EXTERNAL SPECIFICATION
(User Perspective)

TITLE : 120 / 170 System Diagnostic External Specification

AUTHOR : Sunny Kirsten

REPORT NO. :

REVISION NO. : @(#) sysdiag.txt 1.27 3/21/85

DATE : April 5, 1985

STATUS : Preliminary

APPROVALS :	DATE
Originator	----------
	Sunny Kirsten
Test Engineering	----------
Manufacturing	----------
Production	----------
Quality Control	----------

Document Review Form

Please make note and initial on this page all corrections and/or proposed amendments by page number and/or section number.

Recommendations, Differences, Construction Errors, and comments:

Typographical Errors:

Attach additional sheet(s) as needed.

TABLE OF CONTENTS

120 / 170 System Diagnostic EXTERNAL SPECIFICATION

1. INTRODUCTION

1.1. Purpose

This document details the structure and operation of the Sun 2 System Diagnostic *sysdiag*, which performs system level test on a Sun workstation, as required for burn-in prior to shipment, and for field-service or customer level diagnostics.

1.2. Applicable Documents

All documents about the UNIX system, and all hardware reference manuals for the hardware included in the system configuration will apply.

1.2.1. Ethernet boards

1.2.1.1. 3Com Ethernet board

1.2.1.2. Sun Ethernet board

1.2.2. TapeMaster board

1.2.3. Xylogics board

1.2.4. Sky Fast Floating Point board

The Sun document *Sky diagnostic* **ffpusr** *External Specification* and the test procedure *Sky FFP Test Procedure* give details about the Sky board diagnostic *ffpusr* and test procedure for this board. The Sun document 800-1104-01 describes the Sky board, its installation, and tells something of the Sky version of *ffpusr*, and associated software. This document is not needed to operate the program, though it would be useful if you wanted to decipher hardware bit patterns mentioned in error messages, to isolate problems within the Sky board. This is rarely done at Sun where we swap bad boards back to Sky for repairs.

1.3. Definitional Conventions

1.3.1. Notations

Program names and document names are usually shown in itallics. Thus command names are also shown usually in itallics, since most commands invoke programs by the same name. Bold is used for a variety of contextual emphasis purposes.

As in the C language, the prefix '0x' is used when a number is in hexadecimal (base 16) format. The notation 'nnnnn' is used when indicating numeric values which vary.

1.3.2. Syntax When dealing with the syntax of terminal input, it is conventional to use the term **key-in** to mean that the operator should type exactly the characters/words/phrases specified. The term **enter** implies that the operator should press the **carriage return** key after **keying-in** the specified data.

1.3.3. Terminology

2. SYSTEM OVERVIEW

2.1. General Description

The System Diagnostic operates on top of the UNIX environment. It consists of test scripts and programs to exercise the various hardware components, and shell scripts which provide a multi-windowing interface where different windows are provided for each major diagnostic function. The test automatically adapts to test the UNIX configuration in **/dev**. The system is configured to default to using **/dev/console** as the system console, a Sun Workstation, and to operate in the multi-windowing mode. It may be manually configured to use a standard alphanumeric terminal in non-windowed mode, attached to **port A**, which conforms to the **RS423** standard.

2.2. Features

2.2.1. User Interface

2.2.1.1. Operation on an alphanumeric terminal

sysdiag intermixes all the output from all the windows normally displayed on the Sun workstation, on a line by line basis, on the standard alphanumeric terminal. Error messages are tagged by the generating program to prevent ambiguity when the messages appear all in one "window". The command *killme* issued at the console will terminate *sysdiag* when operated in the non-windowing mode. The command *setterm* issued by root on any tty will set the terminal type of the console.

2.2.1.2. Operation on a Sun workstation

Sysdiag is partitioned into three active test windows. In addition, it also displays windows for date/time, the performance monitor graphs, and a CONSOLE window which displays UNIX system error messages. Most commonly, the following messages will appear on the CONSOLE window:

NOTICE: Window display lock broken after time limit was exceeded by process n
WARNING: You may see display garbage because of this action

Note that the above messages are **normal** on a system which is loaded as heavily as *sysdiag* loads it.

The three active test windows are dedicated as follows:

2.2.1.2.1. Disk test

In the upper left corner we run a disk diagnostic *disk* which constructs random disk data images in memory, exercises the disk via file system I/O, and compares two the two identical files it wrote as it reads them back from the disk. *disk* uses new memory routines, and stops on errors for post-mortems instead of filling the file system with error logs.

2.2.1.2.2. Memory test

In the window at the bottom left corner are run two memory exercisers.

2.2.1.2.2.1. pmem uses a virtual page which it remaps repeatedly thru memory, in order to read all physical memory, instead of allocating a huge block of physical memory, in order to leave some space for UNIX to continue to function.

2.2.1.2.2.2. vmem allocates, writes, and reads a 2MB array of **virtual** memory.

2.2.1.2.3. Peripherals tests

In the lower right corner we run the peripherals tests. Sysdiag automatically configures itself by probing via *devtop* for all xylogics and scsi disks, and for all scsi, 1/2 in., and archive tape controllers, and for a sky board. For each such device it first checks to see that if it's a tape drive, that the tape drive is online, and rewound, then it automatically tests (via *dev*), with a generic read/write diagnostic *devtest*, which uses the biggest allowable block size for most of a file transfer, followed by some small blocks for the balance of the file. In the case of the Sky board, *dev* calls *ffpusr*. Note that the **SCSI tape** and the **archive tape** are only tested every 5th pass, in order to limit the accumulated run time during the burn-in period, to less than 8 hours (the head cleaning period).

2.2.1.2.3.1. devtop finds the hardware configuration by looking in **/dev**, which it passes to:

2.2.1.2.3.2. dev tests each of the peripherals found by *devtop* by calling:

2.2.1.2.3.3. devtest which reads or writes/reads a device for a given number of blocks (used in place of dd and tar in the device tests).

2.2.1.2.3.4. ffpusr which tests the Sky Fast Floating Point processor board.

2.2.2. Error Logs

Sysdiag produces a variety of error logs. In the case of the Sky board, an execution log is produced, named "logsky". In the case of the peripherals tests, error logs are created if required by each test. In the case of the disk test and the memory tests, each produces it's own log. The error logs are all created in the sysdiag directory, and all have a name which begins with *log*. Most end with a number to uniquely identify them, which is particularly useful in case of multiple invocations of *sysdiag*. When *sysdiag* is terminated, it automatically reviews the logs for the operator, via the "more log*" command. This same command may be manually entered by the operator, at any time, in the CONSOLE window, during the execution of *sysdiag*, to observe the current status of tests.

2.3. Required Configuration

sysdiag will automatically adapt to test different configurations of peripherals, including various combinations of disk and tape, and the Sky board. *sysdiag* itself is dependent on the hardware configuration for which UNIX is built. Thus it is very important that the UNIX system be built properly for the specific hardware configuration to be tested correctly. Specifically, raw /dev files should reflect the devices attached.

NOTE: At the current time, the hardware combination of **tapemaster** board and **SCSI** board does not work in UNIX software, because the **SCSI** controller eats **tapemaster** command blocks.

2.4. Error Handling

General class error messages are displayed in the appropriate window on the system console, while detailed error messages are logged to disk files for later review by the operator. Error logs may be reviewed during *sysdiag* operation, and are automatically displayed to the operator when the test is terminated.

2.5. General Performance Characteristics

sysdiag tends to load the UNIX system rather heavily. It is CPU, I/O, memory, disk, and swap intensive. On a one megabyte system, UNIX will thrash tediously. On a two megabyte system, some reasonable performance levels will be achieved, so that the system appears to be responsive. Mouse and keyboard response may seem to be absent, when in actuality the response is just very slow.

2.6. Planned Extensions

The sky board diagnostic will be rewritten to be more thorough. Add a serial-port self-loopback test. Modify *devtop* to test for the existance of devices by testing for response from the hardware status registers, rather than believing the configuration in **/dev**.

2.7. Limitations

sysdiag does not test the serial ports, which it could be doing. It would seem to be a good idea to add a self-loopback test for each port, and the test could be disabled for **serial port A** when it was in use as the system CONSOLE when operating in the non-windowed mode.

sysdiag does not handle the color board, which is tested separately.

It takes a while to terminate *sysdiag* given the slowness of keyboard / mouse response.

3. 120 / 170 System Diagnostic SPECIFICATION

3.1. User Interface

The user interface for sysdiag is based on multiple windows. Different windows display and control various tests. In the case of the alphanumeric terminal version, there is the equivalent of only one window, in which all messages are intermixed, in time order.

3.2. Input/Output

There are no program paramaters at the top level of sysdiag, so user input is usually restricted to the functions of terminating the test, and reviewing the error logs. The individual tests within sysdiag may be used independently by calling them with the appropriate paramaters, described below. Since the SunWindows environment is used, a mouse is required, in order to direct keyboard input to the appropriate window. There is one error condition which requires operator intervention and interaction with the **peripherals test** window.

3.2.1. Setting the Terminal Type

3.2.2. Use of the Mouse

3.2.3. Termination of Sysdiag

3.2.3.1. killme to terminate terminal version

3.2.3.2. ^C to terminate windows version

3.2.4. Responding to *devtop* on mag tape errors

3.3. Operation

To use *sysdiag*, the first step is to configure the operator's terminal, if it is other than a Sun workstation. To configure another terminal as the console for sysdiag, login as root and use

the *setterm* command to set your terminal type.

Log-in as the user named sysdiag. The login procedure for sysdiag will display the system's idea of the current date and time, requesting verification via a simple carriage-return, or update in the standard format of the unix date command. Once the date has been verified, the SunWindows environment is initiated, and various tests initiated in the corresponding windows.

3.3.1. User Interface

3.3.1.1. Operation on an alphanumeric terminal

To operate *sysdiag* from an alphanumeric terminal, you will first need to login as root and use the *setterm* command to set the terminal type of the system console, to other than a Sun workstation. Once this is accomplished, then you may proceed as follows.

3.3.1.2. Operation on a Sun workstation

To operate *sysdiag* you will need to login as the user **sysdiag**. The systems version of the current date and time is displayed, and if it is correct, simply key-in a carriage return. Otherwise enter the date/time in YYMMDDHHMM[.SS] format.

3.3.1.2.1. Disk test

Typical startup messages appearing in this window are:

```
Disk REV 1.3 5/21/84 starting
Wed May 23 16:18:31 1984
Pass 1
Pass 2
Pass 3
.....
```

3.3.1.2.2. Memory test

```
Starting mem
starting scanner
[1] 104
Started scanner Wed May 23 16:18:37 1984
scanner: started with 0x3e0000
bytes to check Wed May 23 16:18:39 1984
scanner: pass 1 errors 0
scanner: pass 2 errors 0
scanner: pass 3 errors 0
vmem: testing 0x200000 bytes.
scanner: pass 4 errors 0
scanner: pass 5 errors 0
scanner: pass 6 errors 0
scanner: pass 7 errors 0
scanner: pass 8 errors 0
vmem: Written
scanner: pass 9 errors 0
scanner: pass 10 errors 0
scanner: pass 11 errors 0
scanner: pass 12 errors 0
scanner: pass 13 errors 0
vmem: Read
scanner: pass 14 errors 0
```

```
scanner: pass 15 errors 0
scanner: pass 16 errors 0
scanner: pass 17 errors 0
scanner: pass 18 errors 0
Pass 1, no errors
```

3.3.1.2.2.1. pmem

3.3.1.2.2.2. vmem

3.3.1.2.3. Peripherals tests

Typical startup messages for the peripherals test window are:

3.3.1.2.3.1. devtop

3.3.1.2.3.1. If no peripherals attached

```
Probing ..
So why bother me at all?
```

3.3.1.2.3.1. If all peripherals attached

```
Probing .. xy0c st0 mt0 sky
Thu May 24 16:00:00 PDT 1984 Starting testing of xy0c st0 mt0 sky
769120 blocks to do on xy0c
84150 blocks to do on sd0c
Testing sky
end of pass 1
769120 blocks to do on xy0c
84150 blocks to do on sd0c
Testing sky
end of pass 2
769120 blocks to do on xy0c
84150 blocks to do on sd0c
Testing sky
end of pass 3
769120 blocks to do on xy0c
84150 blocks to do on sd0c
Testing sky
end of pass 4
769120 blocks to do on xy0c
84150 blocks to do on sd0c
Testing st0
Testing sky
end of pass 5
```

3.3.1.2.3.2. dev

Disk drives are tested in read-only mode. Tape drives are tested in write/ read mode.

3.3.1.2.3.3. devtest

3.3.1.2.3.4. ffpusr

3.4. Error Handling

When massive quantities of errors occur in a given subsystem, it's usually pretty easy to determine that a specific subassembly needs to be replaced. Most of the subsystems are prone to transient errors which require a good judgement call to determine their severity. Experienced floor support personell are required in these cases.

3.4.1. Error Messages (displayed on console)

3.4.2. Error Logs (written to disk files)

3.4.2.1. Error Log contents

3.4.2.1.1. sysdiag *logtimes*$$

Thu May 24 16:00:00 PDT 1984 window version started
Thu May 24 18:00:00 PDT 1984 window version stopped

3.4.2.1.2. disk *logdisk*$$

Disk REV 1.3 5/21/84 starting Thu May 24 16:00:00 1984
disk: ending pass 189 Thu May 24 16:00:00 1984

3.4.2.1.3. pmem *logpmem*$$

Started scanner Thu May 24 16:00:00 1984
scanner: started with 0x100000 bytes to check Thu May 24 16:00:00 1984
errors 0 stopping at pass 9 SIGINT Thu May 24 16:00:00 1984
errors 0 stopping at pass 9 SIGHUP Thu May 24 16:00:00 1984

3.4.2.1.4. vmem *logmem.*$$

starting scanner
Thu May 24 18:00:00 1984 mem stopped pass 100

3.4.2.1.5. devtop *logdevtop*$$

There is normally no log produced by *devtest.*

3.4.2.1.6. dev *logdev*$$

Thu May 24 16:00:00 PDT 1984 Starting testing of sd0C
Thu May 24 16:00:00 PDT 1984 Test stopped on pass 999

3.4.2.1.7. devtest *logdevtest*$$

There is normally no log produced by *devtest.*

3.4.2.1.8. ffpusr *logsky*

3.5. Performance

Performance is everything.

APPENDIX

4. APPENDIX SECTION

4.1. Files comprising *sysdiag*

INDEX

5. INDEX SECTION