

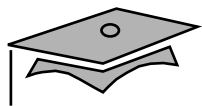


Sun Educational Services

Sun Systems Fault Analysis Workshop

ST-350





Copyright 1999 Sun Microsystems, Inc., 901 San Antonio Road, Palo Alto, California 94303, U.S.A. All rights reserved.

This product or document is protected by copyright and distributed under licenses restricting its use, copying, distribution, and decompilation. No part of this product or document may be reproduced in any form by any means without prior written authorization of Sun and its licensors, if any.

Third-party software, including font technology, is copyrighted and licensed from Sun suppliers.

Parts of the product may be derived from Berkeley BSD systems, licensed from the University of California. UNIX is a registered trademark in the U.S. and other countries, exclusively licensed through X/Open Company, Ltd.

Sun, Sun Microsystems, the Sun Logo, Solaris, OpenBoot, Ultra, SunSolve, SunVTS, AnswerBook, NFS, SyMON, UltraSPARC, Ultra Enterprise, SunDiag, and SPARCstorage are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. in the U.S. and other countries. Products bearing SPARC trademarks are based upon an architecture developed by Sun Microsystems, Inc.

The OPEN LOOK and 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 who implement OPEN LOOK GUIs and otherwise comply with Sun's written license agreements.

U.S. Government approval required when exporting the product.

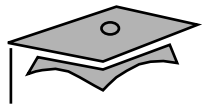
RESTRICTED RIGHTS: Use, duplication, or disclosure by the U.S. Government is subject to restrictions of FAR 52.227-14(g) (2)(6/87) and FAR 52.227-19(6/87), or DFAR 252.227-7015 (b)(6/95) and DFAR 227.7202-3(a).

DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS, AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID.



Preface

About This Course

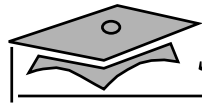


Course Goal

Learn a systematic fault analysis technique to troubleshoot intermediate and advanced Solaris™ system faults.

ST350 is intended for

- System administrators
- Support engineers
- Programmers



Course Map

General Methods, Tools, and Files

Fault Analysis
and Diagnosis

Diagnostic
Tools

POST
Diagnostics

OpenBoot PROM

OBP Diagnostics
and Commands

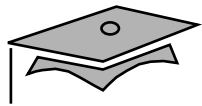
Software Packages and Services

SunSolve
Database
Information

SunVTS
System
Diagnostics

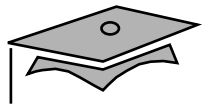
Crash Dump Analysis

Kernel Core
Dump Analysis



Module-by-Module Overview

- Module 1 – Fault Analysis and Diagnosis
- Module 2 – Diagnostic Tools
- Module 3 – POST Diagnostics
- Module 4 – OBP Diagnostics and Commands
- Module 5 – SunSolve Database Information
- Module 6 – SunVTS System Diagnostics
- Module 7 – Kernel Core Dump Analysis



Course Objectives

- Define a comprehensive fault analysis method
- Solve problems using the fault analysis method
- Differentiate hardware and software problems
- Use a comprehensive set of Solaris tools to analyze and solve problems
- Interpret POST diagnostic output
- Perform remote diagnostics using the TIP interface



Course Objectives

- Maintain and troubleshoot system problems using OBP commands
- Look up technical information in the SunSolve™ database
- Analyze hardware problems using SunVTS™ system diagnostics
- Perform an initial crash dump analysis
- Analyze hung systems

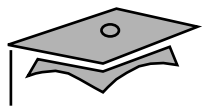


Skills Gained by Module

Meaning of:

- Black boxes
- Gray boxes

Skills Gained	Module			
	1	2	3	4
Skill or Objective 1				
Skill or Objective 2				
Skill or Objective 3				
Skill or Objective 4				



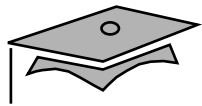
Guidelines for Module Pacing

Module	Day 1	Day 2	Day 3	Day 4	Day 5
About This Course	A.M.				
Module 1 – Fault Analysis and Diagnosis	A.M.				
Module 2 – Diagnostic Tools	A.M./ P.M.				
Selected Workshops	P.M.				
Module 3 – POST Diagnostics		A.M.			
Module 4 – OBP Diagnostics and Commands		A.M.			
Selected Workshops		P.M.			
Module 5 – SunSolve Database Information			A.M.		
Module 6 – SunVTS System Diagnostics			A.M.		
Selected Workshops			P.M.		
Module 7 – Kernel Core Dump Analysis				A.M./ P.M.	
Selected Workshops				P.M.	A.M./ P.M.



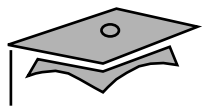
Topics Not Covered

- Basic system administration
 - Software installation
 - User account configuration
 - Printer management
 - Basic security policies



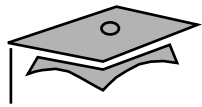
Topics Not Covered

- Advanced system administration topics
 - Device naming conventions
 - NFSTM, cacheefs and automounter administration
 - NIS/NIS+ administration
 - Disk and file system configuration
 - Network configuration
 - The SAF utility



How Prepared Are You?

- Have you completed
 - SA-235: *Solaris 2.x System Administration I* or SA-237: *Solaris 7 System Administration I*
 - SA-286: *Solaris 2.x System Administration II* or SA-287: *Solaris 7 System Administration II*
 - SA-271: *Solaris 1.x to 2.x System Administration*
- or
- Do you have six months of field system administration or system maintenance experience in Sun environments?



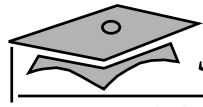
Introductions

- Name
- Company affiliation
- Title, function, and job responsibility
- Troubleshooting experience
- Fault analysis and diagnosis experience
- Application tools experience
- Reasons for enrolling in this course
- Expectations for this course



How to Use Course Materials

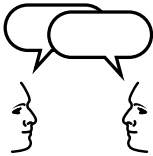
- Objectives
- Relevance
- Overhead image
- Lecture
- Exercise
- Check your progress
- Think beyond



How to Use the Icons



Additional resources



Discussion



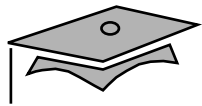
Exercise objective



Caution



Warning



Typographical Conventions and Symbols

- `Courier` is used for the names of command, files, and directories, as well as on-screen computer output.
- **`Courier bold`** is used for characters and numbers that you type.
- *`Courier italic`* is used for variables and command-line placeholders that are replaced with a real name or value.
- *Palatino italics* is used for book titles, new words or terms, or words that are emphasized.



Attention VARs

- Retain your signed course certificate.
- Schedule yourself to take the appropriate Drake test by contacting:

John Shedaker
Sun Microsystems
2550 Garcia Ave., MS UMIL06-01
Mountain View, CA 94043

Fax Number (408)-945-9476
Phone Number (408)-276-1315



Module 1

Fault Analysis and Diagnosis



Objectives

- Use a total system approach for fault analysis
- Write accurate problem statements
- Describe system problems
- Identify commonly available resources
- Generate and test likely causes on a per fault basis
- Communicate and document information gathered during fault analysis
- Use fault analysis worksheets to document facts



The Fault Analysis Method

- Fault analysis – Identify the problem and organize fact gathering and comparisons
- Diagnosis – Organize the actual discovery, testing, repair, and reporting of the problem

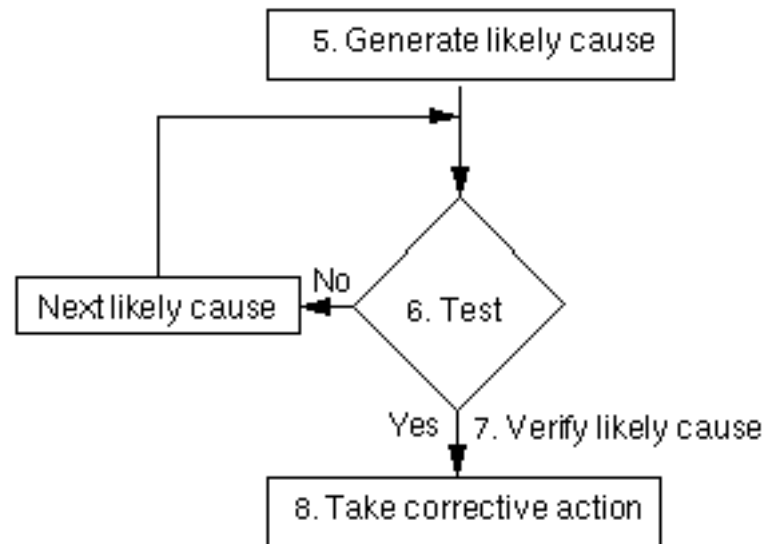


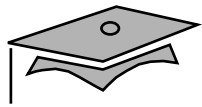
Eight Steps of Fault Analysis and Diagnosis

1. State the problem.
2. Describe the problem.
3. Identify differences.
4. List relevant changes.
5. Generate likely causes.
6. Test likely causes.
7. Verify the most likely cause.
8. Take action to correct the fault.



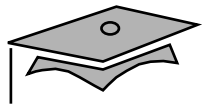
Eight Steps of Fault Analysis and Diagnosis





Stating the Problem

- Writing a problem statement
 - Determine which object, device, or subsystem exhibits the problem
 - Hypothesize what is wrong
- Guidelines for a problem statement
 - Identify the exact object with the exact defect
 - Be certain that the cause is not already known
 - Limit the problem statement to a single object and a single defect



Describing the Problem

- List all observed facts
- Establish comparative facts
- Identify the unique elements of the problem
- List any explicit error messages displayed
- Describe conditions and symptoms
- Identify any timely and related changes on the system



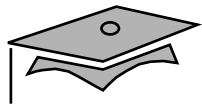
Listing All Observed Facts

- Who observed the problem?
- What is the problem?
- Where is the problem observed?
- What is the magnitude or size of problem?
- Are there any other questions?



Listing All Observed Facts

- Customer complaints
- Customer interviews
- Interviews of others involved
- Diagnostics
- Dumps
- Other



Establishing Comparative Facts

- Do other systems with similar OS revision, architecture, configuration have this problem?
- What parts of system are functional?
- When was the problem first observed?
- Was the problem observed at the same time on more than one system?
- What events occurred in the environment that may have contributed to the problem?
- Are the systems used similar enough to compare?



Identifying Differences

- Focus on one set of comparisons at a time
- List unique differences
- State the facts, not opinions
- Analyze comparisons for contrasts
- Analyze comparisons for what is the same
- Keep records



Listing Relevant Changes

- Examine differences.
- Describe each change and the date of its occurrence.
- Determine if each change represents something new or unusual about a difference.
- Determine if the problem occurred during a certain period of time.



Generating Likely Causes

- Stating your hypothesis

How could the *fault analysis element* have caused this problem?

- Example
 - Problem – Slow system response
 - Relevant change – Administrator added four users on Friday
 - Hypothesis – How could adding four users on Friday have caused slow system response?



Testing Likely Causes

- Closely examine doubtful answers.
- Investigate assumed facts.
- Eliminate likely causes that cannot be the cause.



Verification

- Factual and logical
- Realistic
- Result-oriented



Verifying the Most Likely Cause

Use a method that is

- Least disruptive
- Least expensive
- Least time-consuming
- Most conclusive



Taking Corrective Action

1. Complete the repair.
2. Test and verify the repair.
3. Document results.
4. Obtain confirmation and acceptance.



Fault Analysis Worksheet Example

Analysis Phase

Initial Customer Description

Problem Statement

Resources

Problem Description

Error Messages	Symptoms and Conditions	Relevant Changes	Comparative Facts



Fault Analysis Worksheet Example

Diagnostic Phase

Test and Verification

Likely Cause(s)	Test(s)	Result(s)	Verification(s)

Corrective Action

Final Repair	Communication	Documentation



Exercise: Fault Analysis and Diagnosis

- Exercise objective
- Tasks
- Preparation
- Exercise summary



Check Your Progress

- Use an organized total system approach for fault analysis and diagnosis
- Write accurate problem statements
- Describe a system problem in terms of error messages, symptoms, relative comparisons, and technical conditions
- Identify and use commonly available resources to solve technical problems
- Generate a list of likely causes on a per fault basis
- Design and implement tests to validate likely causes
- Communicate and document information gathered in fault analysis
- Use the Fault Analysis Worksheet to gather and document facts



Think Beyond

- Can you describe a way to improve the current fault analysis procedure used in your company environment?
- What on-line resources are available at your company site?



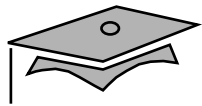
Module 2

Diagnostic Tools



Objectives

- Differentiate watchdog resets, panics, and system hangs
- Differentiate hardware and software problems
- Provide examples of fatal and non-fatal error conditions
- Identify a set of Solaris commands and utilities which are useful in fault analysis
- Identify Solaris system files which contain information that is useful in fault analysis



Objectives

- Describe the syntax, function, and relevance of each command or system file
- Use Solaris commands and files to determine system configuration and status information
- Solve workshop problems using Solaris utilities and system files



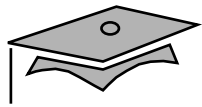
Introduction to Hardware and Software Errors

- Error categories – Software, hardware-corrected, recoverable, fatal, and critical
- Error reporting mechanisms – Bus errors, interrupts, and resets



Types of Errors

- Software errors
- Hardware-corrected errors
- Recoverable errors
- Critical errors
- Fatal errors



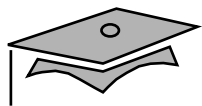
Error Reporting Mechanisms

- Bus errors
- Interrupts
- Watchdog resets
 - ▼ CPU watchdog reset
 - ▼ System watchdog reset



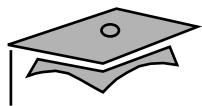
Differentiating Hardware and Software Faults

- Fault frequency
- History
- Stack trace routines
- Watchdog resets
- Log files
- Panics



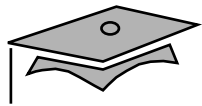
Diagnostic Commands and Tools

Command or Tool	Use
adb	Analyze dumps and a running system.
AnswerBook TM TM	Display online reference manuals in areas of hardware, user, system administration, and other
arch	Display architecture and kernel architecture information.
arp	Display the Address Resolution Protocol tables
aset	Use automated Security Enhancement Tool
catman -w	Create the /usr/share/man/windex database for use with index function available through the apropos command
crash	Analyze crash dumps
devlinks	Create symbolic links in /dev using information in /devices
df -k	Display disk space usage in Kbytes, including free space
dfmounts	Display remote file system mount information
dfshares	Display shared file system information
diff	Compare file contents
dmesg	Analyze recent log messages
disks	Create symbolic links in /dev/dsk and /dev/rdisk



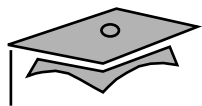
Diagnostic Commands and Tools

Command or Tool	Use
<code>drvconfig</code>	Configure the devices directory and the device information tree
<code>eeeprom</code>	Analyze and change programmable read-only memory (PROM) settings
<code>file</code>	Determine a file's type
<code>find</code>	Search for specific files in the file system structure
<code>format</code>	Analyze or modify disk partition information
<code>fsck</code>	Check UFS file systems for inconsistencies
<code>fsdb</code>	Use file system debugger (see <code>fsdb_ufs</code> in man pages)
<code>fstyp</code>	Display extensive file system parameters for a specified file system
<code>grep</code>	Analyze file contents, and search for specific patterns
<code>groups</code>	Display group definitions for a given user
<code>grpck</code>	Check the <code>/etc/group</code> file for syntax errors or inconsistencies
<code>ifconfig</code>	Analyze the status of network interfaces



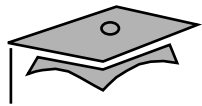
Diagnostic Commands and Tools

Command or Tool	Use
infocmp	Compare tic formatted files
iostat	Analyze I/O performance issues
kadb	Trap kernel and low-level faults
last	Display history of system login information
ls	Analyze file properties
ndd	Get and set named device driver parameters
netstat (-i, -r)	Analyze network tuning information, including active routes.
newfs	Create and examine file system parameters
nfsstat	Analyze NFS performance information
nm	Display symbol table information for (unstripped) executables
pagesize	Print the size of a memory page in bytes
perfmeter	Provide graphic display of performance metrics
ping	Contact network hosts by sending Internet Control Message Protocol (ICMP) request and reply datagrams
pkgchk	Check file integrity and accuracy of installation



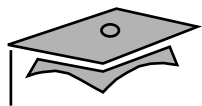
Diagnostic Commands and Tools

Command or Tool	Use
prtdiag (on sun4u and sun4d)	Display system configuration and diagnostic information
prtconf -v	Get system device information from Power-on Self Tests (POST) probe
ps	List properties of running processes (/bin/ps, /usr/ucb/ps)
pwck	Check the /etc/passwd file for errors and inconsistencies
route	Add, remove, and display kernel route table information
rpcinfo	Display information about Remote Procedure Call (RPC) services
A running system	Compare symptoms and results between functioning and faulting systems
sar	Analyze system performance information
shells	Use -x and -v options to provide debugging information
showrev -p	List currently installed patches
snoop	Display and analyze network traffic
strings	Search object and binary files for ASCII strings



Diagnostic Commands and Tools

Command or Tool	Use
SunSolve SM databases	List bug, patch, release, and general technical information for hardware and software
sysdef	Analyze device and software configuration information
swap	Add, delete, and monitor system swap areas
sum	Calculate and print a checksum value for a named file
SyMON TM	Use system monitor utility package with interfaces to the diagnostics package, SUNvts.
tail -f	Leave file open for reading and display what is there
tapes	Create logical links to device special files in /devices for tapes
tic	Terminfo compiler; translate a terminfo file from the source format into the compiled format
timex	List runtime and system activity information during command execution
traceroute	Show the routes followed by packets transferred in a subnetted environment
truss	Trace system calls issued and used by a program or command



Diagnostic Commands and Tools

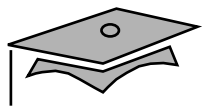
Command or Tool	Use
tunefs	Modify file system parameters that affect layout policies
/usr/proc/bin/*	Exercise the functions of the /proc file system
uname	Print platform, architecture, operating system, and system node information
vmstat	Analyze memory performance statistics
who am i	Display the effective current user name, terminal line and login time



Open Discussion

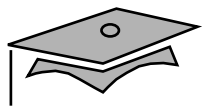
What other tools have you used that could be added to the list?

- _____
- _____
- _____
- _____
- _____
- _____
- _____



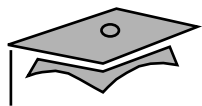
Diagnostic Files

File	Contents and Use
/etc/defaultdomain	Name the current domain; read and set it at each boot by the script /etc/init.d/inetinit
/etc/default/cron	Determine logging activity for the cron daemon through specification of the <i>cronlog</i> variable
/etc/default/login	Control root logins at the console through specification of the <i>console</i> variable, and other defaults
/etc/default/su	Determine /etc/hostname.le0 logging activity for the su command through specification of the <i>sulog</i> variable
/etc/dfs/dfstab	List which distributed file systems will be shared at boot time
/etc/dfs/sharetab	List currently shared NFS files and directories
/etc/hosts	Associate IP addresses to particular system names
/etc/hostname.le0 /etc/hostname.hme0	Assign a system name, and through cross-referencing the /etc/hosts file, and an IP address to a particular network interface
/etc/inetd.conf	List information for network services that can be invoked by the <i>inetd</i> daemon
/etc/minor_perm	Specify permissions to be assigned to device files



Diagnostic Files

File	Contents and Use
/etc/mnttab	List currently mounted file systems
/etc/name_to_major	List configured major device numbers
/etc/netconfig	Display the network configuration database read during network initialization and use
/etc/netmasks	Display the netmask value; read at boot time during network initialization
/etc/nsswitch.conf	List the database configuration file for the name service switch engine
/etc/path_to_inst	List the contents of the system device tree using the format of physical device names and instance numbers
/etc/protocols	List known protocols used in conjunction with Internet
/etc/rmtab	List current remotely mounted file systems
/etc/rpc	List of available RPC programs
/etc/services	List well-known networking services and associated port numbers; maintained by the Network Information Center (NIC)
/etc/vfstab	List local and remote file systems mounted at boot time



Diagnostic Files

File	Contents and Use
<code>/var/adm/messages</code>	List recent console window and/or boot messages
<code>/var/adm/sulog</code>	Display a record for each invocation of the su command
<code>/var/adm/utmpx</code>	List user and accounting information for the who and login commands
<code>/var/adm/wtmpx</code>	Maintain history of user information for the accounting package and report facility
<code>/var/lp/log</code>	List print services activity
<code>/var/sadm/install/contents</code>	List installed software packages
<code>/var/saf/_log</code>	List activity of the Service Access Facility (SAF)



Open Discussion

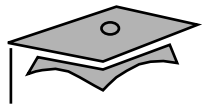
What other files have you used that could be added to the list?

- _____
- _____
- _____
- _____
- _____
- _____
- _____
- _____



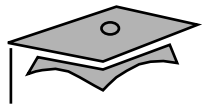
Applying the Tools

- Crash debuggers
- SunSolve databases
- OpenBoot™ PROM (OBP) commands



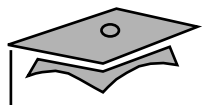
Device Management Utilities

- The `drvconfig` utility – Creates the device directory tree in `/devices`
- The `devlinks` utility – Creates links in `/dev` according to entries in the `/devices` directory tree.
- The `tapes` utility – Creates symbolic links in the `/dev/rmt` directory.
- The `disks` utility – Creates symbolic links for disks that are attached to the system.



Device Management Utilities

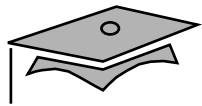
- The `ports` utility – Creates links in `/dev/term` and `/dev/cua` for the serial port entries in `/devices`.
- The `modinfo` utility – Displays information about loaded kernel modules.



Device Management Utilities

modinfo

Id	Loadaddr	Size	Info	Rev	Module Name
6	f59ddb30	2bc8	1	1	TS (time sharing sched class)
7	f59e06f8	4a4	-	1	TS_DPTBL (Time sharing dispatch table)
8	f5ab9000	235ac	2	1	ufs (filesystem for ufs)
9	f59e0ba0	dc4b	226	1	rpcmod (RPC syscall)
10	f59ee7f0	277ef	0	1	ip (IP Streams module)
13	f5a16c98	530	62	1	dma (Direct Memory Access driver)
14	f5a171c8	aeb	59	1	sbus (SBus nexus driver)
15	f5a17cb8	19a6	76	1	iommu (iommu nexus driver)
16	f5a19660	14e8	12	1	sad (Streams Administrative driver's)
17	f5a1ab48	538	2	1	pseudo (nexus driver for 'pseudo')
18	f5b1d000	f0e1	32	1	sd (SCSI Disk Driver 1.257)
19	f5a1b080	5f26	-	1	scsi (SCSI Bus Utility Routines)
20	f5a20fa8	ca68	61	1	esp (ESP SCSI HBA Driver 1.250)
25	f5a2da10	11fd6	5	1	procfs (filesystem for proc)
27	f594c000	bb40	8	1	sockfs (filesystem for sockfs)
30	f5a40058	15108	2	1	tcp (TCP Streams module)
31	f5a55160	3fb4	3	1	udp (UDP Streams module)
32	f5a59118	364d	4	1	icmp (ICMP Streams module)
33	f5a5c768	4ea3	5	1	arp (ARP Streams module)
34	f5a61610	390c	6	1	timod (transport interface str mod)
36	f5a64f20	819a	29	1	zs (Z8530 serial driver V4.116)
37	f5a6d0c0	76d	58	1	obio (obio nexus driver)
39	f5a6d830	1684	7	1	ms (streams module for mouse)
42	f5a73268	9b0	16	1	conskbd (Console kbd Multiplexer driver)



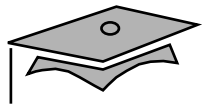
The modload Utility

- Loads a kernel module into a running system
- Example command:

```
# modload -p misc/obpsym
```

- In the /etc/system file:

```
forceload: misc/obpsym
```



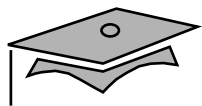
The modunload Utility

- Unloads a kernel module from a running system
- Example commands:

```
# modinfo | grep obpsym
```

```
89 f5a60d48 e1e - 1 obpsym (OBP symbol callbacks)
```

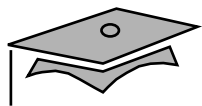
```
# modunload -i 89
```



The ifconfig Utility

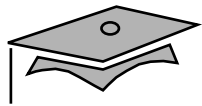
```
# ifconfig le0 inet 128.50.101.121 netmask 255.255.255.0  
-trailers up broadcast +
```

```
# ifconfig -a  
lo0: flags=849<UP,LOOPBACK,RUNNING,MULTICAST> mtu 8232 inet  
127.0.0.1 netmask ff000000  
le0: flags=863<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST>mtu  
1500 inet 129.151.28.5 netmask fffffff00 broadcast 129.151.28.25  
ether 8:0:20:7e:b9:cd
```



The `ifconfig` Utility

- FLAGS
- MTU
- INET
- NETMASK
- BROADCAST
- ETHER



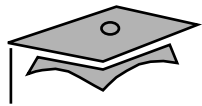
The netstat Command

- Returns the contents of network data structures and tables, including status of active sockets, interfaces, routing tables, and DHCP
- Examples
 - ▼ `netstat -i` – Lists statistics per interface
 - ▼ `netstat -r` – Lists routing table statistics



Debugging Utilities

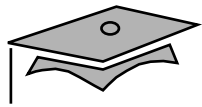
- The `truss` utility – Traces system calls, library calls, and signal activity for the program passed to it as an argument on the command line.



Error-Free truss Command Output

```
# truss ls
```

```
execve("/usr/bin/ls", 0xEFFFFFF510, 0xEFFFFFF518)
open("/dev/zero", O_RDONLY) = 3
mmap(0x00000000, 4096, PROT_READ|PROT_WRITE|PROT_EXEC, MAP_PRIVATE, 3, 0) = 0xEF7C0000
open("/usr/openwin/lib/libc.so.1", O_RDONLY)
open("/usr/lib/libc.so.1", O_RDONLY) = 4
fstat(4, 0xEFFFFFF024) = 0
mmap(0x00000000, 4096, PROT_READ|PROT_EXEC, MAP_PRIVATE, 4, 0) = 0xEF7B0000
mmap(0x00000000, 700416, PROT_READ|PROT_EXEC, MAP_PRIVATE, 4, 0) = 0xEF700000
munmap(0xEF793000, 61440) = 0
mmap(0xEF7A2000, 27680, PROT_READ|PROT_WRITE|PROT_EXEC, MAP_PRIVATE|MAP_FIXED, 4, 598016)
= 0xEF7A2000
mmap(0xEF7A9000, 5936, PROT_READ|PROT_WRITE|PROT_EXEC, MAP_PRIVATE|MAP_FIXED, 3, 0) = 0xEF7A9000
close(4) = 0
open("/usr/openwin/lib/libdl.so.1", O_RDONLY) Err#2 ENOENT
open("/usr/lib/libdl.so.1", O_RDONLY) = 4
fstat(4, 0xEFFFFFF024) = 0
mmap(0xEF7B0000, 4096, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_FIXED, 4, 0) = 0xEF7B0000
close(4) = 0
.....
```



Interpreting truss Command Output

General guidelines

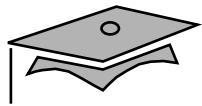
- `exec( )`
- `open( )`
- `mmap( )`
- `/dev/zero`



Interpreting truss Command Output

The truss ls example

- `ioctl()`
- `brk()`
- `lstat()`
- `getdents64()`
- `ioctl()` and `write()`



Example of truss Command Output With Errors

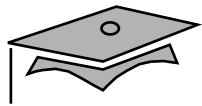
```
# truss ls rep
....
open("/dev/zero", O_RDONLY)           = 3
mmap(0x00000000, 4096, PROT_READ|PROT_WRITE|PROT_EXEC, MAP_PRIVATE, 3, 0) = 0xEF6C0000
munmap(0xEF6C0000, 4096)               = 0close(3)
= 0
munmap(0xEF6F0000, 4096)               = 0
time()                                = 909175996
ioctl(1, TCGETA, 0xEFFFF39C)          = 0
ioctl(1, TIOCGWINSZ, 0x000241B2)      = 0
brk(0x00026720)                        = 0
brk(0x0002E720)                        = 0
lstat64("rep", 0xEFFFF318)           Err#2 ENOENT
repwrite(2, " r e p", 3)               = 3
: write(2, " : ", 2)                   = 2
No such file or directorywrite(2, " N o   s u c h   f i l e".., 25)      = 25
write(2, "\n", 1)                       = 1
open("/dev/zero", O_RDONLY)           = 3
mmap(0x00000000, 4096, PROT_READ|PROT_WRITE|PROT_EXEC, MAP_PRIVATE, 3, 0) = 0xEF6F0000
llseek(0, 0, SEEK_CUR)                 = 20650
_exit(2)
....
```



Interpreting Errors in truss Command Output

General guidelines

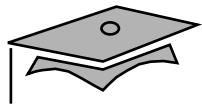
- Two main categories of errors:
 - ▼ System call error
 - ▼ Missing file errors
- Significant errors usually appear toward the end of the truss output



Interpreting Errors in truss Command Output

The truss ls example

- The output text includes the string `Err#2 ENOENT`.
- `ENOENT` indicates that a named file or directory does not exist.
- The final invocation of the `write()` system call outputs the string `No such file or directory`.
- The `ls` program terminates with `lstat()` failing since the named file, `rep`, does not exist and cannot be found.



The errno.h Header File

```
# cat /usr/include/sys/errno.h

/* Error codes */

#define EPERM1/* Not super-user*/
#define ENOENT2/* No such file or directory*/
#define ESRCH3/* No such process*/
#define EINTR4/* interrupted system call*/
#define EI05/* I/O error*/
#define ENXIO6/* No such device or address*/
#define E2BIG7/* Arg list too long*/
#define ENOEXEC8/* Exec format error*/
#define EBADF9/* Bad file number*/
#define ECHILD10/* No children*/
#define EAGAIN11/* Resource temporarily unavailable*/
#define ENOMEM12/* Not enough core*/
#define EACCES13/* Permission denied*/
.....
```



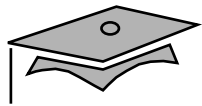
Exercise: Using Solaris Troubleshooting Tools

- Exercise objective
- Preparation
- Tasks
- Exercise summary



Check Your Progress

- Differentiate watchdog resets, panics, and system hangs
- Differentiate hardware and software problems
- Provide examples of fatal and non-fatal error conditions
- Identify a comprehensive set of Solaris commands and utilities which are useful in fault analysis
- Identify a comprehensive list of Solaris system files which contain information that is useful in fault analysis



Check Your Progress

- Describe the syntax, function, and relevance of each command or system file
- Use Solaris commands and files to determine system configuration and status information
- Solve workshop problems using Solaris utilities and system files



Think Beyond

- What on-line resources are at your disposal when seeking information on a system?
- What types of problems might you now be able to solve that you could not before?



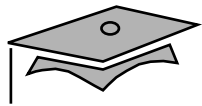
Module 3

POST Diagnostics



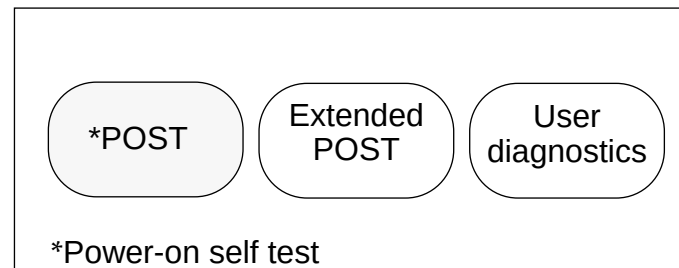
Objectives

- Describe capabilities and limitations of POST in identifying and resolving system faults
- Describe different ways to view the POST
- Configure the file `/etc/remote` on a console server to enable the use of `tip` in a remote diagnostic session
- View and interpret POST output
- Describe the functionality of the `prtdiag` command

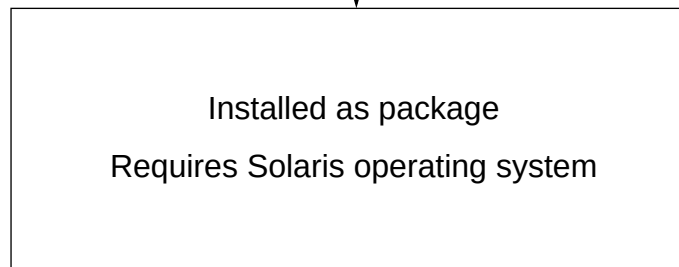


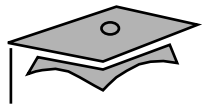
Diagnostic Overview

Boot PROM-based diagnostics



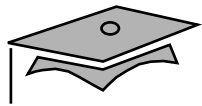
SunVTS diagnostics





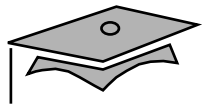
Boot PROM POST

- Is invoked automatically at the power-on sequence
- Differs slightly between workstation models
- Differs slightly between boot PROM revisions
- Conducts error detection and hardware verification for each system board
- Conducts all hardware bus probes, and saves information for the operating system's automatic reconfiguration



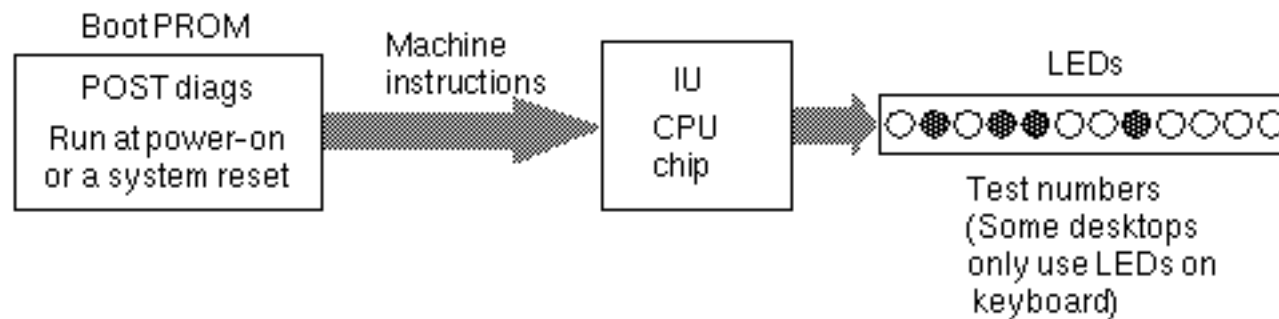
SunVTS Diagnostics

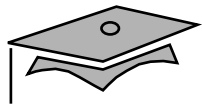
- Requires Solaris operating environment
- Is installed as a package
- Is used for system verification
- Runs in a window or non-windowed environment



POST Viewing Methods

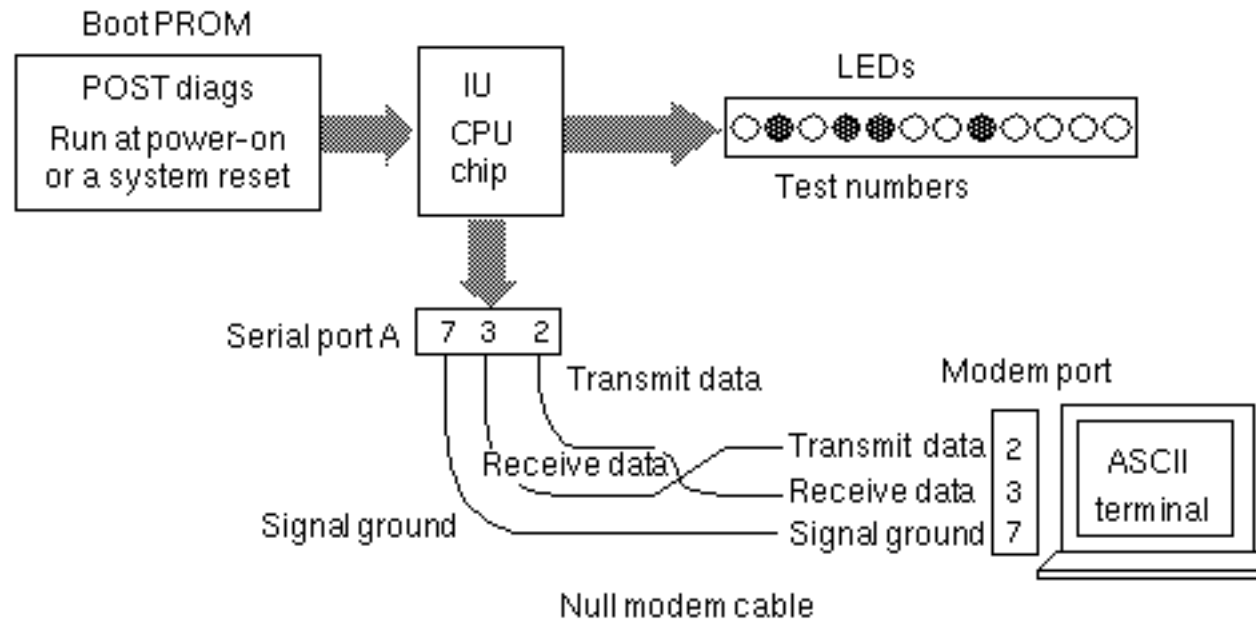
- Viewing POST from the CPU board LEDs





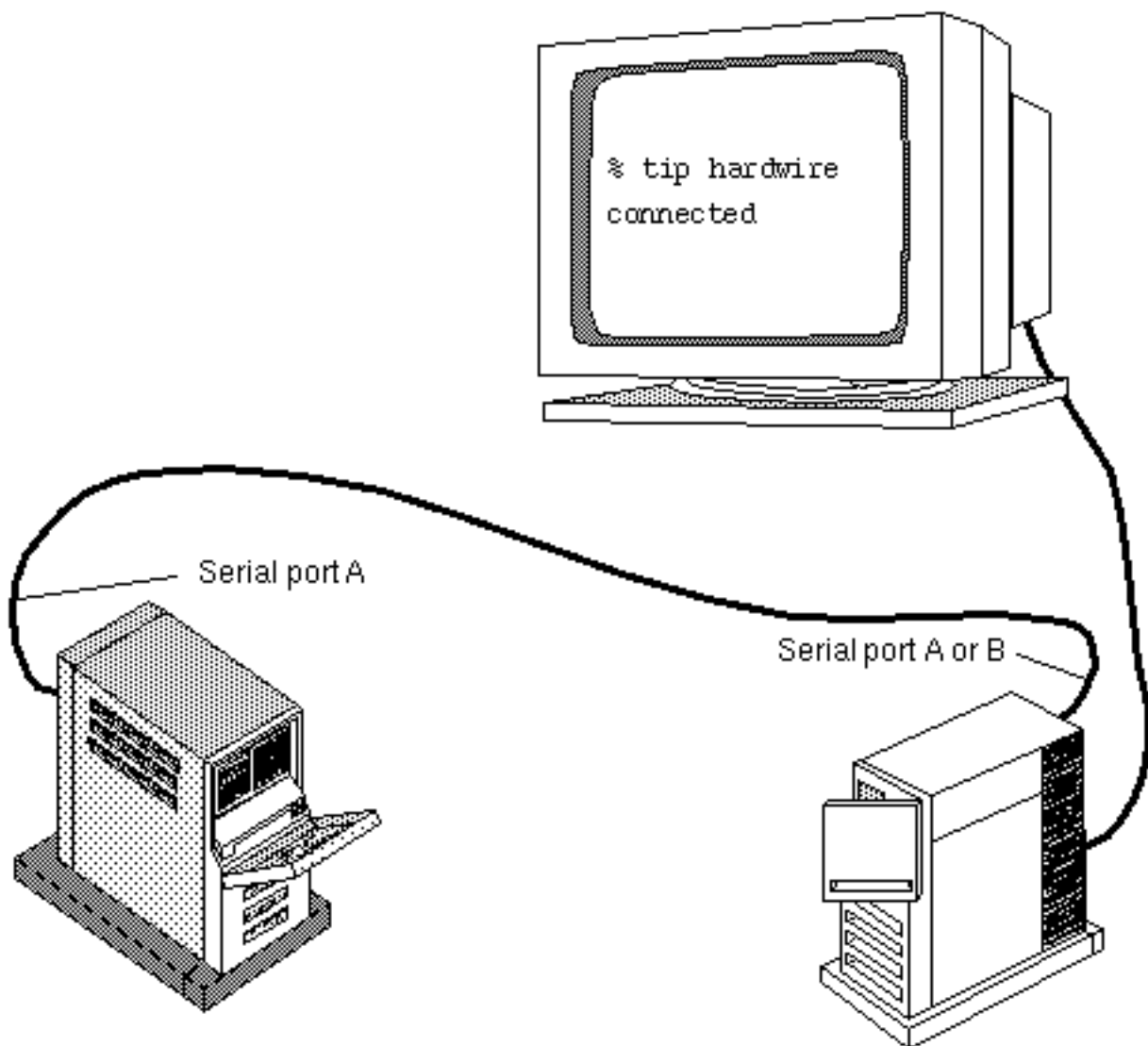
POST Viewing Methods

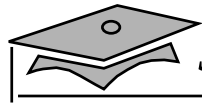
- Viewing POST with a serial port terminal



The ASCII terminal will list, in English text, the current executing POST diagnostic.

Viewing POST Using tip hardware

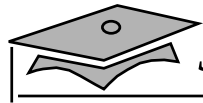




UltraSPARC Example

Using tip

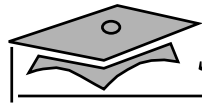
```
# tip hardware
connected
Hardware Power ON
@(#) Sun Ultra 5/10 UPA/PCI 3.11 Version 12 created 1998/05/19 11:30
Probing keyboard Done
%00 = 0000.0000.0000.4001
Executing Power On SelfTest
@(#) Sun Ultra 5/10 (Darwin) POST 2.2.9 (Build No. 502) 17:54 on 02/
19/98
CPU: UltraSPARC-LC (MHz: 271 Ecache Size: 256KB)
Init POST BSS
  Init System BSS
  NVRAM
  NVRAM Battery Detect Test
  NVRAM Scratch Addr Test
  NVRAM Scratch Data Test
DMMU TLB Tags
  DMMU TLB Tag Access Test
DMMU TLB RAM
  DMMU TLB RAM Access Test
Probe Ecache
  Probe Ecache
Ecache Tests
  Ecache RAM Addr Test
  Ecache Tag Addr Test
  Ecache RAM Test
  Ecache Tag Test
All CPU Basic Tests
  V9 Instruction Test
  CPU Tick and Tick Compare Reg Test
  CPU Soft Trap Test
  CPU Softint Reg and Int Test
All Basic MMU Tests
```



UltraSPARC Example

Using tip

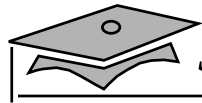
- DMMU Primary Context Reg Test
- DMMU Secondary Context Reg Test
- DMMU TSB Reg Test
- DMMU Tag Access Reg Test
- DMMU VA Watchpoint Reg Test
- DMMU PA Watchpoint Reg Test
- IMMU TSB Reg Test
- IMMU Tag Access Reg Test
- All Basic Cache Tests
 - Dcache RAM Test
 - Dcache Tag Test
 - Icache RAM Test
 - Icache Tag Test
 - Icache Next Test
 - Icache Predecode Test
- Sabre MCU Control & Status Regs Init and Tests
 - Init Sabre MCU Control & Status Regs
 - Initializing SC registers in SabreIO
- Memory Probe and Init
 - Probe Memory
 - bank 0: 0MB
 - INFO: 128MB Bank 2
 - INFO: MC0 = 0x00000000.80001484, MC1 = 0x00000000.06459acb
 - Ecache Access Test
 - Malloc Post Memory
 - Memory Addr with Ecache
 - Load Post In Memory
 - Run POST from MEM
- loaded POST in memory
 - Map PROM/STACK/NVRAM in DMMU
 - Update Master Stack/Frame Pointers
- All FPU Basic Tests
 - FPU Regs Test
 - FPU Move Regs Test
 - FPU State Reg Test



UltraSPARC Example

Using tip

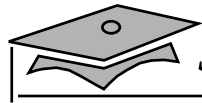
```
FPU Functional Test
FPU Trap Test
UPA Data Bus Line Test
Memory Tests
  Init Memory
    Memory Addr with Ecache Test
    ECC Memory Addr Test
    Block Memory Addr Test
    Block Memory Test
    Write 0x33333333.33333333
.....
Read .....
    ECC Blk Memory Test
Write 0xa5a5a5a5.a5a5a5a5 .....
Read .....
  All Basic Sabre MMU Tests
    Init Sabre
    PIO Decoder and BCT Test
    PCI Byte Enable Test
    Interrupt Map (short) Reg Test
    Interrupt Set/Clr Reg Test
    Sabre IOMMU Regs Test
    Sabre IOMMU RAM Address Test
    Sabre IOMMU CAM Address Test
    IOMMU TLB Compare Test
    IOMMU TLB Flush Test
    PBMA PCI Config Space Regs Test
    PBMA Control/Status Reg Test
    PBMA Diag Reg Test
    Sabre IO Regs Test
All Advanced CPU Tests
  DMMU Hit/Miss Test
  IMMU Hit/Miss Test
  DMMU Little Endian Test
  IU ASI Access Test
  FPU ASI Access Test
```



UltraSPARC Example

Using tip

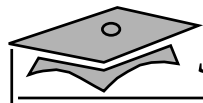
```
Ecache Thrash Test
All CPU Error Reporting Tests
CPU Data Access Trap Test
  CPU Addr Align Trap Test
  DMMU Access Priv Page Test
  DMMU Write Protected Page Test
All Advanced Sabre IOMMU Tests
  Init Sabre
  Consist DMA Rd, IOMMU miss Ebus Test
  Consist DMA Rd, IOMMU hit Ebus Test
  Consist DMA Wr, IOMMU miss Ebus Test
  Consist DMA Wr, IOMMU hit Ebus Test
  Pass-Thru DMA Rd, Ebus device Test
  Pass-Thru DMA Wr, Ebus device Test
  Consist DMA Rd, IOMMU LRU Lock Ebus Test
  Consist DMA Wr, IOMMU LRU Locked Ebus Test
All Basic Cheerio Tests
  Cheerio Ebus PCI Config Space Test
  Cheerio Ethernet PCI Config Space Test
  Cheerio Init
All Sabre IOMMU Error Reporting Tests
  Init Sabre
  PIO Read, Master Abort Test
  PIO Read, Target Abort Test
Status of this POST run:PASS
manufacturing mode=OFF
Time Stamp [hour:min:sec] 15:11:41 [month/date year] 12/06 1998
Power On Selftest Completed
Power ON
@(#) Sun Ultra 5/10 UPA/PCI 3.11 Version 12 created 1998/05/19 11:30
Clearing E$ Tags Done
Clearing I/D TLBs Done
Probing Memory Done
MEM BASE = 0000.0000.1000.0000
MEM SIZE = 0000.0000.0800.0000
```



UltraSPARC Example

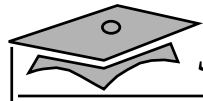
Using tip

```
11-Column Mode Enabled
MMUs ON
Copy Done
PC = 0000.01ff.f000.200c
PC = 0000.0000.0000.2050
Decompressing into Memory Done
Size = 0000.0000.0006.e3e0
ttya initialized
Reset Control: BXIR:0 BPOR:0 SXIR:0 SPOR:1 POR:0
UltraSPARC-IIi 2-2-2 module
Probing Memory Bank #0  0 +  0 :  0 Megabytes
Probing Memory Bank #2 64 + 64 : 128 Megabytes
Probing UPA Slot at 1e,0 Nothing There
Probing /pci@1f,0/pci@1,1 at Device 1 pci108e,1000 network
Probing /pci@1f,0/pci@1,1 at Device 2 SUNW,m64B
Probing /pci@1f,0/pci@1,1 at Device 3 ide disk cdrom
Probing /pci@1f,0/pci@1 at Device 1 scsi disk tape scsi disk tape
Probing /pci@1f,0/pci@1 at Device 2 Nothing there
Probing /pci@1f,0/pci@1 at Device 3 Nothing there
Probing /pci@1f,0/pci@1 at Device 4 Nothing there
Sun Ultra 5/10 UPA/PCI (UltraSPARC-IIi 270MHz), No Keyboard
OpenBoot 3.11, 128 MB memory installed, Serial #10416478.
Ethernet address 8:0:20:9e:f1:5e, Host ID: 809ef15e.
```



SPARCstation 5 POST Example Using tip

```
# tip hardware
connected
Power-ON Reset
MB86907 POST 2.2.3 03SEP96
Probing system memory: 32 32  0  0  0  0  0  0
Config = 8800000A
512Kb ecache detected
1.1.1 mem    ram        walking ones        Pass
1.1.2 mem    ram        address                Pass
1.1.3 mem    ram        post r/w region        Pass
1.1.4 mem    ram        obp r/w region        Pass
1.2.1 mem    control    parity                Pass
2.1.1 srmmu  regs       read/write          Pass
2.2.1 srmmu  ram        io-tlb              Pass
2.2.2 srmmu  ram        d-tlb              Pass
2.2.3 srmmu  ram        pdt cache          Pass
3.1.1 iommu  reg        read/write          Pass
3.1.2 iommu  reg        flush individual    Pass
3.1.3 iommu  reg        flush all           Pass
3.2.1 iommu  timeout    ebus                Pass
3.2.2 iommu  timeout    sbus                Pass
4.1.1 fpu    reg        regfile            Pass
4.1.2 fpu    reg        misalign            Pass
4.1.3 fpu    reg        single precision    Pass
4.1.4 fpu    reg        double precision     Pass
4.2.2 fpu    exceptions double precision     Pass
5.1.1 cache  dcache     ram                 Pass
5.1.2 cache  dcache     address             Pass
5.1.3 cache  dcache     tag                 Pass
5.1.4 cache  dcache     clear               Pass
5.2.1 cache  icache     ram                 Pass
5.2.2 cache  icache     address             Pass
5.2.3 cache  icache     tag                 Pass
5.2.4 cache  icache     clear               Pass
5.3.1 cache  ecache     ram                 Pass
```

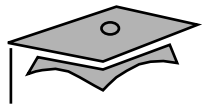


SPARCstation 5 POST Example Using tip

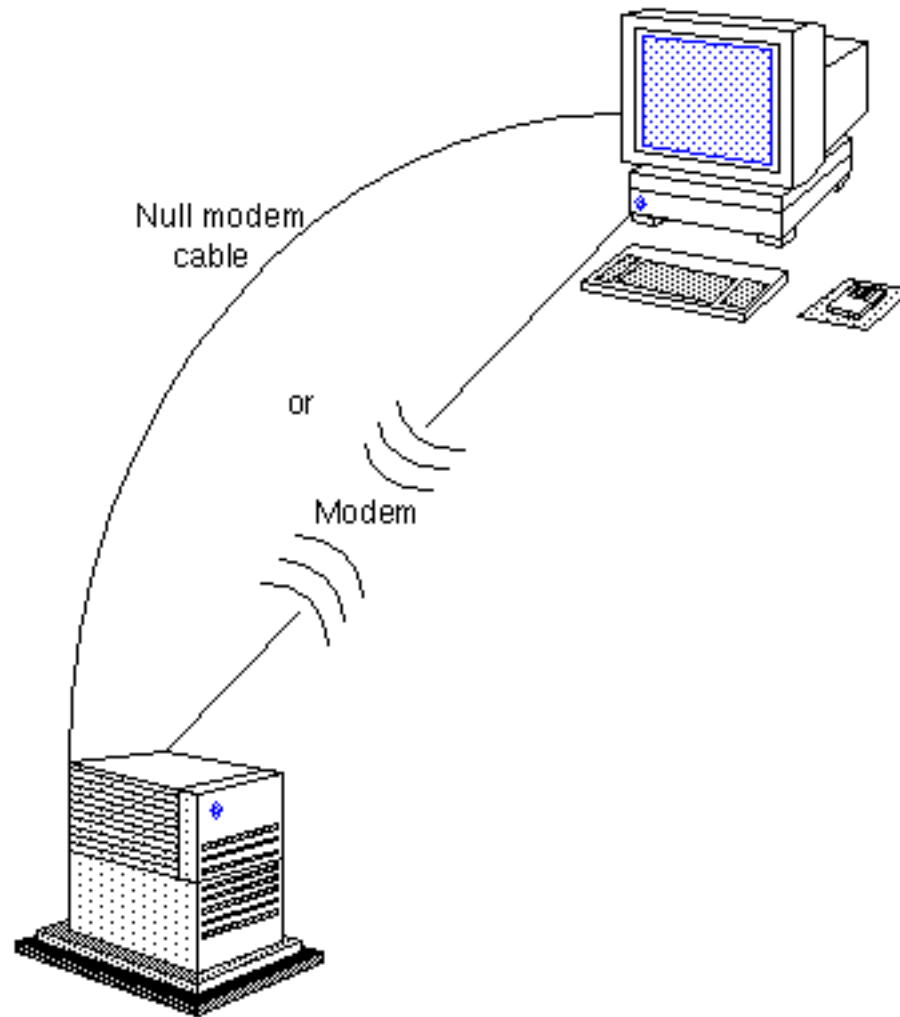
```

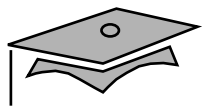
5.3.2  cache  ecache      address      Pass
5.3.3  cache  ecache      tag           Pass
5.4.1  cache  snoop       ram           Pass
6.1.1  i/o    counters    processor user timer Pass
6.1.2  i/o    counters    processor counter Pass
6.1.3  i/o    counters    system counter Pass
6.2.1  i/o    lance       getid        Pass
6.2.2  i/o    lance       csr          Pass
6.2.3  i/o    lance       rap          Pass
6.2.4  i/o    lance       rdp          Pass
6.3.1  i/o    esp         register r/w  Pass
6.3.2  i/o    esp         config reg   Pass
6.3.3  i/o    esp         fifo access  Pass
6.3.4  i/o    esp         command reg  Pass
6.4.1  i/o    pp          register access Pass
6.4.2  i/o    pp          io readback  Pass
6.4.3  i/o    pp          tcr readback Pass
6.5.1  i/o    tod         regs test    Pass
6.5.2  i/o    tod         nvram access Pass
7.1.1  intr   regs        sys          Pass
7.1.2  intr   regs        proc         Pass
7.2.1  intr   software    interrupts disabled Pass
7.2.2  intr   software    interrupts enabled Pass
7.2.3  intr   software    multi        Pass
7.3.1  intr   pp          Interrupts   Pass
7.4.1  intr   timer       system counter Pass
7.4.2  intr   timer       process counter Pass
8.1.1  dma    apc         bypass       Pass
initializing TLB
initializing cache
Allocating SRMMU Context Table
Setting SRMMU Context Register
Setting SRMMU Context Table Pointer Register
~.

```



Post Diagnostic Workshop Using tip





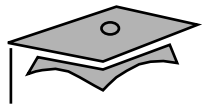
POST `tip` Commands

- To send a break through the `tip` window, type
`~#`
- To interrupt a test, press Control-c.
- To exit from `tip`, type
`~.`
or
`~^D` (tilde Control-d)
- To see a list of `tip` commands, type
`~?`



Exercise: Using tip to Observe POST Diagnostics

- Exercise objective
- Preparation
- Tasks
- Exercise summary



Check Your Progress

- Differentiate watchdog resets, panics, and system hangs
- Differentiate hardware and software problems
- Provide examples of fatal and non-fatal error conditions
- Describe the capabilities and limitations of the POSTs in identifying and resolving system faults
- Describe different ways to view the POST
- Configure the file `/etc/remote` on a console server to enable the use of `tip` in a remote diagnostic session
- View and interpret POST output
- Describe the function of the `prtdiag -v` command



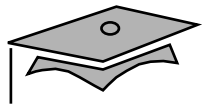
Think Beyond

- Are there hardware conditions that cannot be analyzed through POST diagnostics?
- Is it possible to run the remote diagnostic procedure presented in this module with an ASCII terminal instead of a console monitor?



Module 4

OBP Diagnostics and Commands

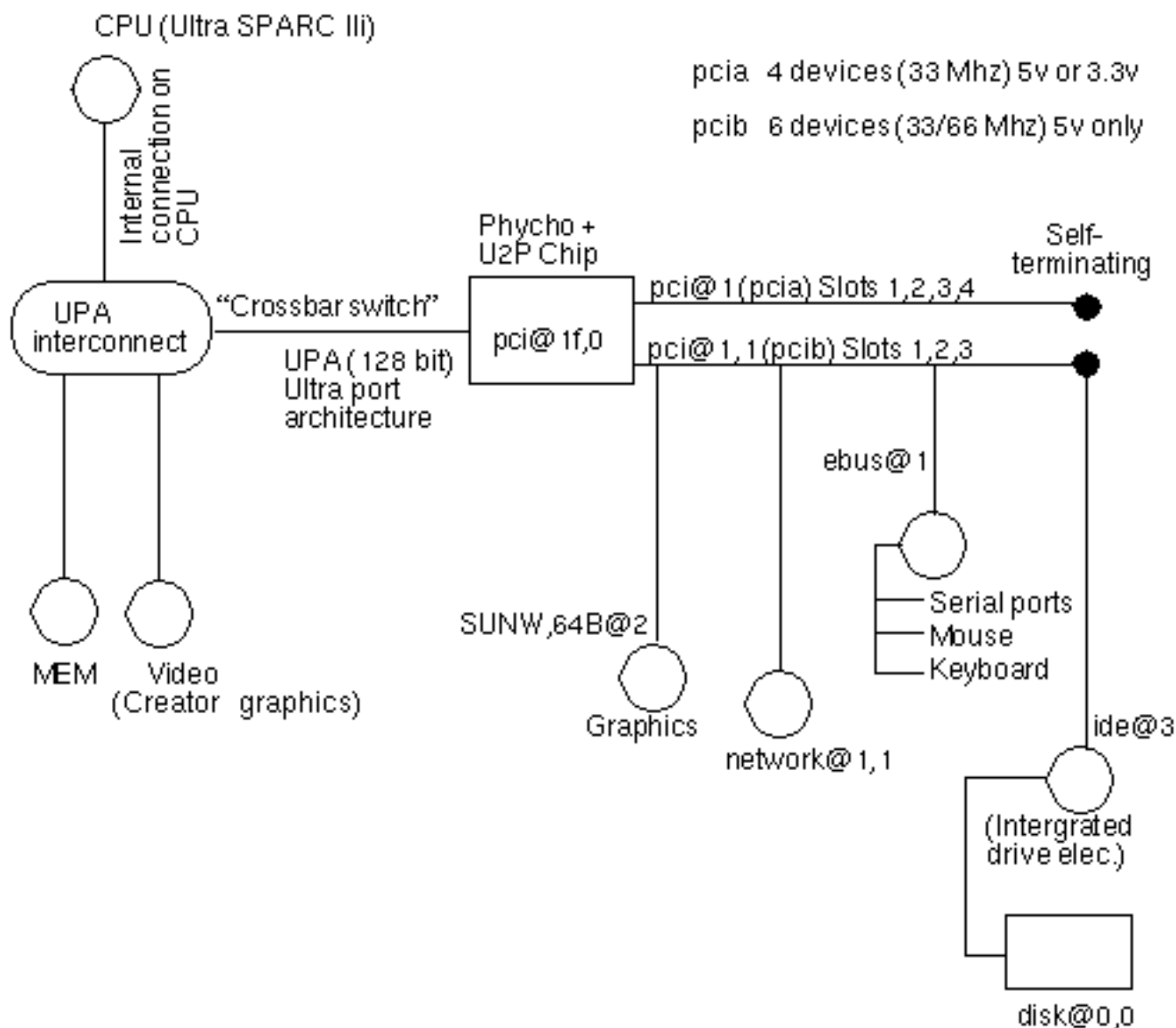


Objectives

- Gather general information about the system
- Define the meaning of the NVRAM variables
- Display and capture the names of the devices in the system device tree, and display their attributes
- Test devices using device path, node name, and alias
- Generate and test a PROM device alias
- Alter and display NVRAM settings, and reset to the defaults
- Use eeprom to examine and define NVRAM



Ultra 5 and Ultra 10 Architecture

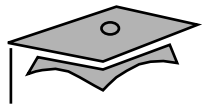




Functions and Capabilities of OBP

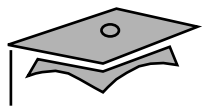
Two chips on each system board:

- The boot PROM itself
- A non-volatile random access memory (NVRAM)

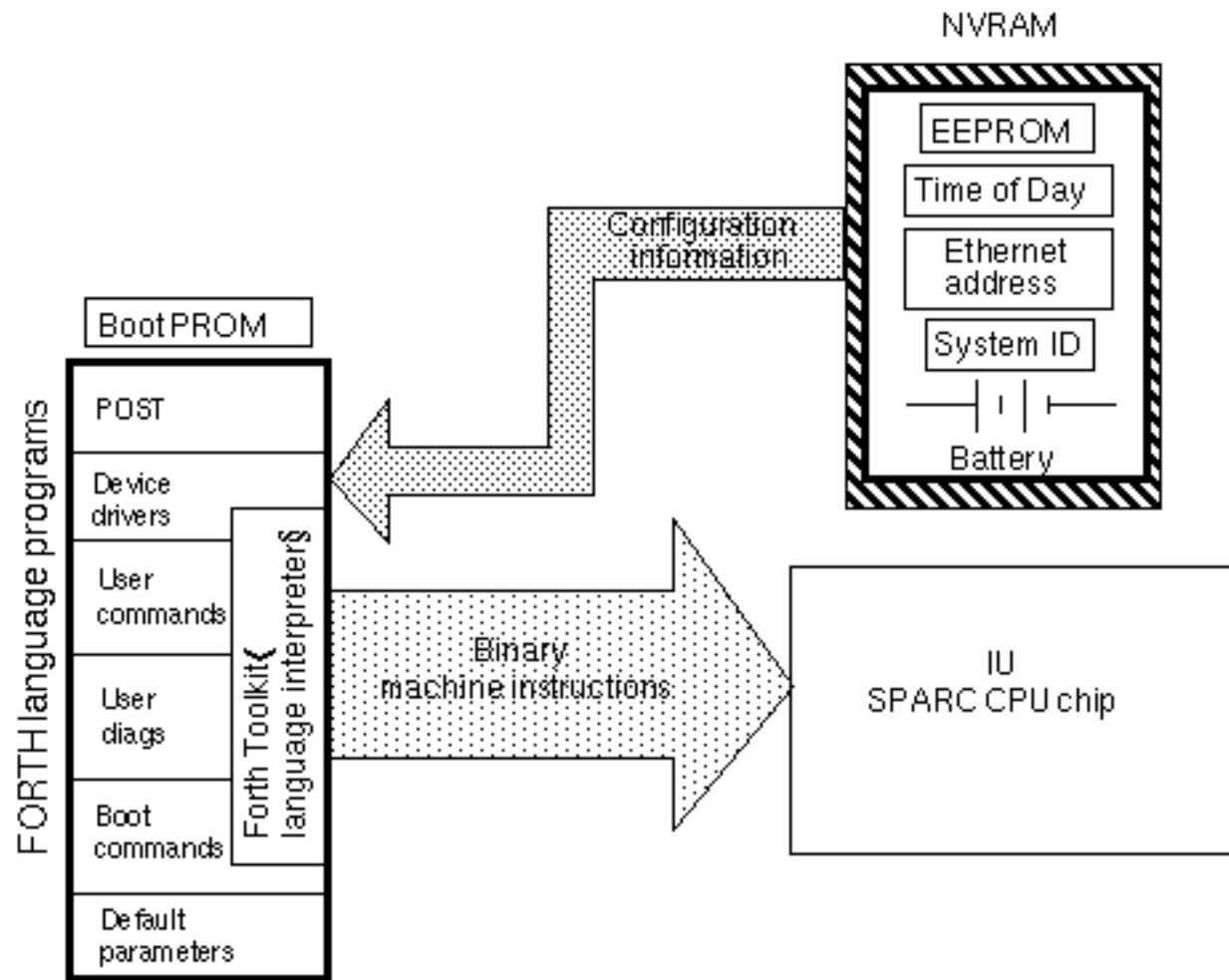


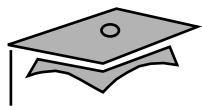
OBP Features

- Ability to read plug-in drivers and diagnostics from probed devices
- F(ORTH) code interpreter
- Device tree hierarchy
- Diagnostic, informational commands, and PROM variables
- Restricted monitor
- System initialization
- Power-On Self Tests (POSTs)



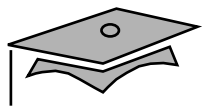
Open Boot PROM





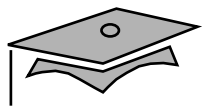
NVRAM Contents – Ultra 5

Variable Name	Value	Default Value
scsi-initiator-id	7	7
tpe-link-test	true	true
keyboard-click?	false	false
keymap		
ttyb-rts-dtr-off	false	false
ttyb-ignore-cd	true	true
ttya-ignore-cd	true	true
ttyb-mode	9600,8,n,1,-	9600,8,n,1,-
ttya-mode	9600,8,n,1,-	9600,8,n,1,-
pcia-probe-list	1,2,3,4	1,2,3,4
pcib-probe-list	1,2,3	1,2,3
mfg-mode	off	off
diag-level	max	max
#power-cycles	10	



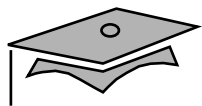
NVRAM Contents – Ultra 5

Variable Name	Value	Default Value
system-board-serial#	5013082149391	
system-board-date	356b9fd2	
fcode-debug?	false	false
output-device	screen	screen
input-device	keyboard	keyboard
load-base	16384	16384
boot-command	boot	boot
auto-boot?	true	true
watchdog-reboot?	false	false
diag-file		
diag-device	net	net
boot-file		
boot-device	disk net	disk net
local-mac-address?	false	false
ansi-terminal?	true	true



NVRAM Contents – Ultra 5

Variable Name	Value	Default Value
screen-#columns	80	80
screen-#rows	34	34
silent-mode?	false	false
use-nvramrc?	false	false
nvramrc		
security-mode	none	
security-passwd		
security-#badlogins	0	
oem-logo		
oem-logo?	false	false
oem-banner		
oem-banner?	false	false
hardware-revision		
last-hardware-update		
diag-switch	false	false



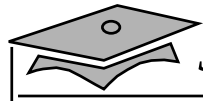
The eeprom Command

Parameter	Meaning
auto-boot?	If true, boot automatically after power-on or reset. Default is true.
diag-device	Diagnostic boot device to use when diag-switch is set to true. Default is net.
diag-switch?	If true, run in diagnostic mode. Default is false.
oem-logo?	If true, use custom OEM logo. Default is false (use Sun logo)
sbus-probe-list pcia-probe-list pcib-probe-list	List which system bus slots to probe, in what order. Default is 0123.
security-mode	Firmware security level. If set to <code>command</code> or <code>full</code> , the system prompts for PROM security password.
selftest#megs	Mbytes of RAM to test. Default is one.
tpe-link-test?	Enable 10baseT link test for built-in twisted pair Ethernet. Default is true.
watchdog-reboot?	If true, automatically reboot after a watchdog reset. Default is false.

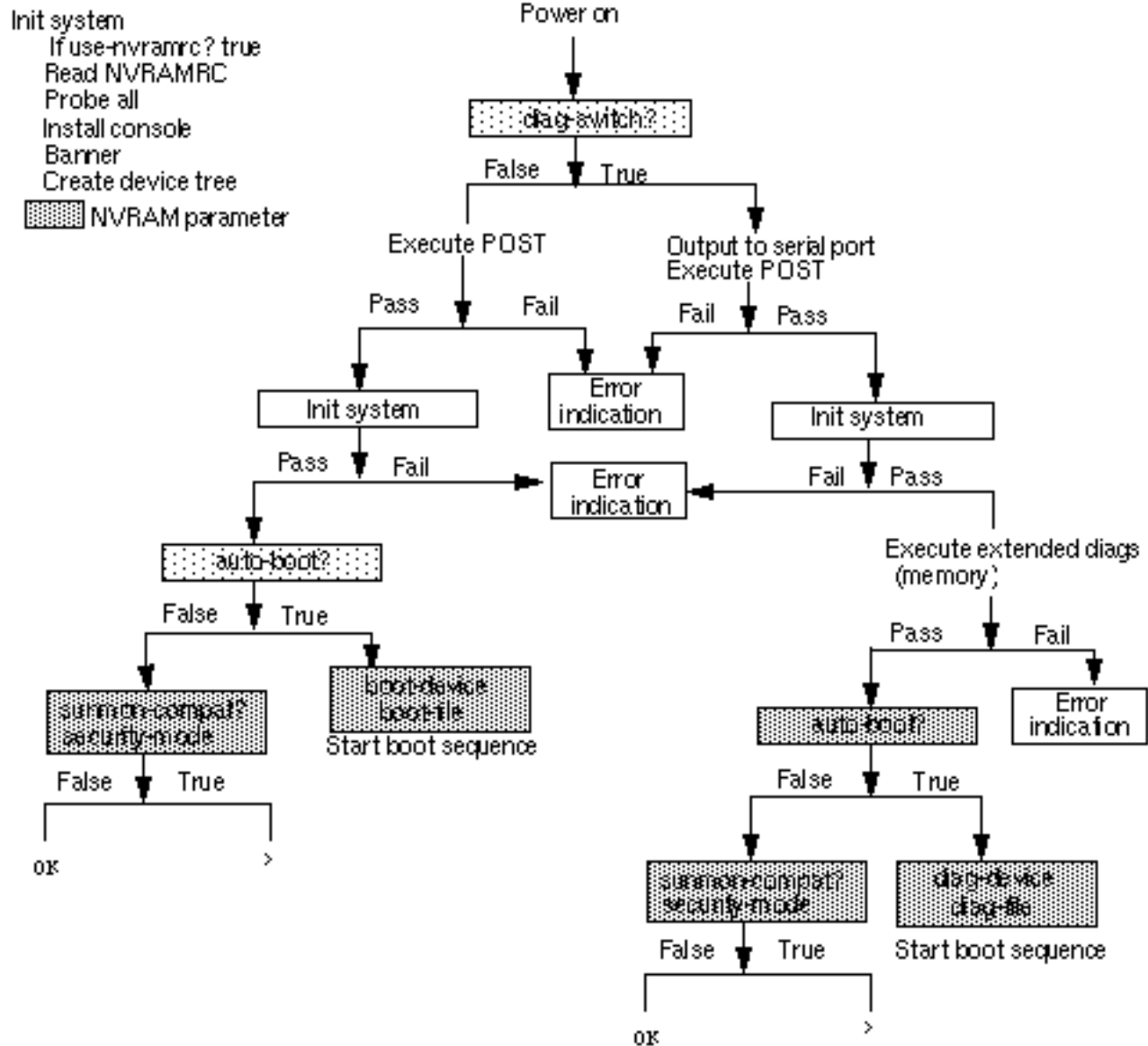


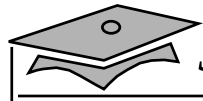
Changing PROM Variables With eeprom

- `/usr/sbin/eeprom` (*lists current settings*)
- `/usr/sbin/eeprom diag-switch?=true`
(*changes settings*)

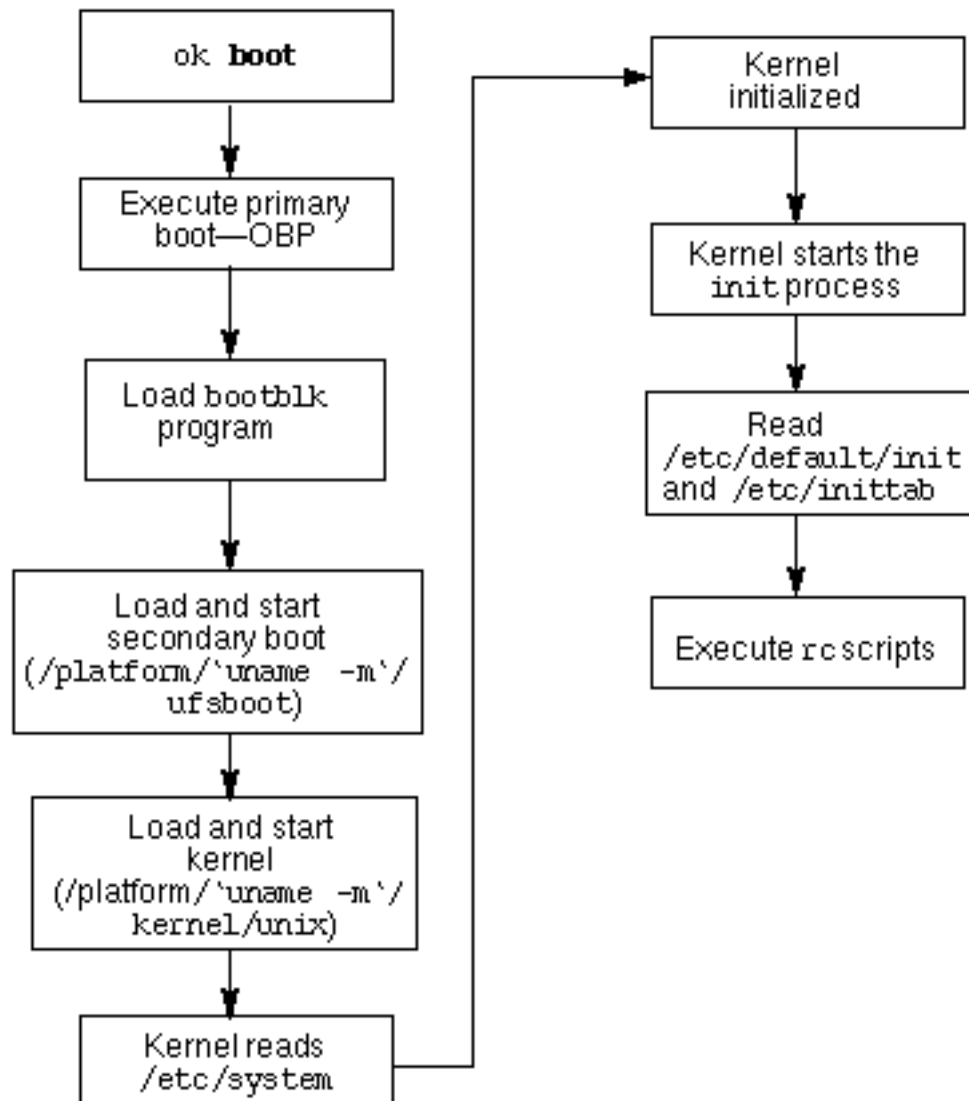


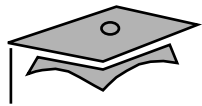
Events During Power On





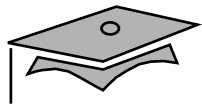
Default Boot Sequence





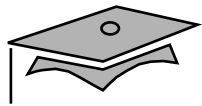
Default Boot Sequence

1. Enter the boot command at the ok prompt.
2. Use the boot command to locate the primary boot code.
3. Use the ufsboot program to load kernel core image files.
 - ▼ 32-bit SPARC and X86 –
/platform/*platform_name*/kernel/unix and
/kernel/genunix
 - ▼ 64-bit SPARC –
/platform/'uname -m'/kernel/sparcV9/unix and
/kernel/genunix



Default Boot Sequence

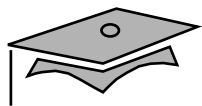
4. Read the `/etc/system` file as part of kernel initialization.
5. After creating the `sched` process (PID 0), the kernel creates process PID 1, which is `init`. The `init` daemon reads `/etc/inittab` and proceeds according to instructions in that file.
6. Use the `rc` scripts contain instructions to run additional scripts in the `rc` directories `/etc/rc0`, `/etc/rcS`, `/etc/rc2`, and `/etc/rc3`.



The rc Files and Directories

```
# ls /etc/rcS.d
```

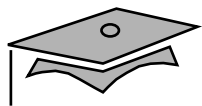
K07dmi	K39lp	K41rpc	S35cacheos.sh
K07snmpdx	K39spc	K42inetsvc	S40standardmounts
K10dtlogin	K40cron	K43inet	S41cachesfs.root
K28nfs.server	K40nsod	README	S50drvconfig
K33audit	K40syslog	S10inipcmcia	S60devlinks
K35volmgt	K40xntpd	S30rootusr.sh	S60pcmcia
K36sendmail	K41autofs	S33keymap.sh	S70buildmnttab.sh
K36utmpd			



The rc Directories and Files

```
# ls /etc/rc2.d
```

K07dmi	S69inet	S74syslog	S88sendmail
K07snmpdx	S70uucp	S74xntpd	S88utmpd
K28nfs.server	S71rpc	S75cron	S89bdconfig
ReadME	S71sysid.sys	S75savecore	S92volmgt
S01MOUNTFSYS	S72autoinstall	S76nsd	S93cacheos.finish
S05RMTMPFILES	S72inetsvc	S80lp	S99audit
S20syssetup	S73cachefs.daemon	S80PRESERVE	S99dtlogin
S21perf	S73nfs.client	S80spc	
S30sysid.net	S74autofs		



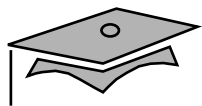
The rc Directories and Files

```
# ls /etc/rc3.d
```

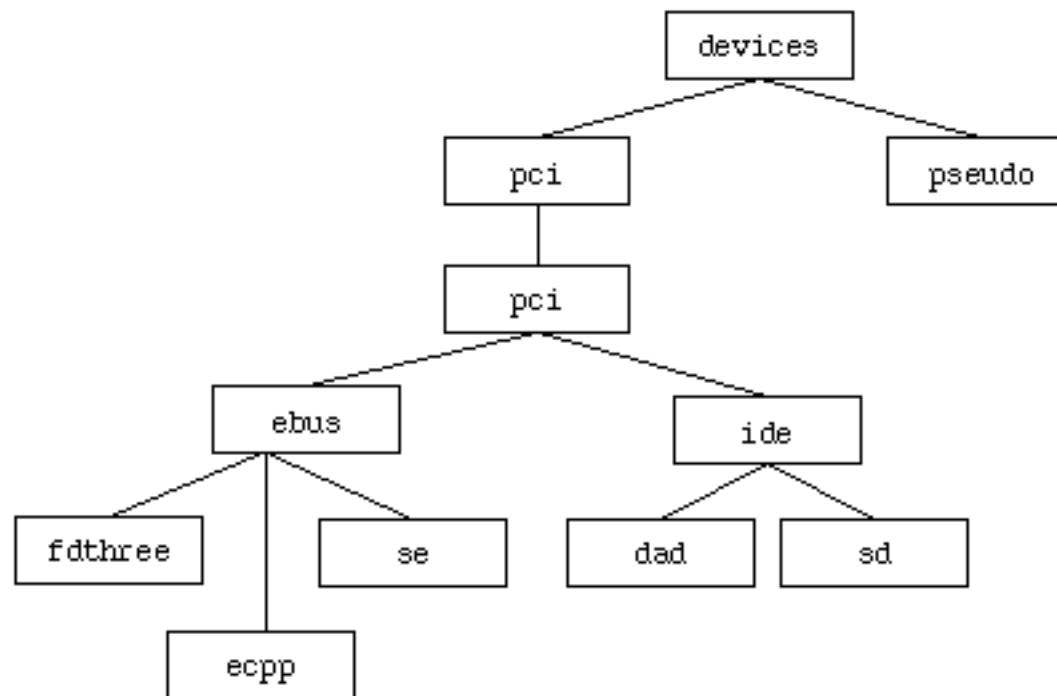
```
README          S15nfs.server  S76snmpdx      S77dmiw
```

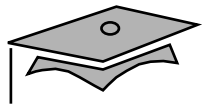
```
# ls /etc/rc0.d
```

```
K00ANNOUNCE    K35volmgt     K40cron        K41nfs.client
K07dmi         K36sendmail   K40nsd         K41rpc
K07snmpdx      K36utmpd      K40syslog      K42inetsvc
K10dtlogin     K39lp         K40xntpd       K43inet
K28nfs.server  K39spc        K41autofs      K78pcmcia
K33audit
```



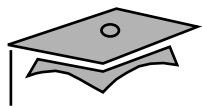
Device Tree Hierarchy





OBP Device Tree Navigation – UltraSPARC Workstation

- The `cd`, `ls`, and `pwd` commands are used to move around the device tree structure.
- The `dev device_pathname` command selects a node for examination.
- The `.properties` command shows the properties that define a particular node
- The `device-end` command unselects a device node.

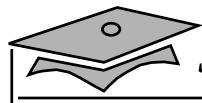


OBP Device Tree Navigation – UltraSPARC Workstation

```
<#0>ok cd /  
<#0>ok ls  
f006ce08 SUNW,UltraSPARC-IIia0,0  
f005f59c pcia1f,0  
f004d448 virtual-memory  
foo4cd68 memory@0,10000000  
f002cc3c aliases  
f002cb04 openprom  
f002ca98 chosen  
f002ca28 package
```

```
<#0>cd pcia1f,0  
<#0>ok ls  
f0060378 pci@1  
f005fd90 pci@1,1
```

```
<#0>ok cd pci@1  
<#0>ok pwd  
/pcia1f,0/pci@1  
<#0>ok ls  
f0083ac8 scsi@1,1  
f007db18 scsi@1
```



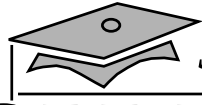
OBP Device Tree Navigation – UltraSPARC Workstation

```
<#0>cd pci@1,1
<#0>ok ls
f007a128 ide@3
f0075828 SUNW,m64b@2
f006d820 network@1
f00610e0 ebus@1

<#0>ok cd ide@3
<#0>ok pwd
/pcia1f,0/pci@1,1/ide@3
<#0>ok ls
f007d0a4 cdrom
f007c9f8 disk
<#0>ok cd ..
<#0>ok cd ebus@1
<#0>ok pwd
/pcia1f,0/pci@1,1/ebus@1

<#0>ok ls
f0066838 SUNW,CS4231@14,200000
f006677c flashprom@10,0
f0066694 eeprom@14,0
f0064ba4 fdthree@14,3023f0f006
f0064aa4 ecpp@14,3043bc
f0064988 su@14,3062f8
f0063168 su@14,3083f8
f0061840 su@14,400000
f00617ac SUNW,pll@14,504000
f00616dc power@14,724000
f006161c auxio@14,726000

<#0>ok
```



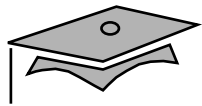
OBP Device Tree Navigation – SPARCstation 1000

```
<#0>ok cd /
<#0>ok ls
ffda476c io-unit@f,e1200000
ffd91c10 io-unit@f,e0200000
ffd8d2f4 mem-unit@f,e1100000
ffd8d210 mem-unit@f,e0100000
ffd8cebc cpu-unit@f,e1800000
ffd8cb68 cpu-unit@f,e1000000
ffd8c814 cpu-unit@f,e0800000
ffd8c4c0 cpu-unit@f,e0000000
ffd839a8 boards
ffd712fc openprom
ffd702bc virtual-memory@0,0
ffd7016c memory@0,0
ffd625cc aliases
ffd6257c options
ffd6252c packages
```

```
<#0>ok cd io-unit@f,e1200000
<#0>ok ls
ffda4d20 sbi@0,0
```

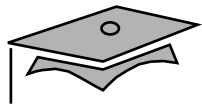
```
<#0>ok cd sbi
<#0>ok ls
ffdb0ffc lebuffer@1,40000
ffdac1f4 dma@1,81000
ffda9ff4 lebuffer@0,40000
ffda51ec dma@0,81000
```

```
<#0>ok cd dma@1,81000
<#0>ok ls
ffdac878 esp@1,80000
<#0>ok cd esp@1,80000
<#0>ok ls
ffdb05b4 st
ffaef4 sd
```



OBP Features – Ultra Enterprise Servers

- Hardware management
- Environmental monitoring
- Clock copy
- Externally initiated reset (XIR)
 - ▼ The clock board button
 - ▼ The key sequence <CR> <CR> <Ctrl-Shift-x>
 - ▼ The `.xir-state-all` command displays CPU status
- POST control



NVRAM Parameters – Ultra Enterprise

Parameter	Meaning
disabled-board-list	Specifies numerically identified boards to be disabled by OBP
disabled memory-list	Specifies numerically identified boards for which memory is disabled
memory-interleave	Enables memory interleave
disable-environmental-monitor	Turns off monitoring of power supplies and boards
copy-clock-tod-to-io-boards	Copies the TOD clock from the clock board to an I/O board
copy-io-board-tod-to-clock-tod	Copies the backup of the TOD clock from an I/O board to the clock board
configuration-policy	Specifies how devices that fail POST are handled (disable system, board, or component level)



Running OBP Commands With Key Sequences

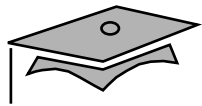
- The L1-f
- The L1-n
- The L1-d
- The L1 key runs POST in INIT mode, independent of security mode settings



Flash PROM Update for Sun4u Architectures

Systems and firmware

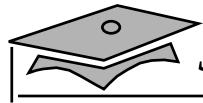
- UltraTM-1, if firmware is lower than Revision 3.11.1, use patch number 104881.
- Ultra-2, if the firmware is lower than Revision 3.11.2, use patch number 104169.
- Ultra-4, if the firmware is lower than Revision 3.7.107, use patch number 106122.
- Ultra EnterpriseTM systems, if the firmware is lower than Revision 3.2.16, use patch number 103346.



Flash PROM Update for Sun4u Architectures

1. Capture and save NVRAM variable settings.
2. Set the write-enable jumper, J2003, on Ultra-1 and Ultra-2 systems, or turn the front panel key switch to the Diagnostic position on Ultra Enterprise systems.
3. Perform the update procedure using the bootable CD-ROM, or use the update script, obtained by pulling the appropriate patch number from:

<http://sunsolve.sun.com/sunsolve/pubpatches/patches.html#Hardware-rec>



Ultra Workstation OBP Command Examples

ok help diag

test <device-specification> Run selftest for specific device

Examples:

test floppy

test net

test scsi

test-all - Execute test for all devices w/selftest enabled

watch-clock - Show ticks of real-time clock

watch-net - Monitor network broadcast packets

watch-net-all - Monitor broadcast packets on all network interfaces

probe-scsi - Show attached SCSI devices

probe-scsi-all - Show attached SCSI devices for all host adapters

ok .version

Release 3.11 Version 12 created 1999 01/19 11:30

OBP 3.11.12 1999 01/19 11:30

POST 2.2.9 1999 01/19 7:54

ok test net

Testing /pci@1f,0/pci@1,1/network@1

Internal loopback test -- succeeded

Transceiver check -- passed

ok test floppy

Testing /pci@1f,0/ebus@1/fdthree@14,3023f0

Testing floppy disk system. A formatted disk should be in the drive.

No diskette or incorrect format.

Selftest failed. Return code = -1

ok .enet-address

8:0:20:9e:06:c2

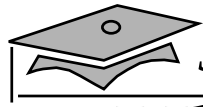
ok .speed

CPU Speed : 300.00MHz

UPA Speed : 100.00MHz

PCI Bus A : 33MHz

PCI Bus B : 33MHz



Ultra Workstation OBP Command Examples

ok **.idprom**

Format/Type: / 80 Ethernet: 8 0 20 9e d6 c2 Date:

Serial: 9ed6c2 Checksum: a9

ok **probe-scsi-all**

/pci@1f,0/pci@1/scsi@1,1

Target 0

Unit 0 Disk Conner CFP1080E SUN1.055150

Target 4

Unit 0 Removable Tape Archive PYTHON 28454-XXX.BL

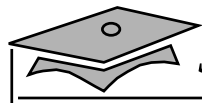
pci@1f,0/pci@1/scsi@1

ok **banner**

Sun Ultra 5/10 UPA/PCI (UltraSPARC-II; 300 MHz), Keyboard present

Open Boot 3.11 128MB memory installed, Serial # 10409666

Ethernet address 8:0:20:9e:d6:c2, Host ID: 809ed6c2



SPARCstation 20 OBP Command Examples

<#0> ok **help diag**

Category: Diag (diagnostic routines)

test device-specifier (--) run selftest method for specified device

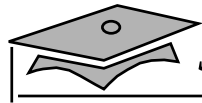
Examples:

```
test /memory      - test memory
test /iommu/sbus/ledma@f,400010/le - test net
test floppy       - test floppy disk drive
test net          - test net (device-specifier is an alias)
test scsi         - test scsi (device-specifier is an alias)
watch-clock      ( -- ) show ticks of real-time clock
watch-net        ( -- ) monitor broadcast packets using auto-
selected
                                interface
watch-aui         ( -- ) monitor broadcast packets using AUI
interface
watch-tpe         ( -- ) monitor broadcast packets using TPE
interface
watch-net-all    ( -- ) monitor broadcast packets on all net
interfaces
probe-scsi        ( -- ) show attached SCSI devices
probe-scsi-all   ( -- ) show attached SCSI devices for all host
adapters
test-all         ( -- ) run test for all devices with selftest
method
test-memory       ( -- ) test all memory if diag-switch? is true,
otherwise
                                test memory specified by selftest-#megs
```

<#0> ok

<#0> ok **show-sbus**

```
SBus slot f SUNW,bpp ledma le espdma esp
SBus slot e SUNW,DBRIe
SBus slot 0
SBus slot 1
SBus slot 2 cgsix
SBus slot 3
```



SPARCstation 20 OBP Command Examples

<#0> ok **probe-scsi**

Target 1

Unit 0 Disk QUANTUM P105SS 910-10-94A.1 08/31/89009030144 GENERIC

Target 3

Unit 0 Disk SEAGATE ST31200W SUN1.05872400795741

Copyright (c) 1994 Seagate

All rights reserved 0000

Target 4

Unit 0 Removable Tape ARCHIVE VIPER 150 21531-003 SUN-03.00.00

Target 6

Unit 0 Removable Read Only device TOSHIBA XM-4101TASUNSLCD108404/18/94

<#0> ok **module-info**

MBus : 50.00 MHz

SBus : 25.00 MHz

CPU#0 : 50.00 MHz SuperSPARC

CPU#2 : 50.00 MHz SuperSPARC

<#0> ok **2 switch-cpu**

<#2> ok **0 switch-cpu**

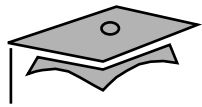
<#2> ok **1 switch-cpu**

Processor #1 is not present!



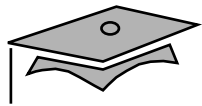
Exercise: Working With Ultra and SPARC 5 Workstations OBP Utilities

- Exercise objective
- Preparation
- Tasks
- Exercise summary



Check Your Progress

- Gather general information about the system
- Define the meaning of the NVRAM variables
- Display and capture the names of the devices in the system device tree, and display their attributes
- Test devices using the device path, node name, and device alias
- Generate and test a PROM device alias
- Alter and display NVRAM settings, and reset to the defaults
- Use the `eeeprom` command to examine and define NVRAM



Think Beyond

- Based on the device and OBP information in this module, how do you think you list disk device names for SPARCstorageTM arrays?
- Why aren't storage array names visible from OpenBoot PROM?



Module 5

SunSolve Database Information



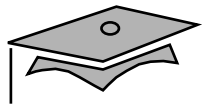
Objectives

- Use the SunSolve database for fault analysis
- Differentiate SunSolve CD-ROMSM and OnlineSM databases
- Describe how to apply for a SunSolve Online account
- Install SunSolve software on a server and share correctly to a network
- Configure and use SunSolve software from a server or CD-ROM



Objectives

- Display the installed patches on a Solaris system
- Display the current patch report
- Install and remove patches on a Solaris system
- Solve a workshop exercise using the SunSolve database



Distribution

SunSolve Online is

- Available to all Sun customers with any level of service agreement
- As a separate purchase through the 1-800-USA4SUN phone number.

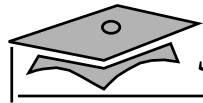
To utilize SunSolve Online, establish an account using your service contract number. Sun employees use their employee ID number.



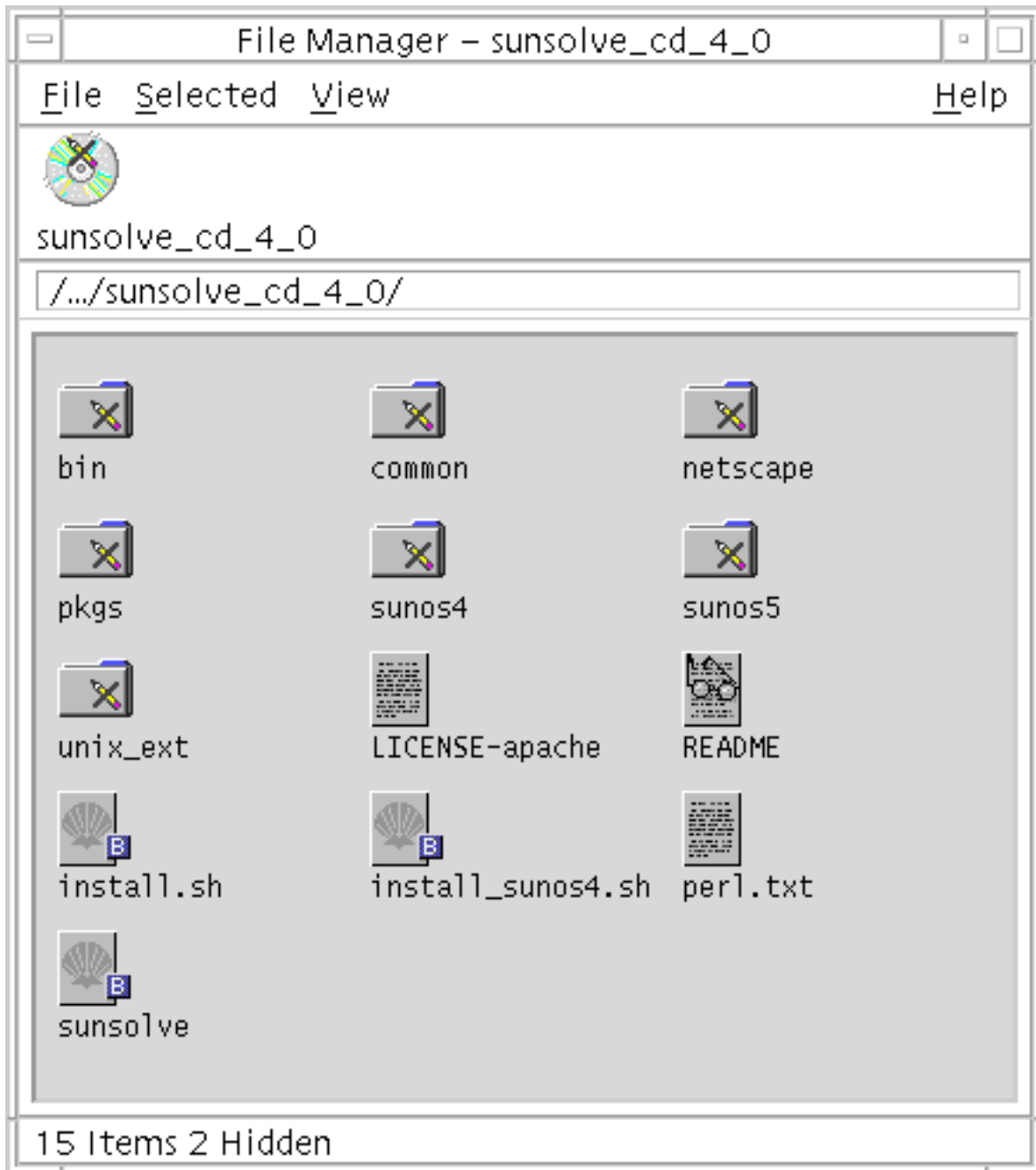
SunSolve Online Account

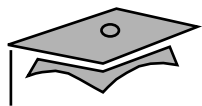
To apply for a SunSolve Online account:

1. Use a browser to access one of the following web sites:
 - ▼ <http://sunsolve.sun.com>
 - ▼ <http://sunsolve1.sun.com>
 - ▼ <http://www.sun.com>
2. Click on the Register button.
3. Click on the Create new account button and answer the questions.



Installing SunSolve Software





Installing SunSolve Software

```
# cd /cdrom/cdrom0
# ./install.sh
```

Script started on Thu Nov 05 10:44:51 1998

The following packages are available:

```
1  SUNWss      SunSolve CD
      (sparc) 4.0.0,REV=Final Release
```

Select package(s) you wish to process (or 'all' to process all packages). (default: all) [?,??,q]: 1

Processing package instance <SUNWss> from </cdrom/sunsolve_cd_4_0/pkg/SUNWss.sparc>

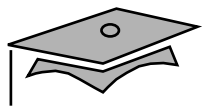
SunSolve CD
(sparc) 4.0.0,REV=Final Release
Copyright 1998 Sun Microsystems, Inc. All rights reserved.

Enter path to package base directory [?,q] /opt/SUNWss

The selected base directory </opt/SUNWss> must exist before installation is attempted.

Do you want this directory created now [y,n,?,q] y

Using </opt/SUNWss> as the package base directory.



Installing SunSolve Software

```
## Processing package information.  
## Processing system information.  
## Verifying disk space requirements.  
## Checking for conflicts with packages already installed.  
## Checking for setuid/setgid programs.
```

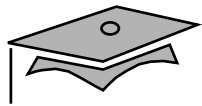
This package contains scripts which will be executed with super-user permission during the process of installing this package.

Do you want to continue with the installation of <SUNWss> [y,n,?] y

Installing SunSolve CD as <SUNWss>

```
## Executing preinstall script.  
## Installing part 1 of 1.
```

```
/etc/rc2.d/S89httpd  
/opt/SUNWss/sso/etc/K89httpd <symbolic link>  
[ verifying class <none> ]  
/etc/init.d/httpd_init <linked pathname>  
/etc/rc1.d/K89httpd <linked pathname>  
## Executing postinstall script.  
installing bin...  
...  
Installation of SUNWss successful.
```



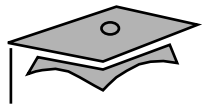
Configuring a SunSolve Server

1. Insert the SunSolve CD-ROM on the server.
2. If needed, perform the installation procedure described previously, or run the software from the CD-ROM.
3. Share the software and, if needed, the CD-ROM.

```
# share -o ro SunSolve_install_dir  
# share -o ro /cdrom/crom0
```

4. Add the following line to /etc/dfs/dfstab:

```
# share -o ro SunSolve_install_dir  
# share -o ro /cdrom/cdrom0
```



Configuring a SunSolve Server

5. Start the NFS server:

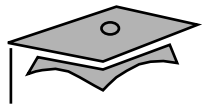
```
# /etc/init.d/nfs.server start
```

6. Check to see if the share command was successful:

```
# dfshares -F nfs server_name
```

7. On clients, remotely mount the software, or provide the URL of `http://server_name`.

```
# mount server_name:/cdrom/cdrom0 /cdrom
```



Running SunSolve Software

There are three ways you can run SunSolve software:

- From the SunSolve CD-ROM
- From a web site
- From an installed server at your site



The SunSolve Online Home Page Selections

- Home Page
- Getting Started
- About the CD
- Browse Collections
- Patch Reports



The SunSolve Online Home Page Selections

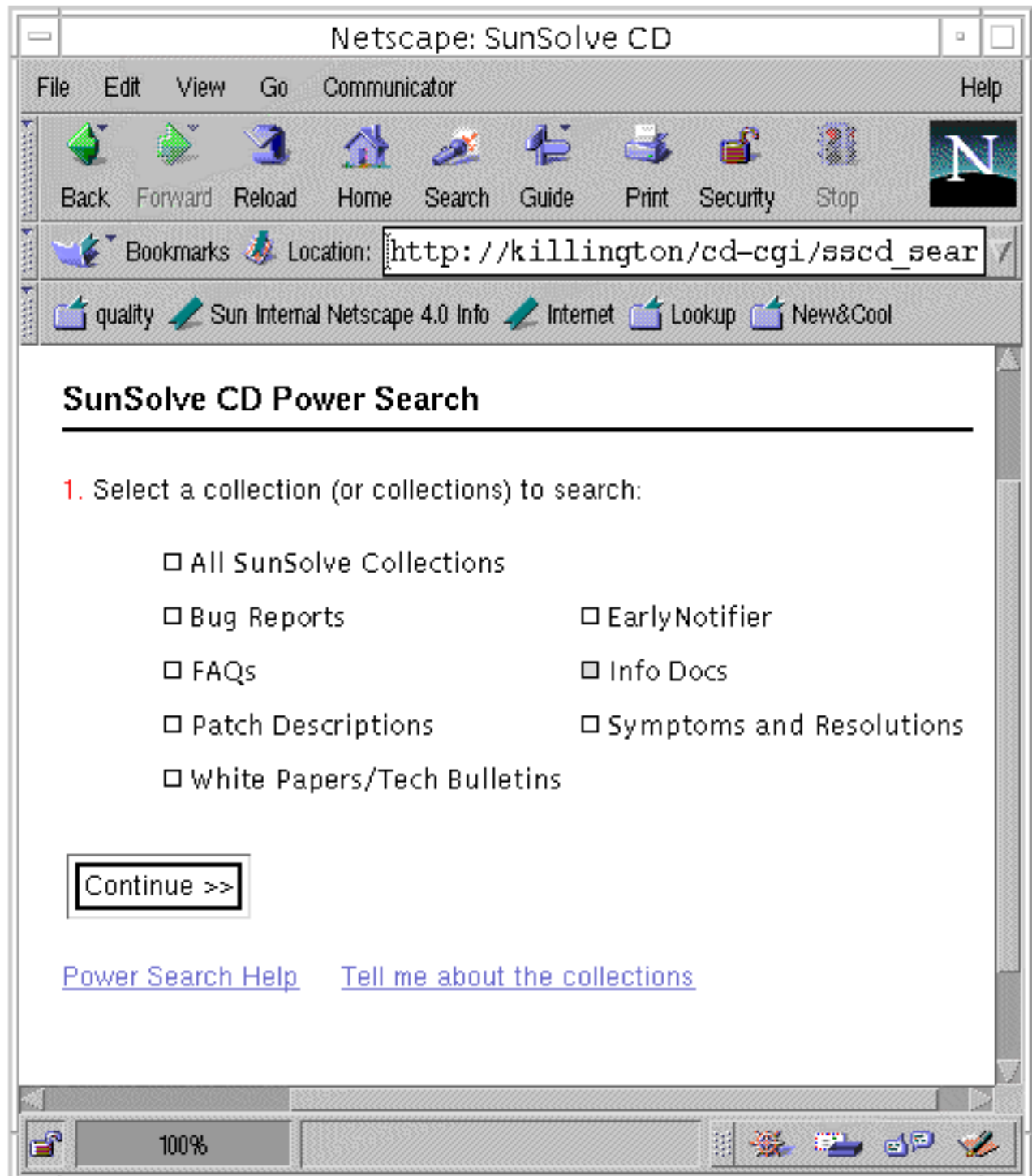
- Power Search
- Patch Diag Tool
- Crash Dump Analysis
- SunCourier



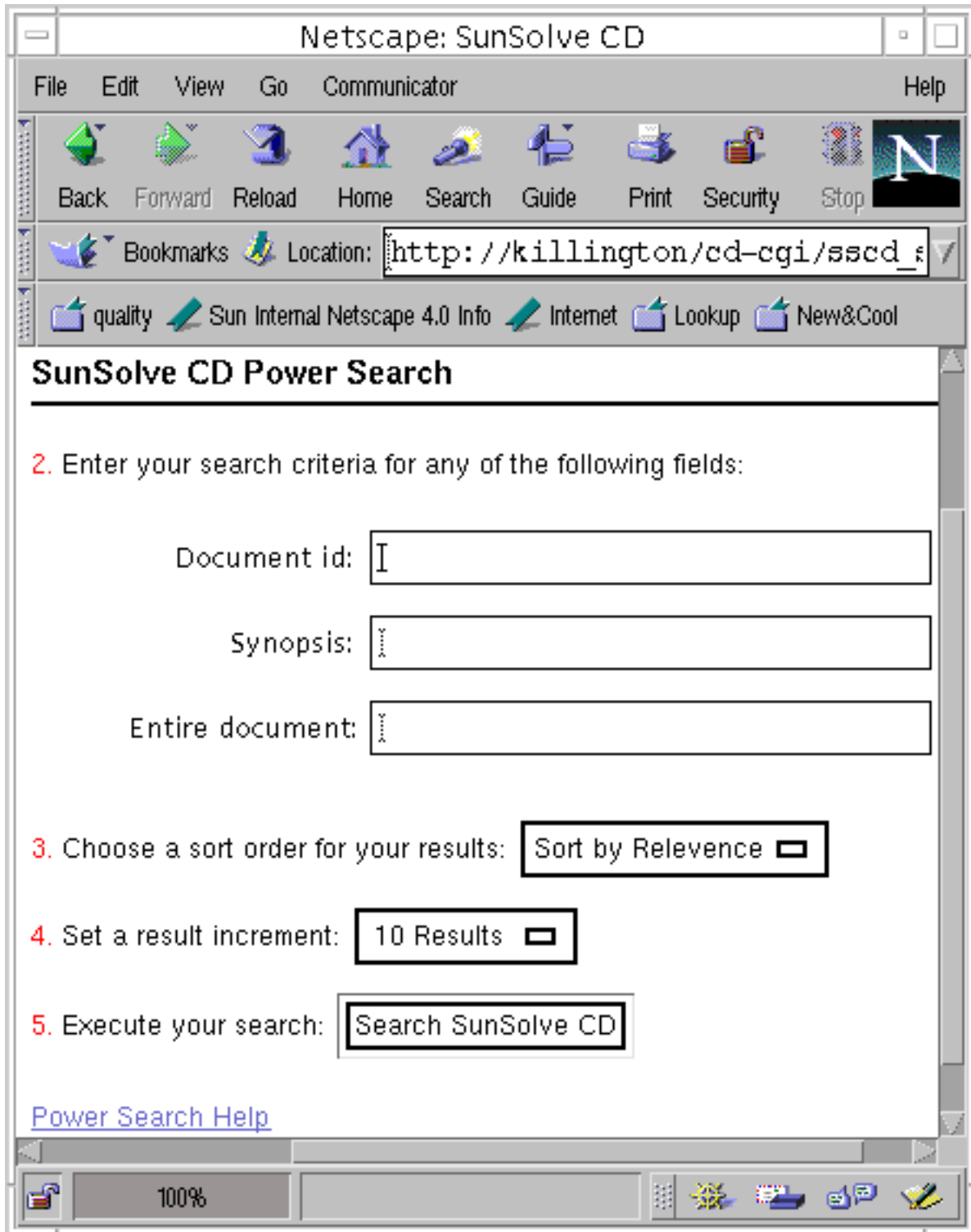
Power Search Tool Overview

- To move from the initial Power Search screen to the display that allows search criteria to be specified, click on Continue.
- Once you have set the criteria, click on Search.
- Having located a document(s), enter a string to refine the search within a particular document.

Power Search Collections



Defining Search Criteria

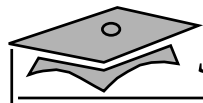


The screenshot shows a Netscape browser window titled "Netscape: SunSolve CD". The address bar contains the URL "http://killington/cd-cgi/sscd_". The browser's menu bar includes File, Edit, View, Go, Communicator, and Help. The toolbar features icons for Back, Forward, Reload, Home, Search, Guide, Print, Security, Stop, and a large 'N' logo. Below the toolbar, there are links for "quality", "Sun Internal Netscape 4.0 Info", "Internet", "Lookup", and "New&Cool".

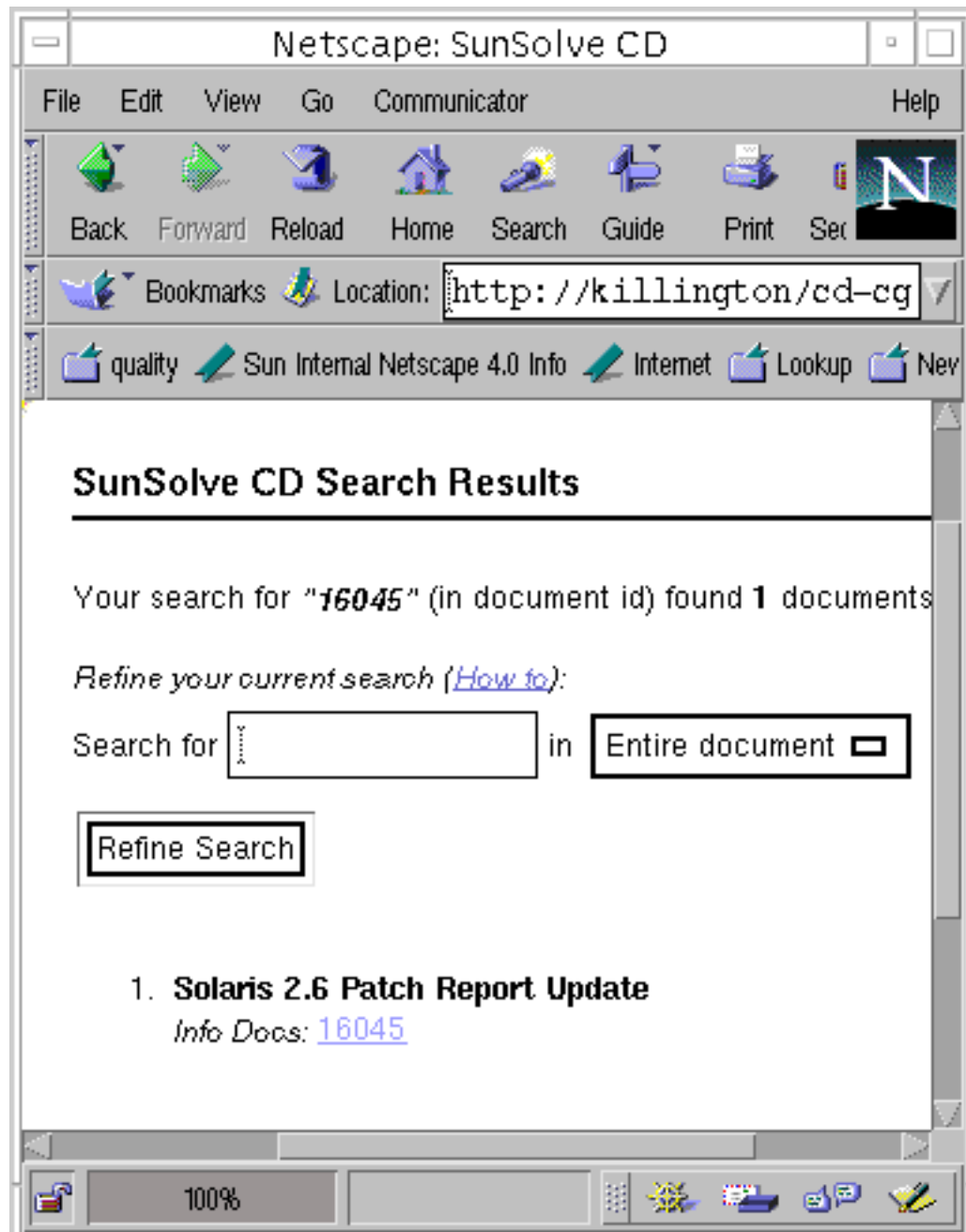
The main content area is titled "SunSolve CD Power Search". It contains the following instructions and form fields:

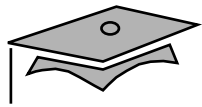
- 2. Enter your search criteria for any of the following fields:
 - Document id:
 - Synopsis:
 - Entire document:
- 3. Choose a sort order for your results:
- 4. Set a result increment:
- 5. Execute your search:

At the bottom of the form, there is a link for [Power Search Help](#). The browser's status bar at the bottom shows a zoom level of 100% and several system icons.



Power Search Results





Boolean Search Syntax

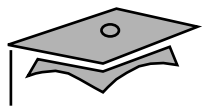
- AND – This is the default operator when two terms are entered on a query line.
- OR – Curly braces ({ }) are used to designate an OR operation.
- NOT – The exclamation point (!) specifies the NOT operation.



The Current Patch Report

Click on the SunSolve Online home page. This provides screen areas with two main selections:

- Patch Report
- Enter a patch ID or keyword

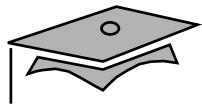


The Solaris 2.6 Patch Report

Solaris 2.6 Recommended Patches:(excerpt)

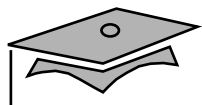
106125-05	SunOS 5.6: Patch for patchadd and patchrm
105401-15	SunOS 5.6: libnsl and NIS+ commands patch
105181-08	SunOS 5.6: kernel update patch
105562-03	SunOS 5.6: chkey and keylogin patch
105210-13	SunOS 5.6: libc & watchmalloc patch
105216-03	SunOS 5.6: /usr/sbin/rpcbind patch
105356-06	SunOS 5.6: /kernel/drv/ssd patch
105357-02	SunOS 5.6: /kernel/drv/ses patch
105375-09	SunOS 5.6: sf & socal driver patch
105379-05	SunOS 5.6: /kernel/misc/nfssrv patch
105621-08	SunOS 5.6: libbsm patch
105393-06	SunOS 5.6: at & cron utility patch
105395-03	SunOS 5.6: /usr/lib/sendmail patch
105407-01	SunOS 5.6: /usr/bin/volrmmount patch
105552-02	SunOS 5.6: /usr/sbin/rpc.nisd_resolv patch
105615-04	SunOS 5.6: /usr/lib/nfs/mountd patch

.....



Patch Diag Tool

- Is one of the newer options within SunSolve Online
- Provides a report indicating the status of installed patches on a system
- Provides
 - ▼ A sample report
 - ▼ Installation instructions
 - ▼ A user's guide



The Patch Diag Report

```
# cd /SunSolve_install_dir/sso/PatchDiag
```

```
# ./patchdiag.sparc
```

```
=====
System Name: killington  SunOS Vers: 5.7          Arch: sparc
Cross Reference File Date: 25/Sep/98
```

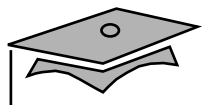
```
PatchDiag Version: 1.0.2
=====
```

```
Report Note:
```

Recommended patches are considered the most important and highly recommended patches that avoid the most critical system, user, or security related bugs which have been reported and fixed to date. A patch not listed on the recommended list does not imply that it should not be used if needed. Some patches listed in this report may have certain platform specific or application specific dependencies and thus may not be applicable to your system. It is important to carefully review the README file of each patch to fully determine the applicability of any patch with your system.

```
=====
INSTALLED PATCHES
```

```
Patch  Installed Latest  Synopsis
  ID    Revision  Revision
-----
```



The Patch Diag Report

UNINSTALLED RECOMMENDED PATCHES

Patch ID	Ins Rev	Lat Rev	Age	Require ID	Incomp ID	Synopsis
-------------	------------	------------	-----	---------------	--------------	----------

All Recommended patches installed!
=====

UNINSTALLED SECURITY PATCHES

NOTE: This list includes the Security patches that are also Recommended

Patch ID	Ins Rev	Lat Rev	Age	Require ID	Incomp ID	Synopsis
-------------	------------	------------	-----	---------------	--------------	----------

All security patches installed!

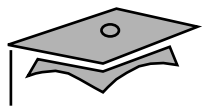
UNINSTALLED Y2K PATCHES

NOTE: This list includes the Y2K patches that are also Recommended

Patch ID	Ins Rev	Lat Rev	Age	Require ID	Incomp ID	Synopsis
-------------	------------	------------	-----	---------------	--------------	----------

=====

All Y2K patches installed!



Installing Patches

```
# ./patchinstall
```

```
=====
patchinstall - install a patch
```

```
Copyright (c) 1993 Sun Microsystems, Inc. All Rights Reserved.
```

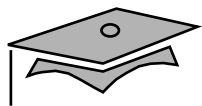
```
Printed in the United States of America.
```

```
2550 Garcia Avenue, Mountain View, California, 94043-1100 U.S.A.
```

```
Patches are distributed to SunService Contract Customers ONLY. Redistribution to unauthorized parties is prohibited by the SunService Contract. Installation of all patches is not suggested as some patches may conflict with one another. Please make sure that the patch you are installing is necessary before actually installing the patch.
```

```
=====
During the installation, default answers will be provided inside brackets '[' Pressing the <Return> key will select the default provided.
```

```
Press <Ctrl-C> at any time to stop the installation.
```



Installing Patches

Continue with patch installation? [Y]

Installation of patches will use several temporary files, and may need several megabytes of space. Please enter the name of a directory that can be used for these temporary files (the directory must exist before running the installation!)

Where should I store temporary files? [/tmp]

The installation for Solaris 2.x patches provides the option of saving the original versions of the software being patched. Unfortunately, this occasionally will cause your /var/sadm/patch directory to grow too large. By default, this installation will NOT save your original versions of software.

Would you like to save the original versions of the software? [no]

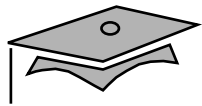
Next you will be prompted for the patches you wish to install. If you would like to install the suggested patches, enter "suggested". Otherwise enter the patch id.

To see a list of the patches you have entered, type a '?' <Return>

To start the installation, press <Return> when prompted for a patch id.

Patch to install (patchid, suggested, ?): suggested

.....
Installation complete!



Removing a Patch

1. Display a list of installed patches on your system.

```
# showrev -p
```

2. Find the installed location of the patch.

```
# find / -name 102044-01 -print
```

3. Change directory to the location of the patch.

```
# cd /var/sadm/patch/102044-01
```

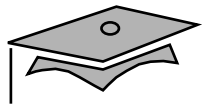
4. Run the script to remove the patch, and reboot the system.

```
# ./backoutpatch .
```



Useful SunSolve Documents

- *Sun High Performance Computing White Paper (1367)*
- *Ultra 450 Architecture White Paper (1396)*
- *Sun Enterprise Cluster Architecture White Paper (1380)*
- *Troubleshooting Guide for Watchdog Resets (1360)*
- *Security in Practice (1390)*



Useful SunSolve Documents

- *ISCDA (13089)*
- *Information to Send to Sun for Analysis (11835)*
- Current patch reports – All releases
- Specific architectures, such as Sun4u or Sun4m
- Common Solaris error messages in alphabetical order



Other Useful SunSolve Online Documents

- _____

- _____

- _____

- _____

- _____

- _____

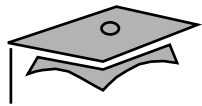
- _____

- _____



Exercise: Using SunSolve Fault Analysis

- Exercise objective
- Preparation
- Tasks
- Exercise summary



Check Your Progress

- Use the SunSolve database for fault analysis purposes
- Differentiate between the SunSolve CD-ROM and SunSolve Online databases
- Describe how to apply for a SunSolve Online account
- Install the SunSolve software and patches software on a server and share them correctly to the network
- Configure and use SunSolve software from an installed server or from the CD-ROM
- Display the installed patches on a Solaris system
- Display the current patch report for a given operating system
- Install and remove patches as needed on a Solaris system
- Solve a workshop exercise using SunSolve databases



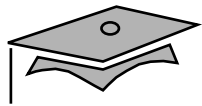
Think Beyond

- Now that you have had some exposure to using SunSolve software, are there any issues at your site that should be analyzed by using the SunSolve database?
- Are there Solaris topics for which there would exist no reference in SunSolve databases?



Module 6

SunVTS System Diagnostics



Objectives

- Install the SunVTS package on a system
- Select, set up, and run SunVTS diagnostic tests
- Run SunVTS software over a network
- Run SunVTS software in TTY mode without a frame buffer
- Analyze SunVTS test results



Introduction

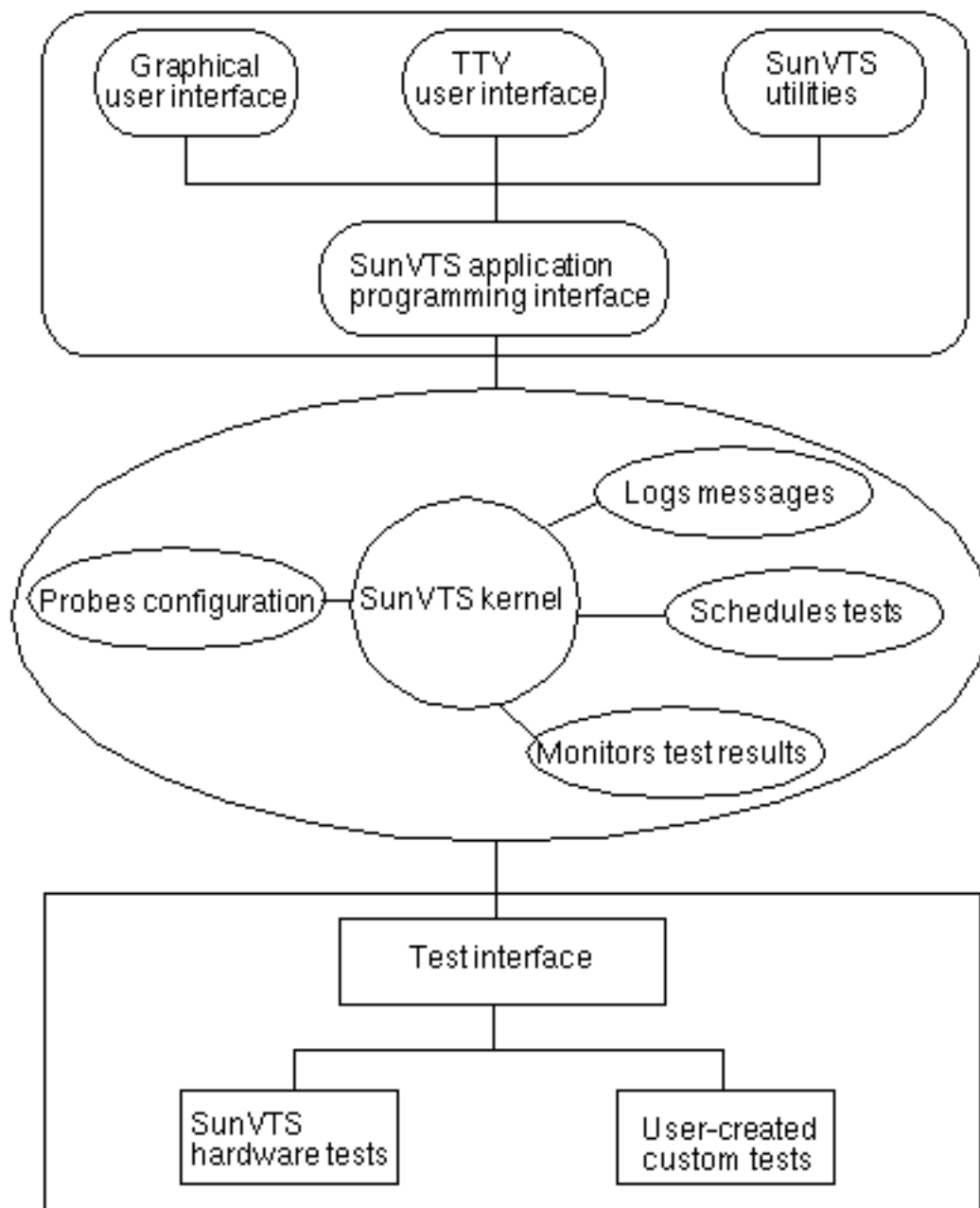
- SunVTS software
 - ▼ Is an on-line validation test suite
 - ▼ Verifies functionality of Sun hardware devices
 - ▼ Is a successor to SunDiag™ diagnostics
 - ▼ Runs on Solaris 2.5 and later releases

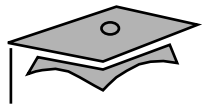


Hardware and Software Requirements

- The Solaris 7 operating system
- The SunVTS 3.0 package
- Operating system kernel configured to support all peripherals that are to be tested
- Superuser access for installation and startup of SunVTS software

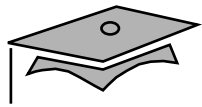
The SunVTS Architecture





Interfaces

- User
- Kernel
- Hardware tests



Installing SunVTS Software

The following packages comprise the SunVTS software:

- SUNwvts
- SUNwvtsx
- SUNwodu
- SUNwvtsmn

The SunVTS software can be installed using either

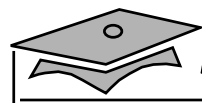
- The pkgadd command
- The installation script



Starting the SunVTS Software

The SunVTS program is started using

- `sunvts`
- `sunvts -l`
- `sunvts -t`
- `sunvts -h host_name`



The SunVTS GI

SunVTS Diagnostic

Commands View Options Reports Help

Start Stop Reset Host Log Meter Quit

Hostname: **killington** Mode: "Ultra-1"

Testing status: **idle**

System passes: 0 Cumulative errors: 0 Elapsed test time: 000:00:00

Select devices

☐ Default
☐ None
☐ All
☐ Intervention

Select mode

☐ Connect on test
☐ Functional test

System map: ☐ Physical ☐ Logical Green = Pass Red = Fail

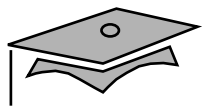
- ☐ ☒ Processor(s)
- ☐ ☒ Memory
- ☐ ☒ Network
- ☐ ☒ Graphics
- ☐ ☐ Comm.Ports
- ☐ ☒ OtherDevices
- ☐ ☒ SCSI-Devices(esp0)

Test messages: Clear



The SunVTS Window Panels

- System Status Panel
- The System Map
- Select Devices
- Select Mode
- Test Messages



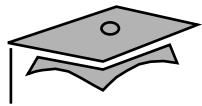
The SunVTS Window Icons

- Start
- Stop
- Reset
- Host
- Log
- Meter
- Quit



The SunVTS Menu Selections

- Commands
- View
- Options
- Reports

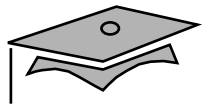


The Schedule Options Menu

The screenshot shows a window titled "Schedule Options". Inside, under the heading "Options:", there are four settings:

- Auto Start:** Two radio buttons, "enabled" (selected) and "dicabled".
- Single Pass:** Two radio buttons, "enabled" (selected) and "dicabled".
- System Concurrency:** A spin box with up/down arrows and a text field containing the value "1".
- Group Concurrency:** A spin box with up/down arrows and a text field containing the value "1".

At the bottom of the dialog, there are three buttons: "Apply", "Reset", and "Cancel".



The Test Execution Menu

Test Execution Options

Options:

Stress: ☐ enabled ☐ disabled

Verbose: ☐ enabled ☐ disabled

Core File: ☐ enabled ☐ disabled

Run On Error: ☐ enabled ☐ disabled

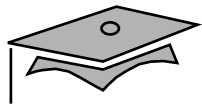
Max Passes:

Max Errors:

Max Time:

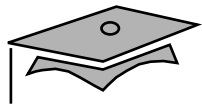
Num. of Instances:

Apply Reset Cancel



The Test Execution Menu

- Stress
- Verbose
- Core file
- Run on Error
- Max Passes
- Max Errors
- Max Time
- Number of Instances



The Advanced Options Menu

Advanced Options

Options:

System Override: ☐ enabled ☐ disabled

Group Override: ☐ enabled ☐ disabled

Group Lock: ☐ enabled ☐ disabled

Test Lock: ☐ enabled ☐ disabled

Apply Reset Cancel

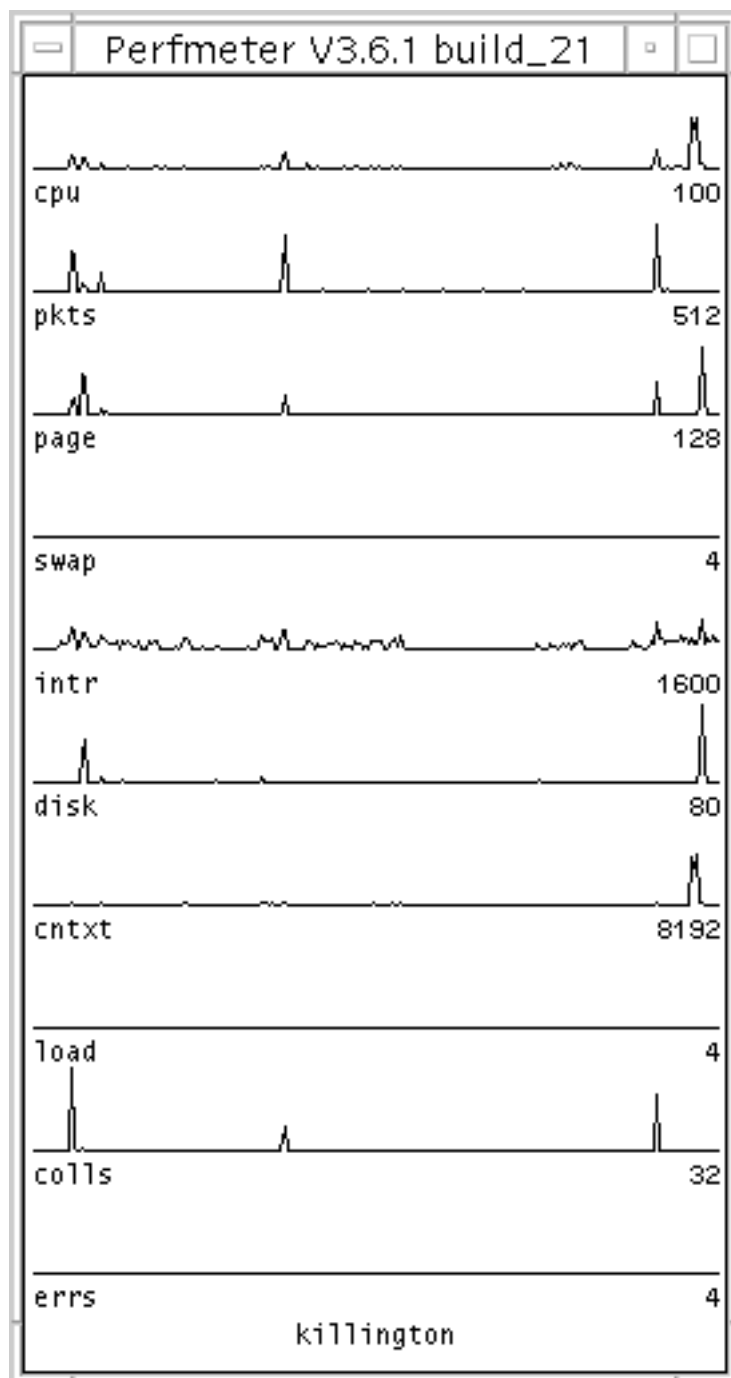


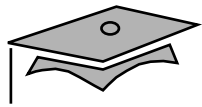
Intervention Mode

Certain tests require that you intervene with

- Loopback connectors needed for certain tests
- Media needed to test tape, diskette, or CD-ROM drives

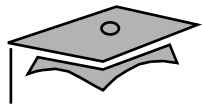
Performance Meter Panel





Performance Monitor Panel

- cpu
- pkts
- page
- swap
- intr
- disk
- cntxt
- load
- colls
- errs



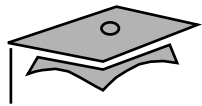
Using SunVTS in TTY Mode

```
SunVTS
*****Hostname:jeeves-----Model:Sun_4_75-----SunVTS version:1.0*****
start      reset      quit      reprobe    tests      SysConf
set_options option-files log-files  connect_to intervention help
*****
                Status
System_Status:idle
System_Passes:0   Total_Errors:0
Elapsed_Time:000:00:00
                Status_view
                Passes Errors

Processor(s)      0      0
 fpu(fputest)     0      0
 fpu(fputest).1   0      0
Memory            0      0
 kmem(vmem)       0      0
 kmem(vmem).1     0      0

                Test_Groups
[*] Processor(s)   Options
[*] Memory         Options
[*] SCSI-Devices(esp0) Options
[*] Network        Options
[ ] Comm.Ports     Options
[*] Graphics       Options
[*] OtherDevices   Options

Console
```



Negotiating the SunVTS TTY Interface

```
SunVTS
*****Hostname:jeeves-----Model:Sun_4_75-----SunVTS version:1.0*****
start      reset      quit      reprobe    tests      SysConf
set_options option-files log-files  connect_to intervention help
*****
                Status
      System_Status:idle
System_Passes:0   Total_Errors:0
      Elapsed_Time:000:00:00
                Status_view
                Passes Errors
Processor(s)      0      0
  fpu(fputest)    0      0
  fpu(fputest).1  0      0
Memory           0      0
  kmem(vmem)      0      0
  kmem(vmem).1    0      0

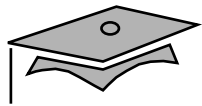
                Test_Groups
[*] Processor(s)    Options
[*] Memory          Options
[*] SCSI-Devices(esp0) Options
[*] Network         Options
[ ] Comm. Ports     Options
[*] Graphics        Options
[*] OtherDevices    Options

Console
```



Running SunVTS Remotely

- Kernel interface
- User interface
- Graphical user interface



The User Interface

- Connecting directly to the remote computer

`/opt/SUNWvts/bin/sunvts -h remote_hostname`

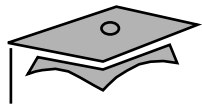
- Using the TTY interface

`/opt/SUNWvts/bin/vtstty`



Exercise: Diagnosing Problems With SunVTS

- Exercise objective
- Preparation
- Tasks
- Exercise summary



Check Your Progress

- Install the SunVTS package on a system
- Select, set up, and run SunVTS diagnostic tests
- Run SunVTS over a network
- Run SunVTS in TTY mode without a frame buffer
- Analyze SunVTS test results



Think Beyond

- How do SunVTS and POST diagnostics differ?
- Which SunSolve database documents are related to SunVTS and SyMON?



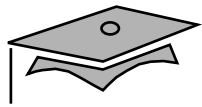
Module 7

Kernel Core Dump Analysis



Objectives

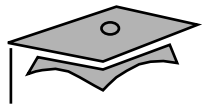
- Describe how a process creates a system dump
- Configure a system to collect and store core files
- Differentiate a system panic from a system hang
- List the steps to perform an initial core dump analysis
- Use `adb` and crash debuggers to analyze crash dumps



Introduction

System panics

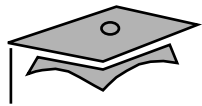
- Display panic message at the console
- Perform stack trace, listing routines that led to panic
- Dump “interesting” portions memory to swap
- Reboot, copying the core file from swap to a file system



Introduction

System hangs

- An application has hung.
- A terminal or window has hung.
- The system has hung.
- The system is overloaded and has a resource bottleneck.



Configuring a System to Collect Crash Dumps

The dumpadm utility

- `savecore` is invoked automatically by the Solaris 7 operating system.
- You can size the primary swap device and `/var/crash`.
- To view current configuration, use `dumpadm`.

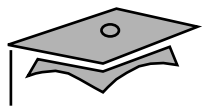
dumpadm

Dump content: kernel pages

Dump device: `/dev/dsk/c0t0d0s1` (swap)

Savecore directory: `/var/crash/killington`

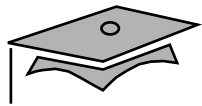
Savecore enabled: yes



The dumpadm Utility

- `-c content_type`
- `-d dump_device`
- `-m min`
- `-n`
- `-s savecore_dir`
- `-U`

```
# dumpadm -c kernel -d swap -s /opt/crash/system1
```



The savecore Utility

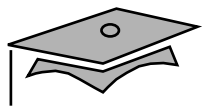
Excerpt from /etc/init.d/syssetup:

```
##  
## Default is to not do a savecore  
##  
#if [ ! -d /var/crash/`uname -n` ]  
#then mkdir -p /var/crash/`uname -n`  
#fi  
# echo 'checking for crash dump...\c '  
#savecore /var/crash/`uname -n`  
# echo ''
```



Debuggers

- adb
- crash
- kadb



Invoking the Debuggers

- adb

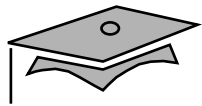
```
# cd crash_directory  
# adb -k unix.n vmcore.n
```

```
# adb -kw /dev/ksyms /dev/mem  
physmem 272a
```

```
# adb -kw -P "adb: " /dev/ksyms /dev/mem  
physmem 272a  
adb:
```

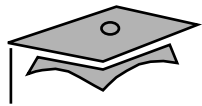
- crash

```
# crash vmcore.n unix.n  
dumpfile = vmcore.0, namelist = unix.0, outfile = stdout  
>
```



General Purpose adb Commands

Command	Function
b	Set a breakpoint
\$b	Displays all breakpoints
d	Delete a breakpoint
\$z	Delete all breakpoints
\$M	Display built-in macros
cpu\$<cpus	Display the cpu structure
cpun \$< cpu	Display the CPU identified by the number <i>n</i>
\$<msgbuf	Display the msgbuf structure
\$c	Display the stack trace
\$C	Show call trace and arguments at the time of the crash, saved fp and the saved pc



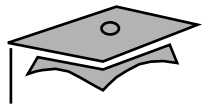
General Purpose adb Commands

Command	Function
\$r	Display the SPARC window registers
<sp\$<stacktrace	Use the sp (stack pointer) address to locate and display a detailed stacktrace
\$q	Quit
w or W	Write a 2- or 4-byte value to address
Z <i>value</i>	Write an 8-byte value to address
\$> <i>file</i>	Redirect output to <i>file</i>
>	Reset standard output to the default



Format Specification With adb

Command	Function
o or O	Display in octal (short, long)
q or Q	Display in signed octal (short, long)
d or D	Display in signed decimal
x or X	Display in hex
u or U	Display unsigned decimal
f or F	Display as floating point (long or double)
K	Print pointer or long in hex (4 bytes for 32-bit kernel; 8 bytes for 64-bit kernel)



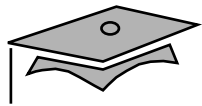
Format Specification With adb

Command	Function
b	Print as an octal byte
s	Print as a null-terminated string
c	Display as a single character
C	Display a single character using escape conventions for nonprinting characters
i	Display as a disassembled instruction (mnemonic code)
0t	Specify base ten or decimal for a value being input to adb



adb Commands

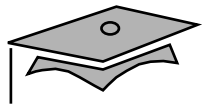
Command	Function
?	Examine code or variables in the object file
/	Examine data from the core file
=	Print values in different formats
\$<	Use as argument for miscellaneous commands, including macro invocations
>	Assign a value to a variable or register
<	Read a value from a variable or register
Return	Repeat the previous command and increment the current location



Syntax Example

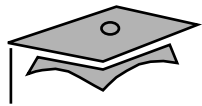
f0cebacd,10/X

- The *address* is f0cebacd. adb goes to this location in memory.
- The *count* is 10 separated by a comma from the address. The command is applied for a count of 10.
- The *command* is / (slash), which specifies to display data from the core file.
- The X specifies 4-byte hexadecimal output.



Register References in adb

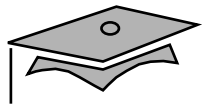
- Eight general purpose registers – %g0 through %g7
- Eight input registers – %i0 through %i7
- Eight output registers – %o0 through %o7
- Eight local registers – %l0 through %l7



Register References in adb

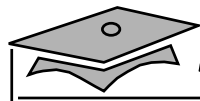
Reserved registers

- o6 – Stack pointer (sp)
- o7 – Program counter (pc)
- i6 – Frame pointer (fp)
- g0 – Always zero
- g7 – Address of the current thread



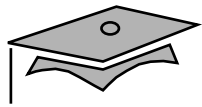
Using Macros in adb

- A macro is a file that contains adb commands.
- Most macros display commonly used structures.
- Macros are bundled with the Solaris operating system.
- Search order for the macros is
 - Current directory
 - ▼ /usr/lib/adb
- Directory specified with the -I option to adb



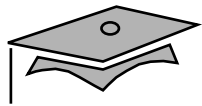
adb Macros

as	inode	proc2u	snode
bootobj	iocblk	procthread	stack
buf	iovec	procthread.list	stackregs
bufctl	itimerval	ptbltohme	stacktrace
bufctl_audit	kmem_cache	ptbltopte	stacktrace.nxt
cachevfs	ksyminfo	ptetoptbl	stat
cachevsmeta	ksiginfo	putbuf	stdata
callout	lwp	putbuf.wrap	strtab
calltrace	mblk	qinit	svcfh
calltrace.nxt	mblk.nxt	qproc.info	sysinfo
cnode	memlist	qthread.info	tcpcb
cpu	memlist.list	queue	tcpip
cpun	memlist.nxt	regs	thread
cpus	memseg	rlimit	thread.trace
cpus.nxt	mntinfo	rnode	threadlist
cred	modctl	rpctimer	threadlist.nxt
ctx	modctl_list	rwindow	tmount
dblk	modinfo	rwlock	tmpnode
devinfo	modlinkage	seg	traceall.nxt
dino	module	segdev	tsdpent
direct	modules	seglist	tsproc
disp	modules.nxt	seglist.nxt	tune
dispq	msgbuf	segmap	u
dispq.nxt	msgbuf.wrap	segn	u.sizeof
dispqtrace	mutex	sema	ucalltrace
dispqtrace.list	netbuf	session	ucalltrace.nxt
dispqtrace.nxt	page	setproc	ufchunk
dumphdr	page2hme	setproc.done	ufchunk.nxt
exdata	page2hme.nxt	setproc.nop	uio
file	pathname	setproc.nxt	ustack
filsys	pcb	sigaltstack	utsname
hat	pid	slab	v
hme	pid.print	sleepq	v_call
hme.sizeof	pid2proc	sleepq.nxt	v_proc
hmelist	pid2proc.chain	slpqtrace	vattr
hmelist.nxt	pollhead	slpqtrace.list	vfs
hmetoptbl	prgregset	slpqtrace.nxt	vnode
ifnet	proc	smap	



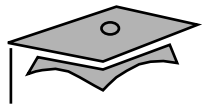
Header Files and adb Macros

- adb macros display information about C programming structures.
- C structures are defined in header files.
- Header files can be used to determine what the information is that adb macros display.



Header Files and adb Macros

- `/usr/include/sys`
- `/usr/include/vm`
- `/usr/include/sys/fs`
- `/usr/platform/arch_name/include/sys`
- `/usr/platform/arch_name/include/vm`
- `/usr/include`



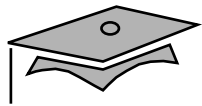
Frequently Used Header Files and adb Macros

- `/usr/include/sys/proc.h` and `proc`
- `/usr/include/sys/thread.h` and `thread`
- `/usr/include/sys/kwp.h` and `lwp`
- `/usr/include/sys/user.h` and `user`
- `/usr/include/sys/cred.h` and `cred`
- `/usr/include/vm/as.h` and `as`
- `/usr/include/vm/seg.h` and `seg`



Core Dump Analysis

- What instruction failed
- What thread was running
- What process was running
- What arguments were passed to the failing process



Core Dump Analysis – Summary for Using adb

1. Obtain a stack trace

\$c

2. Obtain a register dump with the pc address

*second_arg_to_trap***\$<regs**

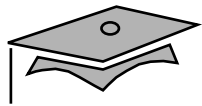
for a 32-bit machine, or

*second_arg_to_die***\$<regs**

for a 64-bit machine

3. Display the faulting instruction

*pc_address***/ai**



Core Dump Analysis – Summary for Using adb

4. Dump the `panic_thread` symbol

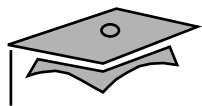
`panic_thread/X` or `panic_thread/K`

5. Run the thread macro

`panic_thread_address$<thread`

6. Display the process structure

`procp_address$<proc2u`



Core Dump Analysis Using adb

```
# adb -k unix.0 vmcore.0  
physmem 1e6e
```

```
$<msgbuf
```

```
BAD TRAP: type=9 rp=f05246f4 addr=30 mmu_fsr=326 rw=1
```

```
BAD TRAP: occurred in module "ramd" due to an illegal access to a user  
address
```

```
mkfs: Data fault
```

```
kernel read fault at addr=0x30, pme=0x0
```

```
MMU fsr=326: Invalid Address on supv data fetch at level 3
```

```
pid=465, pc=0xfc479dbc, sp=0xf0524740, psr=0x40000c2, context=0
```

```
g1-g7: ffffffff98, 0, ffffffff00, 0, f05249e0, 1, fc2dec00
```

```
Begin traceback... sp = f0524740
```

```
Called from f00df9b4, fp=f05247a8, args=1a40000 f0524808 fc38fc80 f0154664 0  
fc479d90
```

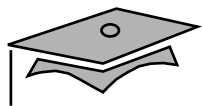
```
Called from f0070258, fp=f05248b8, args=200 f0524920 2 0 4 fc2d5b04
```

```
Called from f0041aa0, fp=f0524938, args=f0160cf8 f0524eb4 0 f0524e90 ffffffff  
ffffffff
```

```
Called from 15cc0, fp=effffae8, args=4 32400 200 0 0 3fe00
```

```
End traceback...
```

```
panic: Data fault
```



Core Dump Analysis Using adb

\$c

complete_panic(0xf026b428, 0xfbfbab98c, 0xf0048ec8, 0x6a, 0xfbfbab818, 0xf0279800) + 108

do_panic(?) + 1c

vcmn_err(0xf0266600, 0xfbfbab98c, 0xfbfbab98c, 0x7, 0xffeec000, 0x3)

cmn_err(0x3, 0xf0266600, 0x1, 0x21, 0x21, 0xf025c000) + 1c

die(0x9, **0xf05246f4**, 0x30, 0x326, 0x1, 0xf0266600) + bc

trap(0xf028a1d8, **0xf05246f4**, 0x0, 0x326, 0x1, 0x0) + 4f8

fault(?) + 84

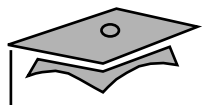
Syssize(via getminor)(0x0, 0x3ffff, 0x20, 0x7fffffff, 0xf5c4b4bc, 0x31585486)

ram_write(0xdc0000, 0xfbfbabbd8, 0xf5a8ed38, 0xdc, 0xf5970d48, 0xf5c54d90) + 1c

write(0x5) + 190

f05246f4\$<regs

0xfc020ac4:	psr	pc	npc	
	110000c4	fv479dbc	f5c98dc0	
0xfc020ad0:	y	g1	g2	g3
	50000000	ffffff98	0	f00bb080
0xfc020ae0:	g4	g5	g6	g7
	40	f5ca0648	1	fc2dec00
0xfc020af0:	o0	o1	o2	o3



Core Dump Analysis Using adb

fc479dbc/ai

ram_write+0x2c: ld [%l1 + 0x30], %l2

panic_thread/X (On a 64-bit system, use /K instead of /X)

panic_thread:

panic_thread: f5c66480

f5c66480\$<thread

link	stk			
	0	fbfab08		
0xf5c6648c:				
	bound	affcnt	bind_cpu	
	f026b494	0	-1	
0xf5c66494:				
	flag	procflag	schedflag	state
	0	0	11	4
0xf5c664a0:	pri	epri	pc	sp
	14	0	f0048ec8	fbfab818
0xf5c664ac:	wchan0	wchan	cid	clfuncs
	0	0	2	f59a0378
0xf5c664d0:				
	nofault	swap	lock	cpu
	0	fbfaa000	ff	f026b494
.....				
0xf5c66524:				
	lwp	procp	next	prev
	f5c11828	f5c0fcc8	f5c665a0	f5c66d80



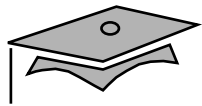
Core Dump Analysis Using adb

f5c0fcc8\$<proc2u

execid	execsz	tsize			
	32581	12e	0		
0xf5c0fe8c:					
	dsize	start	ticks	cv	
	0	31585486	405a4	0	
0xf5c0fe9c:					
	exdata				
0xf5c0fe9c:					
	vp	tsize	dsize	bsize	
	0	0	0	0	
0xf5c0feac:					
	lsize	nshlibs	mach	mag	toffset
	0	0	0	10b	0
0xf5c0febc:					
	doffset	loffset	txtorg	datorg	
	0	0	0	0	
0xf5c0fecc:	entloc ef7d43a8				
0xf5c0fed0:	aux vector				
	7d8	effffffe6	3	10034	

```
0xf5c0ff60:      psargs mkfs /devices/pseudo/ramd@0:0,raw 512 8 1 8192 1024 16 10 60 204 8 t
0 -1 8 -1^@^@^@
```

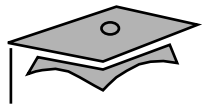
mkfs^@^@^@^@^@^@^@^@^@



Information Gathering – Summary for Using strings

strings vmcore.0 | more

- The panic message
- The program counter address (pc)
- The register pointer address (rp)
- The address of g7
- The cpu id number
- The stack pointer (sp) address



Core Dump Analysis Using strings

```
# strings vmcore.0 | more
```

```
BAD TRAP: type=9 rp=f05246f4 addr=30 mmu_fsr=326 rw=1
```

```
BAD TRAP: occurred in module "ramd" due to an illegal access to a user  
address
```

```
mkfs: Data fault
```

```
kernel read fault at addr=0x30, pme=0x0
```

```
MMU sfsr=326: Invalid Address on supv data fetch at level 3
```

```
pid=465, pc=0xfc479dbc, sp=0xf0524740, psr=0x40000c2, context=0
```

```
g1-g7: ffffffff98, 0, ffffffff00, 0, f05249e0, 1, fc2dec00
```

```
Begin traceback... sp = f0524740
```

```
Called from f00df9b4, fp=f05247a8, args=1a40000 f0524808 fc38fc80 f0154664 0  
fc479d90
```

```
Called from f0070258, fp=f05248b8, args=200 f0524920 2 0 4 fc2d5b04
```

```
Called from f0041aa0, fp=f0524938, args=f0160cf8 f0524eb4 0 f0524e90 ffffffff  
ffffffff
```

```
Called from 15cc0, fp=effffae8, args=4 32400 200 0 0 3fe00
```

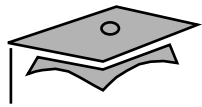
```
End traceback...
```

```
panic: Data fault
```



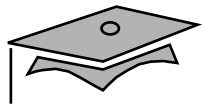
Commonly Used crash Commands

- user
- stat
- proc
- cpu
- stack



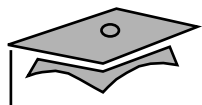
Commonly Used crash Commands

- defthread
- defproc
- file
- inode
- kmastat
- *help command*



Core Dump Analysis – Summary for Using crash

1. Get a status listing with the `stat` command.
2. List the current process with the `u(ser)` command.
3. Get a process table listing with the `proc` command.
4. Use `defthread` and `defproc` to obtain the addresses of the current thread and processes.
5. Enter a question mark (?) any time for help.



Core Dump Analysis Using crash

```
# crash vmcore.0 unix.0
dumpfile = vmcore.0, namelist = unix.0, outfile = stdout
```

```
> stat
```

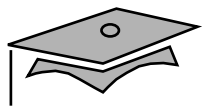
```
system name:    SunOS
release:        5.7
node name:       mustang
version:         Generic
machine name:    sun4m
time of crash:   Fri Nov 29 09:04:19 1998
age of system:   18 min.
panicstr:        Data fault
panic registers:
                pc: f004c13c      sp: f0922808
```

```
> u
```

```
PER PROCESS USER AREA FOR PROCESS 34
```

```
PROCESS MISC:
```

```
    command: mkfs, psargs: mkfs /devices/pseudo/ramd@0:0,raw 512 8 1 8192 1024 16 10 60
2048 t 0 -1 8 -1
    start: Fri Nov 29 09:04:19 1998
    mem: 8a, type: exec
    vnode of current directory: f0bd8688
```



Core Dump Analysis Using crash

OPEN FILES, POFILE FLAGS, AND THREAD REFCNT:

```
[0]: F 0xf06d3db8, 0, 0 [1]: F 0xf06d3db8, 0, 0
[2]: F 0xf06d3db8, 0, 0 [3]: F 0xf06d3938, 0, 0
[4]: F 0xf06d3ae8, 0, 0 [5]: F 0xf06d32a8, 0, 0
[6]: F 0xf06d38d8, 0, 0 [9]: F 0xf06d3878, 0, 0
[10]: F 0xf06d3848, 0, 0
```

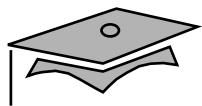
cmask: 0022

RESOURCE LIMITS:

```
cpu time: unlimited/unlimited
file size: unlimited/unlimited
swap size: 2147479552/2147479552
stack size: 8388608/2147479552
coredump size: unlimited/unlimited
file descriptors: 64/1024
address space: unlimited/unlimited
```

SIGNAL DISPOSITION:

```
1: default  2: default  3: default  4: default
5: default  6: default  7: default  8: default
9: default 10: default 11: default 12: default
13: default 14: default 15: default 16: default
17: default 18: default 19: default 20: default
```



Core Dump Analysis Using crash

> **proc**

PROC TABLE SIZE = 4058

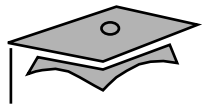
SLOT	ST	PID	PPID	PGID	SID	UID	PRI	NAME	FLAGS
0	t	0	0	0	0	0	96	sched	load sys lock
1	s	1	0	0	0	0	58	init	load
2	s	2	0	0	0	0	98	pageout	load sys lock nowait
3	s	3	0	0	0	0	60	fsflush	load sys lock nowait
24	s	273	269	254	254	0	38	xinit	load
25	s	274	273	274	254	0	59	Xsun	load
26	s	275	273	275	254	0	55	sh	load
27	s	280	1	275	254	0	59	fbconsole	load
28	s	361	318	318	318	0	44	newfs	load jctl
29	s	286	1	275	254	0	59	vkbd	load
30	s	291	147	147	147	0	0	rpc.ttdbserver	load jctl
31	s	289	1	275	254	0	59	ttsession	load jctl
32	s	293	275	275	254	0	59	olwm	load
33	s	294	293	275	254	0	10	olwmslave	load
34	p	363	362	318	318	0	0	mkfs	load
35	s	298	1	298	298	0	59	cmdtool	load jctl

> **defproc**

Procslot=34

> **defthread**

Current Thread = 70632160



The ISCDA Document

The script `iscda` can be run to automatically provide some useful information after a system crash.

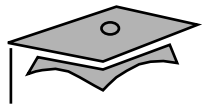
Sample usage

```
# cd /var/crash/mymachine  
# iscda unix.0 vmcore.0 > /tmp/iscda.output
```



Exercise: Performing Kernel Core Dump Analysis

- Exercise objective
- Preparation
- Tasks
- Exercise summary



Check Your Progress

- Describe how a process creates a system dump
- Configure a system to collect and store core files
- Differentiate a system panic condition from a system hang
- List the steps to perform an initial core dump analysis
- Use the `adb` and crash debuggers to analyze crash dumps

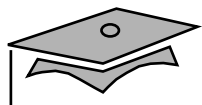


Think Beyond

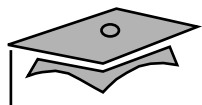
- Now that you have some exposure to analyzing kernel core dumps and have used the various debuggers, how do you correct problems identified?

Course Contents

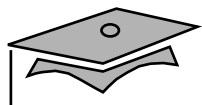
About This Course	Preface-1
Course Goal	Preface-2
Course Map	Preface-3
Module-by-Module Overview	Preface-4
Course Objectives	Preface-5
Skills Gained by Module	Preface-7
Guidelines for Module Pacing	Preface-8
Topics Not Covered	Preface-9
How Prepared Are You?	Preface-11
Introductions	Preface-12
How to Use Course Materials	Preface-13
How to Use the Icons	Preface-14
Typographical Conventions and Symbols	Preface-15
Attention VARs	Preface-16
 Fault Analysis and Diagnosis	 1-1
Objectives	1-2
The Fault Analysis Method	1-3
Eight Steps of Fault Analysis and Diagnosis	1-4
Stating the Problem	1-6
Describing the Problem	1-7
Listing All Observed Facts	1-8
Establishing Comparative Facts	1-10
Identifying Differences	1-11
Listing Relevant Changes	1-12
Generating Likely Causes	1-13



Testing Likely Causes	1-14
Verification	1-15
Verifying the Most Likely Cause	1-16
Taking Corrective Action	1-17
Fault Analysis Worksheet Example	1-18
Exercise: Fault Analysis and Diagnosis	1-20
Check Your Progress	1-21
Think Beyond	1-22
 Diagnostic Tools	2-1
Objectives	2-2
Introduction to Hardware and Software Errors	2-4
Types of Errors	2-5
Error Reporting Mechanisms	2-6
Differentiating Hardware and Software Faults	2-7
Diagnostic Commands and Tools	2-8
Open Discussion	2-14
Diagnostic Files	2-15
Device Management Utilities	2-20
The modload Utility	2-23
The modunload Utility	2-24
The ifconfig Utility	2-25
The netstat Command	2-27
Debugging Utilities	2-28
Error-Free truss Command Output	2-29
Interpreting truss Command Output	2-30
Example of truss Command Output With Errors	2-32
Interpreting Errors in truss Command Output	2-33
The errno.h Header File	2-35
Exercise: Using Solaris Troubleshooting Tools	2-36

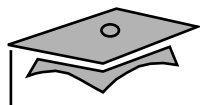


Check Your Progress	2-37
Think Beyond	2-39
POST Diagnostics	3-1
Objectives	3-2
Diagnostic Overview	3-3
Boot PROM POST	3-4
SunVTS Diagnostics	3-5
POST Viewing Methods	3-6
Viewing POST Using tip hardware	3-8
UltraSPARC Example Using tip	3-9
SPARCstation 5 POST Example Using tip	3-14
Post Diagnostic Workshop Using tip	3-16
POST tip Commands	3-17
Exercise: Using tip to Observe POST Diagnostics	3-18
Check Your Progress	3-19
Think Beyond	3-20
OBP Diagnostics and Commands	4-1
Objectives	4-2
Ultra 5 and Ultra 10 Architecture	4-3
Functions and Capabilities of OBP	4-4
OBP Features	4-5
Open Boot PROM	4-6
NVRAM Contents – Ultra 5	4-7
The eeprom Command	4-10
Changing PROM Variables With eeprom	4-11
Events During Power On	4-12
Default Boot Sequence	4-13
The rc Files and Directories	4-16



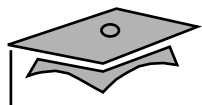
Device Tree Hierarchy	4-19
OBP Device Tree Navigation – UltraSPARC Workstation	4-20
OBP Device Tree Navigation – SPARCstation 1000	4-23
OBP Features – Ultra Enterprise Servers	4-24
NVRAM Parameters – Ultra Enterprise	4-25
Running OBP Commands With Key Sequences	4-26
Flash PROM Update for Sun4u Architectures	4-27
Ultra Workstation OBP Command Examples	4-29
SPARCstation 20 OBP Command Examples	4-31
Exercise: Working With Ultra and SPARC 5 Workstations OBP Utilities	4-33
Check Your Progress	4-34
Think Beyond	4-35

SunSolve Database Information	5-1
Objectives	5-2
Distribution	5-4
SunSolve Online Account	5-5
Installing SunSolve Software	5-6
Configuring a SunSolve Server	5-9
Running SunSolve Software	5-11
The SunSolve Online Home Page Selections	5-12
Power Search Tool Overview	5-14
Power Search Collections	5-15
Defining Search Criteria	5-16
Power Search Results	5-17
Boolean Search Syntax	5-18
The Current Patch Report	5-19
Patch Diag Tool	5-21
The Patch Diag Report	5-22
Installing Patches	5-24

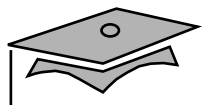


Removing a Patch	5-26
Useful SunSolve Documents	5-27
Other Useful SunSolve Online Documents	5-29
Exercise: Using SunSolve Fault Analysis	5-30
Check Your Progress	5-31
Think Beyond	5-32

SunVTS System Diagnostics	6-1
Objectives	6-2
Introduction	6-3
Hardware and Software Requirements	6-4
The SunVTS Architecture	6-5
Interfaces	6-6
Installing SunVTS Software	6-7
Starting the SunVTS Software	6-8
The SunVTS GI	6-9
The SunVTS Window Panels	6-10
The SunVTS Window Icons	6-11
The SunVTS Menu Selections	6-12
The Schedule Options Menu	6-13
The Test Execution Menu	6-14
The Advanced Options Menu	6-16
Intervention Mode	6-17
Performance Meter Panel	6-18
Using SunVTS in TTY Mode	6-20
Negotiating the SunVTS TTY Interface	6-21
Running SunVTS Remotely	6-22
The User Interface	6-23



Exercise: Diagnosing Problems With SunVTS	6-24
Check Your Progress	6-25
Think Beyond	6-26
 Kernel Core Dump Analysis	7-1
Objectives	7-2
Introduction	7-3
Configuring a System to Collect Crash Dumps	7-5
The dumpadm Utility	7-6
The savecore Utility	7-7
Debuggers	7-8
Invoking the Debuggers	7-9
General Purpose adb Commands	7-10
Format Specification With adb	7-12
adb Commands	7-14
Syntax Example	7-15
Register References in adb	7-16
Using Macros in adb	7-18
adb Macros	7-19
Header Files and adb Macros	7-20
Frequently Used Header Files and adb Macros	7-22
Core Dump Analysis	7-23
Core Dump Analysis – Summary for Using adb	7-24
Core Dump Analysis Using adb	7-26
Information Gathering – Summary for Using strings	7-30
Core Dump Analysis Using strings	7-31
Commonly Used crash Commands	7-32
Core Dump Analysis – Summary for Using crash	7-34
Core Dump Analysis Using crash	7-35
The ISCDA Document	7-38



Exercise: Performing Kernel Core Dump Analysis	7-39
Check Your Progress	7-40
Think Beyond	7-41

Copyright 1999 Sun Microsystems Inc., 901 San Antonio Road, Palo Alto, California 94303, Etats-Unis. Tous droits réservés.

Ce produit ou document est protégé par un copyright et distribué avec des licences qui en restreignent l'utilisation, la copie, la distribution, et la décompilation. Aucune partie de ce produit ou document ne peut être reproduite sous aucune forme, par quelque moyen que ce soit, sans l'autorisation préalable et écrite de Sun et de ses bailleurs de licence, s'il y en a.

Le logiciel détenu par des tiers, et qui comprend la technologie relative aux polices de caractères, est protégé par un copyright et licencié par des fournisseurs de Sun. Des parties de ce produit pourront être dérivées du système Berkeley 4.3 BSD licenciés par l'Université de Californie. UNIX est une marque déposée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company Ltd.

Sun, Sun Microsystems, le logo Sun, Solaris, OpenBoot, Ultra, SunSolve, SunVTS, AnswerBook, NFS, SyMON, UltraSPARC, Ultra Enterprise, SunDiag, et SPARCstorage sont des marques de fabrique ou des marques déposées de Sun Microsystems, Inc. aux Etats-Unis et dans d'autres pays.

Toutes les marques SPARC sont utilisées sous licence sont des marques de fabrique ou des marques déposées de SPARC International, Inc. aux Etats-Unis et dans d'autres pays.

Les produits portant les marques SPARC sont basés sur une architecture développée par Sun Microsystems, Inc.

UNIX est une marque déposée aux Etats-Unis et dans d'autres pays et licenciée exclusivement par X/Open Company, Ltd.

L'interface d'utilisation graphique OPEN LOOK et Sun™ a été développée par Sun Microsystems, Inc. pour ses utilisateurs et licenciés. Sun reconnaît les efforts de pionniers de Xerox pour la recherche et le développement du concept des interfaces d'utilisation visuelle ou graphique pour l'industrie de l'informatique. Sun détient une licence non exclusive de Xerox sur l'interface d'utilisation graphique Xerox, cette licence couvrant également les licenciés de Sun qui mettent en place l'interface d'utilisation graphique OPEN LOOK et qui en outre se conforment aux licences écrites de Sun.

L'accord du gouvernement américain est requis avant l'exportation du produit.

Le système X Window est un produit de X Consortium, Inc.

LA DOCUMENTATION EST FOURNIE "EN L'ETAT" ET TOUTES AUTRES CONDITIONS, DECLARATIONS ET GARANTIES EXPRESSES OU TACITES SONT FORMELLEMENT EXCLUES, DANS LA MESURE AUTORISEE PAR LA LOI APPLICABLE, Y COMPRIS NOTAMMENT TOUTE GARANTIE IMPLICITE RELATIVE A LA QUALITE MARCHANDE, A L'APTITUDE A UNE UTILISATION PARTICULIERE OU A L'ABSENCE DE CONTREFAÇON.