etc/master.d/uart
- /etc/master.d/idisk
- /etc/master.d/kernel
- /etc/master.d/mem
- /etc/master.d/stubs
- /etc/master.d/gentty
- /etc/master.d/s5
- /etc/master.d/ports

Here are files, new for the GENIX V.3 release, that are put into **/usr/old** and kept there if you do a Release Upgrade from a system already running GENIX V.3 (for example, to restore a system if its Shared Libraries become damaged).

- /etc/master.d/disp
- /etc/init.d/ANNOUNCE
- /etc/init.d/MOUNTFSYS
- /etc/init.d/README
- /etc/init.d/RMTMPFILES
- /etc/init.d/autoconfig
- /etc/init.d/cron
- /etc/init.d/disks
- /etc/init.d/firstcheck
- /etc/init.d/sysetup
- /etc/init.d/uucp
- /etc/rc0.d/K00ANNOUNCE
- /etc/rc0.d/K70cron
- /etc/rc2.d/S00firstcheck
- /etc/rc2.d/S01MOUNTFSYS
- /etc/rc2.d/S05RMTMPFILES
- /etc/rc2.d/S10disks
- /etc/rc2.d/S15autoconfig
- /etc/rc2.d/S20sysetup
- /etc/rc2.d/S70uucp
- /etc/rc2.d/S75cron



Here are files that are put into **/usr/old** and are not replaced; if these files already exist on your machine (which depends on the release that you are running before you upgrade), the system puts new versions of the files in a different location:

```
/etc/shutdown.d/ANNOUNCE
/etc/rc.d/0_firstcheck
/etc/rc.d/ANNOUNCE
/etc/rc.d/MOUNTFILESYS
/etc/rc.d/README
/etc/rc.d/RMTMPFILES
/etc/rc.d/acct
/etc/rc.d/autoconfig
/etc/rc.d/cron
/etc/rc.d/disks
/etc/rc.d/ports
/etc/rc.d/sysetup
/etc/rc.d/uucp
```

NOTE: Directories created in **/usr/old** by the Release Upgrade procedure are given mode 777. Also, **/usr/old/README** is given mode 666. This allows any user to remove the files left in **/usr/old** by the upgrade procedure.

Some file modes and owners differ depending on whether you do a "Release Upgrade" or "Full Restore." These differences arise because the old files are kept during a "Release Upgrade." The following files are affected:

```
/dgn/edt_data
/etc/fstab
/etc/gettydefs
```

**Step 8** After you have installed all files from the core floppies, the "Console Login:" prompt will reappear. Log in as **root**. Run **shutdown(1M)**, and then mount all file systems with **mountall(1M)**. (Section (1M) is in the *Administrator's Reference Manual*.)

Remove old utility packages with **sysadm removepkg**. Use the floppies that you used to install the package originally.

Then, install utilities packages with **sysadm installpkg**, which starts another interactive procedure, and tells you if you load utilities out of order. You must have the floppies available for the packages that you want to replace.

### CAUTION

If you do not remove packages before you re-install them, you may not have enough space for the new versions, or the installation may fail, or both.



Install the System Administration Utilities first, then the Directory and File Management Utilities. Install the remaining utilities packages in the order suggested by Table 1-1.

When you install a package that contains a driver, a message containing the following text is displayed at the end of the installation procedure:

Execute "shutdown —i6 —g0 —y"

During a Release Upgrade, Full Restore, or Partial Restore, you do not have to execute **shutdown —i6 —g0 —y** after installing each package. You may wait until all packages are installed to run **shutdown**.

NOTE: You must reinstall the following packages if you have them:

- System Administration Utilities
- Directory and File Management Utilities
- User Environment Utilities
- Inter-Process Communication Utilities
- Editing Utilities
- Performance Measurement Utilities
- Security Administration Utilities (domestic customers only)
- Cartridge Tape Utilities
  - (if your system is equipped with a Cartridge Tape Controller)
- Networking Support Utilities
- Remote File Sharing Utilities

In addition, you must reinstall any other packages that install a device driver. Dependencies between GENIX V.3 utilities are described in Table 1-1.

Step 9 After you finish installing utility packages, compare (for example, with the **diff** command) configuration files in **/etc/master.d** and **/etc/system** (those not saved and replaced) with their copies that still reside in **/usr/old** (See Step 7). Some differences between the old and new files may be because of your tuning rather than the upgrade. There also can be differences because of new and deleted tunables with the GENIX V.3 release. Edit the new files rather than copying them from **/usr/old**.

Step 10 Reboot the system by typing **shutdown —i6 —g0 —y**.

### 1.3.2 Upgrade Procedure 2: Dual-Disk Upgrade

Purpose	To convert the system from a single-disk to a dual-disk system.
Machine	Series 32000 Computers.
Current Release	Release V.3
Media	Only the first diskette of the six Series 32000 Essential Utilities diskette.
Time	40 to 90 minutes.



**TABLE 1-1. GENIX V.3 UTILITIES INSTALLATION CHART**

<b>UTILITIES</b>	<b>SOFTWARE DEPENDENCIES</b>	<b>DRIVERS INSTALLED</b>	<b>FREE BLOCKS NEEDED IN / (root)</b>	<b>FREE BLOCKS NEEDED IN /usr</b>
System Administration	None	No	700	10
Directory and File Management	None	No	190	920
User Environment	Directory and File Management	Yes	190	670
Inter-Process Communication	None	Yes	210	30
Terminal Filters	None	No	20	170
Terminal Information	None	No	170	1200*
Graphics	User Environment Terminal Filters	No	20	4000
Basic Networking	User Environment	No	20	1480
Editing	None	No	20	480
Help	User Environment Terminal Information	No	30	1100
Line Printer Spooling	User Environment	No	20	740
Performance Measurement	None	Yes	70	520
Security Administration	Terminal Information	No	90	70
Spell	Directory and File Management	No	20	470
* The number of free blocks in /usr needed when files provided on the first floppy of the package are installed. If you install all files on the second floppy, an additional 1390 blocks are needed in /usr.				



The Dual-Disk Upgrade procedure allows you to convert the system on your *Series 32000* Computer from a single-disk to a dual-disk system after you have done a Release Upgrade. This procedure does not upgrade your system; use the Release Upgrade procedure to bring your system to GENIX V.3 before doing a Dual-Disk Upgrade.

## 1.4 ADDITIONAL INSTALLATION INFORMATION

### 1.4.1 Misleading Help Message During Release Upgrade

During the Release Upgrade, one **help** message lists the following features:

Upgrade features include:

- Floating boot
- Floating point assist in software support
- XM (expansion module) software support

Floating boot is not provided by the system. Floating point assist and XM support are features that are provided by system software regardless of whether you install the system with the Release Upgrade or with some other procedure.

### 1.4.2 Not Having a Console

If you turn on a *Series 32000* Computer without having a terminal connected to the console port, initialization does not complete. When you connect a terminal, initialization completes.

### 1.4.3 Upgrading the Basic Networking Utilities

The Integral Modem package overwrites the **sysadm** menu from the Basic Networking Utilities package.

Re-install the Integral Modem after you have re-installed the Basic Networking Utilities. Otherwise, you cannot access the Integral Modem. If your Basic Networking Utilities package can access the Integral Modem, it cannot access NPACK, and vice versa; the package cannot access both the Integral Modem and NPACK.

You must install User Environment Utilities before you install the Basic Networking Utilities because the latter depends on **crontab**.



#### 1.4.4 Insufficient Help Message During Full Restore

During a Full Restore, if you respond **help** to the following question, the message displayed does not state the sizes of the default partitions:

Use the default partitioning?

Refer to the "Hard Disk Default Partitions" section in Appendix A of the *Administrator's Guide* for this information.

#### 1.4.5 Messages About Partition Sizes May Be Wrong

When you do a Full Restore on a system with two disks and you do not choose the default partitioning, you get the following message:

How many blocks for the "usr" partition?

Suppose that you asked for **help** after this message. The text that appears after you ask for help contains one statement that may be wrong, and one that could be misleading.

In the following message, the *number* displayed with the message is incorrect when the two disks have different cylinder boundaries and you have **/usr** on the second disk:

NOTE: partition sizes are always rounded up to cylinder *number* boundaries

Nonetheless, partition sizes are rounded correctly, so ignore the number displayed.

The *number* displayed with the following message refers to the amount of space that **/usr** would take if you chose the default configuration, which puts **/usr** on the second disk.

The default "usr" partition size for your disk is *number* blocks

#### 1.4.6 Rebooting the System Takes Time

When you reboot the system after Diagnostics passes, you may wait a long time for a prompt. This does not mean that the system is hung.

#### 1.4.7 Ignorable Error Messages During Upgrade

If you have boards plugged into your *Series 32000* Computer, you see the following message during a Release Upgrade between the installation of the first and second core floppies while the system is rebooting:

SELF CHECK  
UNKNOWN ID CODE 0X6 for device in SLOT 2  
Equipt. dev. table complete will continue  
Check EDT



Ignore these messages.

#### 1.4.8 Unclear Error Message During Installation of Lp Spooler

When you attempt to install the **lp** spooler package for the C-ITOH printer, the following error message is displayed:

Installation of printer dqp101 failed. Save error messages.

Record the error messages showing which printer installation failed.

#### 1.4.9 Existing Data Not Preserved When Installing Lp Spooler

After installing the **lp** spooler package it seems that the existing printer data has been preserved because the command **lpstat -t** reports the expected printer names, but then the command while trying to get the 'accept' status of each.

Make sure existing data are saved before installing the **lp** spooler package.

#### 1.4.10 Printer Uses Wrong Filter

The 5310 printer uses **/usr/bin/53filter** as the default filter even though this file does not exist. Use **/usr/lib/5310**.

#### 1.4.11 TTY Management Menu

The TTY Management Menu of the System Administration Menus has changed for both System V, Release 2 and GENIX V.3. Please refer to your system manual for details.

#### 1.4.12 Installing the Terminal Information Utilities

The install script for the Terminal Information Utilities package (floppies 1 and 2) does not check if there is enough space for the package in the **root (/)** and **/usr** file systems before installing it.

The space requirements for this package are as follows. Floppy 1, which contains the **curses(3X)** library and various support utilities, must be completely installed. The first floppy requires about 150 blocks in the **/** file system and about 1000 blocks in the **/usr** file system. Floppy 2 contains the **terminfo(4)** database that you may install completely or partially (that is, selectively by terminal entry). The entire **terminfo(4)** database requires about 1300 blocks in the **/usr** file system. Each **terminfo(4)** entry requires, on the average, about 1 block.



#### **1.4.13 Adding Printers to a Parallel Port**

The script for adding printers to the lp system does not include a choice for a parallel port. Enter **/dev/tty15** for a parallel port on the first ports board, **/dev/tty25** for a parallel port on the second ports board, and so on, when this script prompts for the device name.



## Chapter 2

### SOFTWARE NOTES

#### 2.1 INTRODUCTION

This chapter offers some additional information about the GENIX V.3 System release. Notes about commands, system calls, or files are listed alphabetically and are organized by the Reference Manual where those commands, system calls, or files appear. For example, Section 2.2 contains notes about commands that are listed in the *User's Reference Manual*. Any further problems may be resolved through the Customer Support Center according to the terms of your maintenance contract.

#### 2.2 USER COMMANDS

##### 2.2.1 Bc

When you enter the following command line and **bc** cannot open *file2*, the error message displayed says that **bc** cannot open *file1*:

```
bc file1 file2
```

Also, when you enter the following command line and **bc** cannot open *file*, the error message displayed contains garbled characters instead of the name of **file**:

```
bc file
```

**Bc** does not handle the following two constructs in the same way:

- (1)    **if ( expr ) {**  
          ...  
          **}**
- (2)    **if ( expr )**  
          {  
          ...  
          **}**

The first case produces what one would expect. The second case is equivalent to an **if** followed by an empty statement, and the compound statement always is executed. If nothing else, the second case should produce a syntax error, but it does not. It dumps core silently.



### 2.2.2 Cdc

The `cdc(1)` command ends abnormally when you invoke it without the `—m` option on an SCCS file which does not have the `v` flag set.

### 2.2.3 Cpio

When you run a series of `cpio` processes in the background, several of them may stop. This occurs because `cpio` creates many child processes, and the maximum number of processes for a user may be exceeded.

You should avoid running too many `cpio` processes in the background.

### 2.2.4 Ct

The `ct` command is not compatible with the Basic Networking Utilities. Do not use this command.

### 2.2.5 Cu

The `call` option under `sysadm uuicpmgmt` leaves two processes running. To correct this, hit `<BREAK>` after exiting the menu.

When you use the `cu` command over a direct line, you should enter a carriage return to get a login prompt. After the login prompt appears, wait about 5 seconds before entering your login. If you do not wait, the first character that you type is not displayed on your terminal screen even though it is accepted by the computer.

This only happens when you use `uugetty`.

The first invocation of `cu` after a power up may fail; the error message “cannot access device” is displayed. Later invocations of `cu` succeed.

If you do a `cu` from machine\_A to machine\_B, and then do a `cu` from machine\_B to machine\_C, the command `~%take file` will *not* transfer *file* to machine\_A.

Transfer files over one link at a time by one of two methods.

- Login to machine\_A, `cu` to machine\_B and then to machine\_C as described above. The command `~%take file` transfers *file* to machine\_B. Then type `~.` (tilde tilde dot). From machine\_B, use the command `~%take file` to transfer *file* to machine\_A.
- If possible, `cu` from machine\_A to machine\_C. You may successfully transfer files over a single link.

Occasionally, `cu` fails with the following message even though all devices are available:

NO DEVICES AVAILABLE



### 2.2.6 File

When you `cc -c main.c`, file `main.o` does not produce the correct output. `File` interprets the optional header, which is not present in the `.o` file.

The `file(1)` command reports that a file containing packed data is “data” instead of “packed data.”

### 2.2.7 Help

The GENIX V.3 system `help` facility is incomplete. Also, the commands of the `help` facility ignore command line options.

The commands of the `help` facility ignore `SIGHUP`, and stay attached to a port that has been dropped or disconnected. This inattentiveness to `SIGHUP` makes the port useless; you cannot login because there are two processes reading the port (`help`'s and `login`'s).

Exit `help`.

### 2.2.8 Ipc

`Ipc(1)` always reports the number of processes attached to shared memory segments, `NATTCH`, as zero, even when running processes are currently attached to shared memory segments.

This information in the `NATTCH` column is incorrect. To obtain the correct information, follow these steps:

- Step 1     Invoke `crash` as `root`.
- Step 2     Type `od shminfo 3`. The last word on the line will be the number (in hexadecimal of shared memory identifiers).
- Step 3     Now type `od shmем 6`.
- Step 4     Examine the first `shmем` identifier. If the leftmost digit of the third word is in the range 8 to f (hexadecimal) the identifier is in use, and you can get the address for the next step in the last word on the second line. If the leftmost digit is not 8-f, skip to Step 6.
- Step 5     Type `od address 3`. The left 4 digits of the last word is a hexadecimal number for the number of processes attached to this identifier.
- Step 6     To go to the next identifier, add (hexadecimal) 30 to the address reported in response to `od shmем 6` and enter `od address 6`.
- Step 7     Repeat Steps 4-6 until all identifiers in use have been displayed. (The data reported by Step 6 will be all zeros.)



### 2.2.9 Login

Intermittently, after executing **exec login** the following warning message is produced:

no utmp entry ... execute from the lowest level shell

If the **who** command shows that someone is still logged in on that line, then you must kill the **getty** and allow it to respawn. Otherwise no action is required.

You can avoid the problem by executing **exec su — xxxx** (where *xxxx* stands for a login), or by logging out (that is, hanging up or executing an **exit** command) and calling back in.

### 2.2.10 Lp

If you issue the **lp** command on *file* within a directory that has 700 permissions, the following error messages are displayed:

lp: can't access file *file*

lp: request not accepted

**cat** or **pr file** and pipe the output to **lp**.

### 2.2.11 Mailx

Interrupts are not handled properly when processing **Cc:**. If interrupts are ignored, nothing happens if an interrupt is received during the composition of the message. However if there is an interrupt during the **Cc:** portion, the **mailx** process dies without saving the letter in **dead.letter**.

### 2.2.12 Od

If a file has an odd number of bytes, **od —c** reports a trailing null byte.

### 2.2.13 Passwd

When you try to change the password for a user that does not exist, **passwd** returns the following error message:

Permission denied.

This incorrect message appears even if you execute **passwd** as **root**.



#### 2.2.14 Sar

The performance package does not report usage activity on devices which use the XDC driver. Accordingly, **sar** and any related functions will not report activity on these devices.

**Sar** produces a table in which CPU time is represented as user, system, waiting for I/O or idle. This idle figure may not be accurate. If memory is being used extensively, the idle time reflected by the output may not be totally idle. Part of the time could be attributed to waiting for memory.

When the command **sar -v** is executed, the field "fhdr-sz" still is reported, even though it should not be.

Ignore this obsolete field that lists all zeros.

#### 2.2.15 Sdiff

When doing an **sdiff**, if one of the files contains more than 196 characters in one line, the **sdiff** output loses the separator symbol (!) or loops infinitely, or both.

Do not **sdiff** a file that contains more than 196 characters in a single line.

#### 2.2.16 Sh

When the command **sh -c string** is executed, the flag value returned is blank. The correct value is **flag=s**. If the **-c** option and the *string* are piped into the shell command, the correct value is returned.

Do not try to execute the **sh** command using the **-t** and **-c** options together. It does not work.

If the pathname of the current directory is close to 512 bytes long, then the shell can overwrite internal data structures and cause inconsistencies.

When the system is under a heavy load, you may be logged off if you hit the <BREAK> key several times in succession.

#### 2.2.17 Shl

If you hang up while in shell layers, intermittently **/etc/utmp** is not cleaned up (that is, the **who** command still shows that you are logged in, and the **ps** command shows a **getty** running on that line). If someone else then calls into that line, the **login** fails because the **utmp** entry cannot be found. To kill the **getty** corresponding to the affected line, follow this procedure:

- Step 1      Execute **ps -ef** to find the process I.D. of the **getty** running on the line erroneously reported as occupied.
- Step 2      Execute **kill -9 PID** to kill the **getty**. When the **getty** automatically respawns, the problem is cleared.

When using the **shl** command, trapping a "hang-up" signal inside a layered shell suspends the layer. This renders the device unusable until the shell is killed. You receive a warning message that a layer is still running, and then you are returned to the GENIX V.3 system shell.



If your terminal does not accept input commands because of the suspended layer, you have to log in at another terminal. To kill the suspended shell, follow this procedure:

Step 1 Find the process number of the shell by executing the following command:

**ps -eaf**

Step 2 Look at the output of the command for the appropriate tty (for example, sxt001) and for the process number of the shell.

Step 3 Enter the following command:

**kill -9 PID**

where *PID* is the process number of the shell that you want to kill.

If you are at the console in **shl** and you execute the following sequence of commands, the shell layer does not respond properly:

1. a **stty** command with a carriage return select style argument (for example, **stty cr3**)
2. an **echo** command
3. a **stty** command with a carriage return select style argument (for example, **stty cr0**)

In particular, the layer's prompt does not return until the <BREAK> or <DEL> key is hit. Then, once the prompt appears, the output of any command executed does not print fully or at all until one or more carriage returns are entered.

Return to the **shl** control layer and use the **shl delete** command to delete the layer.

If you execute a background process not in a shell layer that sends output to your terminal, and then you enter shell layers (**shl(1)**), some screen output from the original process may be lost. When you enter **shl**, the output from the original process temporarily stops printing on the screen. The screen output resumes when you exit **shl(1)**. Loss of data, if it occurs, is noticeable when the screen output resumes.

This is not normal use of the system, and should be avoided.

When resuming a layer of **shl**, the resuming *xyz* prompt is often garbled.

### 2.2.18 Tee

In the following pipeline, the copies were not made on disks 2, 3, and 4.

```
dd if=/dev/rSA/diskette1 |  
tee /dev/rSA/diskette2 |  
tee /dev/rSA/diskette3 >  
/dev/rSA/diskette4
```



The problem is caused by the **tee** command writing in blocks of 16 bytes.

If you use the **block** device names (that is, **/dev/SA/diskette?**), the multiple copies succeed.

#### 2.2.19 Tput

If the value of the **TERM** environment variable is an extremely long string (that is, a string of over 500 characters), **tput(1)** dumps core. Known terminal names (that is, in the **terminfo(4)** database) are, in general, between two and fourteen characters.

#### 2.2.20 Uname

If you use **uname -S** to change the nodename, and then re-boot the system after a shutdown, the nodename may change to what it was before you used **uname**.

Because the node name may be set with **/etc/rc.d/nodename** you enter multi-user mode, use **sysadm nodename** to change the system's nodename.

#### 2.2.21 Uucp

The date stamp included in the mail message by **uucp** to tell you of the arrival of files is always in **EST**, even when the time zone is set differently on the transmitting and receiving machines.

#### 2.2.22 Uulog

If you specify the **f** or **number** option with **uulog** (see **uucp(1C)**), you do not receive a prompt after the execution of **uulog**. To regain access to your terminal, depress the **<BREAK>** key.

#### 2.2.23 Uuto

**Uuto(1C)** does not preserve a directory structure when you try to transfer a directory. For example, suppose that you have two directories, **raleigh** and **durham**, that you want to transfer to user **mth** on a computer named **eagle**. Using two command lines, one for each directory, **uuto** should place the entire contents of the directories on **eagle**, as this example shows:



```
/usr/spool/uucppublic/receive/mth/mycomp/raleigh/ files
```

```
/usr/spool/uucppublic/receive/mth/mycomp/durham/ files
```

where **mth** is the destination user and **mycomp** is the name of your *Series 32000* Computer. Instead, **uuto** places all *files* from both directories under the following directory:

```
/usr/spool/uucppublic/receive/mth/mycomp/ files
```

The fix requires adding one line to the **/usr/bin/uuto** program. After the current line 5 in the shell script, add the following line:

```
export UUP
```

The **uuto** code should now look like this:

```
sub="" (Line 5)
export UUP
mysys='uname -l'
```

#### 2.2.24 Vi

When you use **vi -x** and the terminal shuts off during an editing session, the contents of the buffer are not saved. Contents are saved if the terminal shuts off during an editing session with **vi** without the **-x** option.

The following command line does nothing:

```
$ vi + linenumber
```

Using a named buffer twice in a **vi** map sequence results in the following error message:

```
Can't put partial line inside macro
```

Use named buffers only once in map sequences.



### 2.2.25 /usr/news

The directory **/usr/news** is owned by **bin**, belongs to group **bin**, and has permissions **777**. Anyone can create and delete news.

### 2.2.26 PATH

The default **PATH** environment variable searches the current directory first. A superuser unknowingly may run a program in the current directory.

## 2.3 PROGRAMMER COMMANDS, SYSTEM CALLS

### 2.3.1 Ctime

**Ctime(3C)** reports time in Eastern Daylight Time when the environment variable **TZ** is not set. This may be inconvenient to users outside the Eastern Time Zone.

### 2.3.2 Dial, Undial

The **dial(3C)** and **undial(3C)** routines do not work with GENIX V.3 System release **uucp(1)**.

### 2.3.3 Fork

The limit on the number of attached shared memory segments per process is not cleared on an **exec**, causing a newly **exec**'ed image to be limited to less than the proper amount of segments.

Set the tunable parameter **SHMSEG** to 45.

When no unused entries remain in the system process table, **fork()** system calls fail. No message is printed on the console to show that the system process table is full. When the **fork()** calls fail, **errno** is set to **EAGAIN**.

### 2.3.4 Terminfo

The **terminfo(4)** entry for the AT&T 4426 terminal contains an incorrect definition for the clear all tab stops (**tbc**) capability, rendering an attempt to use this capability ineffective.

The correct definition for the clear all tab stops capability for this terminal is as follows:

<ESC>F<ESC>[3g



### 2.3.5 Unlink

The **unlink** command fails without issuing an error message when you use it to unlink a busy text file.

## 2.4 SYSTEM ADMINISTRATOR COMMANDS

### 2.4.1 Chroot

The error message produced when a non superuser executes the **chroot** command is inaccurate. For example, **chroot /usr/bin command** produces the following error message:

```
/usr/bin: Not owner
```

This error message should be interpreted as “chroot: not superuser.”

### 2.4.2 Crash

The trace option in **crash(1M)** truncates function names to 8 characters.

When you ask for **help** within **crash** to obtain usage information about a function, it provides the definition of a table-entry or the start-address format even though the function does not require it.

### 2.4.3 Cron

The following message means that during shutdown, **cron** processes are killed:

```
cron aborted: SIGTERM
```

This message is normal.

If you install the Basic Network Utilities package, the changes made to the **root** crontab (**/usr/spool/cron/crontabs/root**) are not picked up by **cron**.

To correct this, enter the following commands as **root**:

```
# cd /tmp
# crontab -l > ctemp
# crontab ctemp
# rm ctemp
```



#### 2.4.4 Ctcinfo

The command **ctcinfo -r /dev/rSA/ctape** (which resets drive usage) is available to any user. To restrict use of the **ctcinfo** command (and its options) to superusers or **sysadm tapemgmt** users, the system administrator should use the following command:

```
chmod 550 /etc/ctcinfo
```

#### 2.4.5 Dcopy

The usage message issued by **dcopy** does not list all options appropriately. When summarizing, **dcopy** prints the new gap and cylinder sizes when it says that it is printing the old information. The new sizes are, however, printed correctly.

**Dcopy -f** recreates the file system based on the “block:inode” count given; however, the blocks specified are assumed to be logical blocks (physical blocks = 2 x the number specified) and the inode count is assumed to be 16 x the number of inodes requested. Thus, **dcopy** does not accept the inode or block count specified to be the actual number of inodes or physical blocks required, as other commands do (for example, **mkfs**).

#### 2.4.6 Dd

There are two times when **dd** unjustifiably claims success. The first time occurs when it tries to read from an inaccessible address on a disk. A successful return code is returned with the following message:

```
0+0 records in
0+0 records out
```

The second time occurs when writing to a write protected disk. For each block in the file to be copied, the error message that it cannot write to a write-protected disk is displayed, and then the following message is produced:

```
?+1 records in
?+1 records out
```

#### 2.4.7 Errdump

When no errors have been logged, the **errdump** command does not cleanly handle it.



#### 2.4.8 Ff

The **ff** command lists the options **U** and **S** in the usage message. These options are illegal.

#### 2.4.9 Finc

The following **finc** error message may be displayed during a backup to cartridge tape:

Error occurred, error #=??

The error number is echoed to the console. No further information is provided. Check the console for the error number and look it up in the *Administrator's Guide*.

#### 2.4.10 Fmtflop

If **fmtflop** is used without the **-v** option (verify), then the command completes but any writes to that floppy will result in I/O errors. If **fmtflop -v** is done, the verify phase will indicate errors. This problem is configuration dependent and will rarely occur. If you experience this problem, you can do the following:

1. Save the old **/etc/master.d/kernel**.
2. Save the old **/boot/KERNEL** someplace other than **/boot**.
3. Change the **NPROC** value in **/etc/master.d/kernel** by 38 (either increment or decrement).
4. Type the following commands:  

```
cd /boot
mkboot -k KERNEL
```
5. Reboot from **/etc/system**

#### 2.4.11 Fsck

The **fsck(1M)** manual page says that when the medium (for example, a floppy disk) is write protected, **fsck** assumes a **no** response to all questions asked by the command (equivalent to **fsck -n**). This is not true; in these circumstances, **fsck** waits for you to respond to the questions asked.

If you use **lseek** to skip over entire blocks of a file being written, the kernel assigns a block number of 0 to the skipped blocks. Reads of any bytes in those blocks correctly return a value of 0. However, because the number of "non-zeroed" blocks allocated to the file is consistent with the length of the file, an **fsck** of the file system produces the following message:

POSSIBLE FILE SIZE ERROR

Core dumps sometimes occur in these circumstances.

There is no damage to the file system. If the file size error messages are bothersome, you can eliminate them by determining the name of the file corresponding to the inode that has the "possible error" with **ncheck(1M)** and, after mounting the file system, copying that file to a



temporary file and then back to its original name. Blocks are allocated on the copy to hold all the bytes with values of 0.

If **/tmp** is filled up with enough file names to run the **root** file system out of i-nodes, an **fsck** of the file system may produce a "POSSIBLE DIR SIZE ERROR" message. The file system itself is not damaged.

If the error messages are bothersome, you may eliminate them by running **init 1**, and then moving the contents of the directory that provokes the message to a temporary place, removing and recreating the directory, and then moving the files back. For example,

```
# cd /
# mv tmp badtmp
# mkdir tmp
# find badtmp -print | cpio -pdv tmp
# rm -rf badtmp
# init 2
```

Do not change directory modes and permissions, or the modes and permissions of the files in that directory.

#### 2.4.12 Init

**Init S** drops the remote line and changes speed to 9600 baud when entered from a remote terminal.

Only use **init** from the console.

Starting in **init(1M)** state 3 and going to **init 2** changes the modes of the console so that backspace is no longer an erase character. Going from state 2 to state 3 apparently does the same thing.

Restore the original erase *character* with the following command:

```
stty erase character
```

Executing **init s** from within *shl* layers causes inconsistent results. Sometimes, the machine may hang after printing the following message:

```
init: single user mode
```

Other times, the system may not really change run states. At no time within **shl** layers does **init s** do what it is supposed to do.



When you go to **init 3**, the listener is already active; thus, the following message appears on the console:

```
/usr/bin/nlsadmin: listener already active
```

Ignore this message.

On the other hand, when you shut the system down the listener is killed in more than one spot; thus, a message goes to the console that the listener is not running the second time.

#### 2.4.13 Labelit

**Labelit(1M)** assumes that it is dealing with a disk file system unless the device path name starts with **/dev/mt** or **/dev/rmt**, which are tape special files for use with cartridge tapes.

#### 2.4.14 Mkfs

It is possible to build a file system with **mkfs** (using default inode number) larger than the size of the floppy. It is also possible to create a file as large as 1464 blocks on this file system. A later **fsck** does not complain about either anomaly. It is only when you have to read the file that the read fails.

Create file systems on floppies with **sysadm makefsys** instead of with **mkfs**.

#### 2.4.15 Mountfsys

**Sysadm** does not mount an unlabeled file system. File systems that are created by **sysadm** are labeled.

You must label the file system with **labelit(1M)** or use **/etc/mount** manually instead of through the System Administration menus.

#### 2.4.16 Prtconf

On rare occasions when the system has been powered up, **prtconf** does not display the XDC subdevices. This omission occurs because the XDC device has not spun up completely. Execute **sysadm reboot**.

The problem above also may occur when rebooting. If XDC subdevices are not displayed when rebooting, you have an old version of the firmware. Replace your firmware with the latest version.



#### 2.4.17 Shutdown

You cannot use **shutdown(1M)** to return to **init** state 2 from state 3. If you try, you get the following error message:

```
shutdown: Initstate is not for system shutdown
```

Sometimes when you shutdown to single-user mode, the unmount of the **/usr** file system fails with the busy error. Manually unmount **/usr**.

#### 2.4.18 Sysadm

If the **SIGQUIT** signal (ctrl-|) occurs while the user is in **sysadm**, a core dump occurs. The core file is in **/usr/admin/menu/\*mgmt** (for example, **diskmgmt**, **filemgmt**, as appropriate) and should be removed.

If a file system on a floppy is mounted on top of a non-empty directory (for example, **/lib**) with **sysadm**, and then unmounted with **sysadm**, the mount directory becomes inaccessible to all other users. A file system mounted over **/etc** cannot be unmounted since the **umount(1M)** command is no longer accessible.

Avoid mounting a file system on a non-empty directory. If you must mount a file system on non-empty directory, use the “manual” commands such as **mkfs(1M)**, **labelit(1M)**, **mount(1M)**, and **umount** instead of the subcommands in the System Administration menus. If you must use **sysadm**, then you must change the modes of the mount point directory manually with **chmod(1)** after the file system is unmounted.

System administration partial subcommand names (for example, **for** for **format**) are not accepted from the shell. Do not use partial subcommand names.

The **sysadm** help file for setting a password does not warn of the 8 character limit on passwords. Someone who chooses a password “abcdefgh” is told that the password requires a numeric character but the password is rejected when “abcdefgh9” is chosen because the number does not appear within the first 8 characters.

There is no protection provided against multiple users using the same **sysadm** subcommand at the same time.

If you use **sysadm** for floppy disk or for cartridge tape handling, then you must be certain you have control of the appropriate drives. This is no different from using tape drives on larger GENIX V.3 system machines. If the session involves changing administrative files, the problem is probably one of system management, specifically, only one person should be authorized to make changes to **uucp**, and **passwd** files.



#### 2.4.19 Sysadm autold

**Sysadm autold** defines the device that the system uses to do an automatic load (auto load) if no other device is specified. By default, the auto load device is null. If the auto load default is not null and is not **/unix**, then a Release Upgrade, Partial Restore, or Full Restore fails. Also, if the default device number is 0 (floppy disk), these procedures fail.

Set the default auto load to **/unix**. Set the default device number to 1, which specifies to boot off the hard disk.

#### 2.4.20 Sysadm backup

During a **sysadm backup**, if a bad sector is encountered while writing on a floppy disk, you are not warned of the problem. Furthermore, when a **sysadm restore** is executed, the data from the point of the bad sector throughout the remaining backup disks is unreadable.

NOTE: You should make frequent individual file or directory backups, with periodic complete backups. It is strongly recommended that you verify a file or a directory that you backup. The file and directory storage commands are available in the **store** menu under the **filemgmt** menu of System Administration.

#### 2.4.21 Sysadm firmware

When you do a Partial Restore, files in **/etc/rc?.d** and **/etc/init.d** are copied to **/usr/old**. The original files are not removed. When the machine reboots after the Partial Restore finishes, the scripts in **/etc/rc?.d** are executed, and an error may occur because the system cannot find certain software (that software must be reinstalled).

#### 2.4.22 Sysadm portmgmt

You cannot login on the contty port if the computer is in **init** state 3 (for Remote File Sharing) or 4 (user defined). As root, edit the second field for contty in **/etc/inittab** to be **234** instead of the default state of **2**.

#### 2.4.23 Sysadm portmgmt delete

After executing **sysadm portmgmt delete**, the **inittab** entry for the port is not returned to a usable state. For example, if you connect the modem to **tty21** and you execute **portmgmt delete**, the **/etc/inittab** file entry for **tty21** looks similar to the following line:

```
21:2:respawn:/etc/getty -t 60 tty21 1200H
```

To change the entry to a usable state, log in as **root**, edit the **/etc/inittab** file, and change the entry for **tty21** to look like the following line:

```
21:2:off:/etc/getty tty21 1200
```



#### 2.4.24 Sysadm portmgmt modify

If you are connecting a modem to a port that had a terminal connected to it, **sysadm portmgmt modify** may not start the **uugetty**s. Before connecting a modem to the port, you should check to see if a **getty** is running on the port. Execute the following command and look for the process number of the port to which you want to connect the modem. You can identify the port by its tty number (for example: tty14 or tty22).

**ps —eaf**

After you have identified the process number, execute the following command.

**kill —9 PID**

where *PID* is the process number. You may now connect a modem to the port. The **portmgmt modify** command should execute properly.

#### 2.4.25 Sysadm uucpmgmt

If you add a dialer script (that is, instructions on how to talk with a modem, ACU, or special device) to **/usr/lib/uucp/Dialers**, then you must also edit the **/usr/lib/uucp/Devices** file to create an entry to use that dialer script. The dialer script is not accessible with **sysadm**.

See the “Supporting Data Base” section in the Basic Networking chapter of the *Administrator's Guide*.

Using the **sysadm uucpmgmt** command, the subcommand **systemmgmt** should list all systems known to the current machine. If this list contains more than 825 systems (that is, around 5000 characters), the list begins normally, but eventually the output is garbled and the listing end with an error message.

**Sysadm uucpmgmt** states that you can depress the <CR> key to select the default speed (contty). If you depress the <CR> key, an error message is returned saying that the default speed is not found in the **gettydefs** file. Instead of selecting the default speed, you need to enter a baud rate, for example, 300, 1200, or 9600.

The call option under **sysadm uucpmgmt** leaves two processes running. Hit <BREAK> after exiting the menu.

#### 2.4.26 Sysdef

**Sysdef** does not display the following tunable parameters: SHLBMAX, GPGSMSK, PUTBUFSZ, and ILOGSIZE.



#### 2.4.27 Swap

If there is a heavy process load on a system with a small memory capacity, the **swap** command may silently fail (that is, complete without any error indication) when attempting to add a swap area.

Retry the command when the load on the system decreases. If this is a recurring problem, try to reduce the load on the system or increase the amount of memory on the machine.

#### 2.4.28 Umount

If you attempt to **umount** **—d** a disk file system, the error message returned is as follows:

```
umount: /dev/dsk/c1dxxx not mounted
```

The **—d** option is valid only for remote **umounts**.

#### 2.4.29 Ucheck

You should always use the **—v** option with **ucheck**. You cannot ask for different levels of debugging information with **ucheck** **—x**.

#### 2.4.30 Ucheck, Ucleanup, Utry

Most of the Basic Networking Utilities commands can be executed from a normal user login. The exceptions are **ucheck** and **ucleanup**, which require either an administrative (**uucp**) login or a **root** login.

**Ucheck**, **ucleanup**, and **Utry** are located in the **/usr/lib/uucp** directory, which is not in the search path for most logins, including those for **uucp** or **root**. Therefore, you must give the full path name, or you must be in the **/usr/lib/uucp** directory to execute these three commands.

Another alternative is to link the command where it may be easily accessed, for example, **/usr/bin**.

#### 2.4.31 Uucico

It is possible to get **uucico** into a runaway state when you use it through the NPACK network under extremely heavy network load. For example, the process may accumulate too much time (2500 minutes of CPU time).

For example, suppose on machine\_A you have a **uucico** to machine\_B with suspiciously high CPU time. Login on machine\_B and do **ps —ef**. Look in the output of the **ps** command for a **uucico** process talking to machine\_A (that is, with a command line argument like **—s machine\_A**). If there is such a process, then the connection is still active. If there is not, then the **uucico** on machine\_A is in a runaway state.



If **uucico** is in a runaway state, take the following steps:

- Step 1 Kill the **uucico** process with **kill -9 pid**, where *pid* is the process ID of the looping **uucico**.
- Step 2 Remove the associated lock file with **rm /usr/spool/locks/LCK..*machine*** where *machine* is the system where the **uucico** originated, for example, *machine\_A*.

### 2.4.32 Uutry

There are two legal options to **Uutry(1M)**: **-r** and **-x**. Any other input (for example, illegal options, garbage text) is assumed to be a system name.

Consult the **Uutry** manual page in the *Administrator's Reference Manual*.

### 2.4.33 Volcopy

While executing **volcopy(1M)**, you have the option of hitting **DEL** to obtain a shell. On exiting the shell, **volcopy** fails, dumps core and prints the error message: **bus error- core dumped**.

### 2.4.34 Path for Superuser Inconsistent

If you log in as **root**, the system default **PATH** contains the current directory. If you **su** to **root**, the system default **PATH** does not contain the current directory. Both **PATH** variables should be the same. The **PATH** should not contain the current directory.

Create a **.profile** for the **root** login that sets the **PATH** variable as follows:

```
PATH=/bin:/etc:/usr/bin
```

## 2.5 MISCELLANY

### 2.5.1 Console

If you are running **shl** on the console, and you then run **shutdown -is**, the console hangs.

At times after rebooting the system, someone logging in on the console may receive the following message without being prompted for a password:

```
Login incorrect:
```

This may also occur if the superuser kills **hdelogger** while no one is logged into the console. The console becomes disassociated from **/dev/tty**. The process group of the console has become attached to one of the startup functions that execute out of **/etc/rc.d** or **/etc/inittab**.

Reboot the system.



When a hard-wired terminal is disconnected from the console port, and is reconnected, the console port echoes but does not respond or present a login prompt. Going to another terminal to investigate this problem shows no `getty` or shell process associated with the console. A `getty` runs on the console if you type `init q` or if `hdellogger` is killed. If you kill `hdellogger`, you should be wary of the problem discussed immediately above.

### 2.5.2 Kernel

When you run programs that use all available memory and swap space, the computer prints the following message on the console:

DANGER: out of swap space

However, the programs continue to run.

If a system call fails because a process would exhaust memory or swap space, system error messages (for example, `growreg: cannot allocate ...`) go only to the console. When a system call fails because of lack of memory, the system call always returns an error code and an `errno` value of `EAGAIN`. If these return codes are not checked by your program, you get a core dump when the program tries to use memory.

Problems such as these likely result from programs that have large virtual memory requirements. System administrators should set `MAXUMEM` to some value under the available memory plus swap space so that such processes can fail because they are too big. That notification goes directly to the user.

Change `MAXUMEM` in `/etc/master.d/kernel` to 1024, and then execute the following commands, after which "Console Login:" should appear:

```
# cd /boot
# mkboot -k KERNEL
# cd /
# sync
# sync
# touch /etc/system
# shutdown -i6 -g0 -y
```

Nothing can be done with system administration if an application program is available in object code only, except possibly to allocate more swap space with the `swap(1M)` command. If you have source code, you may put checks for error returns into the code, and recompile the program.

It is possible to get logged off if several interrupts are transmitted during the execution of a pipe. For example, pressing the `<BREAK>` key repeatedly during a pipe could cause you to be logged off.



When a program executes integer division by zero the following error message is displayed:

floating exception - core dumped

This message does not accurately describe the error.

Processes spawned by the kernel at boot time (**sched**, **/etc/init**, **vhand**, **bdflush**) have start times (**STIME**) that are the time the system was last brought down, not the time they were spawned.

Some core dumps may have possible file size errors reported by **fsck**. The reporting of possible file size errors by **fsck** is only a warning, so these errors can be ignored. To determine whether the possible file size errors reported are resulting from core dumps, execute **ncheck -i i-number**, where *i-number* is given in the **fsck** message:

POSSIBLE FILE SIZE ERROR I=*i-number*

**Ncheck** will generate the path name of a file from its inode number, *i-number*. See the “How To Check a File System for Consistency” section of the *Administrator's Guide* and the **ncheck** page in the *Administrator's Reference Manual* for further details.

If the system is reconfigured to adjust the maximum number of open files per process, by changing the value of the tunable parameter **NOFILES**, the value of **NOFILE**, a **#define** in **/usr/include/sys/param.h**, and the value of **\_NFILE**, a **#define** in **/usr/include/stdio.h**, are unaffected. The value of **NOFILE** and of **\_NFILE** is always 20, regardless of the value of the tunable parameter **NOFILES**.

Never use **NOFILE** from **/usr/include/sys/param.h** or **\_NFILE** from **/usr/include/stdio.h** in a user program; use the **NOFILES** tunable parameter instead.

If the operating system runs out of free **clists**, all input/output activity from/to terminal ports and the console will cease. No warning message is printed by the system to show that it is out of **clists**.

The value of the **SHMALL** tunable parameter specifies the maximum number of in-use shared memory segments allowable system-wide. This parameter is not checked by the system (that is, **shmget(2)** does not check this limit).

Under heavy I/O, the system hangs after a panic.

When this happens, hit **<RESET>** and take a dump. The reset wipes out register contents.

### 2.5.3 Manual Page for Fs Format Is Incorrect

**Mount** expects **s\_magic** field of the given argument's superblock to be equal to **FsMAGIC**. If not, **mount** does not mount the file system even if it is a proper one. Every file system is made with **s\_magic** set to **FsMAGIC**, and therefore **mount** always works. However, the manual page for the **fs(4)** format gives the impression that even if a file system's **s\_magic** is not equal to **FsMAGIC**, it can still be a valid file system.



#### 2.5.4 /etc/ports Does Not Insert Gettys in /etc/inittab

After a partial restore, the new **/etc/inittab** does not contain **/etc/getty** entries for the tty lines on the ports board. When **/etc/ports** is run, these entries should be inserted into **/etc/inittab** after creating **/dev/tty??** entries. However, if **/dev/tty??** exists, **/etc/ports** does not insert the getty entries into **/etc/inittab**.

After a partial restore, if **/etc/inittab** does not contain getty entries for the tty lines on the ports board, you should remove **/dev/tty??** and run **/etc/ports** again.

NOTE: Do not remove **/dev/tty**.

**/etc/ports** then inserts getty entries into **/etc/inittab** and creates the appropriate **/dev/tty??** character special devices.

#### 2.5.5 Tunable Parameters

In the "Tunable Parameters" section of the *Administrator's Guide*, the maximum message size of the message queue facility (**MSGMAX**) is documented to be 128 kilobytes. You cannot set **MSGMAX** to 128K. The largest value (power of 2) that **MSGMNB** (maximum queue size) may be set to is 65,535 (64K -1) bytes.

#### 2.5.6 Possibility of Data Corruption After a Power Outage

Although the probability is low, the hard disk data may be severely corrupted after a power outage. If the data are severely corrupted, several warning messages are displayed when you turn on the computer. One of the messages is the following:

WARNING: The file system check has failed very badly. We strongly suggest that you do not use the machine until a service representative has looked at it. I will now turn off power.

The system is then automatically powered off. Before you contact a service representative, try a full restore of the GENIX V.3 operating system using the *Series 32000* Computer core system floppy disks. A Full System Restore corrects the disk corruption, and the computer is again usable. See the *Administrator's Guide* for the Full System Restore procedure.

NOTE: A Full System Restore deletes all data on the disk unless you back them up.



## 2.6 FILE SYSTEM RESTORE

### 2.6.1 Error Messages During Complete Restore

During a “complete” restore of the root (/) and /usr file systems, messages similar to the following are displayed:

```
Cannot link </bin/sh> & </bin/rsh>  
Cannot link </etc/init> & </etc/telinit>  
Cannot unlink current </etc/cron> error 26 text file busy  
Cannot unlink current </etc/getty> error 26  
Cannot create </etc/getty> error 26 text file busy  
Cannot unlink current </etc/hdlogger> error 26  
Cannot create </etc/hdlogger> error 26  
Cannot unlink current </usr/lbin/ncpio> error 26  
Cannot create </usr/lbin/ncpio> error 26
```

It is normal to get these messages; they do not suggest a problem with the restore process.

### 2.6.2 Turn Off User Terminals During a Full System Restore or an Upgrade

During an upgrade or a full system restore, the only terminal that should be turned on is the console terminal. When an upgrade or full system restore is performed, all tunable parameters are set to default values for a 1M system. If several terminals are connected to Expanded I/O Capability feature cards and are turned on, the default parameter values may not be adequate to handle all the terminals. A possible consequence of this condition is that users may not be able to log in after a full system restore or an upgrade. To avoid problems, you need to re-tune the system parameters to their previous values before allowing users to turn on their terminals. A message is printed during an upgrade or restore process telling you to do this.

### 2.6.3 Change Access Permission After Full Restore on Two-Disk System

After doing a full system restore, the access permission on the /usr and /usr2 file systems should be set to 775; that is, not everyone should have write permission. However, if you have a two-disk system and you do a full restore, the access permission on the /usr2 file system is set to 777; that is, everyone has write permission. To change the access permission, log in as root and enter the following command:

```
chmod 775 /usr2
```



## 2.6.4 Reinstall Utilities With Software Drivers After a Partial Restore

NOTE: After doing a partial restore, any installed utilities that contain software drivers must be reinstalled.

See your system manual to identify utilities that contain software drivers.

## 2.6.5 Saving Device Files When Backing Up Root File System

When you backup the root (/) file system, the device files (/dev directory) are not saved as part of the backup. To save the device files, become the superuser, mount a blank formatted floppy that has a file system on it, and enter the following commands:

```
# sysadm mountfsys
# find /dev —print | cpio —pdl /mnt
```

where /mnt is the directory on which the floppy disk file system is mounted. The cpio options are lower-case letters **p**, **d**, and **l**.

To restore the files, insert the floppy on which the files were saved, and enter the following commands:

```
# sysadm mountfsys
.
.
.
# cd /mnt
# find . —print | cpio —pdl /dev
.
.
.
# sysadm umountfsys
```

## 2.6.6 Contents of Sys Crontab Is Wrong

/usr/spool/cron/crontab/sys is intended to contain performance collection commands. The file is installed by core floppy number 5, and should contain only a few comment lines. When the Performance Management Utilities package is installed, it edits /usr/spool/cron/crontab/sys to add the appropriate commands unless it finds that the commands already exist.

This crontab file installed by core floppy number 5 for the GENIX V.3 release contains both the comment lines and the performance commands. Thus, between the time you install the core



floppies and the time you install the performance package, cron sends mail to `/usr/spool/cron/crontab/sys` complaining that it cannot find `/usr/lib/sa/sa1` (one of the performance commands). If you decide not to install the Performance Management Utilities (possibly to save space on `/usr`), the file `/usr/mail/sys` grows by 5,424 characters per day, and the file `/usr/lib/cron/log` grows by an extra 2,520 characters per day (avoid wasting space with mail messages).

Either install the Performance Management Utilities, or manually truncate `/usr/mail/sys` and `/usr/lib/cron/log`. If you do not install the performance package, remove non-comments from `/usr/spool/cron/crontab/sys` (the last three lines).

### 2.6.7 f450 Printer Needs Unsupplied Filters

The f450 printer model needs the filter `/usr/lib/etx`, but this filter does not exist. Also, the f450 printer model needs the filter `/usr/bin/450`. This filter is not provided with the GENIX V.3 release. The source for `/usr/bin/450` comes with the Source Code Provision of GENIX V.3.

The f450 filter is supplied with the Terminal Filters Utilities. Install these utilities before loading the 450 model to ensure that it works correctly.

### 2.6.8 Read/Write Permissions for Basic Networking Do Not Work

The read/write permissions for the Basic Networking Utilities do not work correctly. For a system to be able to read a directory, the target machine must also have granted the system write permissions (`/usr/lib/uucp/Permissions`). Suppose that your system has read permissions in root and write permissions in `/usr/tmp`. The following command fails:

```
uux "A!pr A!/etc/inittab > A!/usr/tmp/B.out"
```

However, if your system has read permissions in root and write permissions in `/usr/tmp` and `/etc`, the above command line succeeds.

### 2.6.9 Primary Server Panics

The primary server panics with the following scenario.

```
# init 3
# rfadmin -r secondary
FAILS -- can't remove secondary
# rtstop
TRAP
PANIC: KERNEL MMU FAULT (F_ACCESS)
```



A dump is available.

Take the system down and reboot.

### 2.6.10 Converting to Getopts by Hand

**Getoptcv** (see **getopts(1)**) adds about 30 lines of code to a shell script, so you may want to convert scripts by hand instead. Converting by hand probably will make the code cleaner and easier to understand. Also, you do not have to worry about parsing option-arguments that are also options.

Follow these guidelines to convert most scripts that currently use the **getopt(1)** command.

- Step 1     Delete the old invocation line, and the **if** statement that checks the exit code.
- Step 2     Change the **for** loop to a **while** loop that invokes **getopt(1)**.
- Step 3     Change the patterns in the **case** statement from **—option** to single option letters.
- Step 4     Delete the case for **—**.
- Step 5     Add a case for **?**. This case may be used to print the usage message and exit with a non-zero exit code. Note that the **?** is quoted since it is interpreted for filename expansion.
- Step 6     Remove all **shift** commands within the **case** statement.
- Step 7     Change **\$2** to **\$OPTARG** for cases that require an option argument.
- Step 8     Add the statement **shift `expr \$OPTIND - 1`** after the **while** loop so the remaining arguments may be referenced as before.



Here is an example of a script before and after conversion:

```
# before conversion
set -- 'getopt abo: $*'
if [ $? != 0 ]
then
    echo $USAGE
    exit 2
fi
for i in $*
do
    case $i in
        -a | -b)    FLAG=$i; shift;;
        -o)        OARG=$2; shift 2;;
        -)         shift; break;;
    esac
done
```

```
# after conversion
while getopts abo: i
do
    case $i in
        a | b)    FLAG=$i;;
        o)        OARG=$OPTARG;;
        ?)        echo $USAGE
                  exit 2;;
    esac
done
shift `expr $OPTIND - 1`
```



If you want your script to work on releases before GENIX V.3 (that is, use either **getopts** or **getopt**), convert it as shown in the following example:

```
if [ "$OPTIND" = 1 ]
then
    while getopts abo: i
    do
        case $i in
            a | b)    FLAG=$i;;
            o)        OARG=$OPTARG;;
            ?)        echo $USAGE
                     exit 2;;
        esac
    done
    shift `expr $OPTIND - 1`
    echo $*
else
    set -- `getopt abo: $*`
    if [ $? != 0 ]
    then
        echo $USAGE
        exit 2
    fi
    for i in $*
    do
        case $i in
            -a | -b)  FLAG=$i; shift;;
            -o)       OARG=$2; shift 2;;
            -)        shift; break;;
        esac
    done
    echo $*
fi
```



## Chapter 3

# COMPATIBILITY NOTES

### 3.1 INTRODUCTION

GENIX V.3 is not completely compatible with previous System V releases. This section details the particular differences so that you can determine if any changes to your software are needed. Some of these changes arise because instances of incorrect behavior in previous System V releases have been fixed. Others arise from bringing GENIX V.3 into compliance with several developing standards, in particular the /usr/group, IEEE, and ANSI standards. The remainder are irreconcilable differences resulting from adding new features.

### 3.2 CHAPTER OVERVIEW

The sections that follow list the changes that affect compatibility with System V, Release 2 compatible operating systems. Some of these changes became effective in System V, Release 2.1. If you are upgrading directly to GENIX V.3, you can refer to just this section to find all the changes that may affect you. Each change listed below is marked with either (V.3) or (2.1) to distinguish in which release the change was made.

Each of the following sections is organized by manual page section, much like the *User's Reference Manual* and *Programmer's Reference Manual*, as follows:

- Changes in Shell Commands
- Changes in System Calls
- Changes in Library Routines
- Changes in File Formats and Contents
- Changes in the C Compilation System
- Miscellaneous Changes

If you will be using the optional Remote File Sharing feature, you should refer to the *Remote File Sharing Utilities Notes* for a list of additional differences.

### 3.3 CHANGES IN COMMANDS

Table 3-1 lists the GENIX V.3 commands affected by changes introduced in Release 2.1 or Release V.3. The first column gives the command, while the second column gives the title of the section in this manual that describes the change. These titles are also the titles of the manual pages in the *User's Reference Manual* or the *Administrator's Reference Manual* that describe the commands. Look in the latter manual if the title is marked with a (1M). The last column lists the names of the packages in which the commands are found.



TABLE 3-1. SUMMARY OF CHANGED COMMANDS

COMMAND	LOOK UNDER
cc(1)	<i>C Programming Language Utilities</i>
crash(1M)	<i>System Administration Utilities</i>
crypt(1)	<i>Security Administration Utilities (Domestic Only)</i>
dirname(1)	<i>User Environment Utilities</i>
df(1M)	<i>Essential Utilities</i>
ed(1)	<i>Essential Utilities</i>
ex(1), vi(1)	<i>Editing Utilities</i>
file(1)	<i>Directory and File Management Utilities</i>
fuser(1M)	<i>System Administration Utilities</i>
mail(1)	<i>Essential Utilities</i>
pr(1)	<i>Essential Utilities</i>
ps(1)	<i>Essential Utilities</i>
sh(1)	<i>Essential Utilities</i>
ex(1), vi(1)	<i>Editing Utilities</i>
who(1)	<i>Essential Utilities</i>



### 3.3.1 Cc(1) 2.1

#### Summary of Change:

Functions in C programs that do not explicitly return a value will now return a random value. Previously, such functions would often have a zero return value.

While NSC never guaranteed the behavior of functions that did not return a value, the value zero was often returned. If you have a program that depends on such behavior, it may now fail.

You should check your C programs to ensure that all functions, other than those declared to be of type **void**, always end with a statement of the form

```
return (X)
```

where *X* is a value of the type appropriate for the function.

### 3.3.2 Crash(1M) V.3

#### Summary of Change:

There are several new and enhanced **crash** commands.

You should refer to the new **crash** manual page in the *Administrator's Reference Manual* for the new commands and the new method of using some of the old commands.

### 3.3.3 Crypt(1) V.3

#### Summary of Change:

The **crypt** command now takes options that begin with a dash (-), such as **-k**. These can no longer be given as keys. This change will only effect those who have the Security Administration Utilities package installed on their system.

### 3.3.4 Dirname(1) V.3

#### Summary of Change:

The **dirname** command now properly parses the names **//** and **//anything/**.

Previous to the GENIX V.3 release, the **dirname** command would return a dot, **.**, if given the argument **//** or **//anything/**. Now it correctly returns a slash, **/**.

You should change any shell scripts that rely on the previous incorrect behavior of the **dirname** command to reflect the correct operation.



### 3.3.5 Df(1M) V.3

#### Summary of Change:

The **df** command now exits with a non-zero return code when it encounters an error, where before it would always exit with a zero return code.

In earlier releases, shell scripts had to check the output of the command to see if it failed. These scripts will still work correctly, but you may want to simplify them.

### 3.3.6 Ed(1) V.3

#### Summary of Change:

The **ed** command now defaults to using the **/usr/tmp** directory to hold temporary files, instead of **/tmp**.

This change does not apply to the **vi** or **ex** programs.

NOTE: You do not need to take any special action to use **ed** in single user mode if the **/usr** file system is not mounted. When **ed** cannot put its temporary file in **/usr/tmp**, it tries **/tmp** instead.

### 3.3.7 Ex(1), Vi(1) V.3

#### Summary of Change:

The structure of the **/usr/preserve** directory used by **vi** and **ex** has changed.

Instead of saving an editing session as a file directly under the **/usr/preserve** directory, it is saved in a subdirectory with the name of the user whose session is saved. Only the same user can access the content of the subdirectory.

In general you will not be able to recover a **vi** or **ex** session preserved before the upgrade to GENIX V.3. All sessions should have been recovered before upgrading.

#### Summary of Change:

The **ex** and **vi** commands now exit with a return code equal to the number of errors encountered during the editing session. Before, no specified return code was used if errors were encountered.

Because all return codes must be between 0 and 255, if more than 255 errors are encountered the return code will not be accurate. If an integral multiple of 256 errors are found, then **ex** and **vi** exit with a zero return code.

#### Summary of Change:

The **ex** and **vi** commands no longer set the eighth bit in the characters of the **%** expansion of the current filename.

When the percent sign, **%**, is used in a shell escape from **ex** or **vi** via the exclamation mark, **!**, the **%** is replaced with the name of the file being edited. Previously each character in this replacement had the eighth bit set to 1 to quote it, now it is left alone.



Generally, you can use older versions of the **ex** or **vi** commands on Release V.3, but you cannot use the percent sign, **%**, in a shell escape via the exclamation mark, **!**, even if the file being edited has no special characters in it.

### 3.3.8 File(1) V.3

#### Summary of Change:

Additions to the **file** command's repertoire of file types (found in the system file **/etc/magic**) make the command incompatible with an older **/etc/magic**.

You must use the **sysadm installpkg** command to ensure that the entire application package is installed. Moving only the new **file** command to an earlier release of GENIX V.3, without moving the **/etc/magic** file as well, will cause the **file** command to fail.

NOTE: The old **file** command will work with the new **/etc/magic**, although no new features are gained by doing so.

### 3.3.9 Fuser(1M) V.3

#### Summary of Change:

The **fuser** command now exits with a return code equal to the number of errors encountered during execution. In previous releases a zero return code was returned if no errors were encountered; one was returned otherwise.

This change should not adversely affect programs using the **fuser** command.

Because all return codes must be between 0 and 255, if more than 255 errors are encountered the return code will not be accurate (see **vi** description above).

### 3.3.10 Mail(1) V.3

#### Summary of Change:

The **mail** command now requires an entry for the group **mail** in the **/etc/group** file.

### 3.3.11 Pr(1) V.3

#### Summary of Change:

The **pr** command now correctly interprets the combined options **-m -k** as an error, where before one option would be ignored.

Shell scripts that took advantage of the earlier fault in the **pr** command must be changed to use the correct option.



### 3.3.12 Ps(1) V.3

#### Summary of Change:

The **ps** command now correctly interprets a non-numeric argument to the **—g** or **—p** options as an error, where previously it was treated as zero.

You should change any shell scripts that rely on the previous incorrect behavior of the **ps** command to reflect the correct operation.

NOTE: Note that the **—g** option is used to examine the processes belonging to a particular process group leader, not to a particular user group identifier.

### 3.3.13 Sh(1) V.3

#### Summary of Change:

A trailing colon in the shell **PATH** variable will cause the current directory to be included in command searches. Previously, a trailing colon was ignored.

The **test** and **[...]** commands now use the effective user and group IDs to determine permissible file access, instead of the real IDs as previously.

The only way to invoke the **test** command with different effective and real IDs is to invoke the command, or a shell script containing it, from a compiled program that has the set user (group) ID on execution permission. Otherwise, the effective IDs are the same as the real IDs, and this change will have no effect. If your program must rely on the test operators to behave as they did previously, which is to have **test** use the real user and group IDs, you should change it to use the **setuid(2)** or **setgid(2)** system calls to set the effective ID to the real ID before invoking the **test** command or shell script.

#### Summary of Change:

The GENIX V.3 Shell no longer treats the eighth bit in the characters of a command line argument specially; it also no longer strips the eighth bit from the characters of an error message.

If you have any program that sets the eighth bit of characters, they will have to be changed. You should use one of the standard Shell quoting mechanisms, such as the backslash, instead of setting the eighth bit.

#### Summary of Change:

The GENIX V.3 Shell **type** command will display backslashes before each character that was quoted initially.

This change is a result of the change described above, where the eighth bit is no longer used by the shell to quote characters.



#### Summary of Change:

The result of a parameter substitution in a command like

```
ls "${a:=xyz abc} lmnop"
```

is now correct.

In general, the parameter substitution

```
${ parameter:=word }
```

when used inside double quotes and when the *word* contains spaces, now works correctly. If you have programs that rely on the previous incorrect behavior, you should change them to reflect the correct behavior.

#### 3.3.14 Who(1) V.3

##### Summary of Change:

The **who —q** command now lists the login names in space padded fields of equal size and no longer sorts the entries.

If you have any programs that process the output of the **who —q** command, you should inspect them to see if they will still work with the new form of the output.

### 3.4 CHANGES IN SYSTEM CALLS

Table 3-2 lists the GENIX V.3 system calls affected by changes introduced in Release 2.1 or V.3. The first column gives the system call, while the second column gives the title of the section below that describes the change. These titles are also the titles of the manual pages in the *Programmer's Reference Manual* that describe the system calls.

#### 3.4.1 Acct(2) V.3

##### Summary of Change:

The **acct** system call now sets **errno** to **EACCESS** when given an argument that is a directory instead of a file, instead of setting it to **EISDIR** as before.



**TABLE 3-2. SUMMARY OF CHANGED SYSTEM CALLS**

<b>SYSTEM CALL</b>	<b>LOOK UNDER</b>
acct	acct(2)
brk	brk(2)
exec	exec(2)
fcntl	fcntl(2)
fork	fork(2)
mount	mount(2)
msgrcv	msgctl(2), msgop(2)
msgsnd	msgctl(2), msgop(2)
plock	plock(2)
ptrace	ptrace(2)
semop	semctl(2), semop(2)
shmat	shmop(2)
shmctl	shmctl(2)
signal	signal(2)
sys3b	sys3b(2)
umount	umount(2)
ustat	ustat(2)



### 3.4.2 Brk(2) 2.1

#### Summary of Change:

The **brk** system call fails, setting **errno** to **EAGAIN**, if allocating memory may cause a deadlock.

In previous releases it was possible for the GENIX V.3 system to enter a deadlock if free swap space or free main memory were not available. When this occurred, you could not “swap in” a runnable process because there was no room in main memory, and it was also impossible to “swap out” a process to free the needed main memory because there was no room in the swap area. While rare, this situation occurred often enough to require us to add checks in the GENIX V.3 release to prevent it.

Several system calls now fail and set **errno** to indicate that a deadlock might have occurred. Table 3-3 shows which system calls have changed and what value is given to **errno**.

**TABLE 3-3. DEADLOCK DETECTING SYSTEM CALLS**

SYSTEM CALL	ERRNO
brk	EAGAIN
exec	EAGAIN
fork	EAGAIN
plock	EAGAIN
shmat	ENOMEM
shmctl	ENOMEM

**NOTE:** Note that this new behavior should only occur when the system has nearly exhausted its memory capacity. If this occurs often, you should consider adding more main memory or increasing the size of the swap space.



### 3.4.3 Exec(2) 2.1

#### Summary of Change:

The **exec** system call fails, setting **errno** to **EAGAIN**, if allocating memory may cause a deadlock.

See discussion of deadlock detection above.

#### Summary of Change:

The **exec** system call no longer checks the **F\_EXEC** bit in the **a.out** file header flags of a program before attempting to execute the program.

This change only affects those programs that were produced using the **—r** option in the **ld** program (also available through the **cc** program.) Such programs are often still executable, so the new behavior allows you to run them. Previously, the **exec** system call would fail with **errno** set to **ENOEXEC**.

### 3.4.4 Fcntl(2) 2.1

#### Summary of Change:

The **flock** structure returned with the **F\_GETLK** command of the **fcntl** system call has changed; the **l\_pid** element has been changed from type **int** to type **short**, and a new element, **l\_sysid** of type **short**, has been added.

This change was made to accommodate cases where a file or record lock has been set by a process on a remote computer. The new structure now uniquely identifies a process.

Programs compiled under Release 2.0 that use record locking will have to be recompiled because **l\_sysid**, which occupies the space that used to be the high order 16 bits of the (old Release 2.0 32 bit) **pid**, can be non-zero for remote locks. This will cause problems if the application tries to kill that **pid**.

#### Summary of Change:

The **fcntl** system call, on a file or record lock request, now sets **errno** to **ENOLCK** when the system runs out of lock resources, instead of setting it to **ENOSPC** or **EMFILE** as before.

This change will only affect programs that currently check for **ENOSPC** or **EMFILE** to learn if the system has run out of lock resources.

NOTE: Note that the effect will only be seen when you run the program and other programs have used up all the available locks.



### 3.4.5 Fork(2) 2.1, V.3

#### Summary of Change:

The **fork** system call now has additional reasons for failing, but still sets **errno** to **EAGAIN** in these cases.

This should not affect a program's attempt to catch cases where the **fork** system call fails, because the same **errno** value is used. You should recognize, however, that the new paging system introduces new ways for **fork** to fail when system resources are running low.

### 3.4.6 Mount(2) 2.1

#### Summary of Change:

The **mount** system call now correctly fails, with **errno** set to **ENOTDIR**, on an attempt to mount a special file on itself.

The **mount** system call now fails and sets **errno** to **EINVAL** if the file system's type is not recognized or if the **mflag** (previously **rwflag**) argument is not correct.

### 3.4.7 Plock(2) 2.1

Summary of Change: The **plock** system call fails, setting **errno** to **EAGAIN**, if locking a process may result in a memory deadlock.

See discussion of deadlock detection above (**brk(2)**).

### 3.4.8 Ptrace(2) V.3

#### Summary of Change:

The **ptrace** system call now allows write access to a shared text segment. This allows a program to be debugged that more than one person may be running.

In earlier releases, the **ptrace** system call would refuse to write into a text segment, or "pure procedure space," if that segment was being shared with another process and the other process was executing in the segment. The **ptrace** system call allows this in the GENIX V.3 release by ensuring that a separate text image is made and that the write access is to that separate image.

### 3.4.9 Shmop(2) 2.1

#### Summary of Change:

The **shmat** system call may fail, setting **errno** to **ENOMEM**, if there is not enough memory to allocate page tables or if attaching the shared segment may cause a deadlock.



Previously the only reason for the **shmat** system call to fail with **errno** set to **ENOMEM** was if there was not enough memory for the shared memory segment. Now the unavailability of additional memory for tables needed to manage the separate pages of the shared segment, or the possibility of a memory deadlock, will also cause the system call to fail. (For a discussion of deadlock detection, see the **brk(2)** section above.)

#### Summary of Change:

**shmat** system call now allows text as well as data segments to be shared.

Previously, a shared memory segment could only be attached to an address in the data segment of a process. With Release V.3, a shared memory segment can be attached to any address in a process.

### 3.4.10 Shmctl(2) 2.1

#### Summary of Change:

The **shmctl** system call now fails, setting **errno** to **ENOMEM**, if locking the shared memory region may cause a deadlock.

For example, the following will fail if the attempt to lock the shared memory segment, identified by *shmid* causes a deadlock:

```
shmctl(shmid, SHM_LOCK)
```

See the discussion of deadlock detection under the **brk(2)** section above.

#### Summary of Change:

The **shmctl** system call ignores an attempt to unlock a shared memory segment that is already unlocked, instead of failing as before.

This change is not likely to cause a problem unless a program deliberately tries unlocking a shared memory segment without knowing if the segment is already locked in memory. If such a program looked for the **EINVAL** error return to indicate that the unlock attempt was not needed, it will no longer work as expected.

### 3.4.11 Signal(2) V.3

Summary of Change: The **signal** system call now returns a pointer to a function of type **void** instead of a pointer to a function of type **int** as before.

This change was made to bring GENIX V.3 closer to conforming with the IEEE standard on the GENIX V.3 operating system. Since the function to which the return value of the **signal** system call points does not itself return a value, **void** is its correct type, not **int**.

No source code changes are required for GENIX V.3, although continued use of the **int** type instead of the **void** type will cause a warning message from the GENIX V.3 cc compiler or the



**lint** program checker. While this message is harmless and the code will compile correctly, you should start changing source code now to ensure compatibility with future GENIX systems. All previously compiled programs and application packages that use the **signal** system call will still work with this release.

#### Summary of Change:

The signal **SIGIOT** is being phased out to be replaced with the signal **SIGABRT**.

This change was made to bring GENIX V.3 closer to conforming with the IEEE standard on the GENIX V.3 operating system.

Currently, both names are supported so source code is compatible. In the future the name **SIGIOT** will no longer be supported, so you should start changing your source code now. However, the value of **SIGIOT** and the value of **SIGABRT** are the same, which means that all compiled programs, including application packages you may have purchased, will continue to work, even in the future. For example, the **abort(3C)** library routine is now described as issuing the **SIGABRT** signal instead of the **SIGIOT** signal as before. You should therefore write new source code to expect the **SIGABRT** signal. However, since the values are same, a program previously compiled to expect the **SIGIOT** signal from **abort** will continue to work when linked with the new **abort** routine.

#### Summary of Change:

The **signal** system call now may also fail, setting **errno** to **EINVAL**, if the *func* argument is invalid.

Previously the **signal** system call did not check its second argument, *func*, to ensure that it was one of **SIG\_DFL**, **SIG\_IGN**, or a valid function address.

### 3.4.12 Umount(2) V.3

#### Summary of Change:

On an attempt to unmount a special device whose major and minor numbers do not exist, the **umount** system call now sets **errno** to **EINVAL**, instead of **ENXIO**.

There should not be many programs affected by this change, since special devices are usually mounted and unmounted using the **mount(1M)** and **umount(1M)** shell commands. You should see if any of your programs that use the **umount(2)** system call check **errno** for the value **ENXIO** when the system call fails. Any that check for **ENXIO** should be changed to check for **EINVAL**.

#### Summary of Change:

For **umount**, **EBUSY** is now returned if the device of the filesystem to be unmounted is the default pipe device.

The default pipe device is that section of hard disk used for unnamed pipes. If it is overridden and placed in a section of disk belonging to a filesystem that can be unmounted (for example, **/usr**), and you attempt to unmount that filesystem, the unmount will fail with the above error.



### 3.4.13 Ustat(2) V.3

#### Summary of Change:

A new error return has been added. If the root inode of the mounted file system that you are doing the **ustat** on is NULL, **ENOENT** is set.

## 3.5 CHANGES IN LIBRARY ROUTINES

Table 3-4 lists the GENIX V.3 library routines affected by changes introduced in Release 2.0 or Release V.3. The first column gives the routine, while the second column gives the title of the section below that describes the change. These titles are also the titles of the manual pages in the *Programmer's Reference Manual* that describe the routines.

### 3.5.1 Abort(3C) V.3

#### Summary of Change:

The **abort** routine now issues the **SIGABRT** signal instead of the **SIGIOT** signal.

See **signal(2)** in the section on system calls.

#### Summary of Change:

The **abort** routine no longer closes files when the **SIGABRT** (previously **SIGIOT**) signal is being caught or ignored.

Previously the **abort** routine would close all open files before issuing the **SIGIOT** signal that would normally cause the program to halt. If, however, the program had arranged to trap or ignore the **SIGIOT** signal then it would have to reopen the closed files before continuing.

With GENIX V.3 the **abort** routine closes the files only if the program will halt on receiving the **SIGABRT** signal (which has the same value as the **SIGIOT** signal).

If you have a program that used the **abort** routine and trapped or ignored the **SIGIOT** signal, then you should check to see if the new action by **abort** of keeping files open will not cause a problem.

### 3.5.2 Ctime(3C) V.3

#### Summary of Change:

The type of the argument **clock** in the **ctime**, **gmtime**, and **localtime** routines have been changed from "pointer to **long**" to "pointer to **time\_t**".

This is another change made to bring GENIX V.3 closer to conforming with the IEEE standard on the GENIX V.3 operating system.

No source code changes are required for GENIX V.3, but you should start changing source code now to ensure compatibility with future GENIX systems. All previously compiled programs and application packages that use these routines will still work with this release.



**TABLE 3-4. SUMMARY OF CHANGED LIBRARY ROUTINES**

ROUTINE	LOOK UNDER
abort	abort(3C)
ctime	ctime(3C)
fputs	puts(3S)
fread	fread(3S)
fwrite	fread(3S)
gmtime	ctime(3C)
localtime	ctime(3C)
puts	puts(3S)
setvbuf	setbuf(3S)
strncat	string(3C)
strncmp	string(3C)
strncp	string(3C)

### **3.5.3 Puts(3S) V.3**

#### **Summary of Change:**

The **fputs** and **puts** routines now correctly return **EOF** if the attempt fails, instead of zero as before.

If you have a program that checks for a zero return from **puts** or **fputs** to indicate a write error, you should change it to check for **EOF**.



### 3.5.4 Fread(3S) V.3

#### Summary of Change:

The type of the “size” argument to the `fwrite` and `fread` routines and the `strncat`, `strncpy`, and `strncmp` string manipulation routines, has been changed from `int` to `size_t`.

This is another change to bring GENIX V.3 closer to conforming with the IEEE standard on the GENIX V.3 operating system. By changing the type to `size_t` larger buffers or longer strings can be handled by these routines.

No source code changes are required for GENIX V.3, but you should start changing source code now to ensure compatibility with future GENIX systems. All previously compiled programs and application packages that use these routines will still work with this release.

### 3.5.5 Setbuf(3S) V.3

#### Summary of Change:

The `setvbuf` routine now behaves correctly as described in the System V, Release 2 manual pages.

The correct description of the parameters for the `setvbuf` routine is

```
setvbuf (stream, buf, type, size)
FILE *stream;
char *buf;
int type, size;
```

This is consistent with Release 2.0 and Release V.3 documentation, but differs from the implementation on Release 2.0. The implementation now correctly matches the documentation.

If you have a program that uses `setvbuf` with the second and third arguments switched to match the earlier implementation, you will need to switch the arguments back to have the program work on Release V.3.



### 3.5.6 String(3C) V.3

Refer to the **fread(3S)** section above for the changes made to the **string** routines.

### 3.5.7 Changes in the C Compilation System

There are three general changes in the C Compilation System utilities, or C Programming Language Utilities as they are now called.

- The recognition of a new C preprocessor directive, **#ident**, and the addition of it to GENIX V.3 header files.
- The addition of new error codes and system calls returning new or different error codes.
- Changes to GENIX V.3 header files.

Your programs may be affected by these changes.

### 3.5.8 #ident 2.1

Summary of Change: The new C preprocessor recognizes a new directive, **#ident**, that provides a better way of controlling software versions. This directive has been added to all header files.

The change will only cause a problem if another C preprocessor is used that does not recognize the **#ident** directive, and any of the standard header files are included in the code processed by the other preprocessor. In such cases you will have to arrange for the header files to be stripped of the **#ident** directives before being passed to the other preprocessor.

## 3.6 NEW ERROR CODES

Table 3-5 shows the error codes introduced in Release 2.0 and Release V.3.

The codes **ENONET** through **ECOMM** and the **EMULTIHOP** code are specific to the Remote File Sharing feature. The **ENOSTR**, **ETIME**, **ENOSR**, **EPROTO**, and **EBADMSG** codes are related to the use of Streams and the Transport Level Interface features. The **ELIBACC** through **ELIBEXEC** codes are related to the use of shared libraries. You should not see any of these error codes unless you are using one of the related features.

Several system calls now include additional error codes as possible reasons for failure. Some of these codes are related to the new features as mentioned above, but some are older error codes that are now appropriate failure reasons in GENIX V.3. In some cases, system calls simply have a different error code for the same failure. These changes are described in Section 3.5.2.



**TABLE 3-5. ERROR CODES**

<b>ERROR CODE</b>	<b>VALUE IN RELEASE V.3</b>
ENOSTR	60
ENODATA	61
ETIME	62
ENOSR	63
ENONET	64
ENOPKG	65
EREMOTE	66
ENOLINK	67
EADV	68
ESRMNT	69
ECOMM	70
EPROTO	71
EMULTIHOP	74
EBADMSG	77
ELIBACC	83
ELIBBAD	84
ELIBSCN	85
ELIBMAX	86
ELIBEXEC	87



### 3.7 CHANGES IN HEADER FILES 2.1, V.3

Table 3-6 lists the header files changed, dropped, or introduced in System V, Release 2.1 compatible operating systems and GENIX V.3. The manual pages for the system calls and routines listed in the *Programmer's Reference Manual* describe where some of these header files are needed. The names listed are those referred to in the manual pages, and are the names used in **#include** statements. The files are found starting in the **/usr/include** directory. Some of these files are or were only available if you have a source license.

Most of the changes should not affect normal user programs. However, some header files changed greatly as a result of adding new features. These header files are marked with an asterisk (\*) in the figure below. Programs that use these files will, at minimum, have to be recompiled. In some cases, especially if a particular order was assumed in a structure or a particular offset was assumed for an element of a structure, the programs will have to be recoded to remove such dependencies. Such programs are extremely system dependent and, as such, cannot be expected to be fully portable to new releases of GENIX V.3.

#### 3.7.1 Core.h 2.1

Summary of Change:

The macros defined in **core.h** that specify the location of the stack in a core dump file were changed to reflect the change in the stack location.

#### 3.7.2 Errno.h 2.1, V.3

Summary of Change:

Several new error codes have been added for the new features of Releases 2.1 and V.3. See other parts of this document for details, in particular Table 3-5, Error Codes.

#### 3.7.3 Fcntl.h 2.1, V.3

Summary of Change:

The **flock** structure missing from the **fcntl.h** file in some Release 2.0 computers is now included. Also, a copy of this file is now in **sys/fcntl.h**.

These changes should not affect any programs. However, we encourage you to start changing your programs to include **sys/fcntl.h** instead of **fcntl.h**.



TABLE 3-6. CHANGED, DROPPED, OR ADDED HEADER FILES

HEADER FILE	RELEASE 2.1	RELEASE V.3
core.h	changed	
errno.h		changed
fcntl.h	changed	changed
ldfcn.h	changed	
limits.h		new
signal.h		changed
stdio.h	changed	
string.h		changed
unistd.h		new
sys/buf.h		changed
sys/dir.h		changed
sys/fblk.h		changed
sys/fcntl.h		new
sys/filsys.h		changed
* sys/flock.h	changed	changed
* sys/immu.h	changed	
sys/inline.h		new
* sys/inode.h	changed	changed
sys/ipc.h	changed	
sys/lo.h		dropped
sys/map.h	changed	
* sys/mount.h	changed	changed
sys/nvram.h	changed	
sys/open.h	changed	
* sys/param.h	changed	changed
* sys/pfdat.h		new
* sys/proc.h	changed	
sys/psw.h	changed	
* sys/reg.h	changed	
sys/sbd.h	changed	
sys/shm.h	changed	
sys/stat.h		changed
* sys/stream.h		new
* sys/stropts.h		new
sys/sys3b.h	changed	
* sys/sysinfo.h	changed	changed
sys/sysmacros.h		changed
sys/systm.h	changed	changed
sys/tty.h	changed	
sys/types.h	changed	
* sys/user.h	changed	changed
* sys/var.h	changed	changed
sys/fs/s5dir.h		new
sys/fs/s5filesys.h		new
sys/fs/s5param.h		new
sys/fs/s5inode.h		new
sys/fs/s5fblk.h		new
NOTE:	Because of the extensive changes made to the header files marked with an asterisk, the changes are not discussed in the sections that follow. Instead you are referred to the files themselves for a description of the changes.	



### 3.7.4 Ldfcn.h 2.1

#### Summary of Change:

The declarations of some functions already declared in **stdio.h** have been removed from **ldfcn.h**

Since the proper use of the **ldfcn.h** header file requires the inclusion of the **stdio.h** header file, this change should not affect any programs.

### 3.7.5 Signal.h V.3

#### Summary of Change:

New macros for the GENIX V.3 enhanced signal handling have been added. Also, the macro **SIGABRT**, equivalent to **SIGIOT**, has been added. The **SIG\_DFL** and **SIG\_IGN** macros have been changed from values of type **(int(\*)())** to values of type **(void(\*)())**.

The latter two changes are discussed under the **signal(2)** section above.

### 3.7.6 Stdio.h 2.1

#### Summary of Change:

Several functions that return values of type **int** are now explicitly declared in **stdio.h** rather than leaving them implicitly declared.

Unless a program has already declared or defined symbols with the same name but of different type, this change should not present a problem and should help program developers find improper use of the functions. The functions that are now explicitly declared are listed below:

<code>fclose</code>	<code>fflush</code>	<code>fgetc</code>	<code>fprintf</code>
<code>fputc</code>	<code>fputs</code>	<code>fread</code>	<code>fscanf</code>
<code>fseek</code>	<code>fwrite</code>	<code>getw</code>	<code>pclose</code>
<code>printf</code>	<code>sprintf</code>	<code>puts</code>	<code>putw</code>
<code>scanf</code>	<code>setvbuf</code>	<code>sscanf</code>	<code>system</code>
<code>ungetc</code>	<code>vfprintf</code>	<code>vsprintf</code>	<code>vprintf</code>

### 3.7.7 String.h V.3

#### Summary of Change:

All the definitions found in the **memory.h** header file are now also included in the **string.h** header file.



This change should not cause a problem unless a program includes the **string.h** file but defines its own versions of the symbols from the **memory.h** file. The symbols added to the **string.h** file are **memccpy**, **memchr**, **memcpy**, **memset**, and **memcmp**. These are not macros, so they cannot be undefined. If your program already defines one or more of these symbols, it is best if you redefine them with different names.

The **memory.h** file still exists—programs that refer to it can still be compiled on this release. The change will not affect programs already compiled.

### 3.7.8 Sys/buf.h V.3

#### Summary of Change:

The file system dependent fields, **b\_filsys** and **b\_dino** have been removed from the **buf** structure defined in the **sys/buf.h** header file.

### 3.7.9 Sys/dir.h V.3

#### Summary of Change:

The contents of **sys/dir.h** and **sys/fblk.h** have been moved to the header files **sys/fs/s5dir.h** and **sys/fs/s5fblk.h**, respectively. The old files contain a **#include** directive to include the contents from the new files. Also, the type of the **d\_ino** field in the structure has changed from **ino\_t** to **ushort**.

Because the standard System V file system type is just one of, potentially, several others, the contents of **sys/dir.h** and **sys/fblk.h** were moved to the **sys/fs** directory under different names where there will be similar files for other file system types provided by NSC in the future. Thus we encourage you to change statements such as:

```
#include <sys/dir.h>
#include <sys/fblk.h>
```

in any C programs you may have to the following statements.

```
#include <sys/fs/s5dir.h>
#include <sys/fs/s5fblk.h>
```

The old **sys/dir.h** and **sys/fblk.h** files may be removed in the future.

The type change should not cause a problem because the header file that defines the **ino\_t** type, **sys/types.h**, had to be included previously, and, secondly, the **ino\_t** type was and is the same as the **ushort** type, (namely **unsigned short**.)



### 3.7.10 Sys/fblk.h V.3

See the **sys/dir.h** section above for information.

### 3.7.11 sys/fcntl.h V.3

See the **fcntl.h** section above for information.

### 3.7.12 Sys/filsys.h V.3

Summary of Change:

The contents of **sys/filsys.h** have been moved to the header file **sys/fs/s5filsys.h**. The old file contains a **#include** directive to include the contents from the new file. Also, the type of the **s\_inode** and **s\_tinode** elements in the **filsys** structure have changed from **ino\_t** to **ushort**; and **getfs** is now declared as a macro instead of a function.

Because the standard System V file system type is just one of, potentially, several others, the contents of **sys/filsys.h** were moved to the **sys/fs** directory under a different name, where there will be similar files for other file system types provided by NSC in the future.

The type change should not cause a problem because the header file that defines the **ino\_t** type, **sys/types.h**, had to be included before, and second, the **ino\_t** type was and is the same as the **ushort** type, (namely **unsigned short**.) However, the change in **getfs** from a function to a macro requires that you recompile any programs that use it. We encourage you to start changing your programs to include **sys/fs/s5filsys.h** instead of **sys/filsys.h**, as this file may be dropped in a future release.

### 3.7.13 Sys/inline.h V.3

Summary of Change:

Some of the macros defined in **sys/inline.h** are not defined when using the **cxref** command to cross-reference a program that includes this file.

### 3.7.14 Sys/inode.h 2.1, V.3

Summary of Change:

The contents of **sys/inode.h** have been changed extensively, and now contains only inode information that is file system independent. The inode information that depends on the System V file system is now contained in **sys/fs/s5inode.h**.

Any programs that include **sys/inode.h** will also have to include **sys/fs/s5inode.h**. Because of the extensive changes, it is suggested that you compare the old file with the two new files to determine what coding changes, if any, are required.



### 3.7.15 Sys/ipc.h 2.1

#### Summary of Change:

The definitions of the macros **SHM\_LOCK** and **SHM\_UNLOCK** have been moved from **sys/ipc.h** to **sys/shm.h**

Since the **SHM\_LOCK** and **SHM\_UNLOCK** macros are only used in programs that already include the **sys/shm.h** file, this change should not have any affect.

### 3.7.16 Sys/lo.h V.3

#### Summary of Change:

The **lo.h** file has been dropped in GENIX V.3.

### 3.7.17 Sys/map.h 2.1

#### Summary of Change:

The declarations for **swapmap**, **coremap**, and **shmmap** have been deleted from **sys/map.h** in GENIX V.3.

### 3.7.18 Sys/open.h 2.1

#### Summary of Change:

A macro for another type of driver-level open, **OTYP\_SWP**, has been defined in Release V.3; it takes the same value as the previous **OTYP\_LYR** macro, which now takes on a new value. The macro **OTYPCNT**, which gives the total number of types, has been increased by one.

Any drivers that use **OTYP\_LYR** will have to be recompiled.

### 3.7.19 Sys/param.h 2.1, V.3

#### Summary of Change:

Parts of **sys/param.h** have been changed extensively; most of the changed content has been moved to the header file **sys/fs/s5param.h**. The old file contains a **#include** directive to include the contents from the new file.

You should examine the **sys/param.h** and **sys/fs/s5param.h** files to see if the changes will affect any of your programs. Since the new file is automatically included by the old file, your programs that need the new file's content should not also include the file.



### 3.7.20 Sys/pcb.h 2.1

#### Summary of Change:

The macros that define the kernel interrupt PSWs have been moved from **sys/pcb.h** to the **sys/psw.h** header file.

Any of your programs that use the **ZPSW** or **KPSW** *nn* macros for the kernel interrupt PSWs should now include the proper file with a line such as the following.

```
#include <sys/psw.h>
```

### 3.7.21 Sys/proc.h 2.1, V.3

#### Summary of Change:

While **sys/proc.h** has changed extensively, the offsets of certain elements in the **proc** structure (the process table) have not.

### 3.7.22 Sys/psw.h 2.1

#### Summary of Change:

The macros **PS\_V** and **PS\_Z** are now correctly defined; the definitions in earlier releases were reversed. Also, several definitions have been moved here from the **pcb.h** and **types.h** header files.

### 3.7.23 Sys/sbd.h 2.1

#### Summary of Change:

The default shared memory attach address, defined by the macro **UVSHM**, has changed.

### 3.7.24 Sys/shm.h 2.1

#### Summary of Change:

The macro **SHM\_CLEAR** has been renamed to **SHM\_INIT**, although the value remains the same, and the **shm\_reg** element in the **shmid\_ds** structure has changed from a segment descriptor structure to a pointer to such with sufficient padding so as not to affect other elements. Also, the definitions of the macros **SHM\_LOCK** and **SHM\_UNLOCK** have been moved here from the **ipc.h** header file.

The renaming of **SHM\_CLEAR** to **SHM\_INIT** will not affect compiled programs or application packages that use this macro because the same value has been kept. However, source code should be changed to reflect the new name.



### 3.7.25 Sys/stat.h V.3

#### Summary of Change:

The type of the **st\_ino** element in the **stat** structure has changed from **ino\_t** to **ushort**. Also, new definitions have been added to the **stat.h** header file.

The type change should not cause a problem because the header file that defines the **ino\_t** type, **sys/types.h**, had to be included before, and, secondly, the **ino\_t** type was and is the same as the **ushort** type (namely **unsigned short**.)

The addition of the new definitions should not cause a problem unless a program includes this file, using a C program line such as the following:

```
#include <sys/stat.h>
```

and the program already uses a symbol now defined in the header file. The new symbols are listed below:

<b>S_ENFMT</b>	<b>S_IRWXU</b>	<b>S_IRUSR</b>	<b>S_IWUSR</b>
<b>S_IXUSR</b>	<b>S_IRWXG</b>	<b>S_IRGRP</b>	<b>S_IWGRP</b>
<b>S_IXGRP</b>	<b>S_IRWXO</b>	<b>S_IROTH</b>	<b>S_IWOTH</b>
<b>S_IXOTH</b>			

### 3.7.26 Sys/sysmacros.h V.3

#### Summary of Change:

The macro **brdev**, that would mask the lower 17 bits of a device number, has been removed from **sys/sysmacros.h**.

### 3.7.27 Sys/systm.h 2.1, V.3

#### Summary of Change:

The declarations for the variable **Maxmem** and the functions **bmap**, **ialloc**, **owner**, and **maknode** have been removed from **sys/systm.h**.

### 3.7.28 Sys/tty.h 2.1

#### Summary of Change:

A declaration for a new variable, **cfreecnt**, of type **int**, has been added to **sys/tty.h**.



### 3.7.29 Sys/types.h 2.1

#### Summary of Change:

The definition of the **psw** structure has been moved to the **sys/psw.h** header file, and the definition of the **sde\_t**, **SRAMA**, and **SRAMB** types have been moved to the **sys/immu.h** header file. Also, the definitions for the **psram** and **mmuproc** structures have been removed.

### 3.7.30 Sys/user.h 2.1, V.3

#### Summary of Change:

The **sys/user.h** header file has changed extensively, but the offsets of the following fields in the **user** structure have not changed.

<b>u_base</b>	<b>u_count</b>	<b>u_error</b>	<b>u_gid</b>
<b>u_offset</b>	<b>u_proc</b>	<b>u_qsav</b>	<b>u_r</b>
<b>u_rgid</b>	<b>u_ruid</b>	<b>u_segflg</b>	<b>u_ttyn</b>
<b>u_uid</b>			

### 3.7.31 Sys/fs/s5dir.h V.3

See the **sys/dir.h** section above for more information.

### 3.7.32 Sys/fs/s5filsys.h V.3

See the **sys/filsys.h** section above for more information.

### 3.7.33 Sys/fs/s5param.h V.3

See the **sys/param.h** section above for more information.

### 3.7.34 Sys/fs/s5inode.h V.3

See the **sys/inode.h** section above for information.



### 3.7.35 Sys/fs/s5fblk.h V.3

See the **sys/dir.h** section above for information.

### 3.7.36 NSC Miscellaneous Changes

### 3.7.37 Brc Entry in /etc/inittab V.3

Summary of Change:

The **brc** entry in the **/etc/inittab** file is now executed on the **bootwait** action, not the **sysinit** action as before.

This change should not have any affect on your use of the system.

### 3.7.38 Longest Allowed Pathnames V.3

Summary of Change:

The longest pathname is now restricted to 1024 bytes. System calls that require pathnames as arguments will now fail, setting **errno** to **ENOENT**, if a longer pathname is given.

The length of a file's full pathname is now restricted to 1024 bytes. While previously the pathname was not restricted by the GENIX operating system, most programs gave an ad hoc limit to the length. Generally these limits were well below 1024 bytes, so most programs should not be affected by this change.

The **limits.h** file defines a macro **PATH\_MAX** to be the longest length of a pathname. While in GENIX V.3 this file incorrectly sets the macro to 256, it will probably be changed in a future release to 1024. Local system administrators can safely change the value for **PATH\_MAX** to 1024 without harm, since the Release V.3 system internally uses the longer limit.

We encourage you to include the **limits.h** file with a statement like

```
#include <limits.h>
```

and refer to the **PATH\_MAX** macro for the longest pathname allowed.

## 3.8 CHANGES UNDER SHARED LIBRARIES

The new shared libraries feature allows several processes to use routines in common without each having a separate copy of the code. However, you must recompile or relink a program to take advantage of this feature; The following sections describe some things to watch for. If you do not recompile or relink a program, then the shared libraries feature will not affect the program and it should continue to work as before.



### 3.8.1 Chroot(1) V.3

#### Summary of Change:

Using either the **chroot** shell command or the **chroot** system call may not work when the program involved has been built with a shared library.

This will not affect any existing programs. However, new programs compiled in this release, or old programs recompiled in this release, may have a problem if the libraries linked into the program are shared. When such a program is run, the GENIX V.3 system tries to find the shared library using the original pathname of the library, but if the root has changed places the system may be unable to find the library.

The only way to avoid this problem when using either the **chroot** command or the **chroot** system call is to arrange beforehand for copies of the required shared libraries to be found in the correct place under the new root directory.

For example, suppose a program named **xyz** has been compiled to use the shared library **/shlib/libc\_s**. The following commands will let this program run under a new root directory called **/abc**.

```
mkdir /abc/shlib # only if /abc/shlib does not exist yet
cp /shlib/libc_s /abc/shlib/libc_s
chroot /abc xyz
```

### 3.8.2 Exec(2) V.3

#### Summary of Change:

The **exec** system call will now fail if the program to be run requires a shared library for which you do not have execute permission (**errno** set to **ELIBACC**), or if you try to exec a shared library directly (**errno** set to **ELIBEXEC**), or if you try to exec a shared library that doesn't exist.

### 3.8.3 Mount(2) V.3

#### Summary of Change:

The **mount** system call now fails and sets **errno** to **EINVAL** if the file system's type is not recognized or if the **mflag** (previously **rwflag**) argument is not correct.



### 3.9 DOCUMENTATION

The following documents are sent with the release of the system.

PUBLICATION NUMBER	MANUAL TITLE
424510771-110	Administrator's Guide
424510771-120	Programmer's Guide
424510771-130	User's Guide
424510771-140	Network Programmer's Guide
424510771-150	STREAMS Primer
424510771-151	STREAMS Programmer's Guide
424510771-210	Administrator's Reference Manual
424510771-220	Programmer's Reference Manual
424510771-230	User's Reference Manual
424510771-310	Doc Roadmap
424510771-320	Product Overview
424510771-410	GENIX V.3 Utilities Notes
424510771-420	Networking Support Utilities Notes
424510771-430	Remote File Share Utilities Notes



## INDEX

### A

abort(3C) V.3, change summary	3-14
Access permission	2-23
full restore on two-disk system	2-23
acct(2) V.3, change summary	3-7
Additional features	
crash	1-5
encryption	1-5
help	1-5
Additional installation information	1-15

### B

Backing up root file system	
saving device files	2-24
Basic networking utilities	
commands, enhanced	1-3
read/write permission do not work	2-25
upgrading	1-15
bc	2-1
Before you upgrade	1-7
brc entry in /etc/inittab V.3	3-28
brk(2) 2.1, change summary	3-9

### C

C compilation system, changes in	3-17
cc(1) 2.1, change summary	3-3
cdc	2-2
Changed commands, summary	3-2
Changes in commands	3-1
Changes in header files 2.1, V.3	3-19
Changes in library routines	3-14
Changes in system calls	3-7
Changes in the C compilation system	3-17
Changes under shared libraries	3-28
chroot	2-10
chroot(1) V.3, change summary	3-29
Command changes	1-5
crash	1-5
Command syntax standard	1-4
Commands, changes in	3-1
Compatibility notes	3-1
Console	2-19
Console, not having a	1-15
Contents of sys crontab is wrong	2-24
Conventions	1-1
Converting to getopts by hand	2-26
core.h 2.1, change summary	3-19
cpio	2-2
crash	1-5, 2-10
crash(1M) V.3, change summary	3-3
crash	1-5
command changes	1-5
cron	2-10
crypt(1) V.3, change summary	3-3

ct	2-2
ctcinfo	2-11
ctime	2-9
ctime(3C) V.3, change summary	3-14
cu	2-2

### D

Data corruption after a power outage possible	2-22
dcopy	2-11
dd	2-11
Deadlock detecting system calls	3-9
Device files	2-24
saving when backup up root file system	2-24
df(1M) V.3, change summary	3-4
dial	2-9
dirname(1) V.3, change summary	3-3
Documentation	3-30

### E

ed(1) V.3, change summary	3-4
encryption	1-5
mechanisms repackaged	1-6
Enhanced basic networking commands	1-3
errdump	2-11
errno.h 2.1, V.3, change summary	3-19
Error codes	3-18
new	3-17
Error messages	
during complete restore	2-23
ignorable during upgrade	1-16
unclear during installation of lp spooler	1-17
/etc/ports does not insert	
gettys in /etc/inittab	2-22
ex(1) V.3, change summary	3-4
exec(2) 2.1, change summary	3-10
exec(2) V.3, change summary	3-29
Existing data not preserved	1-17
when installing lp spooler	1-17

### F

f450 printer needs unsupplied filters	2-25
fcntl(2) 2.1, change summary	3-10
fcntl.h 2.1, V.3, change summary	3-19
Features, additional	
crash	1-5
encryption	1-5
help	1-5
Features of GENIX V.3	1-2
ff	2-12
file	2-3
File system restore	2-23
file(1) V.3, change summary	3-5
Filter	







## R

Read/write permissions for	2-25
basic networking do not work	2-25
Rebooting the system takes time	1-16
Reinstall utilities with software drivers	2-24
after a partial restore	2-24
Release upgrade	1-7
Remote file sharing	1-2
Root file system	2-24
saving device files when backing	2-24

## S

sar	2-5
sdiff	2-5
setbuf(3S) V.3, change summary	3-16
sh	2-5
sh(1) V.3, change summary	3-6
Shared libraries	1-4
changes under	3-28
shl	2-5
shmctl(2) 2.1, change summary	3-12
shmop(2) 2.1, change summary	3-11
shutdown	2-15
Signal mechanism enhancements	1-4
signal(2) V.3, change summary	3-12
signal.h V.3, change summary	3-21
Software drivers	2-24
retinstall after partial restore	2-24
Software notes	2-1
stdio.h 2.1, change summary	3-21
string(3C) V.3, change summary	3-17
string.h V.3, change summary	3-21
Summary of changed commands	3-2
Summary of changed library routines	3-15
Summary of changed system calls	3-8
Superuser	2-19
path inconsistent	2-19
swap	2-18
sys crontab	2-24
contents is wrong	2-24
sysadm	2-15
sysadm autold	2-16
sysadm backup	2-16
sysadm firmware	2-16
sysadm portmgmt	2-16
sysadm portmgmt delete	2-16
sysadm portmgmt modify	2-17
sysadm uucpmgmt	2-17
sys/buf.h V.3, change summary	3-22
sysdef	2-17
sys/dir.h V.3, change summary	3-22
sys/fblk.h V.3, change summary	3-23
sys/fcntl.h V.3, change summary	3-23
sys/filsys.h V.3, change summary	3-23
sys/fs/s5dir.h V.3, change summary	3-27
sys/fs/s5fblk.h V.3, change summary	3-28
sys/fs/s5filsys.h V.3, change summary	3-27
sys/fs/s5inode.h V.3, change summary	3-27
sys/fs/s5param.h V.3, change summary	3-27
sys/inline.h V.3, change summary	3-23
sys/inode.h 2.1, V.3, change summary	3-23
sys/ipc.h 2.1, change summary	3-24
sys/lo.h V.3, change summary	3-24

sys/map.h 2.1, change summary	3-24
sys/open.h 2.1, change summary	3-24
sys/param.h 2.1, V.3, change summary	3-24
sys/pcb.h 2.1, change summary	3-25
sys/proc.h 2.1, V.3, change summary	3-25
sys/psw.h 2.1, change summary	3-25
sys/sbd.h 2.1, change summary	3-25
sys/shm.h 2.1, change summary	3-25
sys/stat.h V.3, change summary	3-26
sys/sysmacros.h V.3, change summary	3-26
sys/systm.h 2.1, V.3, change summary	3-26
System administrator commands	2-10
System calls	
changes in	3-7
deadlock detecting	3-9
programmer commands	2-9
summary of changed	3-8
sys/tty.h 2.1, change summary	3-26
sys/types.h 2.1, change summary	3-27
sys/user.h 2.1, V.3, change summary	3-27

## T

tee	2-6
Terminal	
improved facilities for supporting	1-4
information utilities	1-5
information utilities, installing	1-17
turn off during full system restore	2-23
turn off during upgrading	2-23
terminfo	2-9
tput	2-7
TTY management menu	1-17
Tunable parameters	2-22

## U

umount	2-18
umount(2) V.3, change summary	3-13
uname	2-7
Unclear error message during installation	1-17
of lp spooler	1-17
undial	2-9
unlink	2-10
Upgrading to GENIX V.3	1-2
Upgrading	
basic networking utilities	1-15
before you	1-7
dual-disk upgrade	1-13
ignorable error messages during	1-16
misleading help message during release	1-15
procedures	1-6
release upgrade	1-7
turn off user terminals	2-23
your current release	1-9
User commands	2-1
/usr/news	2-9
ustat(2) V.3, change summary	3-14
uucheck	2-18
uucico	2-18
uucleanup	2-18
uucp	2-7
uulog	2-7
uuto	2-7



Uutry

2-18, 2-19

## V

vi

2-8

vi(1) V.3, change summary

3-4

volcopy

2-19

## W

who(1) V.3, change summary

3-7



READER'S COMMENT FORM

In the interest of improving our documentation, National Semiconductor invites your comments on this manual.

Please restrict your comments to the documentation. Technical Support may be contacted at:

(800) 538-1866 - U.S. non CA

(800) 672-1811 - CA only

(800) 223-3248 - Canada only

Please rate this document according to the following categories. Include your comments below.

	EXCELLENT	GOOD	ADEQUATE	FAIR	POOR
Readability (style)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Technical Accuracy	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Fulfills Needs	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Organization	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Presentation (format)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Depth of Coverage	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Overall Quality	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

NAME \_\_\_\_\_ DATE \_\_\_\_\_

TITLE \_\_\_\_\_

COMPANY NAME/DEPARTMENT \_\_\_\_\_

ADDRESS \_\_\_\_\_

CITY \_\_\_\_\_ STATE \_\_\_\_\_ ZIP \_\_\_\_\_

Do you require a response? ☐ Yes ☐ No PHONE \_\_\_\_\_

Comments:

---

---

---

---

---

---



**BUSINESS REPLY MAIL**

**FIRST CLASS PERMIT NO. 409 SANTA CLARA, CA**

POSTAGE WILL BE PAID BY ADDRESSEE

**■ National Semiconductor Corporation  
Microcomputer Systems Division  
Technical Publications Dept. 8278, M/S 7C261  
2900 Semiconductor Drive  
P.O. Box 58090  
Santa Clara, CA 95052-9968**

NO POSTAGE  
NECESSARY  
IF MAILED  
IN THE  
UNITED STATES

