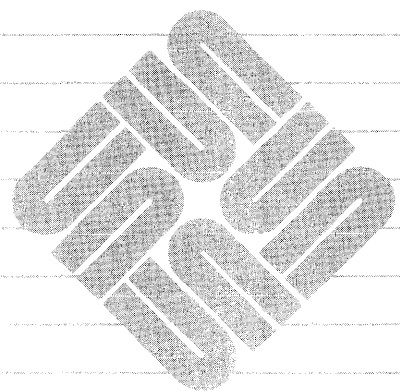




SunOS 4.0.3 Release Manual



The Sun logo, Sun Microsystems, and Sun Workstation are registered trademarks of Sun Microsystems, Inc.

Sun, Sun-2, Sun-3, Sun-4, Sun386i, SunInstall, SunOS, SunView, NFS, NeWS, and SPARC are trademarks of Sun Microsystems, Inc.

UNIX is a registered trademark of AT&T.

All other products or services mentioned in this document are identified by the trademarks or service marks of their respective companies or organizations.

Copyright © 1983-1989 Sun Microsystems, Inc. – Printed in U.S.A.

All rights reserved. No part of this work covered by copyright hereon may be reproduced in any form or by any means – graphic, electronic, or mechanical – including photocopying, recording, taping, or storage in an information retrieval system, without the prior written permission of the copyright owner.

Restricted rights legend: use, duplication, or disclosure by the U.S. government is subject to restrictions set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 52.227-7013 and in similar clauses in the FAR and NASA FAR Supplement.

The Sun Graphical User Interface was developed by Sun Microsystems Inc. for its users and licensees. Sun acknowledges the pioneering efforts of Xerox in researching and developing the concept of visual or graphical user interfaces for the computer industry. Sun holds a non-exclusive license from Xerox to the Xerox Graphical User Interface, which license also covers Sun's licensees.

This software and documentation is based in part on the Fourth Berkeley Software Distribution under license from the Regents of the University of California. We acknowledge the following individuals and institutions for their role in its development: The Regents of the University of California, the Electrical Engineering and Computer Sciences Department at the Berkeley Campus of the University of California, and Other Contributors.

Contents

Chapter 1	Introduction	3
1.1.	Documentation Conventions	3
Chapter 2	Installing SunOS 4.0.3 and Pre-Configured Kernels	7
2.1.	Installation or Upgrade?	7
	Installing SunOS 4.0.3	7
	Upgrading to SunOS 4.0.3	7
2.2.	The sunupgrade Utility	7
	Who Upgrades?	8
2.3.	Pre-Configured Kernels	8
	Need for Configuration	8
	The GENERIC Kernel	8
	Choices	8
	Supplied Small System Kernels	9
	Supplied Configuration Files	9
Chapter 3	Software Changes in SunOS 4.0.3	13
3.1.	New Feature—mt Command Options	13
3.2.	Enhanced Diagnostic Software	13
	The sundiag Program	13
3.3.	Programs Moved to /usr/old	14
	sysdiag	14
	perfmon	14
	setkeys	14

3.4. New Software for New Hardware	14
New arch Options	14
Enhancements to make	14
Entries in /etc/fstab and /etc/exports	15
/usr: New and Changed Directories	15
New eject Utility	15
New fdformat Utility	15
Change to mon/eeeprom.h	15
New Devices	15
The GENERIC Configuration File	15
Chapter 4 The Sun-3/80 and Sun-3/470, Sun-3/480	
Workstations and Servers	19
4.1. The Sun-3/80 Desktop Workstation	19
Floppy for the Sun-3/80	19
4.2. Sun-3/470 Deskside Workstation and Sun-3/480 Server	19
Cartridges for the 150-megabyte 1/4" tape drive	20
4.3. Differences Between Sun-3 (68020-based) and Sun-3x (68030-	
based) Workstations	20
User programs	21
Drivers	21
Other Differences	21
Compiling Kernel-Dependent Code	22
Chapter 5 SPARCsystem 300 Deskside Workstations and	
Servers	25
5.1. SPARCsystem 300 Overview	25
5.2. SPARCsystem 330	25
Cartridges for the 150-megabyte 1/4" tape drive	25
Chapter 6 GPSI Enhancements	29
6.1. GPSI Features	29
6.2. Software Integration	30
Colorcube	30

Programmer's Guide	30
Chapter 7 CG6 Graphics Accelerator Board	33
7.1. Supported Workstations	33
7.2. Software Interface to CG6	33
Supported Graphics Application Software	33
Compatibility	33
Documentation	34
Chapter 8 Type-4 Keyboard	37
8.1. Keyboard Features	37
Type-4 Advantages	37
Changes for the Type-4	37
Unsupported Features	37
Chapter 9 24-Bit-Color Support	41
9.1. 24-Bit Color Frame Buffer Board, CG8	41
Chapter 10 SunDials	45
10.1. SunDials	45
Chapter 11 FPU2 Floating-Point Unit	49
Floating-Point Unit, FPU2	49
Chapter 12 FPA+	53
12.1. Description	53
Chapter 13 High-Performance SMD Disk Drive and Controller	57
13.1. 688-Megabyte SMD Disk Drive	57
13.2. VME/SMD Disk Controller	57
Chapter 14 Bug Fixes From SunOS 4.0.1 Through SunOS 4.0.3	61
14.1. Introduction	61
14.2. Graphics Bugs Fixed	61

System Hung When GP2 Rendered Polygons	61
Extra Pixels Affected Texturing	61
Incorrect Vector Identified for Picking	61
Overlay Colormap	62
_XF_LINE_FLT Left Extra Pixel	62
clip_flag Error on Sun-3/260	62
Polygons with Many Nonhorizontal Edges	62
SunPHIGS Programs Died	62
GP+ Drew Textured Polygons with PICK_NODRAW Set	62
Auto Arrays in pr_texvec.c with Same Name	62
Invisible Cursor with CG4 Frame Buffer	62
cframedemo Dumped Core	62
Incorrectly Drawn Cursor in suntools Reverse Video	62
Missing CG8 Driver Functions	62
Cursor Was Drawn Wrong in Reverse Video with CG8	63
pr_polyline Destroyed GP2 Context Line Texture	63
14.3. SunOS Bugs Fixed	63
/usr/ucb/w	63
filemerge	63
fpa_download	63
ldcore	63
libc	63
liblwp, -llwp	64
Misleading syslog(3) Manual Page	64
Daylight Savings Time Incorrectly Handled	64
386i Security Hole	64
14.4. Kernel Bugs Fixed	64
Diskless Client Boot Panic	64
Driver Mapping Requests	64
Pseudo-ttys Could Hang csh or shelltool	64
Diskless Clients Froze When Booted Simultaneously	64
Bus Error	64
KERNELBASE and SYSBASE Definitions	64

Missing Initialization in Socket Code	65
Vnode for Console Missing	65
exec Failures May Not Clean Up Properly	65
Interrupt Priorities on Sun-4s	65
Tapemaster Driver	65
Calls Improperly Nested	65
st Driver	65
Drivers Test Returns from kmem_alloc	65
sysdiag Test Hangs:	65
uucico	65
NIT STREAMS Buffering Module	65
Multiple Contiguous Writes Hung Bus	65
Null mblok Dereferenced	66
Reselect Interrupt Recovery	66
VPC-2200 Device Driver	66
Read and Write System Calls	66
System Crashed when Exiting Mailtool	66
mknod on Customized Kernel Caused Panic	66
Inverted Test in makenfsnode	66
Adaptec ACB-4000 Controllers	66
File Position Reported Incorrectly	66
Login Ports	66
kadb Hung When Booted on 4/110	66
asrtfail System Panic	66
Watchdog Reset with Bus Error on 4/110	67
Guards Added to Header Files in /usr/include	67
Kernel Prints Fault Information Carefully, Avoids Watchdog Reset	67
Interrupt Loss from 7053 Under Heavy I/O Caused System to Hang	67
Kernel Crashed when Synclloop Ran on Asynchronous MCP Port	67
ipc	67

On NFS, Vnode Unrecognized as <code>umount</code> Directory	68
kbdriver	68
lgkernel	68
Kernel Restarts Instruction that Caused Segmentation Fault	68
p0br, p0lr References Removed from Kernel	68
/movc.s:bcopy() Transferred Byte by Byte	68
ftruncate on /dev/zero Changed Major and Minor Device Numbers	68
mb	68
mmpeekio	69
UDP	69
skyr	69
Lost Files and vm_hat	69
yppush	69
HZ Exists in <code><sys/param.h></code> for Older System Compatibility	69
sizecheck Now Owned by Builder	69
lint Did Not Work for Standalone Drivers	69
14.5. SCSI Drivers Bugs Fixed	70
Length Errors on SCSI Tape	70
biowait Hang with SCSI ¼" Tape	70
SCSI-3 Driver Module si_cmd() Did Not Catch Bus Error	70
No ID on Arbitration of SCSI and SCSI-3 Driver	70
Missing #include Statement	70
SCSI Tape End-of-File Flag Not Cleared	70
SCSI Tape Driver: fsf Errors	71
Sysgen Tape Controllers: File Positioning	71
Tape File Positioning Errors in st.c	71
Misleading Messages when Cartridge Not in Drive	71
Extra File Marks on SCSI Tape after writi.sp	71
SCSI Error Recovery Generated Bus Error in scdone	71
tapetop Test Would Hang with SCSI Tape and SMD Disk	71

SCSI Driver Refused Send-Data's Odd-Length Transfer	71
sw0_cmd: Reselect failure on SunOS 4.0 with SCSI Patch	72
Bus Failure Prevented Recovery of Disk I/O for Swap	72
SCSI Failed when Dumping Core	72
14.6. Serial Drivers Bugs Fixed	72
Redirecting Console to Serial Device	72
Software Carrier Detect Flag Turned Off by MCP	72
DTR Asserted on Modem Line while Port Is Not Open	72
DTR Dropped Too Soon after Serial Port Close	72
Modem Status Unreadable	72
Flow-Control Problems with ALM Board	73
4.0 Crashed on Reboot If ALM Board Was Missing	73
ALM-2 Acted Up when Switching Between Raw and Cooked Mode	73
Multiple ALM-2 Boards Did Not Work Reliably	73
UUCP Unworkable at Transmissions Greater Than 9600 Baud	73
Timeout Table Overflow by MTI Driver	73
pty Driver Could Hang on Writes to Master Side	73
Intermittent Flow Control Problems on ALM with Plotter	73
ALM-2 Driver Was Dereferencing a Null Pointer	74
lpr Panicked System Using ALM-2 Parallel Port	74
Serial I/O Did Not Work Reliably	74
14.7. Network Bugs Fixed	74
Security Hole in yppasswd	74
PC-NFS Search Paths	74
Sun-4 Gateways Hung	74
Spurious VME Interrupts	74
Internet Name Server Crashes	74
Checksum Not Occurring	74
Broadcast Address Not Reset	75
Limited Maximum Segment Size	75

Break-out Routine Called in Error	75
passwd and yppasswd Did Not Reencrypt Correctly	75
makedbm Returned Inappropriate Message	75
ftp Bug	75
nfs Destroyed Files	75
ld error -98 with make_doc from gnuemacs	75
NFS-Based open Did Not Generate Network Status	75
Shared Locks Did Not Turn Off NFS Caching	76
NFS Data Caches Not Always Flushed	76
NFSD Swapped Out	76
rpc.lockd	76
rpc.mountd	76
Bad IP Checksums on Some Packets	76
Broadcast Requests Looped Back to Originating Machine	76
ypserv with Interdomain Resolver Hangs mount	76
ypbind	77
yppasswd	77
rsh and rlogin Failed Due to Order of hosts.equiv Entries	77
panic: dirremove During PC-NFS Writes to NFS- Mounted Filesystems	77
14.8. Compiler Bugs Fixed	77
No Recursive Check for Shared Libraries	77
ld Error Message	77
ld: Library Name and Common Name Collided	77
ld Option Could Not Be Turned Off	77
Nonspecific Exit Codes in ld.so Caused Clumsy Recoveries	78
ld.so Operated Incorrectly in Dynamic Binding to a Common	78
Nonstandard Link Command Generated adb and rm Failures	78
ld with -A, -x Options Produced Bad Output File	78
Watch Points Not Working Correctly	78

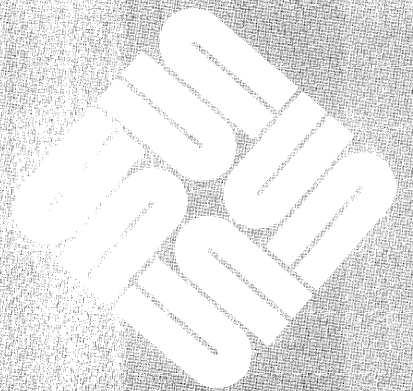
Bit Fields Displayed Incorrectly	78
Compile with <code>-Bstatic</code> Option Failed on 4.0 EXPORT	78
<code>newkey</code> Failed under 4.0 EXPORT	78
Older Sun-2 Object Files Failed to Link	79
Dynamically Linked Programs Require Page-Aligned Stack Limit to Execute	79
<code>a.out</code> Died without Error Message	79
<code>ld</code> Error Message when Linking Two-Object Executable and Sys V Shared Library	79
<code>opreg</code>	79
<code>target</code>	79
Sun-4 <code>/bin/as</code> Dumped Core with <code>/.stabs -01 -S</code>	79
14.9. Debugger Bugs Fixed	80
<code>dbxtool</code>	80
<code>dbx</code>	80
14.10. Utilities Bugs Fixed	81
<code>syslogd</code>	81
<code>catman</code>	81
<code>lpr</code>	81
<code>graph (1G)</code>	81
Security Holes	81
<code>tar</code>	81
<code>login (1)</code>	81
<code>passwd</code> Created Password Entry if <code>/etc/passwd</code> Contained Blank Lines	81
<code>uuxqt</code>	81
<code>lpd</code> Dumped Core when Filename Exceeded 132 Characters	81
<code>lpd</code> Passed <code>-w0</code> Default when <code>lpr</code> Width Unspecified	82
<code>Chesstool</code> Froze	82
Segmentation Violation Occurred In <code>chesstool</code>	82
<code>/usr/kvm/m68k</code> on Sun-3 Running 4.0 Is Incorrect	82
Printer Log File Error Messages Not Shown	82
<code>lpr -s</code> Rejected Files on Remote Filesystems	82

lpd Mailed Remote-Print Error Report to Illegal Address	82
14.11. System Administration Bugs Fixed	82
format	82
format Corrupted Kernel Map and Dumped Core with Incorrect Disk Type Selection	83
format Aborts if Defect Spans into Next Also-Mapped- Bad Sector	83
Defect List Difficult to Recreate	83
Long Hostnames Caused Boot to Fail	83
Sundiag Test Could Fail	84
Security Holes	84
Remote shutdown Failed	84
Shutdown Truncates the /etc/nologin Warning Message	84
mount Returned Incorrect Line Number for Illegal Entry in fstab	84
rpc.mountd Rejected Request for Exporting Filesystem to everybody	84
SCSI Defect List not Dynamically Allocated by format	84
Bus Error Resulted When Logical Defect Was Added to Null Defect List	84
System Accounting Bug Fixed	84
scsi.h File not included in 4.0 SCSI/ALM Patch Tape	85
14.12. SunView Bugs Fixed	85
sunview Exited Leaving Video Reversed	85
Shared Libraries Had Incorrect Version Number Warnings	85
perfmeter	85
14.13. SunInstall Bugs Fixed	85
Default ip Address Illegal	85
Root Partition Could be Designated Free Space Hog	85
Free Space Hog Not Used Soon Enough	85
syslog.conf Caused Loop	85
The setup_client Script	85
/export/swap Partition Not Used Properly	86

Setting Terminal Type to 'other' Dumps Core	86
Screen Forms Changed	86

Introduction

Introduction	3
1.1. Documentation Conventions	3



Introduction

This manual briefly describes the new features included in the SunOS™ 4.0.3 release.

SunOS 4.0.3 is a full SunOS 4.0 operating system, and is fully compatible with SunOS 4.0 and 4.0.1. Integrated into this new SunOS release is support for many new hardware devices and some software features. Included also are descriptions of bugs fixed since the release of SunOS 4.0.

SunOS 4.0.3 supports the Sun-2, Sun-3, Sun-4, and SPARCsystem 300 workstations and servers. While the Sun-3 family now includes the Sun-3x kernel architecture group (see Chapter 4 in this manual), differentiating the Sun-3x from the Sun-3 is necessary only when discussing kernel issues.

Note that two different sets of tapes exist for SunOS 4.0.3: one supports full installs only and one supports upgrades only. Please make sure you are using the correct tape set for your installation.

Read the first three chapters of this manual to get an overview of this document, a discussion of SunOS 4.0.3 installation, and a description of software changes in this SunOS release. The remaining chapters describe new features and bugs fixed.

1.1. Documentation Conventions

This manual follows these conventions:

- System commands and messages, as well as SunOS filenames, appear in `listing font like this`.
- Information you type in response to the system is shown in **bold listing font like this**.
- Information you type that differs between users and systems is shown in *bold italics like this*. These variables often are in tables in the text, where you select the correct entry for your system.

- Document titles show in plain, nonbold, *italic* font.
- Dialogues between you and the system are enclosed in gray boxes, like this command to remove a file and the system request for confirmation:

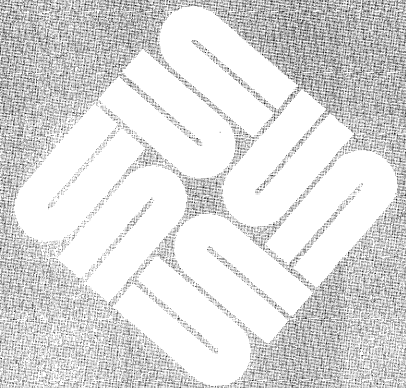
```
hostname% rm filename  
rm: remove filename? y
```

- Sections of program code show in plain listing font, enclosed in clear boxes:

```
int test[100];  
main()  
{  
    register int a, b, c, d, e, f;  
  
    test[a] = b & test[c & 0x1] & test[d & 0x1];  
}
```

Installing SunOS 4.0.3 and Pre-Configured Kernels

Installing SunOS 4.0.3 and Pre-Configured Kernels	7
2.1. Installation or Upgrade?	7
Installing SunOS 4.0.3	7
Upgrading to SunOS 4.0.3	7
2.2. The sunupgrade Utility	7
Who Upgrades?	8
2.3. Pre-Configured Kernels	8
Need for Configuration	8
The GENERIC Kernel	8
Choices	8
Supplied Small System Kernels	9
Supplied Configuration Files	9



Installing SunOS 4.0.3 and Pre-Configured Kernels

2.1. Installation or Upgrade?

This chapter explains when to fully install SunOS 4.0.3 on your system, and when to upgrade your system to SunOS 4.0.3. Refer to the *Installing the SunOS 4.0.3* guide for system upgrade and complete system installation instructions. *Installing the SunOS 4.0.3* also provides descriptions, instructions, and walk-through examples for the installation options.

Note The SunOS distribution tapes are labeled. *If the label specifies "Upgrade," you must perform a system upgrade. If it does not, you must perform a full installation instead.* See the next sections for more information.

Installing SunOS 4.0.3

To migrate from SunOS 3.x to SunOS 4.0.3, or to install a new system, perform a *full installation*. A full installation installs an entire operating system, overwriting the existing system files.

See the *Installing the SunOS 4.0.3* guide for full installation instructions.

Upgrading to SunOS 4.0.3

To migrate from SunOS 4.0 or 4.0.1 to this release, 4.0.3, perform a system upgrade using the upgrade utility, `sunupgrade`.

When you perform a system upgrade, you selectively replace those files that are different between the release you are running and the new release to which you are migrating.

See the upgrade chapter in *Installing the SunOS 4.0.3* for upgrade instructions and walkthrough examples.

2.2. The `sunupgrade` Utility

This section describes the new `sunupgrade` utility.

The SunOS utility `/usr/etc/sunupgrade` exists in the miniroot on the SunOS 4.0.3 release tape set. Once the miniroot files are copied over to disk, you can run `sunupgrade` to perform a system upgrade. The `sunupgrade` utility provides the necessary files to upgrade your system from SunOS 4.0 or 4.0.1.

The upgrade process is straightforward, with an interface that requires a minimum of user input. User-customizable files are preserved.

During the upgrade, `sunupgrade` prompts you for the following information:

- Disk device of the `root` partition
- Type of configuration: file-server, standalone, dataless client
- Local or remote installation
- Tape type
- Any changed directory names, if you are upgrading a server
- Other information, particularly if performing a remote installation

In addition, `sunupgrade` runs in “no rewind” mode for 1/4” tape, greatly speeding system upgrades. Changes to the SCSI tape drivers in SunOS 4.0.3 make this mode of operation possible. Running `sunupgrade` from the SunOS 4.0.3 miniroot allows “no rewind” operation, but remote tape installation from a tapehost running a release prior to SunOS 4.0.3 `sunupgrade` disables “no rewind.” You can explicitly disable “no rewind” mode by invoking `sunupgrade` with the “-n” flag.

As a safety feature when installing dataless clients, `sunupgrade` does not upgrade a dataless client to SunOS 4.0.3 if the server does not also run 4.0.3. The *SunOS 4.0.3 Release Manual* contains the `sunupgrade` (8) manual page.

Who Upgrades?

SunOS 4.0.3 supports all combinations of Sun-2, Sun-3, and Sun-4 clients (both standalone and dataless), and servers.

2.3. Pre-Configured Kernels

In addition to the large default GENERIC kernel supplied with the SunOS 4.0.3, four pre-configured kernels for small systems are also provided. Kernel configuration options are briefly covered below, and are fully discussed in the *Installing the SunOS 4.0.3* guide.

Need for Configuration

Configuring your kernel or using one of the supplied configurations can significantly improve system performance.

The GENERIC Kernel

After either an upgrade to SunOS 4.0.3 or a full SunOS installation, you have a large default GENERIC kernel. The GENERIC kernel contains all device drivers and options, including many that you might not need.

Choices

You have four choices of kernels to run after installation:

- The large default GENERIC kernel (not recommended)
- Four preconfigured small system kernels that you do not need to build
- Sun-supplied kernel configuration files that simplify building a custom kernel
- A custom kernel that you build

**Supplied Small System
Kernels**

Pre-configured small system kernels are listed below.

- Sun-2: 2/50 Diskless or with up to two SCSI disks and one SCSI tape
- Sun-3 (68020-based systems): 3/50 and 3/60 diskless or with up to two SCSI disks and one SCSI tape
- Sun-3x (68030-based systems): 3/80 diskless or with up to two SCSI disks, one SCSI tape, and one floppy drive
- Sun-4: 4/110 or SPARCsystem 330 with up to four SCSI disks and one SCSI tape

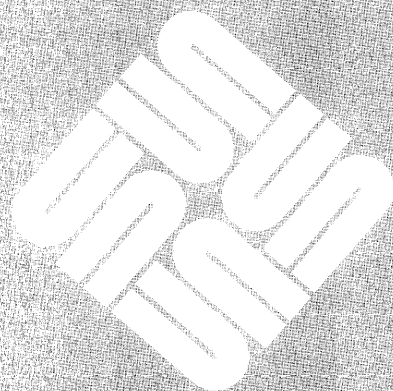
Supplied Configuration Files

Many convenient kernel configuration files are supplied in `/usr/sys/sunARCH/conf`. *ARCH* is **2**, **3**, **3x** or **4** for Sun-2, Sun-3 (68020-based), Sun-3x (68030-based), and Sun-4 systems, respectively. These configurations are for common systems, and allow you to rebuild a custom kernel more easily.

In `/usr/sys/sunARCH/conf`, file names beginning with `DL` are for diskless systems. Names beginning with `SDST` are for SCSI disk and tape, and `XD` and `XY` files support Xylogics disk controllers.

Software Changes in SunOS 4.0.3

Software Changes in SunOS 4.0.3	13
3.1. New Feature—mt Command Options	13
3.2. Enhanced Diagnostic Software	13
The <code>sundiag</code> Program	13
3.3. Programs Moved to <code>/usr/old</code>	14
<code>sysdiag</code>	14
<code>perfmon</code>	14
<code>setkeys</code>	14
3.4. New Software for New Hardware	14
New <code>arch</code> Options	14
Enhancements to <code>make</code>	14
Entries in <code>/etc/fstab</code> and <code>/etc/exports</code>	15
<code>/usr</code> : New and Changed Directories	15
New <code>eject</code> Utility	15
New <code>fdformat</code> Utility	15
Change to <code>mon/eeprom.h</code>	15
New Devices	15
The <code>GENERIC</code> Configuration File	15



Software Changes in SunOS 4.0.3

This chapter describes software changes in SunOS 4.0.3.

3.1. New Feature—`mt` Command Options

The `mt (1)` command supports these new options. For complete information, see the updated manual page shipped with SunOS 4.0.3.

- bsfm** Back space count file marks. The tape is positioned on the beginning-of-tape side of the file mark.
- asf** Absolute space to count file number. 1/4" cartridge drives skip directly from the current tape position to the requested file number. On 1/2" drives, the command is equivalent to a rewind followed by an `fsf` count.
- com** Space to the end of recorded media on the tape (SCSI only). This is useful for appending files onto previously written tapes.

3.2. Enhanced Diagnostic Software

SunOS 4.0.3 includes a new diagnostic system, `sundiag`. This is a SunView-based user interface that tests system devices and peripherals.

The `sundiag` Program

The `sundiag` program is an online system exerciser for testing peripheral devices. Incorporating a SunView-based interface, `sundiag` can be used on any Sun-2, Sun-3, or Sun-4 hardware configuration running SunOS 4.0 and later. Configure the system kernel to support all peripheral devices to be tested.

`sundiag` executes system tests that formerly were performed by `sysdiag`, and runs all `sundiag` tests written for new Sun products. Read the *Sundiag User's Guide*, Part Number 800-3804, for complete information about using this test program. The `sundiag` program is in the `/usr/diag/sundiag` directory.

3.3. Programs Moved to /usr/old

- sysdiag** The `sysdiag` utility is being replaced with the new `sundiag` utility, a window-based diagnostic exerciser. For SunOS 4.0.3, `sysdiag` is moved to `/usr/old`.
- perfmon** The `perfmon` utility was moved to `/usr/old/perfmon` in SunOS 4.0.
- setkeys** The `setkeys` program is being phased out of SunOS and replaced by the *input* function in the `defaultsedit` program. The `setkeys` utility was moved to `/usr/old/setkeys` in SunOS 4.0.

NOTE: `/usr/old/setkeys` does not work with the Type-4 keyboard shipped with the new Sun hardware.

3.4. New Software for New Hardware

New `arch` Options

The following software features have been added to, or changed in, SunOS 4.0.3.

The `arch` command includes the new option, `-k`. Executing `arch -k` prints the kernel architecture type, such as `sun3` (for 68020-based systems) or `sun3x` (for 68030-based systems). This indicates the UNIX[®] kernel that runs on the machine, and is important only for programs explicitly depending on the kernel architecture.

The *archname* option is also new to the `arch` command. The command `arch archname` returns `true` when the machine can run the application binaries of the *archname* machine architecture, and returns `false` when it cannot run those binaries.

For example, `arch sun3` returns `true` on both Sun-3 and Sun-3x kernel architectures because these machines are binary compatible for application software.

See Section 4.3, *Differences Between Sun-3 (68020-based) and Sun-3x (68030-based) Workstations* for an explanation of the term “kernel architecture.” See the new `arch(1)` manual page in the *Change Pages and Addenda to the Docubox* document set for details of the new `arch` command options.

Enhancements to `make`

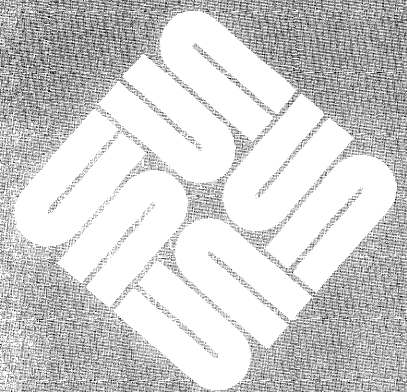
The `make(1)` command is enhanced in the way it assigns the output of shell commands to variables. With these changes, the `make` command takes advantage of new `arch(1)` features.

See the new `make(1)` manual page in the *Change Pages and Addenda to the Docubox* document set for new information.

Entries in <code>/etc/fstab</code> and <code>/etc/exports</code>	Because of changes to the directory structure of SunOS 4.0.3, system administrators must now mount <code>/usr/kvm</code> explicitly as an additional entry in the client's <code>/etc/fstab</code> file, and must explicitly export <code>/export/exec/kvm/client_kernel_arch</code> in the server's <code>etc/exports</code> .
Note to System Administrators	System Administrators: See the <i>System Administration Addenda</i> in the <i>Change Pages and Addenda to the Docubox</i> for full explanations of SunOS 4.0.3 system administration issues.
<code>/usr</code>: New and Changed Directories	Read about changes to the <code>/usr</code> directory structure in the <i>Change Pages and Addenda to the Docubox</i> document set, in the System Administration Addenda.
New <code>eject</code> Utility	The new <code>eject</code> utility supports software ejects for floppy disks on the Sun-3/80. Read the <code>eject(1)</code> manual page in the <i>Change Pages and Addenda to the Docubox</i> document set for a full description of <code>eject</code> features.
New <code>fdformat</code> Utility	The new <code>fdformat</code> utility supports floppy disk formatting on the Sun-3/80. Read the <code>fdformat(1)</code> manual page in the <i>Change Pages and Addenda to the Docubox</i> document set for a full description of <code>fdformat</code> features.
Change to <code>mon/eeeprom.h</code>	<code>mon/eeeprom.h</code> is changed to support the Sun-3/80.
New Devices	MAKEDEV now creates the <code>fd</code> , <code>pp</code> , and <code>ppdiag</code> devices. <ul style="list-style-type: none">□ <code>fd</code> is the Sun-3/80 floppy drive.□ <code>pp</code> is the Sun-3/80 printer port.□ <code>ppdiag</code> is a diagnostics special device file.
The <code>GENERIC</code> Configuration File	The standard <code>GENERIC</code> configuration file has been changed to support new devices.

The Sun-3/80 and Sun-3/470, Sun-3/480 Workstations and Servers

The Sun-3/80 and Sun-3/470, Sun-3/480 Workstations and Servers	19
4.1. The Sun-3/80 Desktop Workstation	19
Floppy for the Sun-3/80	19
4.2. Sun-3/470 Deskside Workstation and Sun-3/480 Server	19
Cartridges for the 150-megabyte 1/4" tape drive	20
4.3. Differences Between Sun-3 (68020-based) and Sun-3x (68030-based) Workstations	20
User programs	21
Drivers	21
Other Differences	21
Compiling Kernel-Dependent Code	22



The Sun-3/80 and Sun-3/470, Sun-3/480 Workstations and Servers

4.1. The Sun-3/80 Desktop Workstation

The Sun-3/80 is a low-cost desktop system featuring optional integrated mass storage and an optional floppy disk. Based on a 20 MHz Motorola 68030 processor with a base configuration of 4 megabytes of dynamic random access memory (DRAM), the Sun-3/80 is binary-compatible with most applications based on other Sun-3 products. This compact system has a wide variety of configurations.

The Sun-3/80 has many attractive features for users preferring the familiar 680x0 architecture:

- Compact packaging
- Motorola 68882 floating point unit (standard)
- Optional integrated floppy disk
- Optional integrated 3-1/2" SCSI hard disks
- Many frame buffer and graphics accelerator options
- External mass storage expansion capability

Floppy for the Sun-3/80

SunOS 4.0.3 supports the 3-1/2" flexible diskette (floppy disk) drive used in the Sun-3/80 Workstation. This floppy is an industry standard, non-SCSI device with a 1.44 megabyte capacity.

You use the floppy in the same ways as you use other disks. For example, the floppy disk can contain a UNIX file system and can be mounted like any other disk partition. In addition, floppy disks can be used to boot the system and to copy application software from diskette to your system. SunOS knows the floppy as `/dev/fd0`.

4.2. Sun-3/470 Deskside Workstation and Sun-3/480 Server

The Sun-3/470 and Sun-3/480 are based on the 33 MHz Motorola 68030 processor. Both systems provide many important features and offer a variety of expansion options:

- 33 MHz Motorola 68882 floating point unit (standard)
- Optional high-performance floating point accelerators
- Many frame buffer and graphics accelerator options

- 60- and 150-megabyte 1/4" tape drives
- 1/2" 6250/1600 bpi tape drive
- Up to 1.3 gigabytes of SCSI disk storage
- Any combination of 4 SMD-4 controllers and 8 SMD disk drives on the Sun-3/470
- Any combination of 4 SMD-4 controllers and 16 SMD disk drives on the Sun-3/480

Cartridges for the 150-megabyte 1/4" tape drive

The 150-megabyte tape drive can *write* to the 3M-DC6150 tape cartridge (formerly the 3M-DC600XTD tape cartridge), available as Sun part number 370-1203-01.

Tapes written on the 60-megabyte tape drive, such as the 3M-DC300XL/P tape cartridge (Sun part number 370-0543) and DEI Series II Silver (Sun part number 370-0543) tape cartridge, can be *read* on the 150-megabyte tape drive. **Failure to use the proper tape for the specified tape drives, or use of any other tape cartridges, is not supported by Sun Microsystems.**

To find the type of tape drive installed on your system, load a tape into the drive and enter the following command, replacing *NUM* with the number of your tape device (for example, `/dev/rst8`):

```
mt -f /dev/rstNUM status
```

The following message displays when you have a 150-megabyte 1/4" tape drive:

```
Archive QIC-150 tape drive:
  sense key(0x0)= no sense   residual= 0   retries= 0
  file no= 0   block no= 0
```

4.3. Differences Between Sun-3 (68020-based) and Sun-3x (68030-based) Workstations

The new 68030-based, Sun-3x, computers (Sun-3/80, Sun-3/470, and Sun-3/480) are members of the Sun-3 family and run Sun-3 user-level applications unchanged. However, while the Sun-3 uses the MC68020 and the Sun MMU, the Sun-3x uses the MC68030 with an on-chip MMU. This difference necessitates a unique kernel for each of the two system architectures.

For detailed information about the MC68020, see the *MC68020 User's Manual, MC68020UM/AD Rev 1* and the *Sun-3 Architecture Manual*. To understand the MC68030, read the *MC68030 User's Manual, MC68030UM/AD Rev 1*.

Because the 68020 and 68030 run the same application binaries, they are said to have a *sun3 application architecture*. Because they require different kernel-dependent code, however, their *kernel architecture* is different: the 68020 has a *sun3* kernel architecture, while the 68030 has a *sun3x* kernel architecture.

The application architecture is displayed by `arch`, while the kernel architecture is displayed by `arch -k`. See the `arch(1)` manual page for details.

Following is a list of differences between the Sun-3 and Sun-3x architectures that can require you to relink, recompile, or rewrite a program.

User programs

- Relink, or recompile and relink, programs that read or write kernel data structures. Relink programs that use the `libkvm` library. Recompile programs that depend on kernel data structure offsets or sizes.
- Programs using the 68020 `callm` and `rtm` instructions do not work on the 68030 because these instructions do not exist on the 68030. This should not cause problems, as the compilers never generate these instructions.
- Re-port programs that assume that the user stack starts at a particular location. On a Sun-3x, the user stack starts at `0xe0000000`. On a Sun-3, the user stack starts at `0xf0000000`. In particular, programs assuming the stack pointer is “positive” will not work.

Drivers

- Recompile drivers accessing kernel data structures (such as the user area and the process table). The offsets of a field in these structures can be different between the Sun-3 and the Sun-3x.
- Rewrite drivers that “know” about page table format, such as some graphics drivers, to use the 68030 page table format.
- Rewrite drivers with hard-coded kernel addresses to use the correct addresses on the Sun-3x architecture.
- Change drivers that access the system enable register, such as some graphics drivers. This is a `short` on the Sun-3x architecture rather than a `char`.

The following two items apply to the Sun-3/470 and Sun-3/480 only:

- Insert delays for some drivers (for example, the DES driver) to wait for hardware to be ready. The Sun-3/470 and Sun-3/480 machines need these delays because they are faster than other Sun-3 machines.
- For improved performance, rewrite disk and Ethernet drivers (and possibly others) to take advantage of the I/O cache. To do this, mark the buffer with the `B_IOCACHE` flag in the strategy routine if the buffer is aligned properly (16 bytes), and turn this flag off in the interrupt routine when the I/O completes.

Other Differences

The kernel, `kadb`, the boot blocks, the `tftpboot` program for Sun-3x clients, and `libkvm` are different between the 68020 and the 68030. Also, the `arch -k` command prints `sun3x` on a Sun-3x machine and `sun3` on a Sun-3 machine.

Compiling Kernel-Dependent Code

The following techniques are recommended when the same kernel-dependent source code is to be compiled for both the sun3 and sun3x targets.

- Use the enhancements to `arch(1)` and `make(1)` in your makefile to define either “sun3” or “sun3x.” These `arch(1)` and `make(1)` enhancements are new with SunOS 4.0.3, and the following example does not work with the `arch(1)` and `make(1)` commands from previous releases.

```
ARCH:sh = arch -k
CPPOPTS= -D$(ARCH)
CFLAGS= $(CPPOPTS)
```

- As appropriate, use `#ifdef sun3` and `#ifdef sun3x` in your code.

```
#if defined(sun3) || defined(sun3x)
    code identical for the Sun-3 and the Sun-3x
#endif sun3 || sun3x

#ifdef sun3
    code for the Sun-3 only
#endif

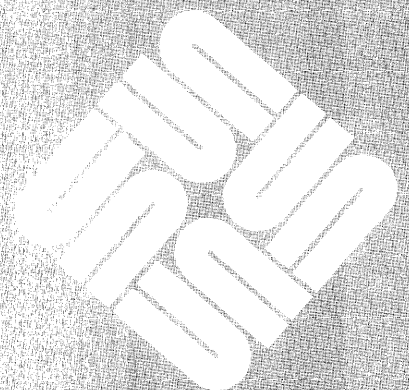
#ifdef sun3x
    code for the Sun-3x only
#endif
```

Note that you must explicitly ask for the definition of C pre-processor %architecture symbols like *sun3* and *sun3x*, invoking the compiler with `cc -Dsun3` or `cc -Dsun3x`. These symbols are not defined by default.

Also, while `-sun3` is a legal `cc` command-line option for both the 68020-based Sun-3 and the 68030-based Sun-3, `-sun3x` is not a legal `cc` command-line option.

SPARCsystem 300 Deskside Workstations and Servers

SPARCsystem 300 Deskside Workstations and Servers	25
5.1. SPARCsystem 300 Overview	25
5.2. SPARCsystem 330	25
Cartridges for the 150-megabyte 1/4" tape drive	25



SPARCsystem 300 Deskside Workstations and Servers

5.1. SPARCsystem 300 Overview

The SPARCsystem 300 is a family of high-performance computer systems based on the 25 MHz Reduced Instruction Set Computer (RISC) processor, SPARC (Scalar Processor ARChitecture). The SPARCsystem 300 packaging options and expansion capabilities meet a diverse set of computing needs.

5.2. SPARCsystem 330

The SPARCsystem 330 compact deskside package offers integrated mass storage and backup capabilities. Features of the SPARCsystem 330 include the following.

- High-performance floating point unit (standard)
- Many frame buffer and graphics accelerator options
- Three 9U × 400mm VME slots
- Two 6U × 160mm VME slots
- 150-megabyte 1/4" tape drive
- Up to 1.3 gigabytes of SCSI disk storage

Cartridges for the 150-megabyte 1/4" tape drive

The 150-megabyte tape drive can *write* to the 3M-DC6150 tape cartridge (formerly the 3M-DC600XTD tape cartridge), available as Sun part number 370-1203-01.

Tapes written on the 60-megabyte tape drive, such as the 3M-DC300XL/P tape cartridge (Sun part number 370-0543) and DEI Series II Silver (Sun part number 370-0543) tape cartridge, can be *read* on the 150-megabyte tape drive. **Failure to use the proper tape for the specified tape drives, or use of any other tape cartridges, is not supported by Sun Microsystems.**

To find the type of tape drive installed on your system, load a tape into the drive and enter the following command, replacing *NUM* with the number of your tape device (for example, /dev/rst8):

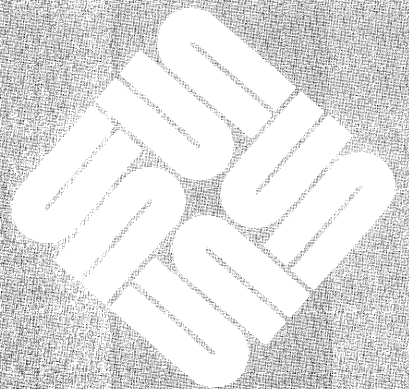
```
mt -f /dev/rstNUM status
```

The following message displays when you have a 150-megabyte 1/4" tape drive:

```
Archive QIC-150 tape drive:  
sense key(0x0)= no sense   residual= 0   retries=  
0  
file no= 0   block no= 0
```

GPSI Enhancements

GPSI Enhancements	29
6.1. GPSI Features	29
6.2. Software Integration	30
Colorcube	30
Programmer's Guide	30



GPSI Enhancements

6.1. GPSI Features

This newest release of GPSI (Graphics Processor Software Interface) microcode adds the following new features to the Sun graphics processor family—the GP2/CG9, GP2/CG5, and GP+/GB/CG3 graphics board sets.

1. Picking
2. Matrix Stack
3. Markers
4. Z-Buffered Vectors
5. Depth-Cued Vectors
6. RGB (Red-Green-Blue) Color Model Support, including Gouraud-shaded, RGB-interpolated polygons, and shading of vectors.
7. Lighting
8. Stroke Text

The first five items are new `GP1_` GPSI commands for the GP2 and GP+ graphics accelerator boards. They are inoperative on the older GP because of microcode space limitations, but will nevertheless run GPSI applications using the new commands on the GP without problems.

Items 6 and 7 extend the GPSI interface to support *true color*: color is specified and processed as RGB (red-green-blue) triplets. The new RGB vector and polygon support makes the GP2 a “true color” graphics engine and provides the foundation for advanced shading and lighting. For compatibility, the “index color” GPSI commands are still available. Items 6, 7, and 8 are provided only for the GP2 (not for the GP or the GP+); to make this clear, these GPSI commands begin with `GP2_` instead of with `GP1_`.

Lighting support is now provided by the GP2 as follows:

- Colored ambient light
- Eight independent colored light sources
- Diffuse light reflection model
- Infinitely distant directional light sources

- Back facing polygon rejection
- Back face lighting
- Specular reflection

In addition, the GP2 now supports the downloading of stroke fonts to provide fast stroke text drawing speeds.

These new GPSI microcode enhancements are supported on the new 24-bit, GP2/CG9 graphics accelerator board set, as well as the existing GP2/CG5 board set and, with exceptions as noted, the GP+/GB/CG3.

6.2. Software Integration

The GPSI microcode is integrated into SunOS 4.0.3, and is found in the `/usr/lib` directory.

Colorcube

To use the new RGB GPSI commands with the CG5 8-bit color board, load the CG5 color map with a 5-9-5 color cube (5 levels of red, 9 levels of green, and 5 levels of blue).

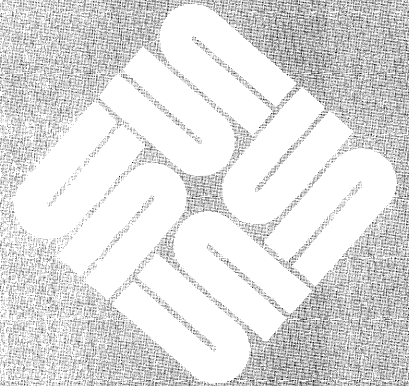
This 5-9-5 color map is in the header file
`/usr/include/sunwindow/cms_colorcube.h`.

Programmer's Guide

An addendum to the GPSI programmer's guide, *GPSI Programmer's Guide Addendum* (800-3616-01), is available.

CG6 Graphics Accelerator Board

CG6 Graphics Accelerator Board	33
7.1. Supported Workstations	33
7.2. Software Interface to CG6	33
Supported Graphics Application Software	33
Compatibility	33
Documentation	34



CG6 Graphics Accelerator Board

The CG6 board accelerates the performance of many 2D and 3D graphics applications. This P4 color frame buffer uses two ASICs to accelerate production of eight-bit color images. The GX graphics accelerator is the S4 version of the CG6.

7.1. Supported Workstations

The CG6 graphics accelerator plugs into the P4 bus on these Sun workstations: 3/60, 3/80, 3/470, 3/480, 4/110, and SPARCsystem 330. To install the board, follow the instructions in the hardware installation manual that accompanies the board.

7.2. Software Interface to CG6

The software that supports CG6 functionality is included in SunOS 4.0.3.

Supported Graphics Application Software

The following four types of graphics libraries are available for Sun workstations equipped with CG6 boards.

- PixWin
- Pixrect
- SunGKS
- SunPHIGS

Compatibility

When upgrading applications to the CG6, the following compatibility considerations apply.

- Applications prepared with CGI/CORE are not supported on the CG6.
- Statically linked applications run, but are not accelerated, on the CG6.
- Dynamically linked (or manually relinked) applications are accelerated by the CG6.
- Recoding is necessary in the following cases:
 - Where overlays are required or expected by the application.
 - Where the application depends on the existence of PixWin or Pixrect internal data structures.

- Where the application is written in CGI or CORE.

For more information on performance tuning applications for the CG6, refer to the *GX Technical Note*.

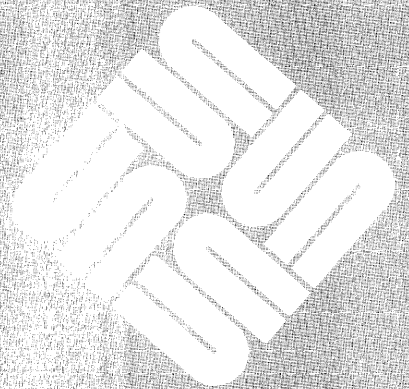
Documentation

The following manuals are general references on CG6-related topics.

- The *SunView Programmer's Guide*, Part Number 800-1783-10, and the *SunView System Programmer's Guide*, Part Number 800-1784-10
- The *Pixrect Reference Manual*, Part Number 800-1785-10
- The *SunPHIGS Reference Manual*, Part Number 800-2475-01, and the *SunPHIGS Programming Guide*, Part Number 800-2476-01
- The *GX Technical Notes*
- The *SunGKS Reference Manual*, Part Number 800-3560-01
- The *SunGKS Software Installation Guide*, Part Number 800-3561-01
- *Read This First* for the *SunGKS Installation Guide*, Part Number 800-3652-01

Type-4 Keyboard

Type-4 Keyboard	37
8.1. Keyboard Features	37
Type-4 Advantages	37
Changes for the Type-4	37
Unsupported Features	37



Type-4 Keyboard

8.1. Keyboard Features

Type-4 keyboards are the new standard for Sun workstations. SunOS releases 4.0 through 4.0.3 allow you to use the Type-4 keyboard as the earlier Type-3 keyboard was used. All earlier Sun keyboards are supported, also.

Type-4 Advantages

Current advantages of the Type-4 keyboard over the Type-3 keyboard are:

- The Type-4 keyboard has three more software-programmable function keys than the Type-3 keyboard.
- The Type-4 keyboard passes standards for electronic emissions in several countries.

Changes for the Type-4

The **[Left]** and **[Right]** keys on the bottom row of the older Type-3 keyboard now show diamond symbols instead of the words *Left* and *Right*. Key function for these “meta” keys is unchanged.

Unsupported Features

Many features of the Type-4 keyboard will be supported in coming SunOS releases. New keys on the the Type-4 keyboard that are not yet supported under SunView are:

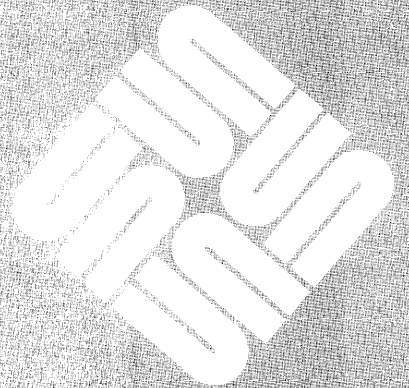
- **[Help]** on the left keypad
- **[Compose]**, **[Alt]**, and **[Alt Graph]** on the bottom keyboard row.
- All keys on the right (numeric) keypad.

Future releases will also support the “Key Lights” (also called LED) panel above the right keypad.

Finally, the `setkeys` utility (`/usr/old/setkeys`) does not work with the Type-4 keyboard and is being phased out. Use the `input` function in the `defaultsedit` program in place of `setkeys`.

24-Bit-Color Support

24-Bit-Color Support	41
9.1. 24-Bit Color Frame Buffer Board, CG8	41



24-Bit-Color Support

9.1. 24-Bit Color Frame Buffer Board, CG8

The new 24-bit frame buffer, the CG8, uses 24-bit-deep pixels to produce true-color images. The CG8 color lookup table allows for color adjustments, such as gamma correction, on the standard 900x1152 resolution display.

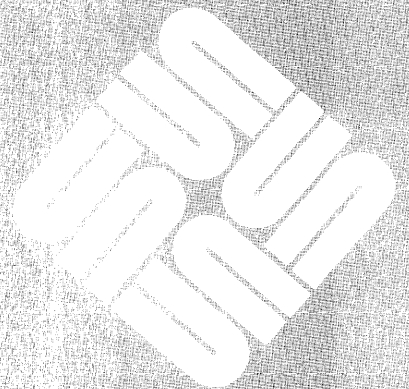
The monochrome overlay plane and enable plane allow displaying either the color or the monochrome plane on a pixel-by-pixel basis.

SunOS 4.0.3 provides a software driver, and code for enhancements to Pixrects and SunView 1 that take advantage of 24-bit color. The code changes are documented in the *Pixrect Reference Manual* and in the *SunView 1 Programmer's Guide*. See also the document set *Change Pages and Addenda for the Docubox* for release notes on 24-bit-color support.

Note: Eight-bit indexed color applications must be modified to run with the 24-bit color frame buffer. Unmodified applications will run, but will display images incorrectly—most likely, with an all-red screen.

SunDials

SunDials	45
10.1. SunDials	45



SunDials

10.1. SunDials

SunDials is an image-manipulation input device for Sun-3, Sun-4, and 386i workstations, using an RS-232 serial interface. The desk-top dialbox is compact, measuring 8.63"×5"×1.15".

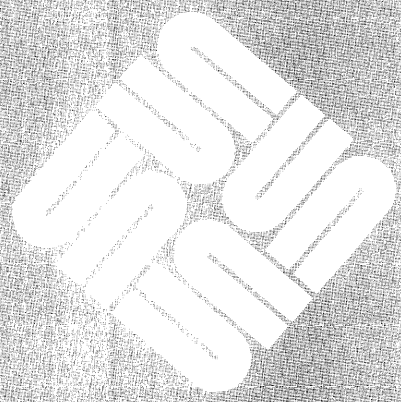
The dials can be programmed to change colors and manipulate images. Image manipulation such as scaling, translating, rotating, and zooming is essential for many CAD applications.

SunView directs SunDials input to the process owning the window (where the cursor is located). SunDials extends the interactivity of the display controller beyond the current capabilities of a mouse or digitizing tablet.

The SunDials device driver is integrated into the GENERIC kernel. No optional software or kernel reconfiguration is required, but kernel reconfiguration is recommended for optimal performance.

FPU2 Floating-Point Unit

FPU2 Floating-Point Unit	49
Floating-Point Unit, FPU2	49



FPU2 Floating-Point Unit

Floating-Point Unit, FPU2

The FPU2 is a new floating-point hardware unit for the Sun-4/110, 150, 260, and 280. The FPU2 has a TI 8847 floating-point unit with an LSI Logic controller chip.

SunOS 4.0/Fortran 1.1 programs can run unaltered after you install the FPU2. The inline expansion template file `/usr/lib/sqrt.il`, included on this tape, may optionally be used to improve performance of Sun-4 FPU2 on problems that perform many square root operations. The inline expansion templates replace calls to `sqrt` subroutines with hardware instructions. Executables created with these templates may run slowly on older Sun-4s with the Weitek 1164/5 floating-point hardware, which has no `sqrt` instructions.

Use the inline expansion template by recompiling with either of these commands:

```
f77 -04 source.f /usr/lib/sqrt.il /usr/lib/libm.il
or
cc -04 source.c /usr/lib/sqrt.il /usr/lib/libm.il -lm
```

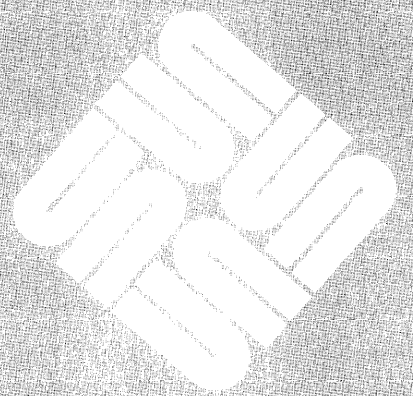
For more information on inline expansion templates, see the `inline(1)` manual page and the *Floating-Point Programmer's Guide* accompanying SunOS 4.0.

A new utility program searches for the FPU2. This utility, `fpuversion4(8)`, determines whether the high-performance floating point components are installed on the system CPU. Detecting the presence of the FPU2, the diagnostic prints a confirming message:

```
Sun-4 floating-point controller version 2 found.
```

FPA+

FPA+	53
12.1. Description	53



12.1. Description

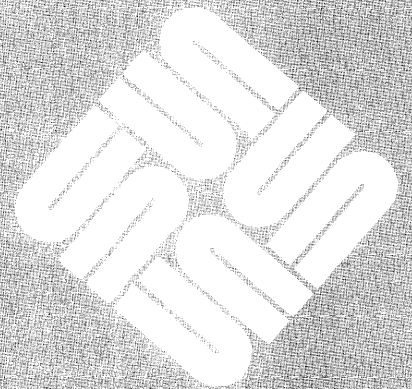
The FPA+ is a high-performance floating point accelerator for use in the Sun-3/470 and Sun-3/480 systems.

While it is object-code compatible with applications written for the Sun-3 FPA, this new unit performs floating point operations faster than the FPA because it runs at the CPU speed of 33 MHz and uses fewer cycles for each operation. While the FPA occupies an entire 9U slot, the FPA+ is a daughter board that attaches directly to the CPU board, and therefore occupies no additional slots.

Both FPA+ and FPA microcode are supplied with this release. The `fpa_download(8)` command is changed to pick up the appropriate version.

High-Performance SMD Disk Drive and Controller

High-Performance SMD Disk Drive and Controller	57
13.1. 688-Megabyte SMD Disk Drive	57
13.2. VME/SMD Disk Controller	57



High-Performance SMD Disk Drive and Controller

13.1. 688-Megabyte SMD Disk Drive

This is a high-capacity, 8" disk drive based on Winchester technology. Called the Storage Pedestal Upgrade Drive (SPUD), the drive has an unformatted capacity of 700 megabytes and formats to 688 megabytes. This is 2.5 times greater than the existing 8" disk drive in the Sun Mass Storage Pedestal, although it occupies the same physical space.

The SPUD also improves on the access time, cost per megabyte of storage, and reliability of the current 8" drive. Performance improvement results in part from SMD-4, the new SMD controller.

13.2. VME/SMD Disk Controller

This high-performance VME/SMD disk controller has a 128-kilobyte read-ahead cache and a pipelined bus DMA architecture. Each controller supports up to four of one type of the following disk drives:

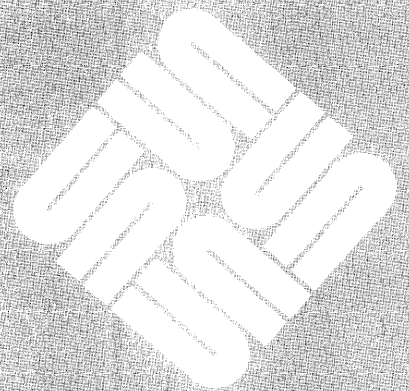
- 8" 688-megabyte disk drive
- 10" 575-megabyte disk drive
- 9" 892-megabyte disk drive

The disk controller also supports the following functions:

- Overlapped seeks
- Read and write optimizations

Bug Fixes From SunOS 4.0.1 Through SunOS 4.0.3

Bug Fixes From SunOS 4.0.1 Through SunOS 4.0.3	61
14.1. Introduction	61
14.2. Graphics Bugs Fixed	61
System Hung When GP2 Rendered Polygons	61
Extra Pixels Affected Texturing	61
Incorrect Vector Identified for Picking	61
Overlay Colormap	62
_XF_LINE_FLT Left Extra Pixel	62
clip_flag Error on Sun-3/260	62
Polygons with Many Nonhorizontal Edges	62
SunPHIGS Programs Died	62
GP+ Drew Textured Polygons with PICK_NODRAW Set	62
Auto Arrays in <code>pr_texvec.c</code> with Same Name	62
Invisible Cursor with CG4 Frame Buffer	62
cframedemo Dumped Core	62
Incorrectly Drawn Cursor in <code>suntools</code> Reverse Video	62
Missing CG8 Driver Functions	62
Cursor Was Drawn Wrong in Reverse Video with CG8	63
pr_polyline Destroyed GP2 Context Line Texture	63
14.3. SunOS Bugs Fixed	63
/usr/ucb/w	63
filemerge	63
fpa_download	63



ldcore	63
libc	63
liblwp, -llwp	64
Misleading syslog(3) Manual Page	64
Daylight Savings Time Incorrectly Handled	64
386i Security Hole	64
14.4. Kernel Bugs Fixed	64
Diskless Client Boot Panic	64
Driver Mapping Requests	64
Pseudo-ttys Could Hang csh or shelltool	64
Diskless Clients Froze When Booted Simultaneously	64
Bus Error	64
KERNELBASE and SYSBASE Definitions	64
Missing Initialization in Socket Code	65
Vnode for Console Missing	65
exec Failures May Not Clean Up Properly	65
Interrupt Priorities on Sun-4s	65
Tapemaster Driver	65
Calls Improperly Nested	65
st Driver	65
Drivers Test Returns from kmem_alloc	65
sysdiag Test Hangs:	65
uucico	65
NIT STREAMS Buffering Module	65
Multiple Contiguous Writes Hung Bus	65
Null mblok Dereferenced	66
Reselect Interrupt Recovery	66
VPC-2200 Device Driver	66
Read and Write System Calls	66
System Crashed when Exiting Mailtool	66
mknod on Customized Kernel Caused Panic	66
Inverted Test in makenfsnode	66
Adaptec ACB-4000 Controllers	66
File Position Reported Incorrectly	66
Login Ports	66

kadb Hung When Booted on 4/110	66
asrtfail System Panic	66
Watchdog Reset with Bus Error on 4/110	67
Guards Added to Header Files in /usr/include	67
Kernel Prints Fault Information Carefully, Avoids Watchdog Reset	67
Interrupt Loss from 7053 Under Heavy I/O Caused System to Hang	67
Kernel Crashed when Syncloop Ran on Asynchronous MCP Port	67
ipc	67
On NFS, Vnode Unrecognized as umount Directory	68
kbdriver	68
lgkernel	68
Kernel Restarts Instruction that Caused Segmentation Fault	68
p0br, p0lr References Removed from Kernel	68
/movc.s:bcopy() Transferred Byte by Byte	68
ftruncate on /dev/zero Changed Major and Minor Device Numbers	68
mb	68
mmpeekio	69
UDP	69
skyrp	69
Lost Files and vm_hat	69
yppush	69
HZ Exists in <sys/param.h> for Older System Compatibility	69
sizecheck Now Owned by Builder	69
lint Did Not Work for Standalone Drivers	69
14.5. SCSI Drivers Bugs Fixed	70
Length Errors on SCSI Tape	70
biowait Hang with SCSI 1/4" Tape	70
SCSI-3 Driver Module si_cmd() Did Not Catch Bus Error	70
No ID on Arbitration of SCSI and SCSI-3 Driver	70
Missing #include Statement	70

SCSI Tape End-of-File Flag Not Cleared	70
SCSI Tape Driver: <code>fsf</code> Errors	71
System Tape Controllers: File Positioning	71
Tape File Positioning Errors in <code>st.c</code>	71
Misleading Messages when Cartridge Not in Drive	71
Extra File Marks on SCSI Tape after <code>writi.sp</code>	71
SCSI Error Recovery Generated Bus Error in <code>scdone</code>	71
<code>tapetop</code> Test Would Hang with SCSI Tape and SMD Disk	71
SCSI Driver Refused Send-Data's Odd-Length Transfer	71
<code>sw0_cmd</code> : Reselect failure on SunOS 4.0 with SCSI Patch	72
Bus Failure Prevented Recovery of Disk I/O for Swap	72
SCSI Failed when Dumping Core	72
14.6. Serial Drivers Bugs Fixed	72
Redirecting Console to Serial Device	72
Software Carrier Detect Flag Turned Off by MCP	72
DTR Asserted on Modem Line while Port Is Not Open	72
DTR Dropped Too Soon after Serial Port Close	72
Modem Status Unreadable	72
Flow-Control Problems with ALM Board	73
4.0 Crashed on Reboot If ALM Board Was Missing	73
ALM-2 Acted Up when Switching Between Raw and Cooked Mode	73
Multiple ALM-2 Boards Did Not Work Reliably	73
UUCP Unworkable at Transmissions Greater Than 9600 Baud	73
Timeout Table Overflow by MTI Driver	73
pty Driver Could Hang on Writes to Master Side	73
Intermittent Flow Control Problems on ALM with Plotter	73
ALM-2 Driver Was Dereferencing a Null Pointer	74
lpr Panicked System Using ALM-2 Parallel Port	74
Serial I/O Did Not Work Reliably	74
14.7. Network Bugs Fixed	74
Security Hole in <code>yppasswd</code>	74
PC-NFS Search Paths	74

Sun-4 Gateways Hung	74
Spurious VME Interrupts	74
Internet Name Server Crashes	74
Checksum Not Occurring	74
Broadcast Address Not Reset	75
Limited Maximum Segment Size	75
Break-out Routine Called in Error	75
passwd and yppasswd Did Not Reencrypt Correctly	75
makedbm Returned Inappropriate Message	75
ftp Bug	75
nfs Destroyed Files	75
ld error -98 with make_doc from gnuemacs	75
NFS-Based open Did Not Generate Network Status	75
Shared Locks Did Not Turn Off NFS Caching	76
NFS Data Caches Not Always Flushed	76
NFSD Swapped Out	76
rpc.lockd	76
rpc.mountd	76
Bad IP Checksums on Some Packets	76
Broadcast Requests Looped Back to Originating Machine	76
ypserv with Interdomain Resolver Hangs mount	76
ypbind	77
yppasswd	77
rsh and rlogin Failed Due to Order of hosts.equiv Entries	77
panic: dirremove During PC-NFS Writes to NFS- Mounted Filesystems	77
14.8. Compiler Bugs Fixed	77
No Recursive Check for Shared Libraries	77
ld Error Message	77
ld: Library Name and Common Name Collided	77
ld Option Could Not Be Turned Off	77
Nonspecific Exit Codes in ld.so Caused Clumsy Recoveries	78
ld.so Operated Incorrectly in Dynamic Binding to a Common	78

Nonstandard Link Command Generated <code>adb</code> and <code>rm</code>	
Failures	78
<code>ld</code> with <code>-A</code> , <code>-x</code> Options Produced Bad Output File	78
Watch Points Not Working Correctly	78
Bit Fields Displayed Incorrectly	78
Compile with <code>-Bstatic</code> Option Failed on 4.0 EXPORT	78
<code>newkey</code> Failed under 4.0 EXPORT	78
Older Sun-2 Object Files Failed to Link	79
Dynamically Linked Programs Require Page-Aligned Stack	
Limit to Execute	79
<code>a.out</code> Died without Error Message	79
<code>ld</code> Error Message when Linking Two-Object Executable	
and Sys V Shared Library	79
<code>opreg</code>	79
<code>target</code>	79
Sun-4 <code>/bin/as</code> Dumped Core with <code>/.stabs -01 -S</code>	79
14.9. Debugger Bugs Fixed	80
<code>dbxtool</code>	80
<code>dbx</code>	80
14.10. Utilities Bugs Fixed	81
<code>syslogd</code>	81
<code>catman</code>	81
<code>lpr</code>	81
<code>graph(1G)</code>	81
Security Holes	81
<code>tar</code>	81
<code>login(1)</code>	81
<code>passwd</code> Created Password Entry if <code>/etc/passwd</code>	
Contained Blank Lines	81
<code>uuxqt</code>	81
<code>lpd</code> Dumped Core when Filename Exceeded 132	
Characters	81
<code>lpd</code> Passed <code>-w0</code> Default when <code>lpr</code> Width Unspecified	82
<code>Chesstool</code> Froze	82
Segmentation Violation Occurred In <code>chesstool</code>	82
<code>/usr/kvm/m68k</code> on Sun-3 Running 4.0 Is Incorrect	82

Printer Log File Error Messages Not Shown	82
lpr -s Rejected Files on Remote Filesystems	82
lpd Mailed Remote-Print Error Report to Illegal Address	82
14.11. System Administration Bugs Fixed	82
format	82
format Corrupted Kernel Map and Dumped Core with Incorrect Disk Type Selection	83
format Aborts if Defect Spans into Next Also-Mapped- Bad Sector	83
Defect List Difficult to Recreate	83
Long Hostnames Caused Boot to Fail	83
Sundiag Test Could Fail	84
Security Holes	84
Remote shutdown Failed	84
Shutdown Truncates the /etc/nologin Warning Message	84
mount Returned Incorrect Line Number for Illegal Entry in fstab	84
rpc.mountd Rejected Request for Exporting Filesystem to everybody	84
SCSI Defect List not Dynamically Allocated by format	84
Bus Error Resulted When Logical Defect Was Added to Null Defect List	84
System Accounting Bug Fixed	84
scsi.h File not included in 4.0 SCSI/ALM Patch Tape	85
14.12. SunView Bugs Fixed	85
sunview Exited Leaving Video Reversed	85
Shared Libraries Had Incorrect Version Number Warnings	85
perfmeter	85
14.13. SunInstall Bugs Fixed	85
Default ip Address Illegal	85
Root Partition Could be Designated Free Space Hog	85
Free Space Hog Not Used Soon Enough	85
syslog.conf Caused Loop	85
The setup_client Script	85
/export/swap Partition Not Used Properly	86

Setting Terminal Type to 'other' Dumps Core 86
Screen Forms Changed 86

Bug Fixes From SunOS 4.0.1 Through SunOS 4.0.3

14.1. Introduction

This chapter describes the bugs fixed since the release of SunOS 4.0. The bug-fix descriptions are categorized as follows:

- Graphics
- SunOS
- Kernel
- SCSI drivers
- Serial drivers
- Network
- Compiler
- Debugger
- Utilities
- System Administration
- SunView

14.2. Graphics Bugs Fixed

The following graphics bugs have been fixed.

System Hung When GP2
Rendered Polygons

GP2 could hang the system when trying to render some polygons.

Extra Pixels Affected Texturing

The GP2 (PHIGS microcode) drew texture lines that appeared to be incorrect. There were additional pixels at each break in the texturing.

Incorrect Vector Identified for
Picking

When lines (usually diagonal) were transformed through GP2 using `_PROC_LINE_FLT_3D`, bit 15 of `clip_flag` was not set if the vector was totally invisible. This caused an incorrect vector to be identified for picking in SunPHIGS.

Overlay Colormap	The colormap of a one-bit-deep frame buffer can now be changed.
<code>_XF_LINE_FLT</code> Left Extra Pixel	<code>_XF_LINE_FLT</code> could leave an extra pixel outside the clipping viewport.
<code>clip_flag</code> Error on Sun-3/260	When running GP2 on a Sun-3/260, there could be an error in <code>clip_flag</code> when you used the command <code>GP1_PROC_LINE_FLT_3D</code> .
Polygons with Many Nonhorizontal Edges	If you used <code>GP1_XF_PGON_INT_2D</code> to draw a polygon with 370 or more nonhorizontal edges, the GP2 could hang the system.
SunPHIGS Programs Died	SunPHIGS programs died (typically, receiving <code>SIGXCPU</code> signal while drawing SOLID-filled fill areas) on a Sun-4/150.
GP+ Drew Textured Polygons with <code>PICK_NODRAW</code> Set	The GP+ rendered textured polygons even though the pick mode flag was set to <code>PICK_NODRAW(2)</code> . This has been fixed so that, in pick mode, primitives are analyzed for picking, but not drawn.
Auto Arrays in <code>pr_texvec.c</code> with Same Name	Several functions in <code>pr_texvec.c</code> declared auto arrays named <code>segarray</code> , which could be used after the declaring functions returned. The problem was noticed when a program using <code>pr_line</code> looped indefinitely in <code>bres_vert</code> after compiling the (SPARC) <code>Pixrect</code> library -02. (The stack layout was sufficiently rearranged to manifest the problem.)
Invisible Cursor with CG4 Frame Buffer	When <code>suntools</code> displayed an alert box, the cursor was invisible on any machine with a CG4 frame buffer.
<code>cframedemo</code> Dumped Core	The <code>cframedemo</code> program compiled, but dumped core when it ran.
Incorrectly Drawn Cursor in <code>suntools</code> Reverse Video	Because of an initially faulty CG8 <code>pixrect</code> implementation, the cursor was not drawn in reverse video. The <code>cg8_colormap.c</code> file has been changed to fix this bug.
Missing CG8 Driver Functions	The <code>cgeight</code> driver functions were missing from the <code>stubs.c</code> file, causing undefined symbols messages in some configurations. Adding <code>cgeight</code> driver functions to the <code>stubs.c</code> file fixed this bug.

Cursor Was Drawn Wrong in Reverse Video with CG8	<i>sunview</i> uses the side-effect of the <code>pixrect</code> library function <code>pr_putcolormap</code> to decide if the screen is in reverse video. It expects the kernel to have the same function with the same side-effect to draw the cursor correctly. CG8 <i>pixrect</i> implementation initially did not do this. This is fixed.
<code>pr_polyline</code> Destroyed GP2 Context Line Texture	The <code>pixrect</code> code has been changed to reset the line type after polymarker primitives.
14.3. SunOS Bugs Fixed	This section describes bug fixes in the SunOS. Those applicable to the operating system are listed first.
<code>/usr/ucb/w</code>	The <code>w</code> program did not display processes on the console.
<code>filemerge</code>	The <code>filemerge</code> program was inadvertently left out of SunOS 4.0 (it should have been moved to <code>/usr/old</code>). The program is now in place, and the 4.0 <i>SunOS Reference Manual</i> contains a manual page, <code>oldfilemerge(1)</code> .
<code>fpa_download</code>	<code>fpa_download</code> , which runs from <code>rc.local</code> at boot time, now outputs data to indicate that it is downloading the FPA or FPA+ programs. The <code>rc.local</code> file is changed with this fix, and the fix file is stored where all clients of a server have access to it. Read and follow the installation instructions carefully to prevent overwriting customized <code>rc.local</code> files.
<code>ldcore</code>	When compiling a set of <code>.o</code> files under SunOS 4.0 and using two <i>lds</i> to link the programs, the second <i>ld</i> dumped core. The problem was associated with overflowing the hard limit for the number of symbols in one <code>.o</code> file. This bug is fixed.
<code>libc</code>	Execution of <code>setlogmask(3)</code> resulted in a core dump and segmentation fault; calling <code>syslog(3)</code> before <code>openlog(3)</code> dumped core; yellow-pages support for the resolver (nameserver) did not work; and <code>fput</code> failed when there was more than one newline on line-buffered streams. This bug was introduced when many library routines were changed. A number of <code>libc</code> (and other) files are now changed, and special rebuild and installation instructions are given in the README file of the 4.0.1 bugfix tape. These bugs are fixed.

liblwp, -llwp	Programs did not link with the lightweight process library; Sun-2 and Sun-3 link attempts caused an error message, REGOFFSET undefined; Sun-4 attempts produced SP_OFFSET, PC_OFFSET, and others. This bug is fixed.
Misleading syslog(3) Manual Page	According to the manual page, "If special processing is needed, openlog() can be called to initialize the log file." In fact, openlog(3) <i>must</i> be called before syslog(3) is used or logging will not occur.
Daylight Savings Time Incorrectly Handled	Daylight savings time was handled incorrectly for the U.K., Australia, and parts of Europe. This has been fixed to match the latest EEC agreement.
386i Security Hole	The -h feature of ypset could breach yp domain security to allow root access.
14.4. Kernel Bugs Fixed	This section describes bug fixes in the SunOS kernel code.
Diskless Client Boot Panic	While booting a diskless machine under 4.0, a sleep occurred that caused the system to panic.
Driver Mapping Requests	The kernel routines in SunN/machdep.c use a locally defined constant MAXMPHYS to limit the maximum DVMA window for raw device I/O. If a driver requested a larger mapping, the kernel would panic.
Pseudo-ttys Could Hang csh or shelltool	Depending on the setting of shell environment variables, a pseudo-tty could cause a shell (csh) or shelltool to exit or hang. Rebooting the system returned the pty to its normal state. This has been fixed.
Diskless Clients Froze When Booted Simultaneously	When booting multiple diskless clients simultaneously from a 3/280 server, some clients froze. This is no longer the case.
Bus Error	Running a program that used TIOCOUTQ ioctl caused a "panic: bus error" under SunOS 4.0. The tty.c program has been changed so this no longer happens.
KERNELBASE and SYSBASE Definitions	KERNELBASE and SYSBASE definitions, along with a problem in buscheck(), precluded mapping in buffers that were allocated in kernel data/bss.

Missing Initialization in Socket Code	The send-related kernel socket code was missing an initialization. This has been fixed.
Vnode for Console Missing	Performing the <code>TIOCCONS ioctl</code> on the controller device of a pty could panic the kernel.
<code>exec</code> Failures May Not Clean Up Properly	In some cases, <code>exec</code> failures did not clean up properly. In particular, the path-name buffer allocated for the initial lookup was not freed, and the vnode for the file was not released.
Interrupt Priorities on Sun-4s	On Sun-4s, the system could hang due to problems with interrupt priorities.
Tapemaster Driver	The Tapemaster driver caused a system panic when <code>tar</code> or <code>dump</code> was used.
Calls Improperly Nested	The exit routines at the end of the runtime code used by <code>boot</code> to save the caller's stack environment did not nest calls.
<code>st</code> Driver	The <code>st</code> driver did not behave as documented in <code>st(4)</code> . When <code>read()</code> sees a tape mark, it does not proceed to the next tape file.
Drivers Test Returns from <code>kmem_alloc</code>	Because <code>kmem_alloc</code> no longer panics if it can't return memory, drivers now must test returns from <code>kmem_alloc</code> .
<code>sysdiag</code> Test Hangs:	The <code>tapetop 1/4"</code> tape test in <code>sysdiag</code> hung at start-up.
<code>uucico</code>	A system panic sometimes occurred due to a data fault at location <code>stropen+dc</code> when running the user program <code>uucico</code> .
NIT STREAMS Buffering Module	The NIT STREAMS buffering module, <code>nit_bug(4m)</code> , failed to collect messages into chunks that arrived on its read queue. Instead, each message was passed to its upstream neighbor upon receipt.
Multiple Contiguous Writes Hung Bus	Multiple contiguous writes to two SCSI disks on the same controller could hang the bus.

Null mblok Dereferenced	The <code>wscn</code> driver could dereference a null <code>mblk</code> .
Reselect Interrupt Recovery	For certain configurations, a message did not indicate that a hardware problem involving a lost reselect interrupt had been recovered in software.
VPC-2200 Device Driver	The device driver for the VPC-2200 did not work correctly.
Read and Write System Calls	Read and write system calls did not always work on a Sun-4 when the buffer was in the VME space (usually obtained via an <code>mmap ()</code> call).
System Crashed when Exiting Mailtool	The system could crash when exiting Mailtool.
mknod on Customized Kernel Caused Panic	Executing <code>mknod</code> on a customized kernel could cause a <code>panic: bus error: if size routines in the xy and sd drivers defered pointers to structures for which memory had not been allocated at configuration time.</code>
Inverted Test in makenfsnode	In <code>makenfsnode ()</code> , one of the tests that determines whether to reuse a cached mode or to create a new one was inverted. Under some conditions, this caused the kernel heap to grow without bound.
Adaptec ACB-4000 Controllers	If you formatted an <code>st-506</code> disk controlled by an Adaptec ACB-4000 controller, the operation failed with a <code>mode select error</code> .
File Position Reported Incorrectly	The <code>st</code> driver did not report the correct file position if <code>fsf</code> failed when skipping files.
Login Ports	If <code>/dev/console</code> and <code>/dev/ttya</code> were both enabled as login ports, a panic crash occurred.
kadb Hung When Booted on 4/110	The kernel debugger, <code>kadb</code> , hung when booted on a Sun-4/110.
asrtfail System Panic	The Sun-2, Sun-3, and Sun-4 kernels were changed to correct the system panel and <code>panic: assertion failed: message</code> . Because <code>uprintf</code> also caused this message when exiting <code>sysdiag</code> , the kernel now checks for a stream pointer before attempting to print to the controlling terminal of a process.

Watchdog Reset with Bus Error on 4/110

On 4/110 systems running 4.0, bus errors that normally kill applications caused a watchdog reset instead. The following code, compiled with '-g' and executed on a 4/110 running 4.0, causes the system to die with a watchdog reset:

```
main()
{
    double *v ;
    int i ;
    i = 1382236159 ;

    v = malloc(120*sizeof(double));
    printf ("%d,%g0, i,v[i]);
}
```

Guards Added to Header Files in /usr/include

Guards have been added to the header files in /usr/include, with these exceptions:

```
/usr/include/f77
/usr/include/pascal
/usr/include/pixrect
/usr/include/suntool
/usr/include/sunview
/usr/include/sunwindow
```

Kernel Prints Fault Information Carefully, Avoids Watchdog Reset

The kernel did not always dump meaningful information as it died from a bus error. To aid developers, `showregs()` is reworked to correct information about a crash.

Interrupt Loss from 7053 Under Heavy I/O Caused System to Hang

On the Sun-3 and Sun-4, spurious interrupts or invalid interrupt vectors hung the system and stopped I/O to the 7053 disk. The only recovery was to reboot the system. A watchdog mechanism was added to correct this bug.

Kernel Crashed when Syncloop Ran on Asynchronous MCP Port

The kernel driver code was changed so that open routines return an error if the `mcp.ops` pointer is already set to another protocol.

ipc

A kernel crash occurred with STREAMS NIT and an IPC board. There may have been corruption of stream buffers and control information when STREAMS NIT was used. This was observed only on the IPC memory Ethernet driver. The bug is fixed.

- On NFS, Vnode Unrecognized as `umount` Directory
- The `umount` system call follows the link and overshoots the mount point. The automounter compensates for this by responding as a directory rather than as a symbolic link. The vnode type is assigned at mount time and assumed not to change.
- A code change now convinces the kernel that this mount point is a directory.
- `kbdriver`
- Kernels configured with some non-Sun keyboard drivers did not work with `kadb` even when a Sun keyboard was used. An `(LI-A)` abort that should have dropped into `kadb` instead dropped into the PROM monitor, bypassing `kadb`. This is fixed.
- `lgkernel`
- Kernels larger than one megabyte did not work. Kernels worked when built smaller than `0xfc000`, while the same sources built on a kernel larger than `0xfc000` did not.
- Kernel Restarts Instruction that Caused Segmentation Fault
- The kernel did not restart the instruction that caused the segmentation violation (SIGSEGV) when `rlimit.rlim_cur` was upped after a stack overflow. Now the processor state is saved when signals occur so that returning from a signal handler causes the processor to continue correctly.
- `p0br`, `p0lr` References Removed from Kernel
- The VAX-compatible terms `p0br` and `p0lr` have been replaced in the kernel with nonVAX-dependent code. This reduces confusion when porting the system.
- `/movc.s:bcopy()` Transferred Byte by Byte
- Improperly aligned `movc.s:bcopy()` transfers were very slow. Code was inserted to move data by longwords before the loop to move by bytes, causing a 3x improvement of longword over byte copying.
- `ftruncate` on `/dev/zero` Changed Major and Minor Device Numbers
- Calling `ftruncate` on a file descriptor corresponding to `/dev/zero` changed the major and minor device numbers so the device eventually became unusable. `itrunc()` now verifies the type of inode before beginning an operation that might be senseless.
- `mb`
- `mbugo` called `xxgo` with an invalid `md->md_mbinfo`. After `DVMA` ran out, `mbrelse` was eventually called to release `DVMA`. This caused a call to `mbugo` to map `DVMA`'s for devices that were waiting while `DVMA` had run out, sometimes resulting in `mbugo` calling `xxgo` with an invalid `md->md_mbinfo`.

mmpeekio	Extra break statements in <code>mmpeekio</code> in <code>mem.c</code> prevented the transfer of more than one short or long per call to <code>mmpeekio</code> . This has been fixed.
UDP	The kernel variable <code>udpcksum</code> was set for UDP packets and turned on the checksum. The checksum did not, however, take effect on NFS, since NFS uses its own path through the network code when sending packets. This is fixed with a change to the NFS packet-sending module.
skyrC	The kernel panicked on a Sun-2 with a Sky floating-point processor board. This bug was fixed earlier, and the new <code>skyrC</code> file is included in this release.
Lost Files and <code>vm_hat</code>	On 128-megabyte Sun-4 systems, the <code>pmem</code> test did not produce one pass in 48 hours. Also, in some circumstances, an NFS server could lose a write to a file if the file was written on a SunOS 4.0 server while being accessed for an NFS read request. The written data would be part of the file for a while, but the same file would have old or spurious data if the data had to be reread from the disk. This is fixed by changes to the <code>vm_hat</code> files.
yppush	A 3/260 yp master server crashed with the message <code>panic: bus error</code> , every time an attempt was made to push a yellow-pages map. An odd set of circumstances in the <code>tcp/ip/ethernet</code> layers of code tickled a bug in <code>ether_pullup()</code> . The bug is fixed.
HZ Exists in <code><sys/param.h></code> for Older System Compatibility	The <code>HZ</code> parameter returns ticks per second according to syscalls that return values in ticks. <code>getrusage()</code> does not return values in ticks. It returns <code>struct timeval</code> , parameters that are independent of the clock resolution.
<code>sizecheck</code> Now Owned by Builder	The makefile attempted to change the file-access permissions of <code>sizecheck</code> during standalone driver build. The make aborted unless you owned the file. All users now have execute permission when <code>sizecheck.sh</code> retrieves from <code>sccs</code> .
<code>lint</code> Did Not Work for Standalone Drivers	The makefiles have been changed so you can run <code>lint</code> on standalone drivers.

14.5. SCSI Drivers Bugs Fixed

The list of bug fixes in SCSI drivers includes 1/4" tape drivers and onboard SCSI and SCSI-3 drivers.

Length Errors on SCSI Tape

Length errors are generated when the blocking factor during a read does not match the blocking factor used when recording. These are no longer fatal errors, and tape operation can continue.

`physio` notices the requested transfer count does not match the actual count and issues another I/O request. This takes care of the problem.

`biowait` Hang with SCSI 1/4" Tape

The following command would hang after writing some files to tape because `biowait` on the `physio` buffer was still marked BUSY.

```
find . -print | cpio -ocB >/dev/rst8
```

This was a general SCSI problem. The code has been changed, and this bug is now fixed.

SCSI-3 Driver Module `si_cmd()` Did Not Catch Bus Error

This module can return a value of `OK`, `FAIL`, `SCSI_FAIL` or `HARD_FAIL`. When called by `sidone()` and `sigo()` of `si.c`, the error checking did not detect the possible value of `SCSI_FAIL`; in the interrupt service routine `siinter`, error checking did not check `SCSI_FAIL` and `HARD_FAIL` after `si_cmd` was called.

In `si_cmd`, `HARD_FAIL` is now returned and this fixes the bug.

No ID on Arbitration of SCSI and SCSI-3 Driver

The driver would enter arbitration phase without asserting proper ID on the SCSI data bus, leaving the contents of the Output Data Register undetermined and with a leftover of the previous SCSI transaction. SCSI driver code was changed, and this bug is now fixed.

Missing `#include` Statement

The following line was missing from `/sc_conf.c` for Sun-4, preventing a build on some machines.

```
include sw.h
```

Two lines of code were added to fix this bug.

SCSI Tape End-of-File Flag Not Cleared

If end-of-file occurred during a read, the EOF flag did not clear when the driver closed.

SCSI Tape Driver: <code>fsf</code> Errors	<p>The driver did not report the location of file-space errors. If such an error occurred at beginning-of-tape, an auto-format search was sometimes triggered.</p> <p>The driver did not report the position of the file when a file-space error occurred. If the error was a blank-check error and occurred at beginning-of-tape, an auto-density check was sometimes triggered.</p>
Sysgen Tape Controllers: File Positioning	<p><code>fsr</code> and <code>bsr</code> file-positioning operations for <code>ioctl</code> functions now work with Sysgen tape controllers.</p>
Tape File Positioning Errors in <code>st.c</code>	<p>End-of-file and end-of-tape error reporting is the same as that of <code>tar</code> and <code>dump</code>: errors are not posted, and only a residual count is returned. Such a report is not seen as an error by the <code>ioctl</code> file-positioning operations <code>fsr</code>, <code>bsr</code>, <code>fsf</code>, and <code>bsr</code>. The file was changed so that EOF/EOT error reports now flag a failure of the requested operation.</p>
Misleading Messages when Cartridge Not in Drive	<p>When the ¼" tape drive is accessed and there is no cartridge in the drive, the SCSI tape driver now prints a sensible and informative screen message.</p>
Extra File Marks on SCSI Tape after <code>writi.sp</code>	<p>If, after a tape write, the SCSI tape driver is closed, then opened and closed again without writing any data, the driver writes an extra file mark.</p>
SCSI Error Recovery Generated Bus Error in <code>scdone</code>	<p>A path through SCSI (<code>sc.c</code>) error handling resulted in a bus error panic (kernel trap, invalid reference) from the <code>scdone()</code> routine. Changes to a critical code section in the <code>sc</code> driver have fixed this bug.</p>
tapetop Test Would Hang with SCSI Tape and SMD Disk	<p>The <code>tapetop</code> ¼" test in <code>sysdiag</code> would hang at startup, showing neither pass nor cycle during 48 hours of run-in with an SMD disk. Rebooting produced the message <code>Some processes wouldn't die</code>. This was determined to be a SCSI driver problem, and is fixed.</p>
SCSI Driver Refused Send-Data's Odd-Length Transfer	<p>The <code>sc.c</code> SCSI drive did not handle the odd transfer length condition for the case of a send-data command. Research found a coding error specifying receive-data instead of send-data in the odd-length routine. The code is changed to fix this bug.</p>

sw0_cmd: Reselect
failure on SunOS 4.0 with
SCSI Patch

The reselect failure message was inaccurate, and was changed to indicate that a lost hardware reselect interrupt was recovered in software. Hardware involved was a SCSI 155-megabyte disk with a 150-megabyte tape on a Sun-4/110 running bootprom 2.8 on Sundiag 1.0beta2.

Bus Failure Prevented Recovery
of Disk I/O for Swap

A bad Archive embedded SCSI tape controller allowed the SCSI bus to fail, and disk I/O for swap was not recovered. The bug was found after a screen message, `Panic: segu_swapin: getpage failed, showed a failed command after missing command phase.`

SCSI Failed when
Dumping Core

Error messages about `sd0:`, `sw0:`, `st0` indicated that `sddump` routine and `sdtimer` failed to reset the bus to clear out all other SCSI I/O before doing SCSI I/O; an **LI-A** abort was required to recover. This is fixed.

14.6. Serial Drivers Bugs Fixed

Serial driver bugs in the following list are now fixed.

Redirecting Console
to Serial Device

A serial driver redirecting the console to a serial device no longer fails with error number 25.

A few lines of kernel code now equate *console device* with *device*; this ensures that the TIOCCONS call is recognized by all serial devices, not just pseudoterminal `pty`.

Software Carrier Detect Flag
Turned Off by MCP

The software Carrier Detect flag was turned off by the MCP driver. This bug is now fixed.

DTR Asserted on Modem Line
while Port Is Not Open

Data Terminal Ready was asserted while the communications port was not open, allowing the modem to answer with no getty waiting on the line. This bug is now fixed.

DTR Dropped Too Soon after
Serial Port Close

The modem did not recognize a less-than-one-second drop of DTR in cases when the port was reopened immediately. Increasing DTR drop time to at least one second fixed this bug.

Modem Status Unreadable

It was impossible to read the modem status flags from an ALM or ALM-2 port.

Flow-Control Problems with ALM Board	Sun-3 users experienced intermittent flow-control problems with printers operating from an ALM board. (The problems occurred when printing graphics and when running the serial port in LLITOUT.) All known ALM flow-control problems have now been fixed.
4.0 Crashed on Reboot If ALM Board Was Missing	When SunOS 4.0 was configured for the ALM board and the board was not installed, a reboot caused the system to crash on a Sun-2, Sun-3, or Sun-4. This has been fixed so that a screen message advises that the board is missing.
ALM-2 Acted Up when Switching Between Raw and Cooked Mode	When using application programs such as <code>vi</code> with an ALM-2, the <code>mcp_async</code> driver was not waiting for data to be properly flushed before changing the mode information for the MCP.
Multiple ALM-2 Boards Did Not Work Reliably	In a system with multiple ALM-2s, only board 0 worked reliably; units 1, 2, and 3 locked up, failed to respond, or responded improperly. A one-line code change in the driver corrected attempts to reset all interrupts as though they were generated on board 0 and fixed this bug.
UUCP Unworkable at Transmissions Greater Than 9600 Baud	The tty driver input to the read queue did not drain fast enough when input was thick and fast; when the read queue filled, all input was discarded. The driver was rewritten to increase queueing, buffering, and drain efficiency. The bug is now fixed.
Timeout Table Overflow by MTI Driver	When characters came too fast, the 4.0 MTI driver queued multiple timeouts and overflowed the timeout table. The bug was fixed by rewriting the driver to note that one timeout is already queued and to avoid queueing additional ones.
pty Driver Could Hang on Writes to Master Side	Large writes to the master side of a pseudoterminal (<code>pty</code>) can be read-queued to the upstream slave-side <code>pty</code> driver. If that queue is full, the master side blocks until it empties, but it sleeps on the wrong event and, if the queue empties again, wakeup never occurs. This is usually masked by a write to the slave side where echoed data causes wakeups. This bug is fixed.
Intermittent Flow Control Problems on ALM with Plotter	The port would eventually hang, and when XON was sent there was no response. The flow control machine was not stripping the parity bit. The new general <code>tty</code> patch tape corrects this bug.

ALM-2 Driver Was Dereferencing a Null Pointer

The ALM-2 driver appeared to be dereferencing a null pointer. A code change in the driver fixed this bug.

lpr Panicked System Using ALM-2 Parallel Port

The `lpr` command caused the system to panic if the printer was using an ALM-2 parallel port.

Rewriting code in the MCP drivers has fixed this bug.

Serial I/O Did Not Work Reliably

Printing stopped and output was truncated after printing two or three pages. Terminals ignored flow control. Toggling printer off, then on, performed a manual restart.

There were intermittent flow-control problems, with an ALM board on a Sun-3 running 4.0 and earlier, while printing graphics. Another user reported truncated terminal output when using a plotter in 4.0.

A new flow control module has been written and integrated into all the serial drivers. The bug is now fixed.

14.7. Network Bugs Fixed

The network bugs were variously found in the network file system, protocols, and other areas.

Security Hole in `yppasswd`

A security hole in `yppasswd` has been fixed.

PC-NFS Search Paths

The search `PATH` of a PC-NFS client could, by mistake, include an NFS-mounted file that was not a directory. An attempt to read this "directory" could crash the server.

Sun-4 Gateways Hung

Sun-4 gateways sometimes hung, requiring a gateway reboot.

Spurious VME Interrupts

Spurious VME interrupts occurred at processor level 5.

Internet Name Server Crashes

Random crashes occurred in the Internet name server `/usr/etc/in.named`.

Checksum Not Occurring

The `udpcksum` variable did not cause NFS to perform a checksum.

Broadcast Address Not Reset	The <code>ifconfig</code> command in <code>/etc/rc.local</code> did not reset the broadcast address; it set the netmask only.
Limited Maximum Segment Size	TCP (Internet Transmission Control Protocol) limited MSS (Maximum Segment Size) to half the window, degrading performance on networks with large package sizes.
Break-out Routine Called in Error	In the routine <code>ether_output</code> , the code called the break-out routine, setting the type variable to the protocol family entry from the registration field.
passwd and yppasswd Did Not Reencrypt Correctly	There was confusion with <code>rpc</code> when <code>passwd</code> encrypted the <i>secret</i> parameter of <code>chkey.c</code> and <code>yppasswd</code> decrypted it. Code changes have cleared the confusion and ensured secure <code>rpc</code> .
makedbm Returned Inappropriate Message	makedbm: Yikes! indicated that the source file was inappropriately formatted when doing a <code>make-netgroup</code> from <code>/var/yp</code> . The message has been changed to be more informative.
ftp Bug	When there was an anonymous <code>ftp</code> account, a command to change the working directory to <code>~root</code> would overwrite the <i>struct passwd</i> returned by <code>getpwnam</code> at the initial login. Code has been changed to copy <i>struct</i> into another variable. This closes the hole that allowed root access to the system, and fixes the <code>ftp</code> bug.
nfs Destroyed Files	When a file was checked out using <code>query.c</code> , edited, and compiled, and then a <code>scs diff</code> made in preparation for checking it back in, the <code>SCCS</code> file was destroyed.
ld error -98 with make_doc from gnuemacs	Creating a <code>make_doc</code> with <code>gnuemacs</code> release 18.51 brought an <code>ld error -98</code> message when the directory was mounted from a 4.3 BSD system onto a Sun-4/110 running SunOS 4.0. A code change has fixed this bug.
NFS-Based open Did Not Generate Network Status	Occasionally the NFS data cache for a file was not flushed, even when data was incorrect; and a network stat did not necessarily take place after an <code>open</code> . Instead, attributes sometimes came out of the local cache. This has been fixed by <code>nfs_open</code> forcing an over-the-wire <code>getattr</code> and by using the attributes to check caches.

- Shared Locks Did Not Turn Off NFS Caching** A process opening a shared lock on a remote file could not always read the most current contents of the file, suggesting that NFS caching was not turned off. Code changes have fixed this bug.
- NFS Data Caches Not Always Flushed** Sometimes the NFS data cache was not flushed, even though the data was incorrect; this generally occurred in `vi`. One line of code was deleted, disallowing the invalidation of the cached attributes, and fixing this bug.
- NFSD Swapped Out** Under heavy NFS usage, NFS daemons were swapped out and could not be killed except by a reboot. Some systems had many `nfstd` running. New code has been written to take care of this bug.
- `rpc.lockd` Exclusive locks always succeeded for local files, but a lock request from a remote client failed.
`rpc.lockd` dumped core after several hours of execution.
The lock manager under 4.0 sometimes failed with the message, `Can't talk to local statd`, causing processes using the lock manager to enter 'D' state and stay there. Code changes to `rpc.lockd` have fixed these bugs.
- `rpc.mountd` The mount daemon did not allow a host to mount a filesystem listed in an `rw=` if it was not also listed in `access=`.
- Bad IP Checksums on Some Packets** In transmission protocols the IP checksum was calculated by an algorithm that made it possible to generate several packets in sequence with bad checksums; this could cause an NFS operation on a soft-mounted file system to fail. A brief code change has fixed this bug.
- Broadcast Requests Looped Back to Originating Machine** Ethernet drivers were looping back broadcast requests through the loopback driver with a different netmask; this led to a machine incorrectly answering its own `ICMP_NETMASK` broadcast. A change of code has fixed this bug in this release.
- `ypserv` with Interdomain Resolver Hangs `mount` If `ypserv` was using the interdomain resolver, the yellow pages library did a bogus match on an empty string, and `ypserv` attempted to resolve the empty string.
Yellow pages library code is changed to ensure that `ypmatch` and `ypserv` do not attempt to operate on an empty string.

- `yplib` Under server loads greater than 8, yellow-pages operations sometimes failed when `yplib` was unable to write the binding file in time and, in effect, became one of the top users of CPU cycles.
The code has been reordered to update the file in a more rational way.
- `yppasswd` It was discovered that the `rpc.yppasswd` program could be breached in SunOS 3.x or SunOS 4.0 if a program was written where `yppasswd` was used in place of `rpc.yppasswd`; this allowed a line entry to be added to the password file.
The code has been changed to fix this bug.
- `rsh` and `rlogin` Failed Due to Order of `hosts.equiv` Entries Remote operations failed when the order of the entries in `/etc/hosts.equiv` did not match the order of entries in `/etc/hosts`, and long host names with dots were not correctly recognized. Code changes have fixed this bug.
- `panic: dirremove` During PC-NFS Writes to NFS-Mounted Filesystems Writing to an NFS-mounted filesystem through PC-NFS sometimes caused a check in `ufs/ufs_dir.c` to generate a `panic: dirremove` condition. This apparently occurred when an application on the PC attempted to remove files in the remote filesystem. (This should only occur when attempting to remove a directory with a null, or zero-length, name.)
- 14.8. Compiler Bugs Fixed** Bug fixes associated with the compiler library and linker are listed in the next sections.
- No Recursive Check for Shared Libraries The link editor `ld` did not recursively check for shared library dependencies.
- `ld` Error Message While linking an executable comprised of two objects (one constructed as a “-r” of several objects) and the System V shared C library, the link editor `ld` returned an error message.
- `ld`: Library Name and Common Name Collided When using `ld`, if a collision occurred between a library name and a common name, all references were relocated to the library name. This bug has been fixed.
- `ld` Option Could Not Be Turned Off The `ld` option, `-assert definition`, could not be turned off. A `-noassert` option has been added to `ld` and the compiler driver updated.

- Nonspecific Exit Codes in `ld.so` Caused Clumsy Recoveries**
The single exit status returned by `ld.so`, 127(10), made for gross recoveries from temporary failures—`mail` aborted instead of being requeued. The linker code is changed to return more specific exit codes, conforming to the standards in `syssexits.h`, and this bug is fixed.
- `ld.so` Operated Incorrectly in Dynamic Binding to a Common**
If a dynamic binding to a symbol that was actually a common occurred, `ld.so` would crash or perform the binding incorrectly. Code has been changed that resets the object identity in the `ld.so` lookup cache and records the link map entry for `main`. This bug is fixed.
- Nonstandard Link Command Generated `adb` and `rm` Failures**
The link command, `ld -N -T 0 -o filenames_filenames`, caused both `adb` and `rm` to fail. The command would create a bad symbol table in some cases. This bug is fixed.
- `ld` with `-A`, `-x` Options Produced Bad Output File**
Work around this by omitting the `-x` option; this produces an executable with a readable symbol table. The executable contains the symbol `_DYNAMIC`, which may also be an error.
- Watch Points Not Working Correctly**
The following data watch points did not work properly: `stop <var>`, `stop IF`, and `when`.
- Bit Fields Displayed Incorrectly**
`dbx` did not display the bit fields of a C structure correctly.
- Compile with `-Bstatic` Option Failed on 4.0 EXPORT**
`ld` produced an obscure error message when linking an executable comprised of two objects and the SystemV shared C library. It was found that no options were passed to `ld` in `Makefile.inter`. The lack of certain options implied that a shared object was being built. These bogus shared objects were included in `libc.a`, producing the error message
`_etext: /lib/libc.a(des_crypt.o): multiply defined`
This bug is fixed.
- `newkey` Failed under 4.0 EXPORT**
The command, `newkey -u root`, followed by a password, resulted in the message `ld.so: call to undefined procedure _cbc_crypt from 0x20196 when the file cbc_crypt in libc.so was missing`. Changes were made to Makefiles in `/usr/src/bin`, `/usr/src/usr.bin`, `/usr/src/usr.etc/yp`, and `/usr/src/Makefile.inter`, and recompiled to fix this bug.

Older Sun-2 Object Files
Failed to Link

Attempting to link archived object files with machine type 0 (`M_OLD SUN2`) produced the message, `ld: <file>: wrong machine type`. This was fixed by expanding the check for `M_68010` to allow an `M_OLD SUN2` object file.

Dynamically Linked Programs
Require Page-Aligned Stack
Limit to Execute

`ld.so` obtains the stack-size limit when building a heap for itself at the stack extreme limit. If the stack-size limit (in bytes) is not page-aligned when passed to the `mmap()` system call, `mmap()` fails the mapping request with an error message similar to this: `ld.so: map heap error (22) for /dev/zero`.

`ld.so` now rounds up the value of the stack-size resource limit to the next whole page before using it.

`a.out` Died without Error
Message

If the `__DYNAMIC` data structure of a program using shared libraries was corrupted, the program died without any error messages from the run-time loader. `ld.so` now checks the version number in `__DYNAMIC` of the `a.out` that calls it.

`ld` Error Message when
Linking Two-Object Executable
and Sys V Shared Library

This happened when `ld` option `-e` (entry point) was placed on the command line after the name of the file to be processed. `ld` allocated more space than needed for the string area. This bug is fixed.

`opreg`

When a C function call returns a structure and includes arguments of type float, the arguments are converted to double; however, SunOS 4.0 and Sys4-3.2 compilers did not convert them. This is fixed by changes to the `/usr/lib/ccom` file, but requires renaming (and saving) the old `lib/ccom` to keep it out of mischief.

`target`

`target` in `a.out.h` and `setjmp.h` conflicted with compiler builds; *Pascal* 1.1 could not build on a 4.0 machine. The files have been changed to fix this bug.

Sun-4 `/bin/as` Dumped Core
with `/.stabs -01 -S`

The Sun-4 assembly/optimizer could dump core if `.stabs` and `.stabn` were present, optimization was enabled (`-01`), and disassembly was enabled (`-S`). This bug is fixed.

14.9. Debugger Bugs Fixed

Both `dbx` and `dbxtool` required changes to fix these problems:

`dbxtool`

`dbxtool` was putting corrupted filenames in the top window.

`dbx`

`dbx` was not dealing with object files with multiple `N_SO` entries, not knowing that `.for` and `.vf` suffixes imply *fortran*, nor correctly handling source files with the same prefix, such as `a.for` and `.f`.

On Sun-4, a *Fortran* `do` loop caused `dbx` to print spurious values; `dbx` was not correctly displaying variables assigned to SPARC 1 registers.

`next` and `step` could not step over floating-point branches on a Sun-4 running 4.0; `step over fp` failed.

`dbx` did not always apply USE-path when searching for source; as an example of the failing, in this program both diagnostics and the failure to set a breakpoint are errors:

```
mkdir a
mkdir b
cd a
cat > a/c << EOF
main ()
{
}
EOF
cc -g a.c
mv a.c ../b
dbx a.out
use ../b
file a.c
>> warning: File 'a.c' has not been comp[iled with the -g option
stop at 3
>> file "../b/a.c" was not compiled with the -g option
status
```

`dbx` was unable to read a core file on a Sun-4 with shared libraries; running `dbx` on an executable that dumped core gave this message: `warning: object file read error: text address not found`

`dbx` could not deal with object files with multiple `n_so` entries; it did not know that `.for` and `.vf` implied Fortran, and never correctly handled multiple source files with different prefixes. Compiler code change has fixed this bug.

The `dbx step` command took longer to execute on a Sun-4 than on a Sun-3. This is no longer the case.

14.10. Utilities Bugs Fixed

System utilities bug fixes are described here.

- `syslogd` `syslogd` failed to define `loghost` name, causing a `syslog` race condition.
- `catman` The utility `/usr/etc/catman -p` failed with an error message.
- `lpr` The usage `lpr -r -s` could conceivably delete files on a remote host and produced an error message such as the following in the printer log file:
`/usr/lib/lpd: imagen: 776 6: open failure <errno = 2>`
- `graph(1G)` When attempting a multiple operation pipe — `spline|graph|plot`— on Sun-4, `graph` went into an infinite loop; `graph` also gave incorrect answers on Sun-4. This is fixed.
- Security Holes** There were security holes in `sendmail` and in the `in.fingerd` daemon.
- `tar` Executing `tar` with the `xcf` options (instead of `xvf`) as in the following example erased the floppy.
`tar xcf /dev/rfd0a`
- `login(1)` A password entry with a negative user ID and no password defaulted to root; `login` has been changed to fix this.
- `passwd` Created Password
Entry if `/etc/passwd`
Contained Blank Lines Library routines `getpwent`, `getpwnam`, `getpwuid` just skipped over and did not do anything about blank lines in `/etc/passwd`. The code has been changed so blank lines are no longer mapped to an empty password, and existing empty password entries are stripped from the file.
- `uuxqt` Attempts to execute a command with `uuxqt` on Sys4-3.2 failed with an `XQT DENIED` message. A 'bad command' debugging error message was issued after checking the 'L-cmds' and `USERFILE` files. This has been fixed.
- `lpd` Dumped Core when
Filename Exceeded 132
Characters `lpd` used a fixed-size buffer for banner lines; when the line was too long, it dumped core. This is changed to dynamically allocate space for banner lines.

- `lpd` Passed `-w0` Default when
`lpr` Width Unspecified The width parameter was set to 0 unless the user specified a different value using the `-w` option of `lpr`. The default has been changed.
- Chesstool Froze The Chesstool would freeze when it was time for the computer to make a move.
- Segmentation Violation
Occurred In chesstool Commands to Start game, Move any piece, Choose "Machine White" or "Human White", Choose either "Cancel" or "Restart" in any combination caused a segmentation violation. It was found that the panel-notify procedures in `chesstool.c` were declared with the wrong argument list, and an argument was dereferenced in `machine_proc`.
The code has been corrected.
- `/usr/kvm/m68k` on Sun-3
Running 4.0 Is Incorrect During a build on a Sun-3 (but not on Sun-4), the file `/usr/kvm/m68k` incorrectly pointed to `false` and propagated the evaluation of `m68k` as false. The link was changed from `/bin/false` to `/bin/true` to correct the problem in 4.0.1 and higher.
- Printer Log File Error Messages
Not Shown `stderr` for the line printer daemon filters other than the output filter was directed to the error file instead of the logfile. This is fixed so all messages from filters are placed in the log file and mailed to the user.
- `lpr -s` Rejected Files on
Remote Filesystems `stat()` returned a device number that was entered as negative in the control file. `lpr` did not recognize negative numbers and rejected the file. Code was changed to enter the device number in unsigned form, and the bug is fixed.
- `lpd` Mailed Remote-Print Error
Report to Illegal Address The local printer daemon did not read user and host names, so when an error occurred, it sent the error report to '@'. Code was changed to have `lpd` read the user and host names from the control file.

14.11. System Administration Bugs Fixed

- `format` There were a number of bugs pertaining to `format` which have been fixed. A few others of miscellaneous character are also corrected.
- When `format` dumped the defect list to a file, the list began with defect number 0. When printed, however, the list began with defect 1. This has been fixed so that defect numbering always starts at 1.
- Disks formatted on Xylogics 450 and 7053 controllers were not compatible.
- `format` aborted if it tried to spare a mapped sector. The program now ignores bad sectors if they are already mapped.

In a spare sector, the `extract` command missed a defect if it was the only one on the track.

`format` did not provide a failure message when the `label` command of the utility failed. The code of `label` has been modified to warn the user of a labeling failure.

An SMD disk drive formatted under 4.0 was ruined if subsequently formatted under 3.5 or Sys4-3.2. This bug has been fixed with the rewriting of `format`.

Defective tracks on the disk drive were not brought to the attention of the formatter. The track-defective bit was not checked when original defect information was read from the disk. In the rewritten `format`, the bit is checked and, if set, an error is returned. This ensures that defective tracks are not used.

`format`, when run with a 451 controller, did not understand a long last sector following the `data` and, when a defect was found in such a sector, the `format` failed. The rewritten `format` now understands drives with long as well as short last sectors.

`format` displayed misleading and unnecessary warning messages when irrelevant data was destroyed in the formatting process. Rewriting `format` has fixed this bug.

`format` Corrupted Kernel Map and Dumped Core with Incorrect Disk Type Selection

When `format` did not find an incorrectly selected disk type in `format.dat`, it dereferenced a null pointer when setting up the default partition table. `format` is rewritten and this bug is now fixed.

`format` Aborts if Defect Spans into Next Also-Mapped-Bad Sector

`format` could not properly handle a bad sector when the defect entry spanned the next sector which was also bad. Program code is changed so the defect entry does not span into the next sector.

Defect List Difficult to Recreate

If forced to re-enter a defect list from hardcopy, you could not recreate the list with an editor (such as `vi`) or an optical scanner and then load the list with the `defect/load` command. (The operation failed with `bad magic number` or `corrupt defect file`.) Instead, you were forced to use a more time-consuming method (the `defect/add` command).

The `format` program now allows either method. You can load any defect list as long as the first line is of the form:

```
0x1 <number of defects> 0x123
```

Long Hostnames Caused Boot to Fail

If you had a server with a `hostname` longer than 16 characters, booting diskless clients failed.

- Sundiag Test Could Fail** The Sundiag `iptext` could fail, telling that the time limit for the test had been exceeded, because the test-start message was not received.
- Security Holes** There were security holes in `passwd`, `getpwent`, `getpwnam`, and `getpwuid`.
- Remote shutdown Failed** When the console received a `root rlogin` from a remote machine followed by a `shutdown -h now` command, the system would hang after displaying `System going down ...` or the `Login: prompt`, and did not do a disk sync. The halt was aborted when `ttyname()` returned a null; and null was returned because `shutdown` kills `rlogin` before it executes `halt`. This bug is fixed by having `halt` use "(no tty)" in the audit record when `ttyname()` returns `NULL`.
- Shutdown Truncates the /etc/nologin Warning Message** The first letter of the shutdown message in `/etc/nologin` was cut off. This has been fixed.
- mount Returned Incorrect Line Number for Illegal Entry in fstab** `mount` obtained the line count from valid entries returned from the `getmntent()` routine which ignores commented-out lines. An added routine now allows `mount` to return the correct line number in the error message, even though one or more lines are commented out.
- rpc.mountd Rejected Request for Exporting Filesystem to everybody** When `rpc.mountd` could not map a caller's address to a machine name, it rejected the request. `rpc.mountd` is rewritten with a bogus host entry structure when the client address cannot be found with regular methods to fix this bug.
- SCSI Defect List not Dynamically Allocated by format** `format` formerly allocated a static amount of memory for SCSI defect lists. If defects exceeded 127, a bus error was issued. Rewriting `format` has fixed this bug.
- Bus Error Resulted When Logical Defect Was Added to Null Defect List** When creating a null defect list and doing an `add` command in the defect menu by selecting the logical defect option, the first defect caused a segmentation violation. This is fixed with the rewriting of `format`.
- System Accounting Bug Fixed** The system accounting utility, `/usr/etc/sa`, could not accommodate a user ID greater than or equal to 65526. When such an ID was encountered, `sa` would loop indefinitely and, at the same time, another `sa` would be restarted daily by `cron`. This quickly degraded system performance. This bug is fixed.

`scsi.h` File not included in 4.0 SCSI/ALM Patch Tape The SCSI/ALM patch tape did not include `scsi.h` and was not compatible with Sun consulting specials. This was not a bug, and the oversight is corrected.

14.12. SunView Bugs Fixed

There were two fixed bugs for SunView.

`sunview` Exited Leaving Video Reversed When `sunview` exited, the screen was left in reverse video and text and prompts in normal video. A change in `suntools.c` now saves `planegroup` `colormaps` when starting, restores them when exiting, and checks the 0th red entry for CG8; this fixes the screen reverse video bug.

Shared Libraries Had Incorrect Version Number Warnings Shared libraries, `libsuntool` and `libsunwindow`, acquired different minor version numbers for Sun-2 and Sun-3; this brought incorrect version number warnings when a binary built on a Sun-2 was run on a Sun-3. Code changes in the shared library files fixed this bug.

`perfmeter` The `perfmeter` occasionally dumped core on startup. This was caused by a divide-by-zero bug, and is now fixed.

14.13. SunInstall Bugs Fixed

Default `ip` Address Illegal The default `ip` address given by `suninstall`, `192.9.200.0`, is a broadcast address and not a legal host address.

Root Partition Could be Designated Free Space Hog `suninstall` allowed you to designate the root partition as the free space hog.

Free Space Hog Not Used Soon Enough If you selected too much software for the size of your `/usr` partition, `suninstall` would wait until the partition was totally full (110%) before starting to steal from the free space hog.

`syslog.conf` Caused Loop The `syslog.conf` file that `suninstall` creates for a standalone machine caused `syslogd` to loop by indefinitely logging an error. This file has been changed to fix this problem.

The `setup_client` Script The `setup_client` script did not always determine whether `yp` was running. The script now performs a more reliable test.

**/export/swap Partition Not
Used Properly**

When installing clients on a server with a /export file system (including a /export/swap partition), suninstall failed if the combined size of the client swapspaces exceeded the size of /export.

**Setting Terminal Type to 'other'
Dumps Core**

On the suninstall form to select terminal type, selecting 4 (other) caused a core dump. This happened because the buffer used to parse the termcap file was too small. The buffer has been increased to the correct size.

Screen Forms Changed

A number of changes were made to suninstall screen forms:

- Partition [a] no longer appears as an option for the Freehog Disk Partition in the Disk Form.
- The name of the Tape Information field in the Software Form has been changed to Media Information.
- Pressing **CTRL-N** in the Software Selection form could dump core. This happened if the tape was rewinding or if the table of contents was being read from the tape.

Notes

Notes

Notes

Notes

Notes

Notes

Systems for Open Computing™

Corporate Headquarters
Sun Microsystems, Inc.
2550 Garcia Avenue
Mountain View, CA 94043
415 960-1300
TLX 37-29639

**For U.S. Sales Office
locations, call:**
800 821-4643
In CA: 800 821-4642

European Headquarters
Sun Microsystems Europe, Inc.
Bagshot Manor, Green Lane
England
0276 51440
TLX 859017

Australia: (02) 413 2666
Canada: 416 477-6745
France: (1) 40 94 80 00

Germany: (089) 95094-0
Hong Kong: 852 5-8651688
Italy: (39) 6056337
Japan: (03) 221-7021
Korea: 2-7802255
Nordic Countries: + 46 (0)8 7647810
PRC: 1-8315568
Singapore: 224 3388
Spain: (1) 2532003
Switzerland: (1) 8289555
The Netherlands: 02155 24888

Taiwan: 2-7213257
UK: 0276 62111

**Europe, Middle East, and Africa,
call European Headquarters:**
0276 51440

**Elsewhere in the world,
call Corporate Headquarters:**
415 960-1300
Intercontinental Sales

