

**ULTRIX-32
Guide to
System Exercisers**

Order No. AA-ME96A-TE

ULTRIX-32 Operating System, Version 3.0

Digital Equipment Corporation

Copyright © 1987, 1988 Digital Equipment Corporation
All Rights Reserved.

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may be used or copied only in accordance with the terms of such license.

No responsibility is assumed for the use or reliability of software on equipment that is not supplied by DIGITAL or its affiliated companies.

The following are trademarks of Digital Equipment Corporation:

DEC	Q-bus	VAX
DECnet	RT	VAXstation
DECUS	ULTRIX	VMS
MASSBUS	ULTRIX-11	VT
MicroVAX	ULTRIX-32	ULTRIX Worksystem Software
PDP	UNIBUS	

UNIX is a registered trademark of AT&T in the USA and other countries.

IBM is a registered trademark of International Business Machines Corporation.

MICOM is a registered trademark of Micom System, Inc.

This manual was written and produced by the ULTRIX Documentation Group in Nashua, New Hampshire.

Contents

About This Manual

Audience	v
Organization	v
Related Documents	vi
Conventions	vi

1 Testing Your System with Exercisers

1.1 Running System Exercisers	1-2
1.2 Exercising More Than One Part of the System	1-2
1.3 Using Exerciser Diagnostics	1-3

2 Exercising File Systems and Memory

2.1 Exercising the Local File System	2-1
2.2 Exercising the System Memory	2-2
2.3 Exercising Shared Memory	2-3

3 Exercising Peripherals

3.1 Exercising the Disk Drives	3-1
--------------------------------------	-----

3.2 Exercising the Tape Drives 3-2

3.3 Exercising the Lineprinters 3-4

4 Exercising Network and Communications Systems

4.1 Exercising the TCP/IP Network 4-1

4.2 Exercising the Terminal Communication System 4-2

Index

About This Manual

The objective of this guide is to provide you with information on system exercisers. It describes how to run system exercisers and explains the exercisers you use for file systems and memory, peripherals, and network and communications.

Audience

The Guide to System Exercisers is written for the person responsible for managing and maintaining an ULTRIX operating system. It assumes that this individual is familiar with ULTRIX commands, the system configuration, and the system's controller/drive unit number assignments. You do not need to be a programmer to use this guide; however, you should be familiar with an ULTRIX editor, such as `vi` or `ed`. You should also be aware of the naming conventions used in this book.

Organization

This manual consists of four chapters and an index. The chapters are:

- Chapter 1: Running System Exercisers
Provides an overview of system exercisers. It describes how to run system exercisers, how to exercise more than one part of the system simultaneously, and exerciser diagnostics.
- Chapter 2: Exercising File Systems and Memory
Describes the exercisers that test file systems, system memory, and shared memory.
- Chapter 3: Exercising Peripherals
Describes the exercisers that test disk drives, tape drives, and line printers.
- Chapter 4: Exercising Network and Communications Systems
Describes the exercisers that test TCP/IP and terminal communications.

Related Documents

This document should be used in conjunction with the ULTRIX Reference Pages.

Conventions

The following conventions are used in this manual:

special	In text, each mention of a specific command, option, partition, pathname, directory, or file is presented in this type.
command(x)	In text, cross-references to the command documentation include the section number in the reference manual where the commands are documented. For example: See the cat(1) command. This indicates that you can find the material on the cat command in Section 1 of the reference pages.
literal	In syntax descriptions, this type indicates terms that are constant and must be typed just as they are presented.
<i>italics</i>	In syntax descriptions, this type indicates terms that are variable.
[]	In syntax descriptions, square brackets indicate terms that are optional.
. . .	In syntax descriptions, a horizontal ellipsis indicates that the preceding item can be repeated one or more times.
function	In function definitions, the function itself is shown in this type. The function arguments are shown in italics.
UPPERCASE	The ULTRIX system differentiates between lowercase and uppercase characters. Enter uppercase characters only where specifically indicated by an example or a syntax line.
example	In examples, computer output text is printed in this type.
example	In examples, user input is printed in this bold type.

% This is the default user prompt in multiuser mode.

This is the default superuser prompt.

>>> This is the console subsystem prompt.

. In examples, a vertical ellipsis indicates that not all of the
. lines of the example are shown.
.

<KEYNAME> In examples, a word or abbreviation in angle brackets indicates that you must press the named key on the terminal keyboard.

<CTRL/x> In examples, symbols like this indicate that you must hold down the CTRL key while you type the key that follows the slash. Use of this combination of keys may appear on your terminal screen as the letter preceded by the circumflex character. In some instances, it may not appear at all.

Testing Your System with Exercisers 1

This chapter describes how to use system exercisers as a troubleshooting tool for your ULTRIX operating system. The exercisers are commands residing in the `/usr/field` directory that allow you to test all or part of your system by exercising specified parts. Note that to run the exercisers you must be logged in as superuser.

The system exercisers test the following system areas:

- System memory and file systems
- Peripherals
- Network and communications systems

Specific exercisers for each area are described in the following chapters.

Each of the exercisers has an on-line help facility which prints a description of how to use the exerciser. To access on-line help, use the `-h` option. For example, to access help for the `diskx` exerciser type:

```
# diskx -h
```

For additional information on each of the exercisers, see the ULTRIX Reference Pages.

Note

If you are running exercisers in a diskless operating system environment, you must either use the `-o` option of the exerciser, or move the `/usr/field` directory to a writable area. Because the `netx` exerciser does not have an `-o` option, you must copy `netx` to a writable area before exercising the TCP/IP network.

By default, the `/usr` file system is mounted read-only in a diskless environment. Since some of the system exercisers in `/usr/field` require their log files to be local to the exerciser, the read-only restriction prevents the creation of the client's log file, causing that test to fail.

To avoid this problem, copy the desired exerciser and its associated files to `/var/tmp` before executing them.

1.1 Running System Exercisers

To run any of the exercisers you must be logged in as superuser and `/usr/field` must be your current directory. For example:

```
% su
password:
# cd /usr/field
```

All of the examples in this guide assume that you are logged in as the superuser and that your current directory is `/usr/field`.

All of the system exercisers, except `netx`, have the `-o` option. The `-o` option allows you to specify a file where diagnostic output is saved when the exerciser terminates.

All of the exercisers can be run in either the foreground or the background and can be canceled at any time by pressing CTRL/C in the foreground. Also, you can run more than one exerciser at the same time. Keep in mind, however, that the more processes you have running, the slower the system will perform. Thus, before exercising the system extensively, make sure there are no other users on the system.

Section 1.2 describes how to run multiple exercisers using the `syscript` utility, and Section 1.3 describes exerciser log files.

1.2 Exercising More Than One Part of the System

To exercise more than one part of the system simultaneously, use the `syscript` maintenance command. The `syscript` command asks you which exercisers you want to run, how long you want to run each exerciser, and how many exercisers you want to run at one time. The `syscript` command allows you to exercise a device, a subsystem, or the entire system. For more information, refer to `syscript(8)` in the ULTRIX Reference Pages.

You can start each exerciser either manually, by specifying the time parameter (`-t` option) and by placing each command in the background before executing the next command, or you can use the `syscript` command. If you are using the `syscript` command, type:

```
# syscript
```

Once the `syscript` command is running, answer the questions that it asks. The `syscript` command then executes the individual exercisers and creates a file called `testsuite`, which contains all of the answers you entered when you first executed `syscript`. You can reexecute the commands in the `testsuite` file by entering:

```
# sh testsuite
```

This causes `testsuite` to execute using the original commands and parameters that you entered.

1.3 Using Exerciser Diagnostics

When an exerciser is halted (by either CTRL/C or timing out), it prints diagnostics on the terminal. These diagnostics are also stored in the exerciser's latest logfile. The diagnostics inform you of the tests results.

Each time an exerciser is invoked, a new logfile is generated in the /usr/field directory. For example, when you execute the cmx command for the first time, a logfile named #LOG_CMX_01 is created. The logfiles are records of each exerciser's results and consist of the starting and stopping times, and of error and statistical information. The starting and stopping times of exercisers are also logged into the system errorlog file, /usr/adm/syserr/syserr.hostname. This file also contains information on errors reported by the device drivers or by the system.

The logfiles are especially useful if you have a video display terminal because they provide a record of the diagnostics. However, after reading a logfile you should delete it, since no more than nine logfiles can accumulate for each exerciser. If you attempt to run an exerciser when it has already accumulated nine logfiles, the exerciser tells you to remove some of the old logfiles so that it can create a new one.

If the exercisers find errors, you can determine which device or area of the system has the difficulty by looking at the system errorlog file /usr/adm/syserr/syserr.hostname, using the appropriate uerf command. For more information on the error logger, see the *Guide to the Error Logger System*.

For the meanings of the error numbers and signal numbers, see intro(2) and sigvec(2) in the ULTRIX Reference Pages.

The following chapters describe each of the system exercisers in detail.

Exercising File Systems and Memory 2

This chapter describes the system exercisers that test local file systems, system memory, and shared memory.

2.1 Exercising the Local File System

Use the `fsx` command to exercise the file systems locally. The `fsx` command exercises the specified file system by initiating multiple processes, each of which creates, writes, closes, opens, reads, validates, and unlinks a test file of random data. For more information on the `fsx` command, refer to `fsx(8)` in the ULTRIX Reference Pages.

Note

Do not run `fsx` on a file system that is on the same partition that `dskx` is currently read/write exercising. The `dskx` exerciser will be overwriting the file system that the `fsx` exerciser is supposed to be using.

Do not test remotely mounted file systems with `fsx`.

The format of the `fsx` command is:

```
fsx -h -ofile -pn -fpath -tmin
```

- `-h` Prints the help message for the `fsx` command.
- `-ofile` Saves the output diagnostics in *file*.
- `-pn` Specifies the number of processes you want `fsx` to initiate. The maximum number of processes is 250. If you do not specify the `-p` option, the default of 20 is assumed.
- `-fpath` Specifies the pathname of the directory of the file system to test, for example, `-f/usr` or `-f/mnt`. If you do not specify the `-f` option, the default is `/usr/field`.
- `-tmin` Specifies how many minutes you want the `fsx` command to exercise the file system. If you do not specify the `-t` option, the `fsx` command runs until you terminate it by pressing CTRL/C in the foreground.

The following is an example of the `fsx` command:

```
# fsx -p5 -f/usr -t60 &
```

The example shows how to test the `/usr` file system with five processes running for 60 minutes in the background.

2.2 Exercising the System Memory

Use the `memx` command to exercise the system memory. The `memx` command exercises the system memory by initiating multiple processes. By default, the size of each process is defined as the total system memory in bytes divided by 20. The minimum allowable number of bytes per process is 4095. The `memx` command runs ones and zeros, zeros and ones, and random data patterns in the allocated memory being tested.

If you are running `memx` in a diskless environment, you must copy the `memx` file and its associated files from `/usr/field` to a writeable area. The files that you need to run the `memx` exerciser include the following:

- `memx`
- `memxr`
- `shmx`
- `shmxb`

For more information refer to `memx(8)` in the *ULTRIX Reference Pages*.

Note

The `memx` command is restricted by the amount of available swap space. The size of the swap space and the internal memory available determine how many processes can run on your system. For example, if there are 16 Mbytes of swap space and 16 Mbytes of memory, all the swap space would be used if all 20 (the default) initiated processes were running simultaneously. This would prevent any other process from executing.

Therefore, on systems with large amounts of memory and small amounts of swap space, you must use the `-p` or the `-m` options, or both, to restrict the number of `memx` processes or to restrict the size of the memory being tested.

The format of the `memx` command is:

```
memx -h -ofile -s -mn -px -tmin
```

- h Prints the help message for the `memx` command.
- ofile Saves the output diagnostics in *file*.

- s Disables automatic invocation of `shmx`, which is performed by default.
- mn Specifies the amount of memory in bytes (*n*) for each process to test. The default is the total amount of memory divided by 20, with a minimum size of 4095 bytes.
- px Specifies the number of processes to initiate. The maximum number is 20, which is also the default.
- tmin Specifies how many minutes you want the `memx` command to exercise the memory. If you do not specify the `-t` option, the `memx` command runs until you terminate it by pressing CTRL/C in foreground.

The following is an example of the `memx` command:

```
# memx -s -m4095 -p5 -t60 &
```

This example disables automatic execution of the shared memory exerciser, `shmx`, which is performed by default. It also initiates five processes that test 4095 bytes of memory, and it runs in the background for 60 minutes.

2.3 Exercising Shared Memory

Use the `shmx` command to exercise shared memory. The `shmx` command spawns a background process called `shmbx`, and together `shmx` and `shmbx` exercise the shared memory segments. They take turns writing and reading each other's data in the segments.

Using `shmx`, you can test the number and the size of memory segments and `shmbx` processes. The `shmx` exerciser runs until the process receives either a CTRL/C or a `kill -15 pid` command in foreground. For more information on the `kill` command refer to `kill(1)` in the *ULTRIX Reference Pages*.

You automatically invoke `shmx` when you start `memx`. You can also run `shmx` by itself.

The `shmx` format is:

- ```
shmx -h -ofile -ti -mj -sk -v
```
- h Prints the help message for the `shmx` command.
  - ofile Saves diagnostic output in *file*.
  - v Uses the fork system call instead of the `vfork` system call to spawn `shmbx`.
  - ti Indicates the run time in minutes (*i*). The default is to run until the process receives either a CTRL/C or a `kill -15 pid` command in foreground.

- mj Specifies the memory segment size in bytes (*j*) to be tested by the processes; *j* must be greater than 0. The default is SMMAX/6. (SMMAX is a system parameter set in the file /sys/conf/param.c.)
- sk Specifies the number of memory segments (*k*). The default is 6. The maximum is also 6.

The following example tests six memory segments (default), each with a segment size of SMMAX/6. The test terminates when you either press CTRL/C or issue the command **kill -15 *pid*** in the foreground.

```
shmx
```

Here is another example of the shmx command:

```
shmx -t180 -m100000 -s3 &
```

This example runs three memory segments of 100,000 bytes for 180 minutes in the background.

## Exercising Peripherals 3

This chapter describes the system exercisers that you use to test the following peripherals:

- Disk drives
- Tape drives
- Line printers

### 3.1 Exercising the Disk Drives

Use the `dskx` maintenance command to exercise the disk drives. The `dskx` maintenance command exercises specified partitions and file systems on the designated disk.

First, `dskx` checks to see if the specified file system is on the disk. If the file system exists and if you are using either the `-p` or `-c` options, `dskx` asks you if you want to continue or stop the program. These options overwrite data on the disk, destroying any file systems on the specified partition or device.

#### Note

The `-p` and `-c` options destroy data on the device you are testing. Use extreme caution when using either of these options.

When using the `-p` option, ensure that the partition you are exercising does not overlap other partitions because you will inadvertently destroy data on another partition. Use the `chpt` maintenance command with the `-q` option to see which partitions overlap on the device you are testing.

For more information on the `dskx` command, refer to `dskx(8)` in the **ULTRIX Reference Pages**.

The format of the `dskx` command is:

```
dskx -h -ofile -pdevpart -cdev -rdev -tmin -dn
-h Prints help message for the dskx command.
```

- ofile       Saves output diagnostics in *file*.
- pdevpart   Performs random seeks, writes, and reads on the specified partition (*part*) of the device (*dev*). Next, it validates the random data and block sizes. The *part* specifies the partition on the device and can be a through h.  
  
You cannot test the c partition because the test would corrupt the bad block information.
- dev*       Specifies the raw or buffered device name and number such as rra2 for an RA device or hp0 for an RM or an RP device.
- cdev       Performs random seeks, writes, and reads on all the partitions of the specified device (*dev*) except for the c partition. Partition c is not tested because the test would corrupt the bad block information.
- rdev       Performs random seeks and reads on all partitions of the device specified (*dev*) except for partition c. The -r option is safe to use on any disk containing file systems because it will not overwrite data.
- tmin       Specifies how many minutes you want the dskx command to exercise the disk. If you do not specify the -t option, the dskx command runs until you terminate it by pressing CTRL/C in the foreground.
- dn       Specifies in minutes how often you want the dskx command to print diagnostics to the terminal. The default is to print diagnostics upon completion of the exercise.

The following are examples of the dskx command:

```
dskx -rra0 -t20 -d5 &
dskx -pra1g -t60 -d10 &
```

The first example tests the first RA disk on the system for 20 minutes in the background. Diagnostics are printed to the terminal every five minutes. Note that this example shows how to seek and read the device; no file systems are overwritten.

The second example tests the g partition of the ra1 device for 60 minutes. It runs in the background and writes diagnostics to the terminal every 10 minutes.

## 3.2 Exercising the Tape Drives

Use the mtm maintenance command to exercise the tape drives. The mtm command writes, reads, and validates random data on a tape device from the beginning of the tape (BOT) to the end of the tape (EOT). For more information refer to mtm(8) in the ULTRIX Reference Pages.

The format of the `mtx` command is:

- `mtx -h -ofile -rn -fn -sdev# -ldev# -vdev# -adev# -tmin`
- `-h` Prints help message for the `mtx` command.
  - `-ofile` Saves diagnostic output in *file*.
  - `-rn` Specifies the record length in bytes for the long record exercise. The range of *n* is 100 through 20480 bytes. The default value is 10240 bytes.
  - `-fn` Specifies the length of the file in numbers of records. The variable *n* is the number of records. The default value is EOT (-1).
  - `-sdev#` Performs a short record test that writes, reads, and validates 512 byte records (the default) on device *dev#*. The variable *dev#* is the raw device name and number such as `rmt0h` for a high-density rewind tape device, or `nrmt0l` for a low-density no rewind tape device.
  - `-ldev#` Performs a long record test that writes, reads, and validates 10240 byte records (the default) on device *dev#*. The variable *dev#* is the raw device name and number such as `rmt0h` for a high-density rewind tape device, or `nrmt0l` for a low-density no rewind tape device.
  - `-vdev#` Performs a variable record length test that writes, reads, and validates random record lengths from 512 to 20280 bytes on device *dev#*. The variable *dev#* is the raw device name and number such as `rmt0h` for a high-density rewind tape device, or `nrmt0l` for a low-density no rewind tape device.
  - `-adev#` Performs short, long, and variable record length tests on device *dev#*. The variable *dev#* is the raw device name and number such as `rmt0h` for a high-density rewind tape device, or `nrmt0l` for a low-density no rewind tape device.
  - `-tmin` Specifies how many minutes you want the `mtx` command to exercise the tape drive. If you do not specify the `-t` option, the `mtx` command will run until you terminate it by pressing CTRL/C in foreground.

The following is an example of the `mtx` command:

```
mtx -r20480 -lrmt0h -t60 &
```

This example shows how to perform a long record test on the tape device `rmt0h`. The record length is 20480 bytes and the test runs for 60 minutes in the background.

### 3.3 Exercising the Lineprinters

Use the `lpx` maintenance command to exercise the lineprinters. The `lpx` command exercises the line printers by printing a rolling character pattern repeatedly to the device. If the device is offline, `lpx` attempts to start the device every 60 seconds until it succeeds or until you terminate it. Once the line printer starts, the `lpx` command prints five pages of a rolling character pattern, pauses for 15 minutes, and then repeats the printing and pausing sequence until the the command terminates. For more information refer to `lpx(8)` in the *ULTRIX Reference Pages*.

#### Note

To prevent other jobs from interfering with the test, you should disable the line printer queue of the printer you are testing before using the `lpx`. You may need to look in the `/etc/printcap` file to determine the spool queue and then use the line printer control command `lpc` to accomplish this task.

The format of the `lpx` command is:

```
lpx -h -ofile -pn -ddev -tmin
```

- h Prints help message for the `lpx` command.
- ofile Saves diagnostic output in *file*.
- pn Specifies the pause period in *n* minutes. The default pause period is 15 minutes and the shortest pause can be zero (0) minutes.
- ddev Specifies the line printer you want to test such as `lp` or `lp1`. The device names are listed in the `/dev` directory.
- tmin Specifies how many minutes you want the `lpx` command to exercise the line printer. If you do not specify the `-t` option, the `lpx` command runs until you terminate it by pressing `CTRL/C` in foreground.

The following is an example of the `lpx` command:

```
lpx -t60 -dlp1 &
```

This example shows how to exercise the `lp1` lineprinter for 60 minutes in the background.

# Exercising Network and Communications Systems 4

This chapter explains the system exercisers used to test the following network and communications systems:

- TCP/IP
- Terminal communications

## 4.1 Exercising the TCP/IP Network

Use the `netx` maintenance command to exercise the TCP/IP network. The `netx` command sets up a stream socket connection with `netx` acting as the client and the `miscd` utility acting as the server in the TCP/IP internet domain. Using the connection, the `netx` command writes random data to the `miscd` server. The server loops the data back to `netx`, and then the data is read and verified against the original data. For more information on the `netx` command, refer to `netx(8)` in the ULTRIX Reference Pages.

The `netx` command uses the port number of the echo TCP service in the `/etc/services` file. The echo TCP service must be enabled in the `/etc/inetd.conf` file. The service is enabled if there are no number signs (#) in front of the echo service.

If you remove the number sign (#) from the echo TCP service in the `/etc/inetd.conf` file, you must start the echo daemon. To do this, first find the PID of `inetd`, using the `ps -ax` command. Then type the command:

```
kill -HUP pid
```

The *pid* argument is the PID of the `inetd` daemon.

The format of the `netx` command is:

```
netx -h -pn nodename -tmin
```

- h Prints the help message for the `netx` command.
- pn Specifies the port number to use in the internet domain. The variable *n* must be less than 32768. By default, `netx` uses the port associated with the echo service, so you should not have to use this option.

- nodename*    The name of the remote or local system host running the server.
- tmin*       Specifies how many minutes you want the `netx` command to exercise the network connection. If you do not specify the `-t` option, the `netx` command runs until you terminate it by pressing CTRL/C in foreground.

The following is an example of the `netx` command:

```
netx suex -t60 &
```

The example shows how to test the network connection to a remote system host named `suez`. The test runs for 60 minutes in the background.

## 4.2 Exercising the Terminal Communication System

Use the `cmx` maintenance command to exercise the terminal communications system. The `cmx` command writes, reads, and validates random data and packet lengths on the communications line or lines specified. The lines you exercise must have a loopback connector attached to the distribution panel or the cable. Otherwise, `cmx` repeatedly prints an error message to the terminal until its time expires or until you press CTRL/C. For more information on the `cmx` command, refer to `cmx(8)` in the ULTRIX Reference Pages.

Also, the lines you specify must be disabled in the `/etc/ttys` file. See the Guide to System Environment Setup for information about the `/etc/ttys` file.

The format of the `cmx` command is:

- ```
cmx -h -ofile -tmin -l line-1 line-2 line-n...
```
- h* Prints help message for the `cmx` command.
 - ofile* Saves output diagnostics in *file* .
 - tmin* Specifies how many minutes you want the `cmx` command to exercise the communications system. If you do not specify the `-t` option, the `cmx` command runs until you terminate it by pressing CTRL/C in the foreground.
 - l line-n* Specifies the line or lines you want to test. The values for *line-n* are found in the `/dev` directory and are the last two characters of the tty device name. For example, if you want to test the communications system for a device named `tty02`, `tty03`, and `tty14`, *line-1* is 02, *line-2* is 03, and *line-3* is 14. Additionally, the *line-n* argument can specify a range of lines to test, such as 00-08.

You cannot test pseudodevice lines or lta device lines. Pseudodevices have a p, q, r, s, t or u as the first character after `tty`, such as `ttyp3`. The lta devices have a major number of 39.

The following examples show how to use the `cmx` command. The first example exercises communications lines `tty22` and `tty34` for 45 minutes in the background. The second example exercises lines `tty00` through `tty07` until you press `CTRL/C`.

```
# cmx -l 22 34 -t45 &
```

```
# cmx -l 00-07
```


Index

C

cmx exerciser, 4-2 to 4-3
 lta lines and, 4-2
 pseudodevice lines and, 4-2

D

disk drive
 testing, 3-1
dskx exerciser, 3-1 to 3-2
 -c option and, 3-1c
 chpt command and, 3-1c
 -p option and, 3-1c

F

file system
 testing, 2-1
fsx exerciser
 and, 2-1c

I

IP network
 testing, 4-1

L

line printer
 testing, 3-4
lpw exerciser, 3-4 to 3-5
 line printer queue and, 3-4n

M

magnetic tape drive
 testing, 3-2
memory
 testing, 2-2
memx exerciser
 swap space and, 2-2n
mtx exerciser, 3-2 to 3-4

N

netx exerciser, 4-1 to 4-2
 echo TCP service and, 4-1
 misc utility and, 4-1

S

shm exerciser
 memx and, 2-3
 shmb and, 2-3

syscript command

testsuite file and, 1-2

system

testing, 1-2

system exerciser

description, 1-1 to 1-3

diagnostics, 1-3

help facility

in a diskless environment, 1-1n

requirements, 1-2

T**TCP network**

testing, 4-1

terminal communications system

testing, 4-2

HOW TO ORDER ADDITIONAL DOCUMENTATION

DIRECT TELEPHONE ORDERS

In Continental USA
and New Hampshire,
Alaska or Hawaii
call **800-DIGITAL**

In Canada
call **800-267-6215**

DIRECT MAIL ORDERS (U.S. and Puerto Rico*)

DIGITAL EQUIPMENT CORPORATION
P.O. Box CS2008
Nashua, New Hampshire 03061

DIRECT MAIL ORDERS (Canada)

DIGITAL EQUIPMENT OF CANADA LTD.
100 Herzberg Road
Kanata, Ontario K2K 2A6
Attn: Direct Order Desk

INTERNATIONAL

DIGITAL EQUIPMENT CORPORATION
PSG Business Manager
c/o Digital's local subsidiary
or approved distributor

Internal orders should be placed through the Software Distribution Center (SDC), Digital Equipment Corporation, Westminister, Massachusetts 01473

*Any prepaid order from Puerto Rico must be placed
with the Local Digital Subsidiary:
809-754-7575

Reader's Comments

Note: This form is for document comments only. DIGITAL will use comments submitted on this form at the company's discretion. If you require a written reply and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Did you find this manual understandable, usable, and well-organized? Please make suggestions for improvement. _____

Did you find errors in this manual? If so, specify the error and the page number.

Please indicate the type of user/reader that you most nearly represent.

- ☐ Assembly language programmer
- ☐ Higher-level language programmer
- ☐ Occasional programmer (experienced)
- ☐ User with little programming experience
- ☐ Student programmer
- ☐ Other (please specify) _____

Name _____ Date _____

Organization _____

Street _____

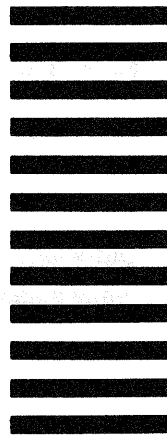
City _____ State _____ Zip Code _____
or _____
Country _____

-----Do Not Tear - Fold Here and Tape-----

digital



No Postage
Necessary
if Mailed in the
United States



BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO.33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

Digital Equipment Corporation
Documentation Manager
ULTRIX Documentation Group
ZKO3-3/X18
Spit Brook Road
Nashua, N.H.

03063

-----Do Not Tear - Fold Here and Tape-----

Cut Along Dotted Line