

decsystem10

OPERATOR'S GUIDE (KL SERIES)

digital



OPERATOR'S GUIDE (KL SERIES)

AA-5104A-TB

May 1977

6.03 MONITOR

This document is a procedural text describing the operation of the KL-series DECsystem-10 hardware, version 6.03 of the TOPS-10 monitor for the main processor, and version 6A of the RSX-20F monitor for the console front-end processor.

To order additional copies of this document, contact the Software Distribution Center, Digital Equipment Corporation, Maynard, Massachusetts 01754.

digital equipment corporation • maynard, massachusetts

First Printing, May 1977

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.

Digital Equipment Corporation assumes no responsibility for the use or reliability of its software on equipment that is not supplied by DIGITAL.

Copyright © 1977 by Digital Equipment Corporation

The postage prepaid READER'S COMMENTS form on the last page of this document requests the user's critical evaluation to assist us in preparing future documentation.

The following are trademarks of Digital Equipment Corporation:

DIGITAL	DECsystem-10	MASSBUS
DEC	DECTape	OMNIBUS
PDP	DIBOL	OS/8
DECUS	EDUSYSTEM	PHA
UNIBUS	FLIP CHIP	RSTS
COMPUTER LABS	FOCAL	RSX
COMTEX	INDAC	TYPESET-8
DDT	LAB-8	TYPESET-10
DECCOMM	DECSYSTEM-20	TYPESET-11

CONTENTS

	Page
PREFACE	ix
CHAPTER 1 THE SYSTEM OPERATOR	1-1
1.1 YOUR RESPONSIBILITIES	1-1
1.2 YOUR DUTIES	1-1
1.2.1 Hardware-Related Duties	1-1
1.2.2 Software-Related Duties	1-1
1.3 DOCUMENTATION	1-2
1.3.1 Supplied by DIGITAL	1-2
1.3.2 Supplied by the Installation	1-2
1.3.2.1 The Operator's Notebook	1-2
1.3.2.2 The System Logbook	1-3
1.3.2.3 Tailored Operation Procedures	1-3
1.3.3 Supplied by You	1-3
CHAPTER 2 SYSTEM COMPONENTS	2-1
2.1 DECSYSTEM-10 HARDWARE	2-1
2.1.1 Central Processors	2-1
2.1.2 Memory Systems	2-1
2.1.3 Front-End Processors	2-2
2.1.4 Peripherals	2-2
2.1.4.1 Line Printer	2-2
2.1.4.2 Card Readers/Punches	2-3
2.1.4.3 DECTapes	2-3
2.1.4.4 Disk Systems	2-3
2.1.4.5 Magnetic Tapes	2-3
2.1.4.6 Plotter	2-4
2.1.5 Terminals and Remote Stations	2-4
2.1.5.1 Hardcopy Terminals	2-4
2.1.5.2 CRT Display Terminals	2-5
2.1.5.3 Remote Stations	2-5
2.2 DECSYSTEM-10 SOFTWARE	2-5
2.2.1 Operating Systems	2-5
2.2.2 Languages	2-6
2.2.2.1 MACRO Assembler	2-6
2.2.2.2 Compilers	2-6
2.2.2.3 Interpreters	2-6
2.2.3 Utilities and Other Software	2-7
CHAPTER 3 STARTING THE SYSTEM	3-1
3.1 POWERING UP	3-3
3.2 LOADING THE CONSOLE FRONT END (RSX-20F MONITOR)	3-3
3.2.1 Loading RSX-20F from Disk	3-3
3.2.2 Loading RSX-20F from DECTape	3-4
3.2.3 Loading RSX-20F via the Switch Register	3-5
3.3 LOADING THE CENTRAL PROCESSOR (TOPS-10 MONITOR)	3-8

CONTENTS (Cont.)

	Page
3.3.1 Loading TOPS-10 from Disk	3-8
3.3.2 Loading TOPS-10 from Magnetic Tape	3-8
3.3.3 TOPS-10 Initialization Dialog	3-9
3.3.4 Loading Second CPU (Dual Processor)	3-11
3.4 LOADING COMMUNICATIONS FRONT ENDS	3-11
3.4.1 Loading via the DL10	3-12
3.4.2 Loading via the DTE20	3-12
3.5 LOADING THE DX10 DATA CHANNEL	3-14
3.5.1 Loading Procedure	3-14
 CHAPTER 4 SYSTEM SERVICES	 4-1
4.1 OPSER	4-1
4.1.1 OPSER Operation	4-1
4.1.1.1 Commands to the OPSER Program	4-2
4.1.1.2 Commands Directly to Subjobs	4-3
4.1.2 OPSER Commands	4-4
4.1.2.1 OPSER Commands to the Operator	4-4
4.1.2.2 OPSER Commands for All Users	4-5
4.1.3 OPSER Operator Messages	4-9
4.2 OMOUNT (FILE AND MOUNT/DISMOUNT)	4-11
4.3 QUEUE	4-14
4.4 GALAXY BATCH SYSTEM	4-16
4.4.1 QUASAR	4-16
4.4.1.1 Components-to-QUASAR Messages	4-17
4.4.2 BATCON	4-19
4.4.2.1 BATCON Commands	4-19
4.4.2.2 BATCON Operation	4-24
4.4.2.3 BATCON Operator Messages	4-26
4.5 OUTPUT SPOOLERS	4-28
4.5.1 LPTSPL	4-34
4.5.1.1 LPTSPL Operator Messages	4-40
4.5.2 SPROUT	4-43
4.5.2.1 SPROUT Operator Messages	4-44
4.6 INPUT SPOOLER (SPRINT)	4-46
4.6.1 SPRINT Commands	4-48
4.6.2 SPRINT Operator Messages	4-49
4.7 OPERATOR/USER COMMUNICATION	4-53
4.7.1 PLEASE	4-53
4.7.2 SEND	4-54
 CHAPTER 5 OPERATOR TASKS	 5-1
5.1 DAILY TASKS	5-1
5.1.1 Update the Message of the Day	5-1
5.1.2 SPACE with /FACT Switch	5-2
5.1.3 Update Account Files	5-4
5.1.4 FILCOM Account Files	5-7
5.1.5 Detached Job Procedure	5-8
5.1.6 BACKUP (Daily)	5-10
5.1.7 SPACE with /BATCH Switch	5-11
5.1.8 GRIPES	5-12
5.1.9 Saving FACT Files	5-12
5.1.10 System Shutdown	5-13
5.1.11 DSKLST	5-14
5.1.12 DSKRAT	5-14

CONTENTS (Cont.)

	Page
5.2 OTHER SCHEDULED TASKS	5-15
5.2.1 Account Control	5-15
5.2.2 BACKUP (Weekly)	5-16
5.3 NON-SCHEDULED TASKS	5-19
5.3.1 Formatting Disk Packs	5-19
5.3.1.1 Formatting RP02s and RP03s	5-19
5.3.1.2 Formatting RP04s and RP06s	5-23
 CHAPTER 6 SYSTEM RECOVERY PROCEDURES	 6-1
6.1 CENTRAL PROCESSOR RECOVERY	6-1
6.1.1 Software Errors	6-2
6.1.1.1 STOPCD (Monitor Error Stop)	6-2
6.1.1.2 RSX-20F Detected Errors	6-3
6.1.1.3 HUNG Condition	6-5
6.1.1.4 Crash Restart Procedure	6-5
6.1.2 Hardware Errors	6-7
6.1.2.1 Parity Error	6-7
6.1.2.2 Altering Your Memory Configuration	6-8
6.1.2.3 Memory Selection and Interleaving	6-9
6.1.3 Environmental Errors	6-14
6.1.3.1 Power Fluctuation	6-14
6.1.3.2 Power Outage	6-14
6.2 FRONT-END PROCESSOR RECOVERY	6-15
6.2.1 Console/Diagnostic Front End	6-15
6.2.2 DTE20-Interfaced Communications Front Ends	6-16
6.2.3 DL10-Interfaced Communications Front Ends	6-17
6.3 DX10 DATA CHANNEL RECOVERY	6-18
 CHAPTER 7 DEVICE OPERATION	 7-1
7.1 LINE PRINTER	7-1
7.1.1 Changing Forms	7-4
7.1.2 Controlling the Vertical Format Unit	7-11
7.1.3 Changing and Reversing the Ribbon	7-16
7.1.4 Cleaning the Line Printer	7-22
7.2 CARD READER	7-22
7.2.1 Loading and Unloading Cards	7-22
7.2.2 Cleaning the Card Reader	7-25
7.3 DECTAPE	7-25
7.3.1 Mounting and Dismounting DECTapes	7-26
7.3.2 Cleaning DECTape Drives	7-26
7.4 MAGNETIC TAPE	7-26
7.4.1 Mounting and Dismounting	7-27
7.4.1.1 TU10 and TU16 Drives	7-27
7.4.1.2 TU40 and TU41 Drives	7-29
7.4.1.3 TU70, TU71, and TU72 Drives	7-29
7.4.1.4 TU45 Drives	7-32
7.4.2 Cleaning Magnetic Tape Drives	7-34
7.5 DISK DRIVES	7-35
7.5.1 Mounting and Dismounting Disk Packs	7-35
7.5.2 Cleaning Disk Drives	7-36
7.5.3 Formatting Disk Packs	7-36

CONTENTS (Cont.)

		Page
APPENDIX A	TOPS-10 OPERATOR-PRIVILEGED COMMANDS	A-1
A.1	INTRODUCTION	A-1
A.2	COMMAND DESCRIPTIONS	A-1
	ATTACH device	A-1
	ATTACH diskunit	A-2
	ATTACH job	A-2
	DETACH device	A-3
	DETACH diskunit	A-3
	SET BREAK	A-4
	SET CORMAX	A-4
	SET CORMIN	A-5
	SET DATE	A-5
	SET DAYTIME	A-6
	SET MEMORY	A-6
	SET NOMESSAGE	A-7
	SET OPR	A-7
	SET SCHED	A-8
	SET TTY	A-8
	SET VMMAX	A-14
	XCHNGE	A-14
APPENDIX B	RSX-20F CONSOLE PROCESSOR COMMANDS	B-1
B.1	ENTERING AND EXITING THE PARSER	B-1
B.2	PARSER CONSOLE MODES	B-1
B.3	PARSER PROMPTS	B-1
B.4	PARSER COMMANDS FOR THE OPERATOR	B-2
B.5	PARSER ERROR MESSAGES	B-7
APPENDIX C	KL INITIALIZATION PROGRAM (KLINIT)	C-1
C.1	KLINIT OPERATOR DIALOG	C-2
C.2	KLINIT MESSAGES	C-5
C.2.1	Informational Messages	C-5
C.2.2	Warning Messages	C-6
C.2.3	Dialog Error Messages	C-7
C.2.4	System Error Messages	C-7
C.3	REPORTS RELATING TO KL MEMORY CONFIGURATION	C-15
C.4	KLINIT DIALOG EXAMPLES	C-16
INDEX		Index-1

FIGURES

FIGURE	3-1	Load Switches and Switch Register	3-2
	7-1	LP07 Line Printer	7-2
	7-2	LP10 Line Printer	7-3
	7-3	LP07 Paper Installation	7-6
	7-4	LP07 Forms Alignment	7-7
	7-5	LP10 Paper Installation	7-9
	7-6	LP07 Ribbon Installation	7-17

CONTENTS (Cont.)

	Page
7-7 LP10 Ribbon Installation	7-20
7-8 CR10-F Card Reader	7-23
7-9 CR10-E Card Reader	7-24
7-10 TU16 Magnetic Tape Drive	7-28
7-11 TU40 Series Magnetic Tape Drive	7-30
7-12 TU70 Series Magnetic Tape Drive	7-31
7-13 TU45 Magnetic Tape Drive	7-33
7-14 RP03 Disk Drive	7-37
7-15 RP04 Disk Drive	7-38
7-16 RP06 Disk Drive	7-39

TABLES

TABLE		Page
3-1	Switch Register Bit Definitions	3-6
4-1	Commands to Set Parameters for Batch Jobs	4-21
4-2	Commands to Display the Current Parameter Settings	4-22
4-3	Commands to Control Batch Subjobs	4-22
6-1	MF10-A (32K) Memory Address Switch Settings	6-11
6-2	MF10-G (64K) Memory Address Switch Settings	6-11
6-3	MF10 Interleave Switch Settings	6-11
6-4	MGL0 and MH10 Memory Bank Selection	6-12
6-5	MGL0 and MH10 Memory Address Switch Settings	6-12
6-6	MGL0 and MH10 Interleave Switch Settings	6-13
7-1	Standard LP10 Line Printer Carriage Tape	7-13
7-2	Standard LP10 Line Printer Vertical Action	7-16
A-1	Number of Fillers by Character and Filler Class	A-11

PREFACE

This operator's guide provides the system operator with a brief introduction to the DECsystem-10's hardware and software components and serves as a task-oriented reference for operating the DECsystem-10.

The procedures documented in this manual are based on a single large in-house installation using all the standard hardware and software. Because individual installations will differ both in the hardware configurations and in the software requirements, you should tailor these procedures to fit the needs of your system and add procedures covering any unique applications.

This publication covers three general areas: the system operator, the system hardware, and the system software.

Chapter 1 discusses the responsibilities and duties of the system operator and introduces the documentation aids that he should have at his disposal.

Chapter 2 gives a brief description of the system components. Chapter 7 provides operating and maintenance instructions for the system peripheral devices.

Chapter 3 covers the loading of the main and front-end processors. Chapter 4 describes the service programs and commands used by the operator. Chapter 5 details some of the scheduled and non-scheduled tasks that the operator is normally responsible for. Chapter 6 contains system recovery procedures to be followed in the event of a system malfunction.

Appendix A contains detailed formats of the operator-privileged commands used in the KL main processor. Usage information and examples are shown throughout the procedures.

Appendix B contains the operator-privileged command subset of the console processor language, PARSE.

Appendix C contains detailed information on the KL processor initialization program, KLINIT.

This publication reflects release 6.03 of the TOPS-10 monitor and release 6A of the RSX-20F front-end monitor.

The following publications in the Software Notebooks are suggested as sources of additional information for system operators:

Getting Started With DECsystem-10	DEC-10-XGSDA-A-D
Getting Started With TOPS-10 Commands	DEC-10-OTSCA-A-D
DECsystem-10 Operating System Commands	AA-0196C-TB
DECsystem-10 Monitor Calls	AA-0974C-TB
DECsystem-10 Galaxy Batch Reference Manual	AA-0967B-TB
DECsystem-10 Galaxy Operation and System Programming Procedures	AA-0966B-TB
Programming Specifications	(Notebooks 10, 11)
Supplementary Operator Procedures	(Notebook 13)

CHAPTER 1

THE SYSTEM OPERATOR

1.1 YOUR RESPONSIBILITIES

As a DECSYSTEM-10 operator, your primary responsibility is to keep the system running and to respond to user requests as quickly and efficiently as possible. You must be familiar with all the resources available to your system, know how to interact with these resources, and recognize potential problems that could reduce the effective use of these resources.

You must know how to communicate with the front-end processor as well as with the main processor. You must be familiar with both the RSX-20F operator commands and the TOPS-10 command language. You will also be responsible for documenting system activities for subsequent shifts and for logging a permanent record of system availability, maintenance, and downtime.

1.2 YOUR DUTIES

Your duties can be classified into two categories: those that are hardware-related and those that are software-related. You may perform these duties on a regular schedule, or on an "on call" or even an emergency basis.

1.2.1 Hardware-Related Duties

Hardware-related duties include the physical upkeep of the system devices and the computer room; the replenishing of consumable supplies such as paper forms, blank cards, and blank paper tape; and the servicing of user requests to mount private tapes or disk packs or to load a card or paper tape reader.

In the event of a hardware malfunction you should be able to remove any damaged or foreign material; assess the extent of the failure and call field service if necessary; and attempt to return the device to its normal operating condition as soon as possible.

1.2.2 Software-Related Duties

Software-related duties include replying to user requests, assigning devices, and running system accounting and control tasks as specified by the system manager.

THE SYSTEM OPERATOR

Due to the complexity inherent in large systems, there will be times when processing will cease to function correctly or will terminate abnormally in what is known as a "crash." This can result from software (programming) problems, hardware problems, or environmental problems. Problems also include the presence of inept or malicious users on the system. When a system crashes or ceases to function correctly, it becomes your responsibility to recognize the problem and take the appropriate corrective measures. You should try to bring the system back up to its previous operating level as soon as possible. When this is not feasible, you should attempt to bring up as much of the system as possible. In either case, you should follow the accepted procedure for recording data in the event of a crash.

1.3 DOCUMENTATION

To make your responsibilities and duties a little easier to handle, there are a number of documents available. You should obtain at least the ones mentioned here. Some are for one-time reading, others are for reference, and still others are for you to document pertinent system information.

1.3.1 Supplied by DIGITAL

DECsystem-10 Software Notebooks are a 13-volume set of loose leaf notebooks containing regularly updated descriptions of all supported DECsystem-10 software. In addition to this manual, the DECsystem-10 Operator's Guide (KL Series), the following publications found in the Notebooks will provide additional information relevant to your function as a system operator.

- Getting Started With DECsystem-10
- Getting Started With TOPS-10 Commands
- DECsystem-10 Operating System Commands
- DECsystem-10 Monitor Calls
- Supplementary Operator Procedures (Notebook 13)

If you are using batch processing, you should also have the following:

DECsystem-10 GALAXY Operation and System Programming Procedures

In addition to the above, you may find helpful ideas in the SPECIFICATIONS located in Notebooks 10 and 11. DIGITAL also supplies some text files on the magnetic tape used for software distribution. These files can be identified by their file extension of .HLP, .MEM, .BWR, or .DOC. If these files have been copied over to your system disk, you can PRINT them on your line printer or at your terminal.

1.3.2 Supplied by the Installation

Installation-supplied documentation usually includes an Operator's Notebook, a System Logbook, and possibly a set of tailored system operations procedures.

1.3.2.1 The Operator's Notebook - The Operator's Notebook is the most practical means for shift-to-shift communications among system operators, and between an operator and the system supervisor. Oral

THE SYSTEM OPERATOR

messages are quickly forgotten, loose notes are often lost; only the Operator's Notebook provides a permanent and centralized message center for efficient system operations.

You should read the turnover entries as you begin your shift. Entries will include scheduled system downtime, new software to try, problems encountered on previous shifts, warnings on potential intermittent malfunctions, and specific instructions for work to be processed on your shift.

At the end of your shift, you should enter the same type of information as it applies to the next shift operator. If you are a non-prime shift operator, you can communicate with the sys\$tem manager via the Operator's Notebook.

1.3.2.2 The System Logbook - The System Logbook is another important piece of documentation. It is the prime source of information about the system and is referred to and used by operators, system programmers, system managers, software support personnel, and field service engineers.

Entries include all monitor loads and reloads, with reason for reloading; all hardware and software problems with symptoms and device identification; and, system shutdowns. For ease in identifying the items, you can follow the convention of entering hardware entries on the left-hand pages and software entries on the right-hand pages. Whatever the format of the entries, it is most important that you check this log when you report for work and that you keep it updated throughout your shift.

1.3.2.3 Tailored Operation Procedures - Your installation may want to generate a tailored set of operating procedures based upon your specific hardware configuration and the software that is actually in use. Sections can be set up for your daily, weekly, monthly and other scheduled tasks. A customized operator's guide such as this would provide you with a compact, easily referenced run book and would be useful in the training of new operators.

1.3.3 Supplied by You

In addition to the entries that you generate in the Operator's Notebook and System Logbook, you should retain and file the system console (CTY) output or log. This chronological record of system events can be very useful in tracking down highly intermittent system problems.

Whenever the system abnormally terminates, you should use one of the memory dump programs to create a hardcopy record of the error conditions. This information will be required by software support personnel or field service engineers when they troubleshoot the problem.

CHAPTER 2

SYSTEM COMPONENTS

To understand the system tasks that you will be performing, you should be familiar with both the hardware and software components of the DECsystem-10. This chapter provides brief descriptions of the hardware devices that you will be operating and the software programs that you and other system users will be running.

2.1 DECSYSTEM-10 HARDWARE

The DECsystem-10 hardware that you will be operating falls into five general categories: central processors, memory systems, front-end processors, peripherals, and remote stations.

2.1.1 Central Processors

The KL10A central processor, used in the DECsystems 1080 and 1088, has a microcoded instruction set of 386 instructions, a high speed 2K cache memory, eight sets of sixteen fast integrated-circuit general purpose registers, virtual memory capability, and a PDP-11 based Console/Diagnostic processor with a 30-character-per-second console terminal. This console terminal allows the system operator to communicate with the central processor as well as with the Console/Diagnostic processor.

The KL10B central processor, used in the high-performance DECsystems 1090 and 1099, offers the same features and functions as the KL10A plus the following:

1. Up to five integrated high-speed data channels (RH20's).
2. Up to three integrated communications processor channels (DTE20's). A fourth DTE20 is used to interface the Console/Diagnostic processor.

2.1.2 Memory Systems

Core memory, on the DECsystems 1080, 1088, 1090, and 1099, consists of combinations of 32K, 64K, 128K and 256K word modules. Up to 16 memory modules may be attached to a system providing a maximum of 4,096K 36-bit words of directly addressable storage.

The storage module capacities for the various storage systems are: 32K and 64K words for the MF10, 64K and 128K words for the MG10, and

SYSTEM COMPONENTS

128K and 256K words for the MH10. Each MF10 memory module contains up to four memory ports while each MG10 and MH10 memory module contains up to eight memory ports. Each port can be further expanded through the use of the MX10 memory multiplexer.

2.1.3 Front-End Processors

The PDP-11 based Console/Diagnostic processor, discussed as part of the central processor, is also referred to as a console front end. It provides the means for the system operator to communicate with the central processor.

Additional PDP-11 based front-end processors, referred to as communications front ends, can be attached to the KL10B processor via the integrated DTE20 interface or DL10 data channel. On the KL10A they are attached via the DL10. Communications front ends provide the facilities for supporting interactive terminals and unit record remote stations. They can be attached locally via direct electrical connections or remotely via dedicated or dial-up telephone lines. A DC76 front end, for example, provides asynchronous line protocol to support terminals such as the LA series DECwriters, LT series Teletypes, and the VT series video terminals.

The DC75NP and DN85 front ends provide synchronous line protocol to support remote stations such as the DC72NP, the DAS80 series, and the DAS92. They attach to the processors via the DL10.

The DN87 and DN87S front ends provide both synchronous and/or asynchronous protocol, depending upon the line options selected. The DN87 attaches to the KL10A and KL10B via the DL10 communication channel. The DN87S attaches to the KL10B via the integrated DTE20 interface.

2.1.4 Peripherals

Peripheral equipment, supported by the DECsystems 1080, 1088, 1090, and 1099, consists of unit record, magnetic disk, and magnetic tape devices.

2.1.4.1 Line Printer - The following line printers are available:

1. The LP10-F impact printer with a 64-character print set, dual speed of 1250/925 lines per minute, optical vertical format control using 12-channel tape, and a quick-change print character drum.
2. The LP10-H impact printer with a 96-character print set, dual speed of 925/675 lines per minute, optical vertical format control using 12-channel tape, and a quick-change print character drum.
3. The LP07 impact printer with a quick-change, dual character set Charaband(R) (64/96 characters), dual speed of 1200/900 lines per minute, and Direct Access Vertical Format Unit.

(R) Charaband is a registered trademark of the Dataproducts Corporation

SYSTEM COMPONENTS

4. The LA180 impact (7 x 7 dot matrix) printer with a 128-print-character set, and a print speed of 180 characters per second resulting in 60-400 lines per minute.

2.1.4.2 Card Readers/Punches - The following card readers and punches are available:

1. The CR10-D table model card reader processing 1000 cards per minute.
2. The CR10-E console card reader processing 1200 cards per minute.
3. The CR10-F table model card reader processing 300 cards per minute.
4. The CP10-D card punch processing 100 cards per minute.

2.1.4.3 DECTapes - The TU56 DECTape drive is a high reliability, bi-directional, random access, magnetic tape storage device available on the DECsystem-10. Each TU56 is a dual tape transport, and, up to 4 of the dual drives can be attached to each TD10-C DECTape controller.

2.1.4.4 Disk Systems - There are five disk systems available on the DECsystem-10: the RHS04 fixed-head swapping system, and the RHP02, RHP03, RHP04, and RHP06 removable disk pack systems.

The RHS04 is a fast access, high transfer rate, swapping system and is used very effectively in implementing the virtual memory option in which pages (512 words) of memory are swapped between core and secondary storage (RHS04). Page swapping can take place directly between one part of core and secondary storage concurrently with program execution in another part of core.

The RHP02, RHP03, RHP04, and RHP06 disk systems offer storage capacities of 5.12, 10.24, 20.48, and 40.96 million words per drive, respectively. The RHP04 and RHP06 disk systems are each available with a dual-port option. This option allows a disk drive to be accessed over either of two paths or both concurrently. In the KL series processors, for example, one dual-ported disk drive can contain files to be accessed by both the main processor and the console front-end processor. If both ports are connected to the main processor, the operating system can maximize I/O throughput by starting simultaneous data transfers on two channels, concurrently.

2.1.4.5 Magnetic Tapes - The following magnetic tape systems and drives are supported:

1. The TU10A-E 9-track and TU10A-F 7-track drives with a recording density of 200, 556, and 800 bits per inch and a tape speed of 45 inches per second.

SYSTEM COMPONENTS

2. The TU16 9-track subsystem with recording densities of 800 and 1600 bits per inch and a tape speed of 45 inches per second.
3. The TU40 9-track and TU41 7-track drives with recording densities of 200, 556, and 800 bits per inch and tape speeds of 150 inches per second.
4. The TU45 9-track drive with recording densities of 800 and 1600 bits per inch and tape speed of 75 inches per second.
5. The TU70 9-track tape drive with recording densities of 800 and 1600 bits per inch and tape speed of 200 inches per second.
6. The TU71 7-track tape drive with recording densities of 200, 556, and 800 bits per inch and tape speed of 200 inches per second.
7. The TU72 9-track tape drive with recording densities of 1600 and 6250 (nominal) bits per inch recording density and tape speed of 125 inches per second.

2.1.4.6 Plotter - The DECsystem-10 supports the following plotters:

1. The XY10-A (CALCOMP Model 565) plotter with .01 inch, .005 inch, or .1 millimeter step size at 18,000 steps per minute, and using 12-inch-wide paper.
2. The XY10-B (CALCOMP Model 563) plotter with .01 inch step size at 12,000 steps per minute or .005 inch or .1 millimeter step size at 18,000 steps per minute, and using 31-inch-wide paper.

2.1.5 Terminals And Remote Stations

The DECsystem-10 supports hardcopy terminals, CRT display terminals, and remote stations comprised of unit-record equipment as well as remote terminals.

2.1.5.1 Hardcopy Terminals - The following hardcopy terminals are supported in both local and remote modes:

1. The LA36 DECwriter II, an impact teleprinter with a 96-character print set, 10/15/30 character-per-second print speed and input keyboard.
2. The LA35RO DECwriter, a receive-only terminal and basically an LA36 without a keyboard.
3. The LA37APL DECwriter, an LA36 with dual APL/ASCII character set. It provides the user with the choice of full APL, full upper/lower case ASCII, or upper case TTY all in one terminal.

SYSTEM COMPONENTS

4. The LA180 DECprinter I, a 7x7 dot matrix impact printer with a 128-character print set and a speed of 180 characters per second.

2.1.5.2 CRT Display Terminals - The following CRT display terminals are supported in both local and remote modes:

1. The VT50 and VT50H video terminals, providing a screen format of 12 lines at 80 characters per line with a 5 x 7 dot matrix of 64 characters.
2. The VT52 video terminal, providing a screen format of 24 lines at 80 characters per line with a 7 x 7 dot matrix of 96 characters.
3. The VT61 video terminal, providing a screen format 24 lines at 80 characters per line with a 7 x 8 dot matrix of 128 characters. The VT61 has a more powerful microprocessor than the VT50 series and can offer more stand-alone capabilities in the area of text editing.

2.1.5.3 Remote Stations - The following devices are supported as remote stations:

1. The DC72 remote station, with peripherals such as card reader, line printer, and hardcopy and display terminals.
2. The DN80 series remote stations that can provide remote job entry facilities (DN80), remote terminal facilities (DN81), or a combination of both (DN82).
3. The DAS92 remote station, which can be configured in combinations of terminal lines and a card reader and/or a line printer.

2.2 DECSYSTEM-10 SOFTWARE

The DECSYSTEM-10 software that you must be familiar with falls into three general categories: operating systems, languages, and utilities.

2.2.1 Operating Systems

The main operating system for the DECSYSTEM-10 is called the TOPS-10 Monitor. It supports time-sharing, batch processing, real-time, and remote communications.

The front-end operating system runs in the Console/Diagnostic processor and aids the central processor and TOPS-10 in handling system tasks such as system loading, console communications, and diagnostic functions. The front-end operating system is called RSX-20F.

SYSTEM COMPONENTS

2.2.2 Languages

The DECsystem-10 supports three types of translators used in symbolic language programming: assemblers, compilers, and interpreters.

2.2.2.1 MACRO Assembler - MACRO is the symbolic assembly program used on the DECsystem-10. It is device-independent; input and output devices can be selected at run time and may vary from run to run. MACRO also contains powerful macro capabilities that allow the user to replace commonly used sequences of instructions with a single statement.

2.2.2.2 Compilers - The ALGO^rithmic Language, ALGOL, is a scientific language designed for the solution of computational processes, or algorithms. The DECsystem-10 ALGOL is composed of the ALGOL compiler for the translation of the source code and the ALGOL object-time system that provides I/O services, a library of mathematical functions, and control of system resources.

The COM^mon Business Oriented Language, COBOL, is an industry-wide data processing language designed for business applications. COBOL programs are written in terms that are familiar to the business user. The English-like statements are easily learned and result in programs that are self-documenting. DECsystem-10 COBOL includes the COBOL compiler; an object time system, LIBOL; a source library maintenance program; a COBOL checkout and debugging program, COBDDT; and other ancillary programs.

The FORM^ula TRAN^slator language, FORTRAN, is a widely used, procedure-oriented language designed for solving scientific-type problems. It has, however, found extensive use throughout the data processing industry and especially in the universities and in timesharing utilities. DECsystem-10 FORTRAN-10 includes the FORTRAN-10 compiler; an object-time system, FOROTS; and a FORTRAN checkout and debugging program, FORDDT.

2.2.2.3 Interpreters - The Algebraic Interpretive Dialogue, AID, is the DECsystem-10 adaptation of JOSS, a program developed by the RAND Corporation. Commands to AID are entered via the user's terminal as imperative English sentences and can be executed immediately or stored as part of a routine to be executed later.

APL (A Programming Language) is a completely conversational system especially suited for dealing with numeric and character array-structured data. However, it is intended for use as a general data processing language as well a mathematician's tool.

The Beginner's All-purpose Symbolic Instruction Code, BASIC, is a problem-solving language, easy to learn, and particularly suited to a time-sharing environment. BASIC contains its own editing facilities; a separate system editor is not required. Although BASIC provides a subset of elementary statements for simple programs, the full language includes statements covering matrix computations, advanced string handling capabilities, mathematical functions, and the generation of user-defined functions.

SYSTEM COMPONENTS

2.2.3 Utilities And Other Software

The DECsystem-10 provides many utilities, as well as text editors, text processors, file manipulators, maintenance programs, and monitor support programs. Some of the more important ones are:

GALAXY	A group of programs that can control up to 127 batch jobs as well as controlling the system unit record devices.
BACKUP	A system program to save disk files on magnetic tape, and later to restore any or all of the files back to disk.
TECO	A character-oriented text editor with the ability to edit any source document without requiring special formatting.
SOUP	A set of programs to facilitate the updating of system or user program source files.
DDT	The Dynamic Debugging Technique used for the on-line checkout and updating of programs.
PIP	A file interchange program to selectively transfer files from one media to another and concurrently perform limited editing and directory manipulation.
RUNOFF	A text processing program that produces a final formatted, paged, right-justified output from an edited file with appropriate RUNOFF commands.
LINK-10	A linking loader to merge independently translated modules into one executable version of a user's program.
OPSER	A service program for the system operator, enabling him to control multiple jobs from a single terminal.

CHAPTER 3

STARTING THE SYSTEM

Before starting the system, you should familiarize yourself with the various operating switches, indicator lights, and other controls located on the console processor panel directly below the DECTape drive. Below these controls, another panel contains a row of 30 switches and additional indicator lights. The switches that are numbered from 0 through 17 comprise the switch register and will be used during the system load procedure. Refer to Figure 3-1 in the following descriptions.

The four load switches are labeled above from left to right as DISK, DECTAPE, SW/REG, and ENABLE. The purpose of each is given below.

LOAD SWITCH	PURPOSE
DISK	Load the system from a disk pack (Section 3.2.1)
DECTAPE	Load the system from DECTape (Section 3.2.2)
SW/REG	Load the system according to the contents of the switch register (Section 3.2.3)
ENABLE	Enable loading via any load switch

There are two power switches. The black switch, labeled POWER ON above and POWER OFF below, is the switch normally used to supply or cut power to the system. The red EMERGENCY POWER OFF switch should be used only in emergencies, such as a fire, where you must turn off the power as quickly as possible. After you press this switch, you must call your field service representative to restore power.

The switch register contains 18 switches representing bit settings of 0 or 1 and numbered 0-17 from right to left according to PDP-11 convention. Note that this is the reverse of the way that bits are numbered in the main PDP-10 processor. The information represented by the switch settings is described in Table 3-1 in Section 3.2.3.

The power light glowing red indicates that power is on. If the power light is blinking, a field service representative at your site has set OVERRIDE. Do not run the system with OVERRIDE set because the system will not power down from overheating.

The fault light should normally be off. If it glows, notify your field service representative. A glowing light indicates a malfunction such as overheating.

STARTING THE SYSTEM

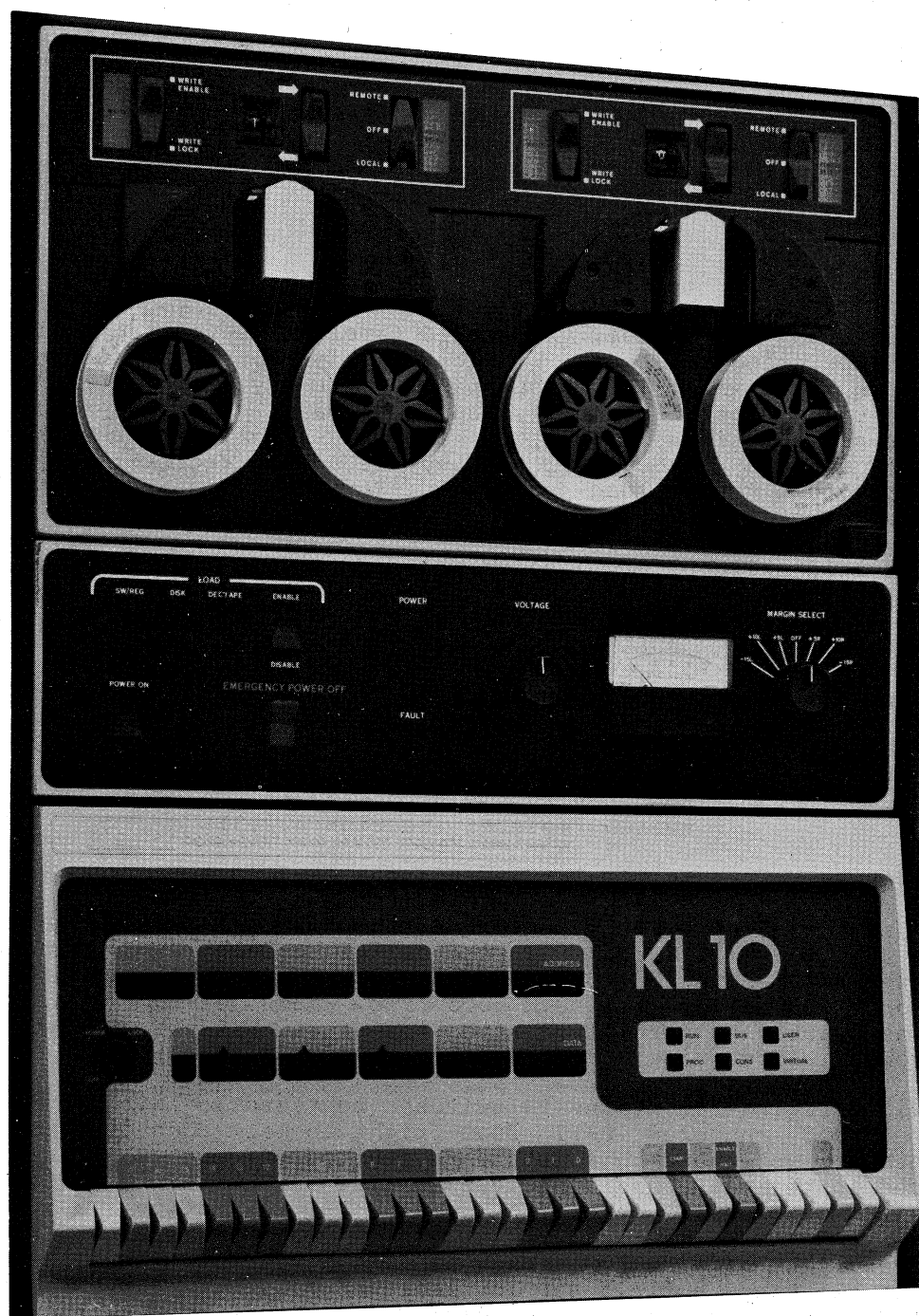


Figure 3-1 Load Switches and Switch Register

STARTING THE SYSTEM

Loading the system involves turning the power ON, loading the console front end with the RSX-20F monitor, and then loading the central processor with the TOPS-10 monitor.

3.1 POWERING UP

In most cases you should only have to check to see that the power light to the right of the load switches is glowing red. Otherwise, press the black power switch to the POWER ON position.

3.2 LOADING THE CONSOLE FRONT END (RSX-20F MONITOR)

After initially powering up the system, or, when a complete reload is indicated, you have a few alternatives in how you load the console front-end processor. You can do a standard load from a disk pack or DECTape, or you can load using the switch register. In most cases, you will load from a disk pack because it is much faster than DECTape. However, if hardware problems prevent loading from a disk pack, or if your system manager wants you to use software residing on a DECTape, you may need to load the system using the DECTape drive. Also, if you need to load the system using some non-standard hardware configurations or special software, you will have to use the switch register. The switch register allows you to load from any disk pack or DECTape and also enter the KL initialization operator dialog (KLINIT). The KLINIT dialog allows you to verify the microcode, reconfigure cache and main memory, and load a non-default bootstrap program. The procedures for loading the front-end processor via a disk pack, a DECTape, or the switch register are described in the next three sections.

3.2.1 Loading RSX-20F from Disk

This is the most common way to load the console front end. It assumes that the system power is ON, that all hardware and software have been correctly installed, and that the disk pack has been mounted properly. Also, the disk pack containing both the TOPS-10 monitor and the RSX-20F front-end monitor must be on a dual-ported drive (controller select switch points to A/B), and the drive must be unit 0 with respect to both the central and front-end processors.

The operation is as follows:

1. Set the ENABLE/DISABLE load switch to ENABLE.

This will enable the other three load switches.

2. Press the DISK load switch.

This will cause the front-end processor to access the disk on drive 0 and load the RSX-20F monitor. The KL initialization program (KLINIT) is loaded and, in turn, loads the KL10 microcode. The default hardware configurations of cache and external memory are set up. The bootstrap program for the TOPS-10 monitor is then loaded into the central processor and started. The following is an example of the output you will receive on the console terminal.

STARTING THE SYSTEM

RSX-20F V006A 0:16 21-JUN-76

```
[SY0: REDIRECTED TO DB0:]
[DB0: MOUNTED]
KLI -- VERSION V002Q RUNNING
KLI -- MICROCODE VERSION 131 LOADED
KLI -- ALL CACHES ENABLED
LOGICAL MEMORY CONFIGURATION:
      CONTROLLER
ADDRESS  SIZE  RQ0  RQ1  RQ2  RQ3      CONTYPE INT
000000   512K  04   FOR  ALL      DMA20   4
KLI -- BOOTSTRAP LOADED AND STARTED
BOOTS Vnn (nnn)
```

BTS>

The first line of output tells you the version and the creation time and date of the RSX-20F monitor that was just loaded into the front end. The next two lines tell you that DB0: (the disk pack on drive 0) is the system device (SY0:) for the front end tasks. The next eight lines are output by the KL initialization program. They document the version of KLINIT that is running, and tell you that the KL10 microcode was successfully loaded, the cache was enabled, external memory was configured, and that the bootstrap program for the TOPS-10 monitor was loaded and started. The next two lines are the bootstrap program's version number and prompt. If an error occurs during the KL initialization program, you will receive an error message preceded by "KLI -- ?" and you will be placed in the KLINIT dialog mode described in Appendix C.

3.2.2 Loading RSX-20F from DECTape

This method of loading the console front end should be used if you cannot load from a disk pack, or if you need a particular version of software that is only on DECTape. It assumes that the system power is ON, that the proper software exists on DECTape and that the DECTape is mounted correctly on unit 0. Be sure to leave the DECTape mounted while the system is running.

The operation is as follows:

1. Set the ENABLE/DISABLE load switch to ENABLE.

This will enable the other three load switches.

2. Press the DECTAPE load switch.

This will cause the front-end processor to access the DECTape drive 0 and load the RSX-20F monitor. The KL initialization program (KLINIT) is loaded and, in turn, loads the KL10 microcode. The default hardware configurations of cache and external memory are set up. The bootstrap program for the TOPS-10 monitor is then loaded into the central processor and started. The following is an example of the output you will receive on the console terminal.

RSX-20F V006A 0:16 21-JUN-76

```
[SY0: REDIRECTED TO DT0:]
[DT0: MOUNTED]
KLI -- VERSION V002Q RUNNING
```


STARTING THE SYSTEM

```

KLI -- MICROCODE VERSION 131 LOADED
KLI -- ALL CACHES ENABLED
LOGICAL MEMORY CONFIGURATION
      CONTROLLER
ADDRESS  SIZE  RQ0  RQ1  RQ2  RQ3      CONTYPE INT
000000   512K  04   FOR  ALL      DMA20   4
KLI -- BOOTSTRAP LOADED AND STARTED
BOOTS Vnn (nnn)

BTS>
```

The first line of output tells you the version and the creation time and date of the RSX-20F monitor that was just loaded into the front end. The next two lines tell you that DT0: (the DECTape on unit 0) is the system device (SY0:) for the front-end tasks. The next eight lines are output by the KL initialization program. They give the version of KLINIT that is running, and tell you that the KL10 microcode was successfully loaded, the cache was enabled, external memory was configured, and that the bootstrap program for the TOPS-10 monitor was loaded and started. The next two lines are the bootstrap program's version number and prompt. If an error occurs during the KL initialization program, you will receive an error message preceded by "KLI -- ?" and you will be placed in the KLINIT dialog mode described in Appendix C.

3.2.3 Loading RSX-20F via the Switch Register

You must load the console front end via the switch register if you need to do any of the following:

1. Enable specific cache or configure external memory yourself instead of using the default configurations.
2. Load a TOPS-10 bootstrap program from a file with a name other than BOOT.EXB; for example, BOOTM.EXB containing BOOTM which loads a monitor from magnetic tape.
3. Load from a disk pack or DECTape that is not on unit 0.

The software for the front-end processor must reside on a DECTape or disk pack connected to the front end. If both the front-end software and the system monitor reside on the same disk pack, the pack must be mounted on a dual-ported drive and the controller select switch must be set to A/B.

The operation is as follows:

1. Set the ENABLE/DISABLE load switch to ENABLE.
This will enable the other three load switches.
2. Set the appropriate switches (or bits) in the switch register.

Switch 0 is mandatory when loading via the switch register. Switches 1 and 2 must be set on if you intend to use the KL initialization dialog (KLINIT). If RSX-20F resides on a disk pack, set switch 7 on; if RSX-20F is on a DECTape, switch 7 must be off. Switches 8-10 must be set to specify the drive number of the disk or DECTape. For a detailed description of all the bit settings in the switch register, see Table 3-1.

STARTING THE SYSTEM

Table 3-1
Switch Register Bit Definitions

17	16	15	14	11	10	8	7	6	3	2	1	0
----	----	----	----	----	----	---	---	---	---	---	---	---

Bit	Meaning
0	If this is set, the remaining bits are interpreted. You must set this to load via the switch register.
2,1	<p>If both 1 and 2 are set, RSX-20F is loaded and the KL initialization operator dialog (KLINIT) is loaded and started. This is the usual case when loading via the switch register.</p> <p>If only one of the two is set, the RSX-20F monitor is loaded; no communication is initiated between the -10 and -11 processors at this time.</p> <p>If both 1 and 2 are not set, the system is loaded much like it is via the DISK or DECTAPE load switch. However, since other bits are interpreted, you can specify the unit number of the bootstrap device in bits 8-10.</p>
6-3	Currently not used, and must not be set.
7	<p>If this is set, the bootstrap device is a disk pack on a dual-ported drive.</p> <p>If this is not set, the bootstrap device is a DECTape drive on the front-end processor.</p>
10-8	These three bits allow you to specify the unit number of the bootstrap device (0 to 7). No bits set indicate unit 0; bits 9 and 8 set indicate unit 3.
14-11	Currently not used, and must not be set.
15	This indicates the action taken when an I/O error occurs during the bootstrapping. If this is set, the operation is retried indefinitely if an error occurs. If not set (the normal case), a halt occurs after ten unsuccessful retries.
17,16	Not used, and must not be set.
Bits are set when the corresponding switch is in the upward position.	

STARTING THE SYSTEM

3. Press the SW/REG load switch.

This will cause the front-end processor to access the disk drive (switch 7 set on) or the DECTape drive (switch 7 set off) with the unit number as specified in switches 8-10. The KL initialization program (KLINIT) is loaded and started and the default hardware configurations of cache and external memory are set up. The standard bootstrap program for the TOPS-10 monitor is then loaded into the central processor and started.

The following is an example of the output you will receive on the console terminal if switches 0, 7, 8, and 9 are set on.

```
RSX-20F V006A 0:16 21-JUN-76

[SY0: REDIRECTED TO DB3:]
[DB3: MOUNTED]
KLI -- VERSION V002Q RUNNING
KLI -- MICROCODE VERSION 131 LOADED
KLI -- ALL CACHES ENABLED
LOGICAL MEMORY CONFIGURATION:
      CONTROLLER
ADDRESS  SIZE  RQ0  RQ1  RQ2  RQ3  CONTYPE  INT
000000   512K  04   FOR  ALL      DMA20    4
KLI -- BOOTSTRAP LOADED AND STARTED
BOOTS Vnn (nnn)

BTS>
```

The first line of output tells you the version and the creation time and date of the RSX-20F monitor that was just loaded into the front end. The next two lines tell you that DB3: (the disk on unit 3) is the system device (SY0:) for the front end tasks. If bit 7 had not been set, the DB3: would have been DT3: for a DECTape. The next eight lines are output by the KL initialization program. They give the version of KLINIT that is running, and tell you that the KL10 microcode was successfully loaded, the cache was enabled, external memory was configured, and that the standard bootstrap program for the TOPS-10 monitor was loaded and started. The next two lines are the bootstrap program's version number and prompt.

If, in the previous example, you had also set switches 1 and 2 on, the console output would have been as follows:

```
RSX-20F V006A 0:16 21-JUN-76

[SY0: REDIRECTED TO DB3:]
[DB3: MOUNTED]
KLI -- VERSION V002Q RUNNING
KLI -- ENTER DIALOG [NO,YES,EXIT,BOOT]?
KLI>
```

Switches 1 and 2 set on specify that you wish to enter the KL initialization program (KLINIT) operator dialog. KLINIT has just asked you the first question and is waiting for an answer. When you have answered all the applicable questions, the dialog will end with the bootstrap program's version number and prompt.

For a complete description of the KLINIT dialog messages, and examples, refer to Appendix C.

STARTING THE SYSTEM

3.3 LOADING THE CENTRAL PROCESSOR (TOPS-10 MONITOR)

When the front-end processor has been loaded using any one of the three methods described in Section 3.2, the system informs you:

```
KL I -- BOOTSTRAP LOADED AND STARTED
```

The actual bootstrap program that has been loaded depends upon the method of loading the front end.

If you used the DISK load, DECTAPE load, or the SW/REG load with switches 1 and 2 off; you will have loaded the bootstrap program found in the file BOOT.EXB. This would usually be BOOTS, a program to load the monitor from a disk pack.

If you had used the KLINIT dialog to specify a non-default BOOT file, you could have loaded a program like BOOTM, which will load the monitor from magnetic tape.

For descriptions of the various bootstrap programs, refer to the Specifications section of the Software Notebooks.

3.3.1 Loading TOPS-10 from Disk

In response to the message:

```
KL I -- BOOTSTRAP LOADED AND STARTED
```

you can press the carriage return key and load the default monitor from the disk file SYSTEM.EXE or you can specify another file such as:

```
DSKB:TSTSYS.EXE
```

and load a different monitor.

When the TOPS-10 monitor is loaded, it will enter into an initialization dialog with you to determine start-up options and conditions. These are covered in Section 3.3.3.

3.3.2 Loading TOPS-10 from Magnetic Tape

In order to load the TOPS-10 monitor from a magnetic tape, the following prerequisites must be met:

1. The magnetic tape containing the monitor should be mounted on drive 0. If this is not possible, mount it on any drive but remember to tell BOOTM which drive by entering the controller type and drive number in the command line.
2. The bootstrap program to load a monitor from magnetic tape, BOOTM, must reside on a front-end load device; disk or DECTape.
3. Load the console front end via the SW/REG load procedure. (Section 3.2.3) with at least bits 0, 1, and 2 set ON. This will allow you to use the KLINIT dialog.

STARTING THE SYSTEM

4. When KLINIT prompts you with:

```
CLI -- LOAD KL BOOTSTRAP[YES,NO,FILENAME]?  
CLI>
```

answer with the name of the file containing the magnetic tape bootstrap. For example, if BOOTM was in the file MTBOOT.EXB, respond with:

```
CLI>MTBOOT.EXB
```

KLINIT will load BOOTM, give you the message:

```
CLI -- BOOTSTRAP LOADED AND STARTED
```

and you would then be under control of the bootstrap program. When BOOTM outputs its prompt characters:

```
BTM>
```

you can give the file specification for the monitor as:

```
BTM>DSKB:MAGSYS.EXE[1,4]/TM02:3
```

to load a specific monitor from tape drive 3 on a TM02 controller or you can simply give a carriage return to get the default file, which is DSKB:SYSTEM.EXE[1,4] on tape drive 0 of a TM10 controller.

Use /TM10:n for TU10, TU40, and TU41 tape drives.

Use /TM02:n for TU16 and TU45 tape drives.

Use /TX01:n for TU70 and TU71 tape drives.

Use /TX02:n for TU72 tape drives.

When the TOPS-10 monitor is loaded, it enters into an initialization dialog with the operator to determine the start-up options and conditions. These are covered in Section 3.3.3.

3.3.3 TOPS-10 Initialization Dialog

When the TOPS-10 monitor is loaded, the system checks to see that the monitor and processor are compatible. If they are not, the following error message is issued:

```
?THIS MONITOR WAS BUILT FOR A xxxxxx  
AND WILL NOT RUN PROPERLY ON A yyyyyy.
```

where xxxxxx and yyyyyy can be KA10, KI10 or KL10. If this message persists, the system manager should review the monitor generation procedure in the Software Notebooks.

If the load time diagnostic program (SYSCHK) was included in the system software at monitor generation time, the system will prompt with:

```
SYSCHK (N,Y):
```

A response of Y runs a 5-second diagnostic program that ensures the accessibility of all configured memory and system devices. A reply of N or a carriage return skips the diagnostic.

STARTING THE SYSTEM

The system then types out the system name (up to 24 characters specified at system generation time) as well as the system creation date; for example:

ABC123 KL10 SYS#1234 07-12-76

When the system prompts:

WHY RELOAD:

reply with one of the following acceptable answers:

OPR	NXM	CM
PARITY	HALT	SA
POWER	LOOP	NEW
STATIC	HUNG	SCHED
HARDWARE	PM	OTHER

If you do not reply within 60 seconds, OTHER is assumed. Refer to the ONCE manual in the Supplementary Operator's Procedures section of the Software Notebooks for the meanings and usage of the above replies. The reply is stored in the system error file (ERROR.SYS) and can be retrieved at a later time using the SYSERR program. For example, a reply of:

WHY RELOAD:SCHED

will record that this particular monitor load was a scheduled reload.

When the system prompts:

DATE:

enter the numeric day and alphabetic month, in either order. The month may be abbreviated to any point where it is still unique. The year is optional. If entered, it must be either the full four digits or the last two. If the year is not entered, it is assumed to be the same as in the monitor creation date. If, for example, the monitor creation date was June 21, 1976; any of the following replies would be recorded as September 8, 1976.

DATE:SEP 8 1976
DATE:S 8 76
DATE:8 SEP

When the system prompts:

TIME:

enter a 4-digit time based on a 24-hour clock. For example,

TIME:2015

represents 8:15 PM.

When the system prompts:

STARTUP OPTION:

reply with one of the following:

QUICK	REFRESH	LONG
GO	UNITID	NOINITIA
DESTROY	CHANGE	

STARTING THE SYSTEM

A complete description of each start-up option is contained in the ONCE manual in the Supplementary Operator's Procedures section of the Software Notebooks. The usual operator's reply is GO to start the system with a minimum amount of dialog or QUICK to start the system immediately without dialog.

3.3.4 Loading Second CPU (Dual Processor)

If your system is a dual processor, this is the time to initialize the second CPU. The master CPU is designated CPU0 while the second or slave CPU is referred to as CPU1. The second CPU needs its own RSX-20F front-end software mounted on a DECTape or disk accessible to its front-end processor.

Load the second front-end processor as described in Section 3.2.3 and enter the KLINIT dialog. Load the microcode and reconfigure cache and main memory. When you receive the message:

```
KL I -- LOAD KL BOOTSTRAP [YES, NO, FILENAME]?
KL I>
```

answer "NO" which will bring up:

```
KL I -- EXIT [YES, RESTART]?
KL I>
```

Reply with YES (or default with a carriage return) and you will exit to the RSX-20F monitor. Proceed as follows:

<u>STEP</u>	<u>FUNCTION</u>
^\ PAR>SET CONSOLE	Type control backslash to enter the Parser. Set console to programmer mode.
PAR>SET NO RELOAD	Inhibit automatic reload of CPU1 on an error.
PAR>SET CONS OPR	Return console to operator mode.
PAR>J 400	Start CPU1 at KL memory location 400.
[CPU1]	This is the response typed on CTY1 when CPU1 has completed initialization.

3.4 LOADING COMMUNICATIONS FRONT ENDS

Communication front ends must be loaded with appropriate software based on the type of remote terminals and/or stations to be serviced.

This section deals only with the loading of a front-end processor with a specific or default file that resides on disk. How the software was generated and placed into the disk file is covered in the System Programming Procedures and Techniques in the Software Notebooks.

It is assumed that you have already loaded the TOPS-10 monitor and have logged in.

The loading procedure varies with the type of interface to the KL processor. The DC75NP, DN85, and DN87 front ends interface via the DL10 Data Channel. The DN87S front end interfaces via the DTE20 Ten-Eleven Data Interface.

STARTING THE SYSTEM

3.4.1 Loading via the DL10

The program used to load a front-end processor via the DL10 is BOOT11. Your system programmer will usually have made this program available on your DSK: [1,2] area or it may reside in the SYS: system library.

Load BOOT11 by typing:

```
.R BOOT11
```

TOPS-10 will load and start BOOT11 which will, in turn, prompt you with:

```
FILE:
```

and expect, as a response, a standard DECsystem-10 file descriptor followed by switches telling BOOT11 what to do. A representative response such as:

```
FILE:DN87.BIN/CLEAR/START/PORTNO:2
```

will clear the memory of the PDP-11 on port number 2 to zeros; load the PDP-11 with the program in the file DSK:DN87.BIN; and start the program.

If you reply with a carriage return, the default action is:

```
FILE:DSK:PDPXI0.BIN/START/PORTNO:0
```

Note however that if there is more than one DL10-interfaced PDP-11 on the system, the port number in the above response must be specified..

BOOT11 keeps you informed of its progress with the following messages:

```
"CLEARING PDP-11
```

```
"PDP-11 LOADING
```

```
"PDP-11 LOADED
```

```
"PDP-11 STARTED
```

Additional information on BOOT11 switch options and a list of error messages are in the BOOT11 specification in the Software Notebooks.

3.4.2 Loading via the DTE20

The program used to load a front-end processor via the DTE20 interface is called DTELDR. Your system programmer will usually have made this program available to you on your DSK: [1,2] area or it may reside in the SYS: system library.

Load DTELDR by typing:

```
.R DTELDR
```

TOPS-10 will load and start DTELDR which will then prompt you with:

```
*
```

DTELDR expects you to reply with a file specification (or assume a default); an action switch, and optional modification switches. The default file specification is:

STARTING THE SYSTEM

DSK:DTELxy.BIN

where:

x is the CPU number (only 0 is allowed in the current implementation). If x is omitted, zero is assumed.

y is the DTE number (0 through 3) of the PDP-11 interface. (DTE0 is reserved for the Console/Diagnostic processor.)

The action switch to load the front end is:

/RELOAD:xy

where x and y, defined as above, determine the front end to be loaded. RELOAD loads the specified file into the appropriate front end and starts the primary protocol. The front end is then enabled for communications.

Modification switches that may be applicable are:

/IMAGE The .BIN file is in image format.

/NODUMP The /RELOAD switch automatically takes a dump of PDP-11 memory before loading a file. Use this switch to cancel the dump.

/NOLOG Error logging is automatically in effect when you load a front end. Use this switch to cancel error logging.

To load a DN87S on DTE number 2 with a program located in DSK:DN87S.BIN and skip the automatic dump, type the following to the DTELDR prompt:

*DN87S.BIN/RELOAD:2/NODUMP

If the program had been located in the default location, DSK:DTEL02.BIN, you need only type:

*/RELOAD:2/NODUMP

Another action switch available with DTELDR is /AUTO. This switch, when set for an active DTELDR job, will automatically reload any front end that crashes with its unique default file. To take advantage of this feature, you must ensure that the default files contain the appropriate programs.

For example:

DSK:DTEL01.BIN must contain the program for the number 1 front end.

DSK:DTEL02.BIN must contain the program for the number 2 front end.

By including the following sequence in the OPR.ATO file (see Section 4.1.1.2), you can provide for both the automatic loading of a front end at system initialization time, and the subsequent automatic reloading of any front end that might crash.

```
:SLOG                ;log on a subjob
:DEF DTE=             ;name it DTE
DTE-R DTELDR          ;load and start DTELDR
DTE-/RELOAD:2/NODUMP  ;load DTE number 2
DTE-/AUTO             ;set automatic reload
```

STARTING THE SYSTEM

For additional information on DTELDR, type:

```
.R DTELDR
*/HELP
```

3.5 LOADING THE DX10 DATA CHANNEL

The PDP-8 in the DX10 data channel must be loaded to enable the operation of the TU70 series of magnetic tape systems.

The bootstrap program, BOOTDX, is designed to load (or dump) programs from the DECsystem-10 disk file system into the PDP-8. These programs must be in PDP-8 image mode file format.

BOOTDX runs in USER mode under a DECsystem-10 timesharing system. Since it does I/O instructions direct to the DX10 hardware, the BOOTDX job must have the TRPSET privilege allowing it to use the TRPSET monitor call. This privilege is assigned from the system accounting file, by the system manager.

3.5.1 Loading Procedure

The program, BOOTDX, normally resides in the system area, SYS:. Load it by typing:

```
.R BOOTDX
```

When BOOTDX is loaded and started, it responds with the prompt:

```
FILE:
```

and waits for a reply consisting of a standard DECsystem-10 file descriptor followed by one or more switches defining the operation to be performed.

If the file descriptor is omitted, the default is:

```
DSK:DXMPA.A8 or DSK:DXMPA.BIN (for a load operation)
```

or

```
DSK:DXMPA.LSD (for a dump operation)
```

If no switches are typed, the default is:

```
/CLEAR/LOAD
```

which first clears the PDP-8 and then loads the channel program into it.

Another commonly used switch is:

```
/CLEAR
```

which sets PDP-8 memory to all HALT instructions.

STARTING THE SYSTEM

----- Example -----

```
.R BOOTDX  
FILE:SYS:/CLEAR/START
```

will load and start the BOOTDX bootstrap program, clear PDP-8 memory, load the file DXMPA.A8 or DXMPA.BIN from SYS: and start it at location 200 octal. While this is going on, you will receive the following progress messages at your TTY:

"CLEARING DX-10

"DX-10 LOADING

"DX-10 LOADED

"DX-10 STARTED

CHAPTER 4

SYSTEM SERVICES

Timesharing requires that several programs be initiated to run throughout the day. Although the list of these programs may vary from installation to installation, the programs discussed in this section generally can be considered necessary for the efficient servicing of user commands.

The programs and commands explained below enable you to handle user-initiated requests such as: disk/DECTape file transfers; disk pack, magtape, and DECTape mount requests; line printer listings; paper tape output; and batch job requests. The variety of user requests will require you to supervise the running of several jobs simultaneously. You, therefore, must become proficient in each procedure and be able to execute each one with a minimum of delay.

4.1 OPSER

The OPSER program enables you to monitor and control up to 14 jobs, simultaneously, from a single terminal. These jobs are known as subjobs because they run under the control of another job; in this instance, OPSER. You, as the operator, can communicate with any active subjob directly or via OPSER.

4.1.1 OPSER Operation

OPSER is normally started automatically when the system is brought up. At this time OPSER reads and processes the file SYS:OPR.ATO (the Auto File). This file contains the OPSER commands that will initiate the running of system service programs such as input and output spoolers, components of the GALAXY batch processing system, and device-servicing programs.

If you want to start OPSER yourself, type:

```
.R OPSER
```

When OPSER responds with an asterisk, tell it to load and process the Auto File by typing:

```
*:AUTO SYS:OPR.ATO
```

OPSER will respond with an asterisk or an exclamation point, at which time you can give it any of the OPSER commands described later in this chapter.

SYSTEM SERVICES

It is important for you to remember to distinguish between typing commands to the OPSER program and typing directly to the subjobs.

4.1.1.1 **Commands to the OPSER Program** - You must precede each OPSER command by a colon (:) and you must type enough letters to make the command unique.

NOTE

The commands :MONITOR, :SLOGIN, :WHAT, and :WHERE are exceptions. The minimum number of letters you must type for each of these commands is as follows:

:MONITO
:SLOG
:W
:WHE

OPSER prompts you with either an asterisk (*) or an exclamation point (!) when it is ready to receive your commands. An asterisk indicates that no subjobs are active or a subjob is waiting for some action on your part. An exclamation point means that at least one subjob is currently active but you can still communicate with other subjobs.

There are four methods of specifying subjobs in an OPSER command string.

1. Type the subjob number (or numbers).
2. Type the mnemonic you assigned to the job with the :DEFINE command.
3. Use the word ALL, which indicates to OPSER that you want the command to apply to all active subjobs.
4. Leave the subjob specification out entirely, in which case OPSER assumes you intend the command to apply to the last subjob you referenced via (1), (2), or (3) above.

If the previous command referenced a list of subjobs, the last one referenced is the default for any subsequent command. For example, in the sequence,

:WHAT A, B, C
:STOP

C is the subjob implicitly referenced in the :STOP command.

If A, B, and C were the only active subjobs, and the following commands were typed,

:WHAT ALL
:STOP

C would again be the implicit argument of the :STOP command.

SYSTEM SERVICES

4.1.1.2 Commands Directly to Subjobs - When you give commands to a subjob, you must indicate for which subjob (or subjobs) the command is intended. Do this by preceding the command with the subjob number (or the name assigned via the :DEFINE command) and a hyphen. For example, to stop subjob 3 (LP1) you could type:

```
!3-STOP          ;using the subjob number
```

or

```
!LP1-STOP        ;using the subjob name
```

There is a method available in OPSER for sending the same command to all the subjobs at once. For example, if you want to know the status within each of the subjobs, instead of typing WHAT to every single subjob, you can do it with a single command. You simply type:

```
ALL-WHAT
```

and OPSER sends the WHAT command to each subjob.

Whenever you enter a command to a subjob, OPSER will timestamp and identify the entry as follows:

```
!hh:mm (xxx)
```

where xxx is the subjob number or name used in the preceding command line. For example, the following is representative of CTY output.

```
!S-R SPRINT      (operator command)
!13:20(S)        (OPSER timestamp)
/               (SPRINT prompt)
```

Sample CTY output included in this manual will usually omit timestamp entries in the interest of conserving space and reducing clutter. Actual OPSER output will be timestamped.

Consider the following sequence of commands:

<u>STEP</u>	<u>EXPLANATION</u>
.R OPSER	Run the OPSER program.
*:SLOG 2,167	Login a subjob under [2,167].
!:DEF FIL=	Associate subjob n with the name FIL.
!:WHAT FIL	Print out the status of FIL.
!FIL-R PIP	Tell FIL to run PIP.
*FILC.EXT=FILA.EXT,FILB.EXT	Use Pip to create one file from data in two other files.
!:STOP FIL	Return FIL to monitor mode.
!:KJOB FIL	Logout FIL.
!:EXIT	Cause OPSER to exit to the monitor.

SYSTEM SERVICES

The example above shows how to log in a subjob, give it a name, run a system program, and log out the subjob. Once the concept of subjobs is understood it is simple to give system commands and program commands to a subjob. The entire range of monitor and program commands can be sent to a subjob as long as the command is preceded by a subjob name or number followed by a hyphen. This becomes very important in explaining the service programs which are run under OPSER.

To CTRL/C a subjob a CTRL/A must be used, since CTRL/C under OPSER is equivalent to :EXIT. For example, to CTRL/C subjob FIL which is running PIP the command is:

```
FIL-^A<CR>
```

Whatever was being done under PIP would be allowed to finish, and then subjob FIL would be put at monitor level.

A useful feature of OPSER is the ability to automatically execute a predefined set of commands contained in a disk file. This is known as the AUTO feature of OPSER. For example, consider the following sequence of commands:

```
:SLOG 2,167
:DEF FIL=
:WHAT FIL
FIL-R PIP
FILC.EXT=FILA.EXT,FILB.EXT
:STOP FIL
:KJOB FIL
```

If the above commands were placed in a file named FIL.ATO, you could accomplish exactly what was done manually in the example above by entering the single OPSER command:

```
:AUTO FIL
```

Commands included in a file named OPR.ATO are automatically executed whenever the system is loaded.

OPSER is used to control a large part of system operation; it is important that you become proficient in its use.

4.1.2 OPSER Commands

Some OPSER commands are available to all users while others are available only to those with OPERATOR privileges.

4.1.2.1 OPSER Commands for the Operator - OPSER commands that are available to you at the operator's console are the following:

:JCONTh	Continues job n that was waiting for operator action.
:KSYS hh:mm	Stops timesharing at the time specified by hh:mm. This must be at least five minutes
:KSYS +hh:mm	later than current time. If +hh:mm is used,
:KSYS	it means the number of hours and/or minutes from the current time and must be at least five minutes. All users logging in receive a

SYSTEM SERVICES

message stating the time at which timesharing will end. Logged-in users receive similar messages when the :KSYS command is given, one hour before timesharing ends, then at 30,15,8,4,2 minutes, and 1 minute before timesharing is scheduled to stop. The messages are of the form: `TIMESHARING ENDS IN n MIN.` :KSYS works only for a [1,2] job running at the OPR terminal.

Note that :KSYS without an argument undoes the effect of :KSYS hh:mm or :KSYS +hh:mm.

The colon in the time specification is optional; both hh:mm and hhmm are valid.

<code>:RESTRICT dev:,...</code>	Restricts the specified devices to the operator's use. The operator can assign the devices to a user, but the user cannot reassign them to anyone but the operator.
<code>:SET BATMAX n</code>	Restricts BATCON (the batch system controller) to run only n jobs simultaneously. The range of n is 0-14 inclusive.
<code>:SET BATMIN n</code>	Reserves n jobs for BATCON subjobs.
<code>:SET LOGMAX n</code>	Restricts the system to run only n jobs.
<code>:SET RUN CPUn</code> <code>:SET RUN NO CPUn</code> <code>:SET RUN ONLY CPUn</code> <code>:SET RUN ALL</code>	Allows the operator of a multiprocessor system to turn processors on and off line. The n is the CPU number (0 or 1). The command :SET RUN adds the named processor to the system pool of running CPUs. :SET RUN NO removes the named CPU from the system pool. :SET RUN ONLY allows only the named CPU to run. :SET RUN ALL puts all the CPUs available into the system pool.
<code>:SET y</code>	Simulates the monitor command .SET y, where y is one of the following: BREAK, CORMAX, CORMIN, DATE, DAYTIME, MEMORY, NOMESSAGE, OPR, SCHED, TTY, VMMAx. See the equivalent monitor commands in Appendix A for an explanation.
<code>:UNRESTRICT dev:</code>	Returns to the free user resource pool a device that has been restricted. It is the complement of :RESTRICT.

4.1.2.2 OPSER Commands for all Users - The OPSER commands that follow are available to all users of the system.

<code>:AUTO filespec</code>	Processes the commands contained in the specified file immediately.
<code>:AUTO /hhmm filespec</code>	Processes the specified file at the next occurrence of hhmm today. If it is past hhmm today, the file will be processed immediately.

SYSTEM SERVICES

:AUTO /<hhmm filespec Processes the specified file at the next occurrence of hhmm today. If it is past hhmm today, the file will not be processed.

:AUTO />hhmm filespec Processes the specified file at the next occurrence of hhmm. If it is past hhmm today, the file will be processed at hhmm tomorrow.

NOTE

In the above commands, "filespec" stands for file specification and consists of a device, filename, extension, and a directory name in the form:

dev:file.ext [proj-prog].

where:

dev:	is any valid input device name. The default is DSK:.
file	stands for any 1- to 6-character filename. There is no default; you must supply a filename.
.ext	is any 1- to 3-character filename extension. The default is .ATO.
[proj-prog]	is the directory area that contains the file. The default is the directory indicated by the project-programmer number you used to log in.
:CLOSE	Closes the disk log file without opening a new one. (This is a transaction file where OPSER logs its activities).
:CONTINUE	Continues processing the AUTO file after being interrupted by a CTRL/C. This allows the operator to gain control of a subjob during AUTO file processing.
:CURRENT	Types the name of the current subjob if one has been defined; if one has not been defined, it types the number of the current subjob (the last one typed in).
:DAYTIME	Types the current date and time.
:DEFINE XXX=n	Gives subjob n the name XXX, which may be from one to six characters long. In general, short names are recommended. The subjob number may be omitted from the command string, in which case OPSER assumes the last subjob you referenced.

SYSTEM SERVICES

Restrictions

The first character of the name must be alphabetic and the second through sixth (if any) must be alphanumeric. The subjob name "B" is reserved as the mnemonic for BATCON and should not be assigned to any other subjob.

:DEVICE dev:log:n	Causes OPSER to exit to the monitor, where you can ASSIGN the device (dev:) on subjob (n) and also associate a logical name (log) with that device. The logical name is optional but if you omit it, you must tell OPSER. You do this by typing a colon (:) where the logical name would have appeared, e.g., :DEVICE DTA: :4. If you issue a .REENTER command before returning to OPSER, you will abort the .ASSIGN command. When you are ready to return to OPSER, type :CONT.
:ERROR n,m,...	Silences all output except error messages from subjobs n,m, etc. The :REVIVE command will resume reporting all messages. When OPSER is first started, :REVIVE is in effect.
:EXIT or CTRL/C	Exits to the monitor if no subjobs are active.
:FREE	Types the number of the first unused subjob.
:HELP	Types a text briefly explaining the commands and their formats.
:KILL n,m,...	See :KJOB.
:KJOB n,m,...	Logs out the specified subjobs n, m, etc., saving all their files. Also, the /Z switch is passed to the KJOB program in order to suppress the normal queuing done at logout time.
:LOGIN p,pn	Logs in a new subjob under project programmer number p,pn. If p,pn is not given, OPSER's project programmer number is assumed.
:MONITOR	Exits to the monitor even if subjobs are active.
:MSGLVL n	Determines whether or not the job status (JOBSTS) bits appear in the response to the :WHAT command. If n equals 0 the job status bits are included in the :WHAT output. If n equals 1 (default), the :WHAT output does not include these bits. The actual letters typed out represent the following:

SYSTEM SERVICES

L for logged in
M for monitor mode
I for input wait
O for output wait
A for JACCT.

:QUEUE line Initiates the first free subjob and sends the line (line) as a QUEUE command to the monitor.

:RESOURCES Lists the available system resources.

:REVIVE n,m,... Clears the effects of the :SILENCE, :TSILENCE and :ERROR commands for subjobs n,m,etc., and causes normal echoing of output from them. When OPSER is first started, :REVIVE is in effect.

:SCHED Types out the schedule bits as set by the operator. Bits are:

- 0 Regular timesharing.
- 1 No further LOGINs except from CTY.
- 2 No further LOGINs from remote terminals, and no answering of data sets.
- 4 Batch jobs only.
- 100 Device MOUNTs can be done without operator intervention.
- 200 Unspooled jobs allowed.
- 400 No operator coverage.

:SEND line Simulates the monitor SEND command.

:SILENCE n,m,... Suppresses all terminal and transaction log output from subjobs n,m,etc.

:SLOGIN n,m Same as :LOGIN, but also silences all terminal output until the job is logged in.

:STOP n,m,... Puts the subjobs n,m,etc., into monitor mode.

:SYSTAT xxx Runs SYSTAT with the optional arguments xxx under the first free subjob.

:TIME n Types out the total running time for job n since the last :TIME n command, followed by the integrated product of running time and core size.

:TLOG dev:file.ext[n,m] Closes the current OPSER transaction log, if any, and opens a new log, on the specified file. If no file is designated, the default is DSK:OPSER.LOG[1,2].

:TSILENCE n,m,... Suppresses all terminal output from subjobs n,m,etc., but continues output to the transaction log.

SYSTEM SERVICES

:TTYTST	Types out a test pattern consisting of the ASCII characters corresponding to octal 40 to 174 inclusive.
:WHAT n,m,...	Types the status of the specified subjob(s). This typeout includes a SYSTAT of each job with the current time, the time of the last input and the last output, a linear listing of the JOBSTS bits (if MSGVLV is set to 0), and the time of the next timed AUTO file.
:WHERE devn:	Types the physical location node/station number of the device dev in subjob n.

4.1.3 OPSER Operator Messages

The following messages from OPSER can appear on your operator terminal.

?OPRAFE AUTO FILE READ ERROR

OPSER encountered a read error while it was trying to process the auto file.

?OPRANI DEVICE CANNOT DO INPUT

Input cannot be performed on the specified device; either because it is not a valid device name or because it is a device that can only do output.

?OPRBFS BAD FILE SPEC

There is a syntax error in the file specification.

?OPRBTS BAD TIME SPECIFIED

There is a syntax error in the time specification. See Section 4.1.2 for information about the format of OPSER commands.

?OPRCKJ CANNOT BE KJOB'ED

OPSER is unable to log the job out. One cause of this condition is a job being over quota.

?OPRCMM CANNOT BE PUT IN MONITOR MODE

The operator attempted to return to the monitor (:EXIT or CTRL/C) but was unable to do so even after several tries. This condition indicates a system or a program problem.

?OPRCNU COMMAND NOT UNIQUE

The operator has entered an abbreviation that does not specify a unique command.

SYSTEM SERVICES

?OPRCUA CANNOT USE ALL

The operator typed a command with "ALL" as the argument. Certain commands do not accept "ALL" as an argument (e.g., :DEFINE). Refer to Section 4.1.2 for information about the OPSER operator commands.

?OPRDCO DEVICE CANNOT DO OUTPUT

The device specified in the :TLOG command is not an output device.

%OPRDNR DAEMON NOT RUNNING

The DAEMON program has not been initialized.

?OPRDUE DAEMON UWO ERROR

An error occurred when OPSER called DAEMON to do accounting. Consult the system administrator.

%OPRFMM :KSYS HAS A MINIMUM OF 5 MIN.

The operator specified an argument that was too small for the :KSYS command. The value of the "+hh:mm" form of the command must be +00:05 or greater. The value of the hh:mm form of the command must be at least five minutes later than current time. See Section 4.1.2.1 for details about the format of this command

?OPRJNA NO JOB NUMBER ASSIGNED

This message generally occurs in response to the :WHAT command. It happens in cases where the monitor does not know about the subjob specified in the command.

?OPRJRO JOB(S) RUNNING ON n,m,... TYPE ":KILL ALL" TO KILL THE SUBJOBS

The operator tried to :EXIT or CTRL/C from OPSER but subjobs n,m, etc. were active. The subjobs must be killed first.

?OPRLEF ENTER FAILURE code ON LOG FILE

An ENTER failure occurred when OPSER tried to open the transaction log file (:TLOG).

?OPRNAS NO ACTIVE SUBJOBS

The operator issued a command to a subjob but no subjobs were running.

?OPRNFP NO FREE PTY'S

There are no more pseudo-teletypes (PTY'S) available under which to run subjobs.

SYSTEM SERVICES

?OPRNFS NO FREE SUBJOBS

This message occurs in response to either the :FREE command or a command that uses a free subjob (e.g., :QUEUE). It happens when there are no free subjobs available.

?OPRNSM NO SUCH MNEMONIC

The mnemonic specified in the command is not associated with any of the subjobs. This is either because it was typed incorrectly or because the name had not previously been associated with a subjob via the :DEFINE command.

[OPRPAF PROCESSING AUTO COMMAND FILE]

OPSER is currently processing the auto file.

?OPRSJW IS STUCK WITH JACCT ON

The operator tried to return to the monitor (:EXIT or CTRL/C) but was unable to do so, even after several tries. This condition indicates a system or program problem.

?OPRSNA SUBJOB n IS NOT ACTIVE

A command was given to subjob n and it is not active.

%OPRSPR SUPERSEDING PREVIOUS :AUTO REQUEST

The operator issued another :AUTO command. This new one supersedes the one previously issued.

4.2 OMOUNT (FILE AND MOUNT/DISMOUNT)

OMOUNT is a program used to process FILE or MOUNT/DISMOUNT commands. FILE commands are used for file transfers to or from disk and DECTape. MOUNT/DISMOUNT commands are used for the physical mounting or dismounting of disk packs, magnetic tapes or DECTapes. The following discussion of OMOUNT is from the operator's viewpoint in responding to such requests. Refer to the DECSYSTEM-10 Operating System Commands manual for a discussion of MOUNT/DISMOUNT and FILE from the user's viewpoint. (Also see the OMOUNT specification in the Software Notebooks for additional operator information.)

Depending on the volume of file transfers, it may be advantageous for you to run two OMOUNT programs; one specifically for FILE commands to minimize the waiting period for completion of a user request. At monitor load time, when the OPSER program is automatically logged in under [1,2], one or more OMOUNT jobs can also be initialized with the appropriate commands in the OPR.ATO file.

You may also start OMOUNT jobs manually if the need arises. In the following example, DECTape unit 6 is assigned to an OMOUNT job named M6.

SYSTEM SERVICES

<u>STEP</u>	<u>EXPLANATION</u>
.R OPSER	Run the OPSER program.
*:SLOG 1,2	Log in a subjob under [1,2].
!:DEF M6=	Give the name M6 to the subjob.
!M6-AS DTA6	Assign DTA6 to job M6.
!M6-R OMOUNT	Run the OMOUNT program.
!M6-START	Cause OMOUNT to accept all FILE and MOUNT requests submitted by users.

A second OMOUNT job could be initiated by repeating all but the first step above and substituting, for example, the name M7 for M6 and DTA7 for DTA6. If this second OMOUNT job was for FILE requests only, the START command would be M7-START:FILE.

In most cases, your response to an OMOUNT request by a user is to perform the necessary action, and then type the subjob name or number followed by a hyphen and a carriage return. A typical example of a FILE command follows:

```
!08:17:56(M6)
  R JOB 13 TTY12 146,504 21 DSKB:,DSKB:CRBDOC.RNO
  MOUNT TAPE 21 [146,504] WRITE LOCKED ON DTA(6):
!M6-<CR>
```

Job M6 requested a DECtape to be mounted with the switch on WRITE LOCK. After you have mounted the tape on DTA6, simply type M6- followed by a carriage return. This one response covers 99% of the OMOUNT FILE requests. Experience and continual use of OMOUNT will familiarize you with the remaining 1% of requests. You can use a drive other than the one suggested (DTA6 in this case). If you choose to mount the DECtape on DTA5 for the above request, type M6-DTA5 or M6-5 instead of M6-<CR>. If you choose to mount it on DTB2, type M6-DTB2.

You will also have to process MOUNT and DISMOUNT requests such as:

```
!08:14:25(M6)
  M JOB 17 TTY201 1,2 1 MOUNT DTA ACCT /WL
  WRITE-LOCKED
  ASSIGN DTA (0)
!M6-<CR>
!08:18:13(M6)
  D JOB 17 TTY201 1,2 1 DISMOUNT DTA0
  PROCEED?
!M6-Y<CR>
```

The response to a MOUNT command is a <CR> as in the FILE command. At dismount time, you are asked the question PROCEED?. Answer with one of the following:

<u>Reply</u>	<u>Action</u>
NO	Cancel the request and delete it from the queue.
YES	Proceed if possible.

SYSTEM SERVICES

<u>Reply</u>	<u>Action</u>
CANCEL	Same as NO.
DEFER:t	Defer processing this request for the specified amount of time, where t=time in the form: mm minutes (1440 maximum) or hh:mm hours and minutes (24:00 maximum).
HELP	Type out list of valid replies to PROCEED question.
PAUSE	Wait for operator input after finishing the current requests.
START:x	Start processing x type requests after completing the current one. "x" can be FILE, MOUNT, NONE, or ALL. START with no argument assumes a default of ALL.
STOP:x	Stop processing x type requests after completing the current one. "x" can be FILE, MOUNT, NONE, or ALL. STOP with no argument assumes a default of ALL.
WAIT	Defer the request and wait for operator input.
WHAT	Retype previous command and all resulting TTY dialogue.

/START and /STOP may accompany other replies.

Responses to OMOUNT requests, either FILE or MOUNT, are normally self-explanatory. There are times, however, when some type of error is encountered and special action is required. When you receive an error message, stop the particular command causing the error, abort it, and notify the user about the problem. The most common errors occur on FILE commands when the DECTape is full, the DECTape directory is full, or an error is encountered on a particular file. The sequence of commands necessary to abort a command is as follows:

!:STOP M6	This OPSER command will cause job M6 to be returned to monitor level.
!M6-START	This passes the monitor command, START, to job M6 to reinitialize the OMOUNT program.
!M6-START	This passes the START command to OMOUNT to reprint the request that initiated the error.

NOTE

If the OMOUNT program was running specific requests, the START command should contain or imply the same argument as the original START command. (For example, START:FILE).

SYSTEM SERVICES

!M6-/C This command cancels the user request causing the error.

You would then send a message to the user informing him of the problem.

Refer to the OMOUNT specification in the Software Notebooks and the OMOUNT.HLP file for further discussion of the OMOUNT program.

4.3 QUEUE

QUEUE is a program which allows a user to add, schedule, list, modify, and delete queue entries in the various system queues. Although the QUEUE command is, for the most part, a user, rather than an operator oriented function, you should become familiar with all of its options and capabilities: QUEUE evokes many questions from users, and you must be ready to offer the maximum amount of assistance. A complete discussion of QUEUE is contained in the GALAXY Operations and System Programming Procedures manual of the Software Notebooks.

The QUEUE command queues user requests that are to be acted upon by corresponding spooler and batch programs. The following system queues are available:

QUEUE MNEMONIC	CONTENTS
LPT	Line printer requests
PTP	Paper tape punch requests
INP	Batch system requests
CDP	Card punch requests
PLT	Plotter requests

Because the list of QUEUE switches is extensive, the entire list will not be included in this document. The more commonly used switches will be discussed along with several examples. However it is important that you become familiar with all the switches contained in the QUEUE specification in the Software Notebooks.

The basic format of the QUEUE command is:

.Q queuename:=filespec/switch

In this format, queuename refers to the queue mnemonic, LPT, PTP, INP, CDP, or PLT. filespec refers to a disk file to be listed, punched on paper tape, card punched, plotted, or input to the Batch controller. switch refers to one or more of the allowable QUEUE switches; the most common ones are listed below.

SYSTEM SERVICES

SWITCH	MEANING
/AFTER:hhmm	The request cannot be processed until after the specified time (24-hour time).
/COPIES:n	Repeat the output n times where n = 1 - 63 inclusive.
/DEADLINE:hhmm	The request must be started before the specified time (24-hour time).
/DISPOSE:DELETE	Delete the file after spooling.
/DISPOSE:PRESERVE	Save the file after spooling.
/DISPOSE:RENAME	Rename the file into area [3,3] and delete after spooling.
/KILL	Remove the specified entry from the specified queue.
/LIMIT:n	Output is limited to the specified number of pages, cards, minutes or feet.
/LIST	List the entries in the queue.
/MODIFY/switch:arg	Change the specified switch entry to the new value "arg". Multiple switches can be altered using /MODIFY.
/PRIORITY:n	Give the request a priority where n=0 - 62 inclusive.
/TAPE:IMAGE	Punch the tape in image mode (PTP only).

The following switches apply only to the INP or Batch queue.

SWITCH	MEANING
/CORE:n	Set the maximum core that the job can use where n=decimal value of core in K.
/RESTART:0 or 1	If 0, job cannot be restarted after a system crash. If 1, job can be restarted.
/TIME:hhmmss	Specify the maximum amount of CPU time allowable for the job in hours-min-sec.

To further simplify the queue commands the following alternatives are available:

PRINT	equals	Q LPT:=
PUNCH	equals	Q PTP:=
SUBMIT	equals	Q INP:=
CPUNCH	equals	Q CDP:=
TPUNCH	equals	Q PTP:=
PLOT	equals	Q PLT:=

SYSTEM SERVICES

Some examples are given below:

<u>EXAMPLES</u>	<u>EXPLANATION</u>
.PRINT FILEA,FILEB	Enter files FILEA and FILEB into the line printer queue.
.PUNCH FIL.BIN/TAP:IMAGE	Enter the file FIL.BIN into the paper tape queue to be punched in image mode.
.SUBMIT FIL/AFTER:1800	Enter the file FIL.CTL into the input queue to be processed after 6 pm.
.SUBMIT [2,167]FIL=/KILL	Kill the entry FIL.CTL for user [2,167] from the input queue.
.Q	List all entries of all queues.
.PRINT FILEA/MOD/COPIES:3/FORMS:NARROW	Modify the print requests for FILEA to produce 3 copies on narrow forms.

It is good practice to examine the queues periodically checking for backlogs or jobs that are stuck in queue because of some hardware or software problem. You can then modify or remove requests or even alter /AFTER and /DEADLINE parameters to suit your operations schedule.

4.4 GALAXY BATCH SYSTEM

The GALAXY Batch system operates under the control of the DECsystem-10 Operating System, TOPS-10, increasing the efficiency of the system by processing jobs that do not require operator interaction. The types of jobs that you should consider for running in a batch environment are: large, long-running jobs; frequently run jobs; jobs requiring large amounts of data; and, jobs requiring no user interaction.

The GALAXY Batch system uses the following system components:

QUASAR	the system queue manager
BATCON	the batch job controller
QUEUE	the system queueing program
LPTSPL	the spooling processor for the line printer
SPROUT	the spooling processor for other output devices
SPRINT	the spooling processor for input devices

You can automatically load and start any of the above programs at monitor load time by including the appropriate OPSE commands in the SYS:OPR.ATO file.

4.4.1 QUASAR

QUASAR is the master controller in the GALAXY Batch system. All user queued and spooled requests are processed by QUASAR and directed to BATCON or the appropriate spooler.

There is no direct interaction between QUASAR and the operator. You communicate with QUASAR via commands to BATCON and the spoolers.

SYSTEM SERVICES

QUASAR communicates with you via messages prefixed with "CTQ" that are issued by BATCON or one of the spoolers.

If QUASAR is not automatically started at monitor load time, you can manually do so as follows:

```
.R OPSER      type this if OPSER is not running
*:SLOG        log in a subjob
*:DEF QUA=    give the subjob the name QUA
*QUA-R QUASAR load and start QUASAR
```

(now log in and load BATCON and the spoolers)

If QUASAR should stop running for any reason, you can simply restart it by reloading it in the same subjob, for example:

QUA-R QUASAR

However, since the new QUASAR does not know about the existence of BATCON and the spoolers, you must go through a routine of killing and restarting them, for example:

```
B-KILL        kills BATCON
L0-KILL        kills the printer spooler
SP-KILL        kills the input spooler
```

Each of the KILL commands causes any processing to stop immediately and sends a message to QUASAR. But since these jobs are unknown to the "new" QUASAR, it sends a message to you through them. The message is

?CTQMFQ MESSAGE FROM QUASAR - NOT A KNOWN COMPONENT

This means that although you have stopped the actual processing of the jobs, the original requests for them still exist in the queues. They will be processed in the normal way when the "new" QUASAR is put in contact with BATCON and the spoolers. This occurs as soon as the programs have been STARTed as follows:

```
B-START
L0-START
SP-START
```

4.4.1.1 Component-to-QUASAR Messages - The following messages are system error messages issued by the interface module which BATCON and the spoolers use to communicate with QUASAR. Most of these messages indicate serious system problems; you should bring them to the attention of the system programmer or system administrator.

?CTQCAP CANNOT ACQUIRE A PID

The program (i.e., BATCON or one of the spoolers) does not have privileges to acquire a system name from INFO. They must be run under [1,2] or a remote operator number [lxx.2].

?CTQFCI FAILURE TO CONNECT TO THE INTERRUPT SYSTEM

A failure occurred when the program (i.e., BATCON or one of the spoolers) attempted to enable a PSI system monitor call for IPCF interrupts. Consult the system programmer.

SYSTEM SERVICES

?CTQIRF IPCF RECEIVE FAILURE

The IPCFR. monitor call failed when the program (i.e., BATCON or one of the spoolers) tried to receive an IPCF message.

?CTQMFQ MESSAGE FROM QUASAR - xxx

The xxx stands for one of the following messages:

ILLEGALLY FORMATTED DEVICE NAME

The operator specified a device name in a format that is unacceptable to the program (i.e., BATCON or one of the spoolers).

ILLEGAL MESSAGE TYPE

The component (i.e., BATCON or one of the spoolers) sent a message to QUASAR in an invalid format. This indicates a problem with the program that sent the message. Consult the system programmer.

INSUFFICIENT PRIVILEGES ENABLED

An attempt was made to run BATCON or one of the spoolers by a user who does not have sufficient privileges to run these programs. BATCON must be run under [1,2]. The spoolers must be run either under [1,2] or through a remote operator project-programmer number. The format for a remote operator project-programmer number is [l_{nn},2] where nn represents the remote station number. For example, a remote operator at station 36 would log in with a [p,pn] of [136,2]. In addition, LPTSPL must always be run from SYS:

NOT A KNOWN COMPONENT

QUASAR is not in contact with the component (i.e., BATCON or one of the spoolers) that tried to send a message. This condition generally indicates one of two problems. Either the component itself is defective or QUASAR crashed and when it was run again, the operator did not put the components back into contact with it. Consult the system programmer.

SPECIFIED REQUEST IS NOT YOURS

QUASAR received a message that does not apply to the request it is currently processing. This condition indicates a difficulty with the program that sent the message. Consult the system programmer.

UNKNOWN QUEUE SPECIFIED

QUASAR received a message to process a queue request but the message specified a queue that is not known to QUASAR. This condition indicates a difficulty with the program that sent the message. Consult the system programmer.

SYSTEM SERVICES

?CTQNGS NO GALAXY-10 SUPPORT IN THIS MONITOR

The monitor currently running does not support the GALAXY system. It should be a 6.02 or later monitor and have VMSER, PSISER, and IPCSER.

?CTQSIF SEND TO INFO FAILED

The program (i.e., BATCON or one of the spoolers) tried to send an IPCF message to INFO and the IPCFS. monitor call failed. This is indicative of an IPCF problem. Consult the system programmer.

?CTQSOF SEND TO QUASAR FAILED

The program (i.e., BATCON or one of the spoolers) tried to send an IPCF message to QUASAR and the IPCFS. monitor call failed. This is indicative of an IPCF problem. Consult the system programmer.

4.4.2 BATCON

BATCON is a system program that reads entries from the system batch queue and initiates and controls batch jobs. Entries in the system batch queue (INP:) consist of user disk files containing monitor commands, system program commands, data normally entered from a terminal, and GALAXY Batch system commands. The actual format of these files will not be discussed here, but the operator should become familiar with these control files by reading the GALAXY Batch System documentation and actually creating them. This section deals with how the operator controls and responds to BATCON to ensure maximum efficiency of the program.

BATCON can be initialized automatically at monitor load time along with the previously mentioned spooler jobs. It is recommended that it be controlled through OPSER. The operator interaction with BATCON consists mainly of changing certain parameters at preset times during the day to regulate BATCON's scheduling of jobs. BATCON schedules jobs based on certain parameters for both individual jobs and total number of jobs able to be run. A list of these parameters along with commonly used BATCON commands is presented below. For a complete discussion of all possible commands the operator is referred to the GALAXY Batch System documentation in the Software Notebooks.

4.4.2.1 BATCON Commands - The general format of a command string directed to BATCON through OPSER is:

Bn-command argument

where

B is the letter that must precede all commands issued to BATCON. B must be the OPSER defined subjob name for BATCON if BATCON is being run under OPSER.

SYSTEM SERVICES

n indicates the BATCON subjob number. If you omit the n in your command string, BATCON assumes the command is intended for the last subjob you specified in a previous command. You may substitute the word ALL for n. In this case, all active batch subjobs are the subjects of the command.

Some BATCON commands are general (i.e., do not apply to a specific subjob) and hence do not require the subjob number. In these cases you should omit the subjob number. The format of the command then becomes:

B-command argument

command is one of the commands described below. Commands may be abbreviated to as few letters as are necessary to indicate a unique command.

NOTE

There is an exception to this rule. ST is an accepted abbreviation for START. Therefore, if the command STOP is desired, at least STO must be typed.

argument varies for each command and may be a number, a string of characters, a key word, or it may not be required.

If a command requires an argument, it must be specified; otherwise, BATCON will issue an error message. The default value for an argument will be assumed only when the command itself is not given.

The commands available to the operator fall into three categories:

1. Commands to set parameters for Batch jobs (refer to Table 4-1),
2. Commands to display the current parameter settings (refer to Table 4-2), and
3. Commands to control Batch subjobs (refer to Table 4-3).

SYSTEM SERVICES

Table 4-1
Commands to Set Parameters for Batch Jobs

Command	Explanation
CORE n	Limits the total core available for all Batch jobs to nK, where n represents a decimal number in the range of zero (0) to the amount of swapping space available. The default is 1.5 times the amount of physical memory the system makes available to the user. (See note 1.)
EXIT	Ceases selection of new jobs, waits for currently active jobs to be completed, and then returns to the monitor.
MCORE n	Limits to nK the core available for an individual Batch job. The n stands for a decimal number in the range from zero (0) to CORMAX (the parameter set by the OPSER command ":SET CORMAX" or the value built into the system at MONGEN time. The default is the value of CORMAX. (See note 1.)
MJOB n	Limits the number of Batch jobs that can be run concurrently to n, which represents a decimal number in the range zero (0) to the value of the JOBMAX assembly parameter. (Refer to the BATCON specification in the Software Notebooks for more information about JOBMAX.)
MTIME n	Limits each individual Batch job to n seconds of CPU time, where n is a decimal number in the range zero (0) to 262,143 seconds. Any Batch jobs in the input queue that have estimates larger than n will not be selected for execution by BATCON. The default is no limit.
NEXT n	Selects the job with the queue sequence number n as the next job to be processed.
RESET	Stops scheduling new jobs, waits for all currently active jobs to be completed, and returns to operator command wait. At this point, all parameters are RESET to their default settings.
START	Begins Batch processing. You can also use this command to cancel the effect of a RESET or EXIT command.
TIME n	Limits the total amount of CPU time available to all Batch jobs to n seconds, where n represents a number in the range zero (0) to 262,143 seconds. The default is no limit.

Note 1: This command is not available if no core enforcement was specified at GALAXY system generation time.

SYSTEM SERVICES

Table 4-2
Commands to Display the Current Parameter Settings

Command	Explanation
CURRENT	Type out the current values of the parameters set by the commands in Table 4-1 above. Also, type out the number of the current default subjob, the number of active job streams and any exceptional conditions that affect the Batch System.
HELP	List all the valid BATCON commands. Only the commands themselves will be listed; detailed descriptions are given in this section of the Operator's Guide.
MONJOB	Type out the correspondence between the Batch subjob number and the monitor job number.
WHAT	Type out a description of the current job: the subjob number, the monitor job number, the job name, the project-programmer number, the program running, its state and the amount of CPU time used. In addition, this command causes the last line of text entered into the specified job's LOG file to be displayed.

Table 4-3
Commands to Control Batch Subjobs

Command	Explanation
EXAMINE n	Display the next n lines of the control file of the specified subjob on the operator's terminal. Only the lines contained in the current core buffers can be displayed. n is a positive, non-zero, decimal number and must be specified. If n exceeds the number of remaining lines in the control file, the display will stop at the end of the file.
GO message	Continue the specified subjob that has either been halted by the STOP command or is waiting for a response. The message is optional, but if present, it will be placed in the LOG file of the specified subjob and be indicated as a comment.
KILL argument;message	<p>Cancel the specified subjob. Argument must be one of the following:</p> <p>FLUSH - cancel without error recovery or automatic queueing of output (passes the switch /Z:0 to KJOB).</p> <p>ERROR - cancel with normal user error recovery.</p> <p>NERROR - cancel without error recovery.</p>

SYSTEM SERVICES

Table 4-3 (Cont.)
Commands to Control Batch Subjobs

Command	Explanation
	<p>If the argument is not specified, KILL ERROR is assumed.</p> <p>The message is optional, but if present, it will appear as a comment in the LOG file of the specified subjob.</p> <p style="text-align: center;">NOTE</p> <p>If the argument to this command is missing, any message must be preceded by a semicolon (;) or an exclamation point (!).</p>
OPERATOR message	<p>Send the supplied message to the specified subjob. This command is the response to dialogue mode and has no effect unless the user has included the Batch command, .OPERATOR, within the control file. Dialogue mode is the operator-program communication facility provided by BATCON.</p>
REQUEUE hh:mm	<p>Return the specified subjob to the system input queue for processing later. The argument hh:mm is used to give the subjob an AFTER parameter (e.g., B1-REQUEUE 1:00, requeues subjob 1 and provides an AFTER parameter of one hour). If the argument is missing, the assembly parameter REQTIM is used. Consult the system administrator to determine the value of the assembly parameters.</p> <p style="text-align: center;">NOTE</p> <p>When specifying an AFTER time, if only one set of numbers is specified, minutes will be assumed. It is only when two sets of numbers, separated by a colon are specified that hours and minutes are assumed. Examples are given below.</p> <p>B1-REQUEUE 20</p> <p>requeues subjob 1 and provides an AFTER parameter of 20 minutes.</p> <p>B11-REQUEUE 1:30</p> <p>requeues subjob 11 and provides an AFTER parameter of 1 hour and 30 minutes.</p>

SYSTEM SERVICES

Table 4-3 (Cont.)
Commands to Control Batch Subjobs

Command	Explanation
B3-REQUEUE 100	requeues subjob 3 and provides an AFTER parameter of 100 minutes (1 hour and 40 minutes).
ROUTE a TO b	Route all output destined for station "a" to station "b". Station "a" refers to the user's logical location as logged in or changed via the LOCATE command. a and b are 2 digit octal node numbers in the range, 01-77.
	WARNING
	Use discretion when building control files using the ROUTE command; this command is likely to be changed or superseded in the future.
STOP message	Halt the specified subjob. The job will not be resumed until a GO command is issued. The message is optional but if present, it will appear as a comment in the LOG file of the specified subjob.
TELL message	Enter the supplied text as a comment in the LOG file of the specified subjob.

4.4.2.2 **BATCON Operation** - BATCON is normally initialized automatically at monitor load time requiring no operator intervention. If for some reason you must start BATCON manually under OPSER, the following procedure, with appropriate commands and arguments, should be followed:

<u>STEP</u>	<u>EXPLANATION</u>
!SLOG 1,2	Log in an OPSER subjob under [1,2].
!DEF B=	Give the subjob the name B. It is very important to note here that the letter B is reserved for BATCON and that no other subjob may be so defined.
!B-R BATCON	Run the BATCON program.
!B-MJOB 4	Limit number of jobs to 4.

SYSTEM SERVICES

<u>STEP</u>	<u>EXPLANATION</u>
!B-MCORE 56	Limit single job core to 56K.(See note 1)
!B-CORE 200	Limit total job core to 200K.(See note 1)
!B-MTIME 3600	Limit single job CPU time to 3600 seconds.
!B-TIME 7500	Limit total job CPU time to 7500 seconds.
!B-START	Start scheduling jobs.

Once BATCON is started, you need only to adjust parameters to maintain optimum system performance. At various times during the day the following parameters might be either increased or decreased depending upon the current system load.

MJOB
MTIME
MCORE(See note 1)
TIME
CORE(See note 1)

The intended effect of adjusting these parameters is to prevent BATCON from overloading the system during peak times, and to defer the larger jobs to the slack times of the day. You should be familiar with the effect the various commands have on BATCON.

A user control file includes all the instructions necessary to complete a specified task. As part of the control file a user may include FILE, MOUNT/DISMOUNT, and PLEASE commands. The FILE, MOUNT, and DISMOUNT commands require no special action by you as far as BATCON is concerned. They are handled through the OMOUNT jobs along with regular interactive users. The PLEASE commands do require a special response in order for the job to continue. When a subjob issues a PLEASE command, the output is directed to the terminal controlling BATCON. Upon receipt of such a command you must take whatever action is necessary and then indicate to the subjob that you have done what was necessary by typing Bn-ESCAPE where n is the subjob name or number. An example of this process follows:

```
!18:42:33 (B)
      SUBJOB# 1      JOB#6
      PLEASE USE DECTAPE 2-167-1 FOR NEXT MOUNT
      WAITING....
!B1-$
```

Your only response in the example above was B1-ESCAPE (ESCAPE prints as \$) to indicate successful completion of the PLEASE request. At times you may need to send a message to the user who submitted the batch request. The TELL command is used to enter such a message into the user's LOG file. The LOG file contains a running transaction of the progress of the Batch job and is printed upon completion of the Batch job. In the above example if you are unable to find the user DECTape you might use the following command to inform the user:

Note 1: Not available if the DEFCOR assembly parameter is zero (0).

SYSTEM SERVICES

B1-TELL CANNOT FIND TAPE 2-167-1 WILL ABORT MOUNT COMMAND
B1-\$

It is very important that you monitor the BATCON job and its subjobs periodically to ensure that all jobs are running and are not hung in a wait or ^C state. If a job is waiting for operator action, periodically a message will be typed on your operator terminal. The message contains information about the job and the text, "WAITING FOR RESPONSE", followed by several bells. The frequency of this message normally is one minute. However, this feature may be changed or removed entirely by resetting the assembly parameter, PROMPT.

To monitor a BATCON job, you should use the following commands:

! :WHAT B	Prints out the status of BATCON (an OPSER command)
!B-CURR	Prints out the current values of the BATCON parameters.
!BALL-WHAT	Prints out the status of all BATCON subjobs
!Bn-WHAT	Prints out the status of BATCON subjob n

You can then determine from the output of these commands any irregularities and take the necessary action to correct the problem. If for some reason you must abort a Batch job, the following action is required:

BALL-WHAT

From the information printed out, you can determine the Batch subjob number. Knowing the subjob number is very important. You must be absolutely sure that you are aborting the correct one.

Bn-KILL

This causes subjob n to be aborted and uses the default argument ERROR which allows for error processing.

Whenever you use the TELL or KILL command a record of this appears in the user's LOG file. It is therefore very important that you use the TELL command to inform the user of any abnormalities encountered. This command does not hinder the processing of the job but serves as a means of communication with the user even though he is not necessarily present on the system.

4.4.2.3 BATCON Operator Messages - The following messages from BATCON can appear on your operator console.

ARGUMENT IS OUT OF RANGE

The numeric argument specified is greater than or lesser than the limits permitted for that command.

BATCON UNABLE TO KJOB

The job could not be KJOBed. This message is issued after BATCON has made two attempts to send a KJOB command. Further messages give alternative actions.

SYSTEM SERVICES

COMMAND IS AMBIGUOUS

The operator has entered an abbreviation that does not specify a unique command.

ILLEGAL NUMBER FORMAT

Illegal characters appear in a field that must contain only numerics.

ILLEGAL SUBJOB SPECIFIED

The operator specified a subjob number that was less than 1 or beyond the assembly parameter JOBMAX.

JOB CANNOT BE PUT IN MONITOR MODE

A subjob did not return to monitor mode after it sent a ^C. Further messages give alternative actions.

MISSING ARGUMENT OR ILLEGAL DELIMITER

A command that requires an argument was entered without one or with a punctuation error.

NO SUBJOBS ARE ACTIVE

A subjob command has been issued but no subjobs are currently running.

RESPOND Bn-GO (or KILL)

Subjob n is waiting for a response. This message usually appears after a PLEASE command.

RESPOND Bn-OPR line

Subjob n is requesting a response to dialogue mode.

SUBJOB IS NOT ACTIVE

"n" specifies a subjob that is not currently running.

SUBJOB #n JOB #n

This is a general heading to clarify the source of more messages.

UNKNOWN BATCON COMMAND

The command entered cannot be recognized by BATCON.

Under certain exceptional conditions, there may be no LOG file for BATCON to write to (e.g., after a KJOB command is issued). When this is the case, any output sent from the subjob is typed (with a heading) on your operator console. Although this can sometimes be annoying, there is a good chance that the output will contain error information that you will need in diagnosing the problem that occurred (e.g., ?SWAP READ ERROR during KJOB/LOGOUT).

SYSTEM SERVICES

4.5 OUTPUT SPOOLERS

The output spoolers access the system output queues and act upon specific requests to create the specified output.

The output spoolers consist of LPTSPL for line printer output and SPROUT for card punch, paper tape punch, or plotter output.

The spooler programs are usually initiated by OPSER automatically at monitor load time (depending on the contents of the auto file) and are controlled from the terminal along with several other system jobs. The commands in the examples discussed in this section apply to LPTSPL, the line printer spooler, and to the card punch option of SPROUT:

In the examples below, The card punch job is called CP. The LPTSPL job which outputs to LPT0 is called L0 and the LPTSPL job which outputs to LPT1 is called L1. These names are arbitrary, but they become very important in the issuing of commands to the various spoolers. Because these jobs are run under OPSER, the job name determines the destination of the command.

If the spoolers are automatically initiated, you need only start these jobs from scratch in the case of some error. The following steps explain how to initialize the line printer and card punch spoolers.

<u>STEP</u>	<u>EXPLANATION</u>
!:SLOG 1,2	Log in a subjob under [1,2].
!:DEFINE L1=	Give this subjob the name L1.
!L1-ASSIGN LPT1	Assign LPT1 to job L1.
!L1-R LPTSPL	Run the line printer spooler program.
!L1-MLIMIT 100	Set the initial page limit on LPT1 to 100.
!L1-START LPT1:	Begin outputting to LPT1.
!:SLOG 1,2	Log in a subjob under [1,2].
!:DEFINE L0=	Give this subjob the name L0.
!L0-ASSIGN LPT0	Assign LPT0 to job L0.
!L0-R LPTSPL	Run the line printer spooler program.
!L0-MLIMIT 2000	Set the initial page limit on LPT0 to 2000.
!L0-START LPT0:	Begin outputting to LPT0.
!:SLOG 1,2	Log in a subjob under [1,2].
!:DEFINE CP=	Give this subjob the name CP.
!CP-R SPROUT	Run the SPROUT program.
!CP-MLIMIT 2000	Set the initial limit on the card punch to 2000 cards.

SYSTEM SERVICES

<u>STEP</u>	<u>EXPLANATION</u>
!CP-LOCK	Pause after each job.
!CP-MESSAGE FILE	Print file identification as each file is processed.
!CP-CARDS	Start the card punch spooler.

Once the spoolers have been started, they will automatically act upon requests that have been queued by the users. You need alter parameters or issue spooler commands only to ensure the most efficient operation of the spooler devices. A list of commands is given below. For each command the list gives the name of the command, an explanation of the effect of the command, and the spooler(s) to which the command applies. LPTSPL or SPROUT implies that the command is applicable only to the line printer or SPROUT devices respectively; ALL implies that the command can apply to any output device.

COMMAND	EXPLANATION	SPOOLER
ALIGN file	Output a test file for a specific form on the line printer. This command is used to aid the operator in checking the alignment of the paper with the carriage tape. The default file is <current form type>.ALP, unless there is a /ALIGN switch for the form in SYS:LPFORM.INI See Section 4.5.1.	LPTSPL
BACKSPACE n	Backspace the file n pages and begin printing there.	LPTSPL
CARDS	Start SPROUT processing files queued to the card punch.	SPROUT
CHKPNT label	Save the line in the control file identified by "label" as a potential restart point in the event of a system failure.	ALL
CURRENT	Print out the current value of all parameters that can be controlled by the operator.	ALL
EXIT	Cause the spooler to return to the monitor if the spooler is idle. If this command is given while the spooler is active, an error message is issued.	ALL

SYSTEM SERVICES

COMMAND	EXPLANATION	SPOOLER
FORMS formsname	Inform the spooler that form "formsname" is now mounted. The default is NORMAL.	ALL
FORWARD n	Forward space the file n pages and begin printing there.	LPTSPL
FREEZE	Cause the currently mounted forms (cards, line printer, paper tape, or plotter paper) to be "frozen". The spoolers will not schedule a different type of form.	ALL
GO	Continue processing after a LOCK, PAUSE, or STOP has been issued.	ALL
HELP	Type out a list of all the spooler commands.	ALL
KILL	Abort the current job and go to the next job.	ALL
LIMIT n	Change the limit for the current job to n pages, cards, feet, or minutes.	ALL
LOCK	PAUSE automatically after each job.	ALL
MESSAGE	Type only the specified messages on the operator's console. where:	ALL
ERROR	prints all fatal errors that occur while processing.	
FILE	prints the file identification of each file processed.	
JOB	prints job information at the beginning and end of each job. (LPTSPL only)	
ALL		
NONE		

SYSTEM SERVICES

COMMAND	EXPLANATION	SPOOLER
	ALL prints the information specified on all three of the above.	
	NONE cancels any previous arguments of the MESSAGE command.	

NOTE

The MESSAGE command supports multiple arguments separated by commas. However, the arguments are processed in order from left to right. Therefore if conflicting arguments are specified, the last valid one will apply. For example:

MESSAGE ALL, FILE No conflict; ALL applies

MESSAGE FILE, JOB, NONE Conflict; NONE Applies

MLIMIT n	Do not run any job which outputs more than n pages, feet, or cards, or which uses more than n minutes of plotter time. This command has no effect on the job currently being processed. (n is a decimal number)	ALL
NEXT n	Select the job with the queue sequence number n as the next job to be processed.	ALL
NOSUPPRESS	Remove the effect of the SUPPRESS command immediately. NOSUPPRESS is in effect at the time the spooler is initialized.	LPTSPL
PAUSE	Finish processing the current job and then stop.	ALL
PLOT	Start SPROUT processing files queued to the plotter.	SPROUT
REPRINT	Stop processing the current copy of the current file immediately and start it over.	ALL

SYSTEM SERVICES

COMMAND	EXPLANATION	SPOOLER
REQUEUE/switches	<p>Stop writing now and place the job back into the queue for later processing.</p> <p>Note that different switches apply to different spoolers as indicated in the right-hand column. The switches are optional and can be omitted.</p> <p>The default for LPTSPL is to start reprocessing at the current position of the file.</p> <p>The default for SPROUT is to start reprocessing at the beginning of the job.</p>	ALL
Switches:		
	/A:n Start reprocessing after n minutes.	ALL
	/B:n Start reprocessing n pages back in the file.	LPTSPL
	/F:n Start reprocessing n pages forward in the file.	LPTSPL
	/H Start reprocessing after 12 hours.	ALL
	/T Start reprocessing at the beginning of the job. This is the default for SPROUT.	ALL
RESET	<p>Reinitialize the spooler.</p> <p>If the spooler is idle when a RESET command is issued, the command will be processed.</p> <p>If jobs are active, an error message is issued.</p>	ALL

SYSTEM SERVICES

COMMAND	EXPLANATION	SPOOLER
SKPCOPY	Stop processing the current copy of the current file immediately and begin the next copy. If the last or only copy is being processed at the time this command is issued, start the next file in the request.	ALL
SKPFILE	Stop processing the current copy of the current file immediately and begin processing the next file in the request.	ALL
START output-dev=queued-dev	<p>Immediately start processing files queued to the queued device and output them on the output device.</p> <p>If no output device is specified, LPTSPL assumes LPT: as the default.</p> <p>If no queued device is specified, LPTSPL assumes the output device as the default.</p>	LPTSPL
STOP	Stop immediately.	ALL
<p style="text-align: center;">NOTE</p> <p>"ST" is an acceptable abbreviation for the START command. Therefore, if STOP is desired, at least "STO" must be typed.</p>		
SUPPRESS/switch	Cause the spooler to ignore all vertical paper motion characters except for the line feed. This command has the following two switch options that control the scope of the command. The FILE option is the default.	LPTSPL
Switch:		
FILE	The SUPPRESS command applies only to the current file. This is the default.	

SYSTEM SERVICES

COMMAND	EXPLANATION	SPOOLER
JOB	The SUPPRESS command applies to the entire print job.	
TAPE	Starts SPROUT processing files queued to the paper-tape punch.	SPROUT
UNFREEZE	Remove the effect of the FREEZE command.	ALL
UNLOCK	Remove the effect of the LOCK command so that the spooler will continue automatically after each job.	ALL
WHAT	Type out on the operator's console the present status of the spooler. This information consists of the job name, the user's name, his project-programmer number, the job's sequence number, and information about the file being processed.	ALL

Once you have familiarized yourself with the general spooler commands it becomes important to know when to use them. Following is a discussion of some applications of the spoolers.

4.5.1 LPTSPL

LPTSPL is the program which handles all PRINT requests submitted for the line printer through QUEUE. In the discussion below, two line printers are assumed to be in use. LPT0 prints, say, at a rate of 1000 lines per minute and LPT1 prints 600 lines per minute. Because of the difference in speed, a higher limit might be set on the job controlling LPT0. This does not mean, however, that the limit on LPT1 must always be lower than LPT0.

The commands which follow could apply to both jobs L0 and L1 although the examples will be based on one or the other.

Initially, when the monitor is loaded, both spooler jobs could be started automatically with a line printer assigned to each and with the following page limits:

L0-MLIMIT 2000

L1-MLIMIT 100

SYSTEM SERVICES

During the normal daily operation, your only interaction with the spoolers would be to increase or decrease the MLIMIT of a particular job. This is not normally done during prime time unless the spoolers are idle and a user must have a listing which is over the current limit. In this case you would type:

```
L0-MLIMIT n
```

where n is the new value of MLIMIT.

LPTSPL is so designed that when an out-of-paper condition on a line printer is detected, it will notify you, and then wait until the situation is corrected. The spooler will then continue automatically. You need not issue any commands. All you must do is correct the problem and the software will do the rest. In the event of such a condition on LPT1, you will receive the following message on the terminal:

```
!13:27:05(L1)
  % DEVICE LPT1 IS NOT READY
!
```

Once the problem is corrected the line printer will automatically continue.

There are some error conditions which may cause the spooler to be returned to monitor level. These conditions are obscure and would serve no useful purpose being explained here. The spooler will output an error message indicating the problem, and the end result is that the spooler will be returned to monitor or ^C level. If this occurs, the spooler program must be restarted before it can continue processing requests. When in doubt as to the state of the spooler, you need only type:

```
:WHAT L0
```

to print out the status of the job. If the spooler is in a ^C state it must be restarted as follows:

<u>STEP</u>	<u>EXPLANATION</u>
!L0-R LPTSPL	Run the spooler program.
!L0-MLIMIT 1020	Set the page limit.
!L0-START LPT0:	Start processing print requests.

It is very simple to remove completed listings without hampering the operation of the line printer. You may, however, stop and start the spooler to remove listings or clear up a jam. Use the commands:

```
!L0-STOP
```

```
!L0-GO
```

These are the only commands needed to stop and continue the spooler program.

The MLIMIT parameter can be raised during a period that is not very busy, to allow large listings to be printed. You can ask the spooler to type out its current status to ensure that MLIMIT has been changed correctly. The following example shows the MLIMIT being raised to 3500 and then checked:

SYSTEM SERVICES

<u>STEP</u>	<u>EXPLANATION</u>
!L0-MLIMIT 3500	Raise the page limit to 3500.
!L0-CURRENT	This causes the spooler to type out the current values of MLIMIT and other operator-set parameters.

There may be a time when the printer is down for repairs. If files still need to be output, perhaps to free some disk space, they can temporarily be output to magnetic tape and printed at a later time. The commands to divert LPT0 output onto MTA0 are:

<u>STEP</u>	<u>EXPLANATION</u>
!L0-R LPTSPL	Have subjob L0 run LPTSPL.
!L0-START MTA0:=LPT0:	Set the output device to MTA0.
Then, after the line printer is repaired, requeue the files onto the printer queue as follows.	
!L0-EXIT	Exit to the monitor at the end of the current job.
!L0-REWIND MTA0:	Rewind the tape.
!L0-R PIP !L0-LPT0:/X=MTA0:*. !L0-^A	Using PIP, transfer the files from magnetic tape to the printer queue. CTRL/A to the subjob acts like a CTRL/C to return to monitor.
!L0-R LPTSPL	Reload LPTSPL.
!L0-START LPT0:	Start the line printer spooler.

Occasionally a user will inadvertently QUEUE a file to be printed which is not in the correct form such as a binary file. If it is discovered before printing of the file begins, the print request may be deleted from the QUEUE via the QUEUE/KILL command discussed earlier. When this is not discovered until the file begins to print, the line printer will rapidly eject pages, print groups of unrecognizable characters, or continually hang up. You must then use the spooler KILL command as follows:

!L0-KILL

This will abort the current print request and proceed to the next request. This command must be used with caution as it can cause the user's disk file to be deleted along with the request.

As mentioned earlier, when a system is operating with operator coverage, the spooler jobs have the line printers assigned. This means that a user may not output directly to the line printer even when the spoolers are idle. While this is reasonable as long as there is operator coverage, it is not desirable during those times when the system is running with no operator coverage. The following steps describe the action necessary to release the line printers from assignment to the spooler jobs and make them available for general use. The spoolers will output to the line printers upon receipt of a queue request even though the line printers are not assigned to the spooler jobs.

SYSTEM SERVICES

<u>STEP</u>	<u>EXPLANATION</u>
!:STOP L0	Stop the spooler.
!L0-FINISH LPT0:	The FINISH command effectively deassigns the line printer.
!L0-R LPTSPL	Run the spooler program.
!L0-START LPT0:	Ready the spooler to output to LPT0 upon receipt of a print request.

The steps above would be repeated for job L1 substituting L1 for L0 and LPT1: for LPT0:.

The special forms handler in LPTSPL provides a means by which various parameters may be set automatically on the basis of form names. This is accomplished through the use of a disk file, SYS:LPFORM.INI, which contains a list of forms and default switches.

Each line in LPFORM.INI is of the form:

formname/sw1/sw2/sw3...

or

formname:locator/sw1/sw2/sw3...

formname	a 1- to 6- character form name. The system administrator is free to specify form names descriptive of the types of forms used at the installation.
----------	----------------------------------------------------------------------------------------------------------------------------------------------------

locator	a field used to describe one or more printers to be affected by various parameters. Legal values for the locator field are:
---------	-----------------------------------------------------------------------------------------------------------------------------

ALL	all line printers
LOCAL	all local line printers
REMOTE	all remote line printers
Snn	all printers at station nn (octal)
LPTn	the specific line printer n

If the locator field is omitted, ALL is assumed. If more than one locator field is used, each must be separated by a comma and the list enclosed in parentheses. Some examples are given below:

formname:LOCAL/sw1/sw2

formname:(LOCAL,S2)/sw1/sw2

/sw1/sw2/sw3...	one or more switches used to describe the manner in which the forms are to be printed.
-----------------	----------------------------------------------------------------------------------------

SWITCH

MEANING

/ALCNT:nn	nn specifies the number of times that the forms alignment file is to be printed. This switch is used in conjunction with the ALIGN command.
-----------	---------------------------------------------------------------------------------------------------------------------------------------------

Typing any character to LPTSPL will terminate forms alignment printing.

SYSTEM SERVICES

SWITCH	MEANING
/ALIGN:ss	ss is a 1- to 6-character ASCII string that specifies the name of the default forms alignment file (extension.ALP) to be used to position this forms type. This switch is used in conjunction with the ALIGN command.
/ALSLP:nn	nn specifies the number of seconds between each printing of the forms alignment file. This switch is used in conjunction with the ALIGN command.
/BANNER:nn	nn specifies the number of banner pages (job headers) desired.
/CHAIN:ss or /DRUM:ss	ss is a 1- to 6-character ASCII string that specifies the chain or drum to be used on the line printer. The switch name and argument are typed on the operator's console when the forms are scheduled. If both the /CHAIN and the /DRUM switches are specified for the same entry in the LPFORM.INI file, the spooler will type only the last one it encountered.
/HEADER:nn	nn specifies the number of file headers (picture pages) desired.
/LINES:nn	nn specifies the number of lines to be printed on the banner, trailer, and header pages. The argument is used for internal calculations and does not affect the file to be printed.
/NOTE:aa	aa is a message of up to 50 characters that is sent to the operator at the time the forms are scheduled.
/PAUSE	Causes the spooler to pause before each job using the type of form specified. This switch overrides the UNLOCK operator command.
/RIBBON:ss	ss is a 1- to 6-character ASCII string that specifies the type of ribbon to be used on the line printer. The switch name and argument are typed on the operator's console at the time that the forms are scheduled.
/TAPE:xxx	xxx is a 1- to 6-character ASCII string that specifies the name of the vertical forms unit control to be used. If the printer has an optical vertical forms unit (uses paper carriage tapes), a message is sent to the operator to mount the carriage tape named xxx. If the printer has a direct access vertical forms unit (DAVFU), LPTSPL will attempt to load the DAVFU with the file SYS:xxx.VFU. If LPTSPL finds the file, it notifies the operator that the file is being loaded. If the file cannot be found, an error message informs the operator that the job has been requeued. If the /TAPE switch is not specified, the name defaults to NORMAL.

SYSTEM SERVICES

SWITCH	MEANING
/TRAILER:nn	nn specifies the number of job trailer (pages) desired.
/VFU:xxx	This switch performs the same function as /TAPE:xxx.
/WHAT	Causes the spooler to type a short WHAT message to the operator for each job printed. This message consists of the job name, the user's name and project-programmer number, and the sequence number.
/WIDTH:nn	<p>The value of nn is used for internal calculations to determine a width class. The width class specifies the number of characters per line to be printed on the banner, header, and trailer pages. There are three width classes.</p> <ol style="list-style-type: none">1. Width class 1 (nn=0-60) specifies that up to 66 characters per line will be printed on the banner, header, and trailer pages.2. Width class 2 (nn=61-100) specifies that up to 90 characters per line will be printed on the banner, header, and trailer pages.3. Width class 3 (nn=101-132) specifies that up to 129 characters per line will be printed on the banner, header, and trailer pages.

The LPFORM.INI file is searched until the first occurrence of a line that contains both the name of the forms currently mounted and a locator field (either implicit or explicit) that is in agreement with the printer specified in the START command. The parameter switch settings on this matching line are then used. So, the sequence,

```
NARROW/sw1/sw2      (equivalent to NARROW:ALL/sw1/sw2)
NARROW:LOCAL/sw1/sw2
```

would never reach the second line, since the first line will always match.

The spooler will request the operator to change the forms only if the first four characters of the new form name differ from the first four characters of the old form name. This allows entries in LPFORM.INI of the form:

```
NARROW/HEADER:1/BANNER:1/TRAILER:1
NARR01/HEADER:0/BANNER:1/TRAILER:1
NARR02/HEADER:0/BANNER:0/TRAILER:0/WHAT
```

The spooler will schedule any of the three types of forms, NARROW, NARR01, or NARR02, without asking the operator to change the forms (provided that one of the specified forms already is mounted). However, the switches will be different for each of the specified forms.

SYSTEM SERVICES

-----Example-----

Assume that the LPFORM.INI file contains the following entry:

LABELS:LPT1/CHAIN:96CHRS/RIBBON:CARBON/VFU:LBLDAV/NOTE:USE 4-UP LABELS

If you issue a command such as:

PRINT NAFILE/FORM:LABELS

The print request is queued on the printer queue. When the request is scheduled to be processed, LPTSPL searches the LPFORM.INI file for a formsname matching the /FORM: switch argument. Finding the LABELS entry, LPTSPL outputs the following:

MOUNT LABELS FORMS THEN TYPE GO

You will also be reminded of the switch settings via the following messages:

[DRUM(CHAIN):96CHRS]
[RIBBON:CARBON]
[NOTE:USE 4-UP LABELS]

If LPT1 has a direct access vertical forms unit (DAVFU), LPTSPL attempts to load it with the file, LBLDAV.VFU. If the file is found, the following message is issued:

[LPTLVF LOADING VFU LBLDAV IN LPT1]

If the file cannot be found, the following message is issued:

%LPTCFV CANNOT FIND VFU LBLDAV FOR LPT1, REQUEUEING JOB

You can then either locate the file, kill the job, or change the /VFU: switch.

If LPT1 does not have a DAVFU, LPTSPL assumes that a paper carriage tape is to be mounted. You will receive the following message:

PLEASE PUT VFU TAPE LBLDAV IN LPT1

4.5.1.1 LPTSPL Operator Messages - The following messages from LPTSPL can appear on the operator's console.

?LPTAWN ACCOUNTING WILL NOT BE DONE - msg

Where msg represents one of the following two messages:

SYSTEM SERVICES

1. NO PRIVILEGES - The operator has no privileges to write the FACT file, and
2. DAEMON NOT RUNNING - The DAEMON program has not been initialized. It must be started by the operator to allow entries to be written into the FACT.SYS file.

?LPTCAF CAN'T ACCESS FILE filename, CODE nnn - xxx

LPTSPL encountered an error while looking up the file it was trying to print. nnn and xxx are the code number and text respectively of one of the error codes listed in the Monitor Calls manual.

%LPTCCF CAN'T CHANGE FORMS IN THE MIDDLE OF A JOB

The operator issued a FORMS command while a job was being processed. Type PAUSE and wait for the current job to be completed.

%LPTCCL CCL ENTRY IS NOT SUPPORTED

LPTSPL does not support CCL entry (i.e., RUN with an increment of 1). It is treated as a standard RUN.

?LPTCFA CAN'T FIND ALIGN FILE: filename

LPTSPL was unable to find the specified alignment file. The file must be on SYS:.

%LPTCFD CANNOT FIND DEFAULT VFU FILE FOR dev, LOADING HARDWARE VFU AS DEFAULT

LPTSPL can't find the default VFU file and is loading the hardware VFU instead. The operator should realign the forms and place the printer on line.

%LPTCFV CANNOT FIND VFU filename FOR dev, REQUEING JOB

LPTSPL cannot find the specified DAVFU file for the device. If the file is not available, kill the job or change the LPFORM.INI file entry to remove that DAVFU reference.

[LPTCPC CLEARING PENDING EXIT, RESET, AND PAUSE COMMANDS]

The operator issued a "START" command while LPTSPL was already started and there were other commands pending. LPTSPL clears the command(s) and continues processing.

?LPTDDE DEVICE dev DOES NOT EXIST

The device specified in the START command does not exist. The device designation was typed incorrectly or the device was not assigned to this job.

SYSTEM SERVICES

?LPTDIS DEVICE dev IS SPOOLED

The device specified in the START command is spooled. LPTSPL can write only on physical devices.

%LPTEWL ERROR WRITING LOG FILE, STATUS nnn

LPTSPL encountered an error while it was trying to write out the LOG file.

[LPTFAF formsname FORMS ARE FROZEN]

Informs the operator that the specified forms are now frozen.

%LPTFPM FORMS PARAMETER FILE LPFORM.INI IS MISSING

LPTSPL can't find the LPFORM.INI file and will use the default values.

?LPTFSD FILE SKIPPED DUE TO I/O ERRORS, STATUS nnn

LPTSPL skipped the file it was processing because it encountered too many I/O errors. The allowable number of error recoveries (%LPTIDE), before this message is issued, is a conditional assembly parameter of LPTSPL and is set by the system administrator. The default number is five.

?LPTICA ILLEGAL COMMAND ARGUMENT arg

The argument is either misspelled or is not a valid argument for this command.

%LPTIDE INPUT DATA ERROR nnn, RECOVERY ATTEMPTED

An error was encountered while LPTSPL was reading the file it was trying to output. The nnn reflects the I/O status.

%LPTLAS LPTSPL IS ALREADY START'ED ON dev

The operator issued a START command to a device that was already running LPTSPL.

[LPTLIR LPTSPL IS RESET ON dev]

The copy of LPTSPL that was processing on another device has been RESET on the specified device by the operator and is awaiting a new START command.

[LPTLVF LOADING VFU filename IN dev]

This message informs the operator that LPTSPL is loading the DAVFU file filename into the device dev.

SYSTEM SERVICES

[LPTLWP LPTSPL WILL PAUSE AT END OF JOB]

The operator issued a PAUSE command to LPTSPL while it was in the middle of a job. This message occurs in response to a WHAT command.

[LPTSIS SPOOLER IS STOP'ED OR PAUSE'ING]

This message is output in response to the WHAT operator command.

%LPTURC cmd IS AN UNRECOGNIZED COMMAND

The operator issued a command that is unknown to LPTSPL.

?LPTVFE VFU ERROR ON dev, PLEASE REALIGN THE FORMS

The device dev had a format error in the DAVFU and the printer is now off-line. Reset forms to top-of-form position before turning printer on-line.

[LPTWFF WAITING FOR formsname FORMS TO BE MOUNTED]

The spooler is waiting for the operator to mount the specified forms. The operator must change the forms and type GO. This message is output in response to the WHAT operator command.

[LPTWFS WAITING FOR A START COMMAND]

The operator must issue a START command to cause LPTSPL to begin processing. This message is output in response to the WHAT operator command.

4.5.2 SPROUT

SPROUT, the spooling processor for output, is a program that handles all files queued to spooled output devices other than the line printer; both those submitted through QUEUE and spooled requests generated by user programs. The spooled devices for which SPROUT processes files are the paper-tape punch, the card punch, and the plotter. Once SPROUT is running, QUASAR, the master controller, handles the scheduling of jobs to be processed.

Your primary responsibilities are to start up one or more copies of SPROUT to process output files on appropriate devices; adjust various system parameters when necessary to ensure optimum system performance; monitor the supply of tape, cards, and/or plotter paper; attend to jams in the devices; and, distribute the output if it is the practice at your installation for the operator to do so.

One or more copies of SPROUT are usually brought up automatically (via the OPR.ATO file, Section 4.1.1.2) when the TOPS-10 monitor is loaded. If it becomes necessary to start SPROUT manually, you may use the procedure(s) given below. Remember that a separate SPROUT job must be loaded and started for each type of spooled output device. You cannot have files queued to more than one type of output device running under a single copy of SPROUT.

SYSTEM SERVICES

If OPSEER is not already running, start it up by typing R OPSEER. Next, type one or more of the following procedures to OPSEER.

```
*:SLOG 1,2      log in a subjob
!:DEF CP=       give the subjob the name CP
!CP-R SPROUT    load SPROUT
```

and when SPROUT prompts with a slash, type

```
!CP-CARDS      start the card-punch spooler
```

To start the plotter spooler:

```
*:SLOG 1,2      log in another subjob
!:DEF PL=       give this subjob the name PL
!PL-R SPROUT    load SPROUT
```

and when SPROUT prompts with a slash, type

```
!PL-PLOT      start the plotter spooler
```

Should any of the spooled devices supported by SPROUT become inoperative, output can be diverted to magnetic tape. For example, if the card punch goes down you can type the following:

```
*CP-R SPROUT
!CP-CARDS MTA0:=CDP0:
```

and the card images intended for the card punch will be diverted to magnetic tape device 0.

For additional information, including error procedures, refer to the SPROUT chapter in the "GALAXY Operations and System Programming Procedures" in the Software Notebooks.

4.5.2.1 SPROUT Operator Messages - The following messages from SPROUT can appear on the operator's terminal.

?SPOAWN ACCOUNTING WILL NOT BE DONE - msg

Where msg represents one of the following two messages:

1. NO PRIVILEGES - The operator has no privileges to write the fact file, and
2. DAEMON NOT RUNNING - The DAEMON program has not been initialized. It must be started by the operator to allow entries to be written into the FACT.SYS file.

?SPOCAF CAN'T ACCESS FILE filename, CODE nnn=xxx

SPROUT encountered an error while looking up the file it was attempting to output. nnn and xxx refer to the number and text respectively of one of the error codes listed in the Monitor Calls manual.

%SPOCCF CAN'T CHANGE FORMS IN THE MIDDLE OF A JOB

The operator issued a FORMS command while a job was being processed. Type PAUSE and wait for the current job to be completed.

SYSTEM SERVICES

%SPOCCL CCL ENTRY IS NOT SUPPORTED

SPROUT does not support CCL entry (i.e., RUN with an increment of 1). It is treated as a standard RUN.

[SPOCPC CLEARING PENDING EXIT, RESET, AND PAUSE COMMANDS!] The operator issued a START class command while the spooler was already STARTed and there were other commands pending. The spooler clears the command(s) and continues processing.

?SPODDE DEVICE dev DOES NOT EXIST

The device specified in the start-class command (i.e., CARDS, PLOT, or TAPE) does not exist; either because it was typed incorrectly or because the device was not assigned to this job.

?SPODIS DEVICE dev IS SPOOLED

The device specified in the start-class command (i.e., CARDS, PLOT, or TAPE) is spooled. SPROUT can write only on physical devices.

[SPOFAF formsname FORMS ARE FROZEN!]

Informs the operator that the specified forms are now frozen.

?SPOFLE FORMS LIMIT EXCEEDED ON DEVICE dev

SPROUT has processed a file on a device which has just exceeded its output forms limit.

?SPOFSD FILE SKIPPED DUE TO I/O ERRORS, STATUS nnn

SPROUT skipped the file it was processing because it encountered too many I/O errors. The allowable number of error recoveries (%SPOIDE) before this message is issued is a conditional assembly parameter of SPROUT and is set by the system administrator. The default number is five.

?SPOICA ILLEGAL COMMAND ARGUMENT arg

The argument is either misspelled or is not a valid argument for this command.

%SPOIDE INPUT DATA ERROR nnn, RECOVERY ATTEMPTED

An error was encountered while SPROUT was reading the file it was trying to output. The nnn reflects the I/O status.

?SPOODE OUTPUT DEVICE ERROR ON dev, STATUS nnn

An error was encountered while SPROUT was trying to write on the specified device.

SYSTEM SERVICES

%SPORUU REQUEST USES UNIMPLEMENTED FORMAT

An output request was queued that specified an option that is not available in the current version of SPROUT.

%SPOSAS SPROUT IS ALREADY STARTED ON dev!]

The operator issued a start-class command (i.e., CARDS, PLOT, or TAPE) when SPROUT was already processing on the device.

[SPOSIS SPOOLER IS STOP'ED OR PAUSE'ING]

This message is output in response to the WHAT operator command.

[SPOSIR SPROUT IS RESET ON dev]

The copy of SPROUT which was processing on another device has been RESET by the operator on the specified device and is awaiting a new start-class command (i.e., CARDS, PLOT, or TAPE).

[SPOSWP SPROUT WILL PAUSE AT END OF JOB!]

The operator issued a PAUSE command to SPROUT while it was in the middle of a job. This message is output in response to the WHAT operator command.

[SPOWFF WAITING FOR formsname FORMS TO BE MOUNTED!]

The spooler is waiting for the operator to mount the specified forms. The operator must change the forms and type GO. This message is output in response to the WHAT operator command.

[SPOWFS WAITING FOR START-CLASS COMMAND!]

The operator must issue a start-class command (i.e., CARDS, PLOT, or TAPE) to cause SPROUT to begin processing. This message is output in response to the WHAT operator command.

4.6 INPUT SPOOLER (SPRINT)

SPRINT is a DECsystem-10 system program which reads any sequential input stream and enters jobs into the BATCON input queue. Normally the input stream is from a card reader, but disk and magnetic tape also can be handled. SPRINT analyzes the input stream for card reader spooler control cards (column 1=\$) and processes them. The remaining cards are copied to various files for processing by the user job. SPRINT creates the job's LOG file and enters a report of processing into this file. Other information may be entered into this file either by means of user control cards or by means of operator intervention.

SPRINT may be run either as a directly-controlled timesharing job or as a subjob of OPSER.

SYSTEM SERVICES

To start SPRINT running as a timesharing job, type:

```
.R SPRINT
```

to which SPRINT responds with:

```
/
```

To start SPRINT running as an OPSER subjob, type the following to OPSE:

```
*:SLOG 1,2
*:DEF S=
*:S-R SPRINT
```

OPSER responds with the timestamp

```
!hh:mm(S)
```

followed by SPRINT's response

```
/
```

When SPRINT responds with a slash, SPRINT is ready to receive commands from the operator's console. The following sample of console log output is the result of loading and starting SPRINT and then querying its status. The SPRINT commands are described in detail below.

!S-R SPRINT	operator command to load SPRINT
!13:20(S)	OPSER timestamp
/	SPRINT prompt
!S-START	operator command to start SPRINT
!13:20(S)	OPSER timestamp
!	SPRINT prompt
!S-WHAT	operator command to query subjob
!13:21(S)	OPSER timestamp
SPRINT IS IDLE ON CDRO	SPRINT response
!	SPRINT prompt

The START command tells SPRINT to begin processing. The format of the START command is:

```
START dev:filename.ext
```

If the entire argument is omitted, the command /START is interpreted as /START CDR:. If only the filename and extension are typed, the command /START MYFILE.SPT is interpreted as /START DSK:MYFILE.SPT.

SPRINT specifically supports three devices for input; the card-reader, magnetic tape and disk. Each of the three input devices is handled differently with respect to input mode and in response to end-of-file as described in Table 4-4.

SYSTEM SERVICES

Table 4-4
Input Device Characteristics

Device	Mode	Response To EOF
CDR	Super-Image	Tries to read next job. If none is found, sleeps for 10 seconds, and tries again.
MTA	ASCII	Waits for another START command.
DSK	ASCII	Waits for another START command.

SPRINT responds to all commands with either a slash (/) or an exclamation point (!). A slash is used when SPRINT is not processing and is awaiting operator input. An exclamation point is used when SPRINT is processing a job but is also waiting for a type-in.

Before each card, disk block, or tape block is read, SPRINT checks for the presence of an operator command. If a command has been typed, SPRINT processes it immediately (except as noted in the description of the command). You may continue a command on a subsequent line (or lines) by using the standard continuation character, the hyphen (-) as the last non-TAB, non-space, or non-comment character on the line (or lines) to be continued. SPRINT responds to a continuation with a number sign (#) prompt. The following sequence of commands illustrates the use of the continuation character;

```
/START DSKB:INPUT.-  
#SPT[77,77]
```

4.6.1 SPRINT Commands

The SPRINT commands are:

COMMAND	DESCRIPTION
EXIT(See note 1)	Go to monitor command level. This command takes effect at the end of the current job.
GO	Continue after a PAUSE or a STOP command.
HELP	Type this list of commands and their actions.
KILL(See note 1)	Kill the current job now and flush the input stream until the next job is encountered.
MCORE n	Set the maximum allowable core limit to nK for all subsequent jobs. This command takes effect at the end of the current job.
MSGVLV abc	Controls which messages SPRINT types to the operator.

Note 1: If this command is issued while SPRINT is stopped in the middle of a job, it will cause an automatic GO.

SYSTEM SERVICES

COMMAND	DESCRIPTION
	<p>a=0 Type short messages to operator. 1 Type long messages to operator. b=0 Suppress user error cards. 1 Type the first fatal error card of the job. 2 Type each error card. c=0 Do not type any cards to the operator. 1 Type each \$JOB card. 2 Type each \$ card. 3 Type all cards.</p> <p>The default is 100 (type long messages but do not type user errors or cards.)</p>
PAUSE	Pause at the end of the current job.
RESET(See note 1)	Re-initialize SPRINT. This has the same effect as CTRL/C CTRL/C START (^C ^C START). This command is delayed to the end of the current job.
START	Start SPRINT processing. This command may not be typed once SPRINT has begun processing unless a RESET is typed first and no jobs are active.
STOP	Stop reading immediately and wait for further commands. The GO command will cause processing to continue with no ill effects.
TELL msg	This command places the message (msg) in the current job's LOG file.
WHAT	Gives the current status of the job.

4.6.2 SPRINT Operator Messages

The following is a list of the SPRINT diagnostic and error messages.

%SPTAWN ACCOUNTING WILL NOT BE DONE - msg

Where msg represents one of the following two messages:

1. NO PRIVILEGES - The operator has no privileges to write the fact file, and
2. DAEMON NOT RUNNING - The DAEMON program has not been initialized. It must be started by the operator to allow entries to be written into the FACT.SYS file.

?SPTBFA BAD FORMAT FOR ACCOUNTING FILE

The accounting file ACCT.SYS is not in a format understood by SPRINT. SPRINT only recognizes ACCT.SYS formats 2, 3, and 4.

Note 1: If this command is issued while SPRINT is stopped in the middle of a job, it will cause an automatic GO.

SYSTEM SERVICES

?SPTCCC CAN'T CREATE CTL OR LOG FILE - msg

SPRINT is unable to create either the control or LOG file. msg represents one of the messages explained in the error message section of the Commands Manual in the Software Notebooks.

?SPTCCI CAN'T CLEAR UFD INTERLOCK

SPRINT invoked a software interlock when creating a user's UFD, and could not release the interlock afterwards. Consult your system programmer about this problem.

%SPTCCL CCL ENTRY IS NOT SUPPORTED

SPRINT does not support CCL entry (i.e., RUN with an increment of 1). It is treated as a standard RUN.

%SPTCDI DEVICE dev CAN'T DO INPUT

Input cannot be performed on the device specified in the START command. For example, input may have been requested for a device that can do only output, such as the line-printer (e.g., START LPT:).

?SPTCEC CAN'T EXPAND CORE

SPRINT requested an amount of core that the monitor was unable to give it.

?SPTCER COMMAND ERROR - RETYPE LINE

There is a syntax error in the command string.

%SPTCME CARD-MOTION ERROR ON CDRn

A card was not read correctly. Reset the last card and press the RESET button on the card reader.

?SPTCOD CAN'T OPEN THE DISK

An OPEN call on device DSK failed. This is generally a symptom of a more serious monitor problem.

?SPTCRA CAN'T READ ACCOUNTING FILE

SPRINT encountered difficulties while attempting to read the accounting file. Consult the system administrator.

?SPTDDI DOUBLE DIRECTORY ILLEGAL

Two directory names cannot appear without an intervening filename.

?SPTDDV DOUBLE DEVICE ILLEGAL

Two names appeared in a row without an intervening filename (e.g., LPT:PTP:) or, two colons appeared in a row (e.g., DSKA::FILEX).

?SPTDEX DOUBLE EXTENSION ILLEGAL

Two extensions cannot appear in a row without an intervening filename or comma (,).

SYSTEM SERVICES

?SPTDFN DOUBLE FILENAME ILLEGAL

Two filenames appeared in a row (e.g., \$COBOL TEST1 TEST2) or, two periods appeared in a row (e.g., TEST..CBL).

[SPTDNR INPUT DEVICE dev NOT READY]

The specified input device requires operator action to prepare it for reading.

%SPTDTM DATA MISSED ERROR ON CDRn

A card was not read correctly. Reset the last card and press the RESET button.

%SPTDUF DAEMON UO FAILED - CODE nnn

An error existed when SPRINT called DAEMON to do job accounting. Refer to the DECsystem-10 Monitor Calls manual for an explanation of the error code nnn. Report any occurrences of this message to your system administrator.

%SPTDVE INPUT DEVICE ERROR ON CDRn

A card was not read correctly. Reset the last card and press the RESET button on the card reader.

?SPTERA ERROR READING ACCOUNTING FILE

SPRINT encountered difficulties while attempting to read the accounting file. Consult your system administrator.

%SPTHPE HOPPER EMPTY OR STACKER FULL ON CDRn

One of the following two conditions exist:

1. The input hopper is empty and SPRINT is expecting more cards because the end of job has not been reached, or
2. The output stacker is full and SPRINT cannot read any more cards until room is made for them.

In either case, correct the condition and press the RESET button.

?SPTIDS ILLEGAL DIRECTORY SPECIFICATION

The format of the directory information enclosed within the square ([]) or angle (<>) brackets is incorrect.

?SPTILU ILLEGAL LUUO IN SPRINT

This is caused by an internal error in SPRINT. Rerun the job.

?SPTIOE I/O ERROR WRITING LOG OR CTL FILE

An I/O error occurred while writing the LOG or the CTL file. The job must be rerun.

?SPTLKE LOOKUP/ENTER ERROR nnn - xxx

SPRINT encountered an error while looking up a file specified by the operator in the START command. nnn and xxx

SYSTEM SERVICES

refer to the number and text respectively of one of the messages explained in the DECsystem-10 Operating System Commands Manual (DEC-10-OSCMA-A-B-D).

?SPTNCA NO CORE FOR ACCT.SYS INDEX

SPRINT builds an in-core index of the accounting file ACCT.SYS. This message indicates that SPRINT could not expand its core area sufficiently to build the index.

?SPTNDV NULL DEVICE ILLEGAL

A colon has been found without a preceding device name.

?SPTNHS NO HIGH SEGMENT - TYPE R SPRINT

SPRINT was not invoked via a RUN or a GET command or a RUN monitor call. Reload, save it, and then START or RUN.

?SPTNJA NO JOBS ACTIVE

A command was given by the operator to perform an action (KILL, TELL, etc.) for the current job, but SPRINT was not processing any jobs at the time the command was issued.

?SPTNSD NO SUCH DEVICE dev

The device does not exist or it was not assigned to this job.

%SPTNST SPRINT IS NOT START'ED

The operator must type START (refer to the START command in section 4.6.1) to cause SPRINT to begin processing.

?SPTNYP SPECIFIED PPN IS NOT YOURS

The project-programmer number on the \$JOB card is different from the one used to log in. This occurs only when a user, rather than the system operator is running SPRINT.

[SPTOIR OPERATOR INTERACTION REQUIRED]

This is the response to the WHAT command if operator interaction is required to continue processing.

%SPTPKF PICK FAILURE ON CDRn

The card about to be read cannot be drawn through the card reader. Check the first card in the input hopper and take appropriate action. To continue reading, replace the card, or a duplicate, back in the input hopper and press the RESET button.

%SPTRCK READ CHECK ON CDRn

A card was not read correctly. Reset the last card and press the RESET button on the card reader.

[SPTRLC RESET THE LAST CARD AND PRESS THE RESET BUTTON]

A card went through the card-reader but was not read correctly. This message is always preceded by one of the following four messages:

SYSTEM SERVICES

1. %SPTCME
2. %SPTDTM
3. %SPTDVE
4. %SPTRCK

?SPTROM REMOTE OPERATOR MUST RUN SYSTEM SPRINT

The operator at a remote Batch station attempted to run a version of SPRINT that did not reside on SYS:.

?SPTSND SFD NESTING TOO DEEP

The directory specification in a filespec required a directory path which contained more than the maximum permissible number of SFD's.

%SPTSPL DEVICE dev IS SPOOLED

The input device specified in the START command is spooled. SPRINT should read only from physical devices.

%SPTSTD SPRINT IS ALREADY START'ED ON dev

A START command was issued when SPRINT was already processing on the device.

?SPTUIE UNRECOVERABLE INPUT DEVICE ERROR - STATUS nnn

SPRINT received an input error indication from which it could not recover. The I/O status was nnn.

4.7 OPERATOR/USER COMMUNICATION

The communication between the operator and an interactive user is handled by two commands: the PLEASE command and the SEND command. Both commands are discussed below.

4.7.1 PLEASE

The PLEASE command provides for two-way communication between you and the user. It is initiated by the user typing at monitor level:

.PLEASE text<CR>

The text is output on your operator terminal (CTY). After you have performed whatever action is requested, you can answer the user by simply typing a message followed by a <CR>.

Whenever you receive a message via the PLEASE command, the user issuing the command is identified by job number, project programmer number, terminal number and name. For example:

User [40,64] types the following line:

.PLEASE MOUNT SA029 ON DTA3 W/E <CR>

The message is transmitted to the CTY and appears as follows:

**PLEASE JOB 44 [40,64] TTY150 USER SPS
MOUNT SA029 ON DTA3 W/E

SYSTEM SERVICES

You mount the tape and answer:

OK GO AHEAD <CR>

Your reply appears on the user's terminal to inform him of the completion of his request.

At this time either you or the user may type ESCape to return both terminals to the monitor state. It is essential that an ESCape be typed to allow your CTY to receive another PLEASE command. The next important point you must remember is that a job must never be logged in on the operator's PLEASE command terminal. This will cause the user to receive a busy message from PLEASE and his command will not be transmitted. It is also important that you respond to the PLEASE commands as quickly as possible for in most cases the user cannot proceed to other work until his request is completed.

4.7.2 SEND

The SEND command provides a means of one-way inter-terminal communication. One line of information at a time is transmitted from one terminal to the other. The format of the SEND command is:

.SEND TTYn:text where n= the TTY line number

.SEND JOB n text where n= the user's job number

You, as the operator, can use this command to transmit information to either a particular user or to all users simultaneously. A user, however, can only initiate a SEND command to the operator or to another user. In the case of sending to one particular user, you would first determine the correct terminal or job number using SYSTAT and then send a message from the CTY as follows:

.SEND TTY150 I CANNOT FIND TAPE 5

The user on line 150 would receive the following message on his terminal:

;;CTY: - I CANNOT FIND TAPE 5

In the event of a prospective monitor reload, you can inform all users as follows:

.SEND ALL THE SYSTEM WILL BE RELOADED IN 5 MINS

All terminals that are turned on will receive:

;;CTY: - THE SYSTEM WILL BE RELOADED IN 5 MINS

When a user sends you a message, the CTY will receive

;;TTY150 - TAPE 5 IS ON TOP OF THE DTA CABINET

You will know which terminal initiated the command by the line number.

You have special privileges that the user does not have. You can send a message to any terminal regardless of the state of the job on that terminal. A user will receive a ?BUSY message if the receiving terminal is not at monitor command level and the GAG switch is set (SET TTY GAG is the initial state).

The SEND command does not require a job to be logged in to be initiated from a terminal.

CHAPTER 5

OPERATOR TASKS

The procedures explained in this chapter are performed periodically to provide the installation with information about system usage and performance and a means of ensuring the integrity of the system. Most of the procedures can be accomplished using the GALAXY Batch System, thereby requiring a minimum of operator intervention. However, there are some procedures that must be performed manually because of specific program responses required of the operator. The tasks as they are described in this chapter include explanations of what the programs do and why they are run, along with the steps required to execute them. It will be your responsibility to be able to run all these procedures without using the Batch System, if the need arises.

5.1 DAILY TASKS

The following operator tasks are usually run on a daily schedule.

5.1.1 Update the Message of the Day

The file NOTICE.TXT, located in the SYS:[1,4] area of the disk, contains messages that are transmitted to users as they log in. These messages, normally preceded by the date on which they first appear, are usually brief and contain important system information. For example, a message may specify the hours that the system will be on the air, or state when the disks will be refreshed, or mention new system features. You can change NOTICE.TXT during the day to warn users of problems on the system, to notify them of your intention to stop timesharing, or to direct users to a more lengthy file for additional information.

To change, add, or delete a message in the file NOTICE.TXT, you can log in on [1,2] and run an editing program such as TECO. (Refer to Introduction to TECO and TECO in the Software Notebooks.) For example, to add a message as of April 12th and delete a 3-line message dated March 18th, you would enter:

```
.R TECO
*ER SYS:NOTICE.TXT$EW SYS:NOTICE.TXT$Y$$
```

Run TECO and specify that file NOTICE.TXT is to be updated. SYS: equals disk area [1,4].

OPERATOR TASKS

```
*IAPR 12 THIS IS AN EXAMPLE OF<CR>
A TWO LINE MESSAGE.<CR>
<CR>
$$
```

The command "I" places you in input mode and inserts everything following it up to, but not including, the first ESCape, into the message file. If "I" is the first command after entering TECO, the insertion is made at the beginning of the file. New messages are always placed at the beginning of the NOTICE.TXT file. ESCape prints as \$. The double ESCape ends the insertion command.

```
*SMAR 18$0L3K$$
```

This command string searches for the line containing the text "MAR 18", uniquely identifying the old message, and deletes that line plus the next two.

```
*EX$$
```

A new NOTICE.TXT is filed; the old version is automatically saved as NOTICE.BAK; and you exit TECO and return to monitor level.

Note that the above commands could have been strung together as follows:

```
.R TECO
*ER SYS:NOTICE.TXT$EW SYS:NOTICE.TXT$Y$$
*IAPR 12 THIS IS AN EXAMPLE OF<CR>
A TWO LINE MESSAGE.<CR>
<CR>
$$
*SMAR 18$0L3K$EX$$
.
```

5.1.2 SPACE with /FACT Switch

The SPACE program is run to provide disk usage information. A listing is produced showing each user's quotas and actual usage. Entries pertaining to disk usage are entered into the system FACT file for later use. SPACE is usually run as part of the daily Batch Job submitted to the batch input queue with a SUBMIT command, e.g., SUBMIT TASKS. To run SPACE manually, you must be logged in under [1,2] and type the following commands:

OPERATOR TASKS

<u>COMMAND</u>	<u>FUNCTION</u>
.R SPACE	Run the SPACE program.
*DSK:=/FACT	Create a disk file with a default filename of SPACE.LST and make appropriate entries in the system FACT file.
*^C	Return to monitor level.
.PRINT SPACE.LST	Queue SPACE.LST to be printed.
If the MONEY program is not to be run at this time, copy the FACT.SYS file to a back-up tape and save it for later processing.	
Use the /HELP switch in the SPACE program or type the monitor command HELP SPACE for more information on this program. Running SPACE with the BATCH switch is illustrated further on in this chapter.	
The MONEY procedure produces a listing showing individual user charges. The command sequence that follows uses DECTape as the back-up medium; if magnetic tape is to be the back-up medium, use the BACKUP program described in the Software Notebooks.	

<u>COMMAND</u>	<u>FUNCTION</u>
.AS DTAn	Assign DECTape unit n (n=0-7) and mount the accumulated FACT files on it.
.COPY FACT.SYS=DTAn:*. *	Create a single disk file, FACT.SYS, from the DECTape files.
NOTE	
If you are processing only the FACT.SYS file that you have just created with the SPACE program, you can omit the above two steps.	
.AS DSK SYS	Assign disk logical name of SYS to cause the MONEY program to look at your disk area for the FACT file.
.RUN MONEY[1,4]	Run the MONEY program on [1,4]. This will produce the MONEY listings on disk.
.DEA SYS	Remove logical name SYS from disk.
.PRINT MONEY.SUM,*.LPT	Queue the money listings to be printed.
.DEL FACT.SYS	Delete FACT.SYS from the disk.

NOTE

When MONEY is run through BATCON, the command AS DTAn is replaced by MOUNT DTA:FACT/VID:FACTFILE and any reference to DTAn: is replaced by FACT:.

OPERATOR TASKS

5.1.3 Update Account Files

CAUTION

No attempt should be made to update the accounting files unless the operator is completely familiar with the operation of the REACT program. The discussion below gives a few common examples using REACT. A complete discussion of the REACT program is contained in the RUNREA specification found in the Supplementary Operator's Procedures of the Software Notebooks. There are no safeguards built into the REACT program and a mistake could prove to be quite serious. For this reason all accounting file updates should be performed by qualified operators only.

Updates are made to the system files on disk and also to backup copies of these files on DECtape or magnetic tape. Following is a list of the current accounting files:

<u>FILE</u>	<u>FUNCTION</u>
ACCT.SYS	Contains the users' project programmer numbers and associated data necessary for users to log onto the system.
AUXACC.SYS	Contains the allowable disk quotas for each user.
QUOTA.SYS	Contains user disk quotas on other than the standard system file structures.
STRLST.SYS	Contains entries for all file structures showing the structure owner, configuration and structure type.

The functions performed by the operator are to insert new entries, delete existing entries, or change existing entries. Several examples are shown below.

Example 1: Inserting an entry in ACCT.SYS.

Add a new user [50,1234] to the ACCT.SYS file.

<u>STEP</u>	<u>FUNCTION</u>
.R REACT	Run the REACT program.
*R	Read ACCT.SYS from SYS into the REACT buffer.
*I 50,1234 NAME:JONES<CR>	Insert 50,1234 into the file. REACT will then prompt you for the user's

OPERATOR TASKS

<u>STEP</u>	<u>FUNCTION</u>
CODE:ABC\$	name. Enter the name followed by a <CR>. REACT then prompts you for the code (password). Enter the password followed by an ESCape. An ESCape character instructs REACT to assume the default conditions for all other parameters.
*W	Write the updated buffer back onto SYS as ACCT.SYS.

Example 2: Inserting an entry in AUXACC.SYS.

Add an entry for user [50,1234] to the AUXACC.SYS file.

<u>STEP</u>	<u>FUNCTION</u>
.R REACT	Run the REACT program.
*A	Read AUXACC.SYS from SYS into the REACT buffer.
*I 50,1234 NUM STRS:1<CR> STR NAME:DSKB<CR> RESRVD:0<CR> FCFS:1000<CR> QUOTA OUT:500<CR> STATUS BITS:0<CR>	Insert 50,1234 into the file. Then answer questions as to number of structures, disk quotas, and status bits. This information is contained on an access form which is prepared in the office.
*W	Write the updated buffer back onto SYS as AUXACC.SYS.

Example 3: Deleting entries.

Delete [2,167] from the ACCT.SYS file.

<u>STEP</u>	<u>FUNCTION</u>
.R REACT	Run the REACT program.
*R	Read ACCT.SYS into the REACT buffer.
*K 2,167	Kill (delete) the entry for 2,167.
*W	Write updated file back onto SYS.

Example 4: Changing entries.

Change the FCFS for DSKB to 1000 for user [2,167] in AUXACC.SYS.

OPERATOR TASKS

<u>STEP</u>	<u>FUNCTION</u>
.R REACT	Run the REACT program.
*A	Read AUXACC.SYS into the ACT buffer.
*C 2,167 CHANGE:FCF ARGS:DSKB:1000\$	Change 2,167. REACT will now prompt for information on what is to be changed and what the new value (ARG) is to be. Reply to the prompts.
*W	Write the updated file back onto SYS.

Example 5: Inserting an entry in QUOTA.SYS.

Add a quota entry for user 10,110 on disk DSKC.

<u>STEP</u>	<u>FUNCTION</u>
.R REACT	Run the REACT program.
*Q DSKC:[1,4]	Read QUOTA.SYS from DSKC.
*I 10,110	Insert user 10,110.
*RESRVD:0	Reply to the prompt with a reserved quota of 0 for the user on this file structure.
*FCFS:1000	Reply to the prompt with a logged-in quota of 1000 blocks.
*QUOTA OUT:500	Reply to the prompt with a logged-out quota of 500 blocks.

Example 6: Inserting a file structure entry in STRLST.SYS.

Add file structure DSKC for user 10,110.

<u>STEP</u>	<u>FUNCTION</u>
.R REACT	Run the REACT program.
*S	Read STRLST.SYS.
*I DSKC	Insert structure DSKC.
PROJ,PROG:10,110	Reply to a series of prompts with p, pn,
NAME:SMITH	User name,
NUM UNITS:2	Number of units in the file structure,
UNIT ID:1RP003	Unit identification,
KONT TYPE:DP	Controller type,
UNIT TYPE:2	Unit type

OPERATOR TASKS

<u>STEP</u>	<u>FUNCTION</u>
CLASS:0	Class of file structure.
UNIT ID:2RP004	REACT then repeats the last four prompts for as many units as specified in NUM UNITS:.
KONT TYPE:DP	
UNIT TYPE:3	
CLASS:0	
*W	Write the updated file back onto SYS.

The entire REACT dialogue is quite complex. You should refer to the RUNREA and REACT documents in the Software Notebooks and become acquainted with all the possible prompts and responses.

5.1.4 FILCOM Account Files

At specified times, run the file compare program (FILCOM) between disk [1,4] versions, and backup tape versions of the ACCT.SYS and AUXACC.SYS files. These files contain the entries necessary for a user to log onto the system as well as the user disk quota entries. This procedure notes any discrepancies in the two files and is used to prevent any user's tampering with the accounting files. For additional security, the REACT program used in the following command sequence can be renamed or removed from disk entirely and kept on tape. The FILCOM can be performed as part of a daily Batch job. The following procedure assumes that backup files are on DECTape; if they are on magnetic tape, replace DTA with MTA.

<u>COMMAND</u>	<u>FUNCTION</u>
.AS DTAX	Assign DECTape drive x where x=0-7. Mount the ACCT DECTape on DTAX.
.AS DSK LPT	Assign the disk a logical name of LPT so that output listings will go to the disk.
.R REACT	Run the REACT program.
*R	Read the system file ACCT.SYS from the disk.
*P	Purge any expired account numbers.
*L ACT.DSK LIST CODES?:Y	Create a disk file ACT.DSK containing all the entries plus passwords from the system file.
*R DTAX:	Read ACCT.SYS from the DECTape.
*P	Purge any expired account numbers.

OPERATOR TASKS

<u>COMMAND</u>	<u>FUNCTION</u>
*L ACT.DTA LIST-CODES?:Y	Create a disk file ACT.DTA containing all the entries plus passwords from the DECTape file.
*A	Read the system file AUXACC.SYS from the disk.
*L AUX.DSK	Create a disk file AUX.DSK containing all the entries from the system file.
*A DTax:	Read AUXACC.SYS from the DECTape.
*L AUX.DTA	Create a disk file AUX.DTA containing all the entries from the DECTape file.
*^C .R FILCOM	Return to the monitor and run the FILCOM program.
*ACCT=ACT.DSK,ACT.DTA	Create a disk file ACCT.SCM containing any differences in the ACCT.SYS files.
*AUX=AUX.DSK,AUX.DTA	Create a disk file AUX.SCM containing any differences in the AUXACC.SYS files.
*^C .PRINT ACCT.SCM,AUX.SCM	Return to the monitor and queue the .SCM files to be printed.
.DELETE *.DSK,*.DTA	Delete the files created above by the REACT program.

NOTE

When this job is run using BATCON, the first command AS DTax is replaced with MOUNT DTA:ACCT/WL and any reference to DTax: is replaced by ACCT:.

5.1.5 Detached Job Procedure

The purpose of this procedure is to rid the system of inactive jobs at specified times during the day. The action described here is for a first shift operator.

At 9 a.m., 12 noon, and 3 p.m., for example, you can run a SYSTAT on the CTY. The SYSTAT will show any jobs that are in a detached state. The 9 a.m. SYSTAT is filed. At 12 noon, a second SYSTAT is run and compared with the 9 a.m. SYSTAT. You then determine if there are any jobs which have been detached and inactive for the complete 3-hour period. This can be determined by comparing job number, project programmer number, program, state, and runtime. The 3 p.m. SYSTAT can likewise be compared to the 12 noon SYSTAT. You can then attach to the inactive jobs, from the CTY, and attempt to log off the job as follows:

OPERATOR TASKS

<u>STEP</u>	<u>FUNCTION</u>
.AT 5[4,111] From Job 2	Attach to job 5, user [4,111]. Monitor response.
.^C .SET SPOOL NONE	Return to monitor and remove spooling to eliminate any output.
.K/F	Log off job with the F option.

Make a note on the SYSTAT output concerning the action taken and file the SYSTAT in the logbook. If you are unable to kill the job, for example, if the user is over quota, you can take the following action:

<u>STEP</u>	<u>FUNCTION</u>
.^C .DET From job 5	Return to the monitor and detach the job. The second shift operator will take care of this job at 10 p.m.

Make a note on the SYSTAT to the effect that the job could not be logged off and file the SYSTAT in the logbook for later use.

The second shift operator can then complete the detached procedure initiated on the first shift. The operator can run a SYSTAT at 6 p.m. and at 10 p.m. These SYSTATs can be compared to the previous SYSTATs to determine if there are any inactive jobs. At 6 p.m. the operator can follow the same procedure as the first shift operator. He can attach to the job and attempt to log it off. At 10 p.m. the operator action becomes more involved. Up until 10 p.m., jobs which cannot be logged off are simply detached. At 10 p.m. it is the operator's responsibility to get rid of these jobs. By checking a list of first and second offenders, the operator can decide the appropriate action to take. If this is the first time a user is detached over quota, the following action might be taken:

<u>STEP</u>	<u>FUNCTION</u>
R REACT	Run the REACT program.
*A	Read AUXACC.SYS in the REACT buffer.
*T proj,prog	Type the disk quotas of the offending user.
*C proj,prog CHANGE:FCF ARGS:DSKB:new value CHANGE:OUT ARGS:DSKB:new value\$	Change the values so that the current logged-out quota becomes the new logged-in value and the current logged-in quota becomes the new logged-out value.
*W	Write the updated file back onto the system.

Once the quotas have been interchanged, the operator can log in a second job under the offender's project programmer number. The original detached job may now be killed and the operator can then kill the second job.

OPERATOR TASKS

If the user in question is a second offender, the operator can take the following action after attaching to the job.

<u>STEP</u>	<u>FUNCTION</u>
.^C K/W/B	Return to the monitor and log off the job with the /W/B option. This will cause the logout to automatically delete files to get the user under quota and log off the job.

It is very important that all terminal output concerning the detached job procedure be filed in the logbook for reference should any questions arise.

5.1.6 BACKUP (Daily)

At some prescribed time of the day, you should arrange to have the contents of the disks saved on magnetic tape using the BACKUP program. This will allow you to reconstruct a system, in the event of an unforeseen crash, and limit the loss of files to those entered or modified since the previous BACKUP procedure.

If saving all files every day seems too time consuming, daily BACKUPS can be run to save only those files that were created or modified since a specific time the day before. The entire system need be saved only once a week; the weekly procedure is explained later in this chapter.

The CTY log output of all BACKUP procedures should be filed in the System Logbook. When files are lost during a system crash, you can refer to the BACKUP logs to determine which magnetic tapes contain the files to be restored. This implies that you set up some system of externally labeling the magnetic tapes and that you enter corresponding tape identification on the CTY logs.

The procedures included below cover only a few of the features available in BACKUP. (For a complete description of all BACKUP commands and switches, refer to the BACKUP specification in the Software Notebooks.)

Example 1: Save system disk DSKB.

<u>STEP</u>	<u>FUNCTION</u>
.R BACKUP	Run the BACKUP program.
/TAPE MTA0	Use tape on MTA0.
/REWIND	Rewind to the beginning of tape.
/LIST DSK:	Produce directory listing on disk (default file name is BACKUP.LOG).
/SAVE DSKB:	Save entire contents of DSKB.
!1,2 DSKB	BACKUP types out each UFD as it begins to save files from that area.
1,3	
1,4	
.	
.	
.	

OPERATOR TASKS

If the tape becomes full before the SAVE is completed, BACKUP stops, types out the full file identification and block number of the current file being saved, rewinds and unloads the tape, and types the message:

\$BKPEOT REACHED EOT -- MOUNT NEW TAPE THEN TYPE "GO"

Remove the full tape, mount a new tape, and type GO. BACKUP continues with the SAVE. (A file may be split across tapes.)

Example 2: Save all files that have been created or modified since 6 p.m., yesterday.

<u>STEP</u>	<u>FUNCTION</u>
.R BACKUP	Run the BACKUP program.
/TAPE MTA0	Use tape on MTA0.
/REWIND	Rewind to the beginning of tape.
/SSNAME WED-PARTIAL	Give the SAVE file a name.
/MSINCE:YESTERDAY:18:00	Condition the save to all files created or modified since 6 p.m. yesterday.
/SAVE	SAVE the files.
!1,2 DSKB	BACKUP types out each UFD that contains a file being saved.
.	
.	
.	

5.1.7 SPACE with /BATCH Switch

The SPACE program with /BATCH switch is run each night immediately after completing the BACKUP save. Its function is to log in a batch job for all users who were on the system during the day and also for users who exceeded their logout quota by more than 100 blocks. A listing is produced showing each users quotas and actual usage. In the case of users who exceeded their logout quota, files are automatically deleted to allow the Batch job to be logged out under quota. SPACE is run while logged in under [1,2] as follows:

<u>STEP</u>	<u>FUNCTION</u>
.R SPACE	Run the SPACE program.
*DSK:=DSKB:/BATCH	Cause SPACE to look only at DSKB and produce SPACE.LST.
*^C	Return to the monitor and queue
.PRINT SPACE.LST	SPACE.LST to be printed.

SPACE.LOG files are also generated for individual users. They can be treated as normal line printer output and returned to the user.

Use the /HELP switch in the SPACE program or type the monitor command HELP SPACE for more information on this program. The /FACT switch is illustrated in a separate paragraph in this section.

OPERATOR TASKS

5.1.8 GRIPEs

During the normal course of timesharing, the user has the ability to run the GRIPE program. Using this program, he can enter suggestions, complaints, or other comments, which are then stored as disk files on area [3,3]. The function of this procedure is to combine all the gripe for the day and produce several listings, which are distributed to various people who will answer them. The procedure is handled completely by Batch. All you need do is mount the GRIPE tape after submitting the GRIPE.CTL file while logged in under [1,2] as follows:

.SUBMIT GRIPE.CTL

The following manual procedure assumes DEctape as a backup medium; for magnetic tape, use the BACKUP program to save and restore the GRIPE files.

<u>STEP</u>	<u>FUNCTION</u>
.DIR *.CMP[3,3]	Check area [3,3] for any CMP (GRIPE) files. If there are none, stop here.
.R PIP	Run the PIP program.
GRIPE=.CMP[3,3]	Create a disk file GRIPE, which contains all the CMP files from [3,3].
*^C .DEL *.CMP[3,3]	Return to the monitor and delete all the CMP files from area [3,3].
.AS DTAn	Assign DTAn (where n=0-7) and mount the previous GRIPE DEctape on it with write enabled.
.R PIP *G=DTAn:*.TXT,DSK:GRIPE	Run the PIP program. Create a disk file G, which consists of the combined GRIPEs from DTAn and DSK.
*DTAn:GRIPEs.TXT=G	Put the updated GRIPEs file back onto DTAn.
*^C .PRI GRIPE/COP:n/DIS:REN	Return to the monitor and queue the n copies of the GRIPE file to be distributed later.
.DELETE G	Delete disk file G.

In the above example, the listing contains the gripe for the current day. The tape contains gripe accumulated according to some schedule determined by your installation. The schedule can be weekly or monthly based upon the gripe frequency, or it can be based strictly on tape capacity.

5.1.9 Saving FACT Files

The system FACT files, containing all the data necessary to compute user charges, must be backed up each night to ensure against their loss. This backup procedure should be the last job done each night. All other system jobs must be logged off and only one [1,2] job left to run this procedure. The FACT files should be saved on both a

OPERATOR TASKS

DEctape (or magnetic tape) and a private disk pack. These files are used both daily and monthly to produce the MONEY listings discussed in Section 5.1.2. The following steps assume DEctape as a backup medium; for magnetic tape, use the BACKUP program.

<u>STEP</u>	<u>FUNCTION</u>
.REN *.*=SYS:FACT.*	Copy the FACT files from SYS to [1,2] and at the same time delete them from SYS[1,4].
.AS DTAn	Assign DTAn (n=0-7) and mount a scratch tape on it.
.R PIP *DTAn:DDMMM.FCT/Z=FACT.*	Run the PIP program to zero the tape and transfer the FACT files onto the DEctape as a single file DDMMM.FCT where DDMMM = the date. For example, APR 10th would be 10APR.FCT.
*^C .MOUNT xxx	Return to monitor level and mount the private pack xxx.
.R PIP *xxx:[p,pn]/X/B=DTAn: *.*	Run the PIP program to transfer the FACT file from the DEctape to the area [p,pn] on the private pack, xxx.

5.1.10 System Shutdown

The termination of timesharing is normally accomplished through the operator-privileged OPSER command :KSYS time, where "time" is the time of shutdown in hours and minutes (hh:mm), or the time from "now" that timesharing will stop (+hh:mm). :KSYS takes care of all system shutdown functions automatically.

<u>STEP</u>	<u>FUNCTION</u>
:KSYS +00:30	This OPSER command causes timesharing to stop in 30 minutes. Periodic warning messages will automatically be sent to users as described in Section 4.1.
You may now stop the system via the console front end and turn the system over to field service or whomever may have the system scheduled next.	
Enter the following:	
.CTRL/\	Control-backslash to get to the front-end PARSER.
PAR%SHUTDOWN	Stop the TOPS-10 monitor.

It is important that you now make a log entry to the effect that you have shut down the system and the reason for the shutdown.

OPERATOR TASKS

5.1.11 DSKLST

The DSKLST program compiles and prints an analysis of the contents and the current status of the system file structures. It provides the information with which the system administrator and you, the operator, can:

1. Analyze the efficiency of disk space usage.
2. Detect and diagnose problem areas on disk.
3. Determine when to refresh various disk areas.
4. Determine optimum cluster sizes for assigning and releasing file storage space.

You can run DSKLST, while logged in under [1,2], as follows:

<u>STEP</u>	<u>FUNCTION</u>
.AS DSK LPT	Assign the disk to receive the DSKLST output rather than the printer.
.R DSKLST FOR HELP TYPE /H *<CR>	Run the DSKLST program, which will prompt with an asterisk. The default response of <CR> will print all DSKLST output for all structures.
*~C .PRINT DSKLST.LST	Return to the monitor and queue the report to be printed.

If you include the above steps in a Batch control file such as DSKLST.CTL, you can submit the job to the Batch system with the command:

.SUB DSKLST.CTL

Additional information on DSKLST can be found in the DSKLST specification in the Software Notebooks.

5.1.12 DSKRAT

The DSKRAT program compiles and prints damage assessment information for level-D file structures. It also has the facility to free up lost disk space. You can run DSKRAT, while logged in under [1,2], as follows:

<u>STEP</u>	<u>FUNCTION</u>
.AS DSK LPT	Assign printer output to the disk.
.AS str STR	Assign the logical device name STR to the file structure "str" that you are analyzing.
.R DSKRAT	Run the DSKRAT program.
.PRINT RAT.LST	Queue the report to be printed.

OPERATOR TASKS

If more than one structure is to be analyzed, repeat the last three steps using different "str's".

To run this procedure under the Batch system, include the above command sequence in a file DSKRAT.CTL and submit it to Batch with:

```
.SUB DSKRAT.CTL
```

Additional information on DSKRAT can be found in the DSKRAT specification in the Software Notebooks.

5.2 OTHER SCHEDULED TASKS

The following operator tasks are usually scheduled at other than daily intervals.

5.2.1 Account Control

The following procedure is usually run weekly. It produces a listing showing all the changes that have been made to ACCT.SYS and AUXACC.SYS during the previous week. The account files, as checkpointed on tape the previous week, are matched against the same files currently on the system. Two reports, one for each account file, are prepared. Changes between the old and the current files are listed. At the end of each processing, the current account files are copied to tape to be used in the next processing.

The command sequence is usually placed in an ACCT.CTL file and run via the Batch system. Should the need arise, you can also run the procedure manually. The following sequence assumes DECTape as the backup medium; for magnetic tape, use the BACKUP program.

<u>STEP</u>	<u>FUNCTION</u>
.AS DTAn	Assign DTAn (n=0-7) and mount the checkpoint DECTape. Set the enable switch to WRITE ENABLE.
.AS DSK LPT	Assign logical name LPT to disk to produce listing files on the disk.
.R REACT	Run the REACT program.
*R	Read the current ACCT.SYS file into the REACT buffer. List it to disk as
*L ACCT.NOW	ACCT.NOW without listing the passwords.
LIST CODES?N<CR>	
*A	Read the current AUXACC.SYS file into
*L AUXACC.NOW	the REACT buffer. List it to disk as
	AUXACC.NOW.
*R DTAn:ACCT.CHK	Read the previous week's ACCT.CHK file
*L ACCT.OLD	from the DECTape and list it to disk as
LIST CODES?N<CR>	ACCT.OLD without listing the passwords.
*A DTAn:AUXACC.CHK	Read previous week's AUXACC.CHK file
*L AUXACC.OLD	from the DECTape and list it to disk as
	AUXACC.OLD.

OPERATOR TASKS

<u>STEP</u>	<u>FUNCTION</u>
*^C .R FILCOM	Return to the monitor and run the FILCOM program.
*ACCT=ACCT.OLD,.NOW	Create a disk file ACCT.SCM containing the differences in the two ACCT files.
*AUXACC=AUXACC.OLD,.NOW	Create a disk file AUXACC.SCM containing the differences in the two AUXACC files.
*^C .R PIP	Return to the monitor and run the PIP program.
*DTAn:ACCT.CHK=SYS:ACCT.SYS	Checkpoint the current ACCT.SYS file to DECTape as ACCT.CHK, replacing last week's copy.
*DTAn:AUXACC.CHK=SYS: AUXACC.SYS	Checkpoint the current AUXACC.SYS file to DECTape as AUXACC.CHK, replacing last week's copy.
*^C .PRI ACCT.SCM,AUXACC.SCM	Return to the monitor and queue the files containing the differences to be printed.
.DELETE *.NOW,*.OLD	Delete the temporary work files.

5.2.2 BACKUP (Weekly)

On a weekly basis, you should save the entire contents of the system disks on magnetic tapes. The procedure is similar to that described under BACKUP (Daily) in Section 5.1.6. Once the files are saved, however, the disk areas can also be refreshed or wiped clean and the saved files then reloaded onto disk. This is done, primarily, to recover unusable disk blocks.

After heavy use of the system, some disk blocks are marked unusable because of system failures and errors. The number of disk blocks marked this way is in reality greater than the actual number of unusable blocks due to the assigning of blocks in clusters. In addition, when the disk is nearly full, disk files become fragmented into non-contiguous parts. When this happens, more disk space is used to maintain the extra retrieval pointers. Also, more overhead time is used because the system has to locate more pieces of the file than when the file is in a contiguous area. Therefore, by saving, refreshing, and restoring the system files, the disk contains the same files as before, but it is better organized. Non-contiguous files are restored into contiguous areas.

You can also take this opportunity to delete files that have been inactive for a specified period of time; this will free up additional room on disk for active files.

The following procedure can be used as a basis for setting up a weekly BACKUP, tailored to your installation.

Stop timesharing operations.

OPERATOR TASKS

<u>STEP</u>	<u>FUNCTION</u>
:KSYS +0030	This OPSEK command will stop timesharing in 30 minutes. Users will be notified, periodically, as described in Section 4.1.

When timesharing has stopped, you can proceed with saving the system.

<u>STEP</u>	<u>FUNCTION</u>
.R BACKUP	Run the BACKUP program.
/TAPE MTA0	Use tape on MTA0.
/REWIND	Rewind tape to load point.
/LIST DSK:	Request a directory listing on disk.
/SAVE DSKB:	Save system disk.
!1,2 DSKB	BACKUP types out each UFD as it begins
1,3	to save files from that area.
.	
.	
.	

If the data overflows a reel of tape, you will receive the message:

\$BKPEOT REACHED EOT -- MOUNT NEW TAPE THEN TYPE "GO"

Replace the full tape with a scratch tape and type "GO". Repeat this procedure until you receive the message:

DONE The SAVE operation is complete.

The above procedure can be repeated for any other system disks to be SAVED.

The monitor can now be reloaded following the procedures in Chapter 3 with one important difference. When the system prompts you for the STARTUP OPTION, you should answer REFRESH. During the ensuing dialog, the system will prompt you as follows:

TYPE STR NAME TO BE REFRESHED (CR IF NONE, ALL IF ALL)

Type in a system structure. For example:

DSKB<CR>

The prompt will be repeated so that you can enter all the structures to be refreshed. A <CR> with no structure name will terminate that set of prompts. The system will then repeat the STARTUP OPTION prompt. If there are no other changes to be made to the file structure parameters, answer with QUICK or GO. You will then receive the message:

TO AUTOMATICALLY LOG-IN UNDER [1,2] TYPE "LOGIN"

The following procedure assumes BACKUP resides on DECTape; if it resides on magnetic tape, change the ASSIGN and RUN commands to conform.

OPERATOR TASKS

<u>STEP</u>	<u>FUNCTION</u>
.LOGIN	Automatically log in under [1,2].
.SET SCHED 1	Set a schedule bit pattern of 1 to prohibit users from logging in.
.AS DTAn	Assign DTAn (n=0-7) and mount a DECTape containing the BACKUP program.
.RUN DTAn:BACKUP	Run the BACKUP program from DECTape because the disk is empty at this point.
/TAPE MTA0	Specify the tape drive containing tape number 1 of the SAVE performed in the previous procedure.
/ASINCE:-30D	This entry is optional and allows you to restore only those files that have been recently accessed (in this case: within the past 30 days).
/RESTORE	Restore all files, except those specifically excluded by prior conditioning.
!1,2 DSKB 1,3 . . .	BACKUP types out each UFD as it begins to restore files into that area.
DONE	RESTORE is complete.
/UNLOAD	Rewind and unload the tape.
/^C	Return to the monitor.

If you should be restoring a multi-reel SAVE, the end of each full tape will result in a message:

\$BKPEOT REACHED EOT -- MOUNT NEW TAPE THEN TYPE "GO"

Mount the next sequentially numbered tape and type GO.

If you desire additional information to be printed at your terminal, for example, tape sequence numbers, SAVE set names, dates and times, include the following entry before the /RESTORE entry.

/LIST CTY:	Direct the printing of the BACKUP.LOG file to the OPR terminal.
------------	-----------------------------------------------------------------

When the RESTORE operation is complete, do the following:

<u>STEP</u>	<u>FUNCTION</u>
/^C	Return to the monitor.
.SET SCHED 0	Set the schedule bit pattern to 0 to allow users to log in.

OPERATOR TASKS

<u>STEP</u>	<u>FUNCTION</u>
.LOG 1,2	Log in again under [1,2] to ensure a correct search list.

You can now initiate your system jobs just as you would after a normal system load.

5.3 NON-SCHEDULED TASKS

The following tasks are performed on an on-call or as-required basis rather than on a formal schedule.

5.3.1 Formatting Disk Packs

Formatting and mapping disk packs is required prior to their first use. This procedure is also recommended if a disk pack develops search errors or other problems such as bad spots on the pack. Formatting writes (or rewrites) a header record at the beginning of each sector so that the hardware can position itself when reading or writing on the disk. Mapping is the process of writing and reading each disk block, or sector, a number of times to search for bad blocks. A log of any bad blocks is kept and is output in two duplicate BAT blocks.

Because the formatting process destroys all data on the disk pack, you should first run the BACKUP program and save any important files. (Refer to Section 5.2.2 for information on saving and restoring files.)

5.3.1.1 Formatting RP02s and RP03s - RP02s and RP03s can all be formatted in a timesharing environment using the DCRPE program. The requirements for running DCRPE are:

1. You must be logged in under [1,2], preferably on a hard copy terminal.
2. The disk pack must be mounted on a private drive; the pack must NOT have been ASSIGNED or MOUNTed.
3. The "write header" switch inside the appropriate RP10 disk controller must be set to enable write headers (LK OUT position).

Proceed as follows:

<u>STEP</u>	<u>FUNCTION</u>
.R DCRPE[6,6]	Run DCRPE, which is usually stored in the hardware and diagnostic area [6,6]. DCRPE will then enter into a dialog with you.
TYPE THE PHYSICAL NAME OF THE DISK PACK DRIVE:	Reply with the physical designation of the drive; for example, DPA0 or DPB2.

OPERATOR TASKS

<u>STEP</u>	<u>FUNCTION</u>
DO YOU WISH TO FORMAT THE PACK? Y or N <CR>-	Y will continue with the formatting; N will skip the formatting and go to the mapping questions that follow the NOTE below.
NAME THE DEVICE FOR ERROR REPORTING:	Enter the logical or physical name of the device for error reports, e.g., TTY. A carriage return defaults to disk.
***ARE YOU SURE YOU WANT TO PROCEED? ALL INFORMATION ON DEVICE dev WILL BE LOST BEYOND THIS POINT. Y or N <CR>-	This is your last chance to exit before losing data or disk. Type Y if all necessary data has been saved.
FORMATTING IN PROGRESS hh:mm:ss	DCRPE notifies you that it has started to format the disk and gives you a timestamp.
HEADERS HAVE BEEN WRITTEN- VERIFICATION NOW IN PROGRESS	DCRPE has completed writing the headers and is now reading them to verify their validity.
FORMATTING COMPLETE- RECOMMEND MAPPING THE PACK TOTAL ERRORS:n TOTAL FORMATTING TIME= hh:mm:ss	DCRPE has completed the verification; you are notified of the number of errors (n) and of the time it took for the formatting process.
Y or N <CR>-	DO YOU WISH TO MAP THE PACK? Y will continue with the mapping; N will exit to the monitor.
At this time you can switch the "write headers" switch on the disk controller back to the WH position.	

NOTE

If you are formatting a new pack, one that has not been used since it came from the factory, it will not have the valid HOM and BAT blocks required by the mapping portion of DCRPE. Use the TWICE program with the LONG option to structure the pack, giving it a scratch name, and initializing both the HOM and BAT blocks. The detailed specifications for running the TWICE program can be found in the Software Notebooks. After this is done, reload the DCRPE program. Answer N to the question DO YOU WISH TO FORMAT THE PACK? You will then be prompted as follows:

OPERATOR TASKS

<u>STEP</u>	<u>FUNCTION</u>
USE OLD BAT BLOCK AS BASE? Y or N <CR>-	You must reply Y.
DO YOU WISH TO RETAIN THE HOME BLOCKS IF YES, THE PACK WILL HAVE TO BE REFRESHED IF NO, THE HOME BLOCK PARAMETERS WILL HAVE TO BE REDEFINED IN ONCE Y or N <CR>-	Again, answer Y.
NAME THE DEVICE FOR CRASH RECOVERY DATA:	The usual reply is DSK.
NAME THE DEVICE FOR ERROR REPORTING:	This question is asked only if you skipped formatting. Answer with TTY.
***ARE YOU SURE YOU WANT TO PROCEED? ALL INFORMATION ON DEVICE dev WILL BE LOST BEYOND THIS POINT Y or N <CR>-	Type Y if all necessary data files have been saved.
MAPPING IN PROGRESS hh:mm:ss	DCRPE notifies you that it is mapping and gives you a timestamp.
END OF PASS 1; ELAPSED TIME =hh:mm:ss n ERRORS THIS PASS CONTINUE? A or Y or N <CR>-	DCRPE has read all the blocks once in the time shown and has found n errors. Type A to continue reading all the blocks a total of ten times without stopping. Type Y to read the blocks one more time and ask the question again. Type N to end mapping at this pass.
TOTAL MAPPING TIME FOR dev=hh:mm:ss HISTORY OF BAD BLOCKS: START OF BAD BLOCK: bbbbbb=ccc,hh,ss SIZE OF BAD REGION IN BLOCKS=n . . . p=BAD BLOCKS TOTAL q=BAD REGIONS TOTAL	After the final pass, DCRPE prints out the total mapping time and the location, size, and number of any bad regions on the disk. Each bad area is located by block number bbbbbbb, as well as cylinder (ccc), head (hh) and sector (ss). The size of each bad region is given by n. After all bad areas have been logged, the total number of bad regions (q) and bad blocks (p) are printed. This same information is also stored in the two duplicate BAT blocks. The program then exits.
EXIT	
Now use the TWICE program with the REFRESH option to re-initialize the two duplicate HOM blocks and clear the Storage Allocation Table (SAT). This procedure frees up any lost blocks and eliminates fragmentation of files since the subsequent BACKUP operation, RESTORE, will reload the files systematically.	

OPERATOR TASKS

RP02 and RP03 disks can also be formatted in exec mode using the DDRPC program. This program gives you the additional capability of formatting single cylinders of a disk pack and single surfaces of a cylinder.

The requirements for running DDRPC are:

1. The "write header" switch inside the appropriate RP10 disk controller must be set to enable write headers (LK OUT position).
2. Drives that are not to be formatted must be placed off line with the STOP switch.

Load and start the DDRPC program; it will initialize and print a description of the disk system.

PDP-10 RP10 Disk Pack Reliability Test (DDRPC) Version 000,002

Memory Map =
From To SIZE/K
000000 077777 32K

POWER FREQUENCY = 60HZ

Write Enable Entire Disk Y or N <CR> Y
DF10 in KA Addressing Mode
Number of Drives= 4
DPA0 = RP03, STATUS-ON LINE
DPA1 = RP02, STATUS-OFF LINE
DPA2 = RP03, STATUS-OFF LINE
DPA3 = RP02, STATUS-OFF LINE
EDDT

You are now ready to format an entire disk or selected portions of it.

Example #1 - Formatting an entire disk pack

EX FORMAT\$X Operator wants to use the formatting capability of the DDRPC diagnostic program. (Note that \$ indicates pressing the ESCape or ALTmode key.)

WANT TO FORMAT THE DISK? Operator types Y.
Y OR N <CR> - Y

WHAT DRIVE?-<CR>0 Format drive 0.

WANT TO FORMAT ENTIRE DISK? Y OR N <CR>-Y The complete disk pack on drive 0 is to be formatted.

VERIFY ERROR, DISK ADDR, DPA=0 CYL=013 SURF=02 SECT=03
VERIFY ERROR, DISK ADDR, DPA=0 CYL=056 SURF=01 SECT=01
VERIFY ERROR, DISK ADDR, DPA=0 CYL=103 SURF=04 SECT=03
VERIFY ERROR, DISK ADDR, DPA=0 CYL=277 SURF=06 SECT=00

EXIT TEST-FORMAT RUN TIME 0:15:20

The error messages above indicate that the disk failed to format the indicated disk addresses successfully. The disk may be completely formatted again or the individual disk addresses (surfaces) may be reformatted as indicated in the example below.

OPERATOR TASKS

Example #2 - Formatting a portion of the disk pack

ACTION	EXPLANATION
EX FORMAT\$X	Format a portion of a disk pack. (Note that \$ indicates pressing the ESC or ALT key.)
WANT TO FORMAT THE DISK? Y OR N <CR>-Y	Operator types Y.
WHAT DRIVE?-<CR>0	Operator specifies drive 0.
FORMAT ENTIRE DISK? Y OR N <CR>-N	Operator wants to format only a portion of the disk.
CYL(8)=100	Cylinder 100 (octal).
SURF(8)=0	All sectors of surface 0.
CYL(8)=102	Cylinder 102.
SURF(8)=A	All surfaces.
CYL(8)=B	B is an illegal character.
?HUH?	
CYL(8)=217	Cylinder 217.
SURF(8)=40	40 is an illegal surface address.
?HUH?	
CYL(8)=217	Cylinder 217 again.
SURF(8)=4	Surface 4.
CYL(8)=E	Either E or X at this point causes program to ask if there is another drive to be formatted (or tested).
TEST WHAT DRIVE?-<CR>E	The formatting for the above cylinders and surfaces will be done, error messages may appear as in the above example, and EXIT TEST will be output.

At this time you can switch the "write headers" switch on the disk controller back to the WH position.

5.3.1.2 Formatting RP04s and RP06s - RP04s and RP06s can be formatted and mapped in either a timesharing environment or stand-alone using the DDRPI program.

DDRPI (Timesharing)

Running DDRPI during timesharing requires the following:

1. You must be logged in under [1,2].
2. Packs to be formatted must not be in the system structure list.

OPERATOR TASKS

3. The following files must be in your area:

DIAMON.SAV
DDRPI.SAV
KLDDT.SAV
SUBRTN.SAV

Proceed as follows:

.ASSIGN DSK LPT	Direct printer output file to disk.
.RUN DIAMON	Load and start the diagnostic monitor.
	DIAMON will then request a command.

DIAMON CMD-	Enter "DDRPI" to start the disk formatting dialog.
-------------	-------------------------------------------------------

The DDRPI dialog for formatting RP04 and RP06 type disk drives is rather lengthy. Print out the DDRPI.HLP file and use it as reference; the dialog questions and their suggested responses are fully detailed.

CHAPTER 6

SYSTEM RECOVERY PROCEDURES

Because of the complexity of the system, there will be times when it will cease to function correctly. Such a condition, commonly referred to as a "crash," may be the result of software problems, hardware problems, or environmental problems. Sometimes, the system will automatically reload itself, record the error conditions, and restart processing. At other times, it becomes your responsibility to recognize the problem and take corrective steps to get the system back up as quickly as possible.

It is very important that you keep an accurate record of any system problem that may arise. In the event of a software crash, a Software Error Report (SER) form must be filled out, in addition to a concise and descriptive entry in the logbook. For hardware problems, a hardware error form should be filled out along with the log entry.

6.1 CENTRAL PROCESSOR RECOVERY

Recovery procedures for the central processor will be discussed under three categories of system failures.

1. Failures due to software errors
2. Failures due to malfunctions in the hardware
3. Failures due to a hostile environment.

The auto-reload feature of the DECsystem-10 will automatically reload the system after a system failure (crash) if the system has been up for at least five minutes. The usual procedure, if possible, is to record the system as it appeared at the time of failure and then reload the previously loaded monitor. System-to-operator dialog provides you with the opportunity to load a different monitor or to change system parameters via the start-up options of the ONCE procedure. (See the ONCE specification in the Software Notebooks.)

For installations where there may be no operator coverage at times, there is a 60-second timeout after the first restart question. If no answer has been received, the system assumes (and records on the CTY) the default answers to the ONCE procedure.

The crash file (a copy of the system at the time it failed) is always written to the same area; if you wish to save this information, it must be copied to another area before the system crashes again.

If the system crashes twice within five minutes, it will not be automatically reloaded.

SYSTEM RECOVERY PROCEDURES

6.1.1 Software Errors

Software errors fall into three general categories:

1. Errors detected by the TOPS-10 monitor, resulting in STOPCD messages
2. Errors detected by the RSX-20F front-end monitor
3. Hung conditions where the TOPS-10 scheduler cannot run any job other than the null job.

6.1.1.1 STOPCD (Monitor Error Stop) - When the monitor detects an internal error, it executes a STOPCD macro, which produces a message at the CTY of the form:

```
?CPUn MONITOR ERROR.  STOPCODE NAME xxx
```

where n is the number of the CPU (0 or 1) that executed the STOPCD and xxx is a mnemonic identifying the error. A complete list of STOPCD error messages and descriptions is part of the STOPCD specification in the Software Notebooks. The list also includes the name of the monitor module containing the STOPCD and the type or severity classification.

There are four types of STOPCDs; in increasing order of severity they are DEBUG, JOB, STOP, and HALT. All but the last initiate automatic recovery procedures.

1. **DEBUG** - If no priority interrupt is in progress, the condition is not immediately harmful to any job or to the system. The monitor transmits the following message to the CTY:

```
?CPUn MONITOR ERROR.  STOPCODE NAME xxx  
CONTINUING SYSTEM
```

and continues. No immediate operator action is required.

If a priority interrupt is in progress, a DEBUG STOPCD acts and counts as a JOB STOPCD.

2. **JOB** - If no priority interrupt is in progress, the condition jeopardizes the integrity of the current job. The monitor transmits the following message to the CTY:

```
?CPUn MONITOR ERROR.  STOPCODE NAME xxx  
?ABORTING JOB
```

and another message to the user's terminal:

```
?MONITOR ERROR;  UUO AT xxx
```

where xxx is one of the following:

```
USER LOC n  
EXEC LOC n; EXEC CALLED FROM EXEC LOC m  
EXEC LOC n; EXEC CALLED FROM USER LOC m
```

The monitor then aborts the current job and continues. No immediate operator action is required.

SYSTEM RECOVERY PROCEDURES

If a priority interrupt is in progress, a JOB STOPCD acts like a STOP STOPCD.

3. STOP - The condition jeopardizes the integrity of the system as a whole. The monitor transmits the following message to the CTY:

```
?CPUn MONITOR ERROR.  STOPCODE NAME xxx
RELOAD MONITOR
```

All jobs are aborted and the system initiates an automatic reload of the monitor.

4. HALT - The condition is so serious that the entire system is affected and the monitor cannot be reloaded automatically. The system halts, requiring that you manually perform the CRASH procedures as described in Section 6.1.1.4.

Note, however, that you will be required to perform the entire CRASH procedure very infrequently, only when the system cannot recover on its own. (Refer to the CRASH specification in the Software Notebooks for a detailed explanation of the procedures.)

In addition, if the monitor can determine the information, the following messages are also printed on the CTY:

```
JOB n ON TTYxxx RUNNING xxxxxx
UUO IS xxxxxxxxxxxx AT USER PC xxxxxx
FILE dev:file.ext[p,pn]
```

They will help you determine the user causing the problem.

6.1.1.2 RSX-20F Detected Errors - Nearly all the error conditions that are not covered by the STOPCDs in Section 6.1.1.1 are detected by RSX-20F, the console front-end monitor. Some of the following are caused by hardware errors, some by software errors, and others by environmental conditions.

Hardware Errors:

CLOCK ERROR STOP	A serious hardware failure has occurred in the KL10 CPU.
E-BUS PARITY ERROR	An E-Bus parity error was detected on the KL10 CPU.
DEPOSIT EXAMINE FAILURE	The front-end software was unsuccessful in an attempt to examine or deposit into KL10 memory.

In each of the above error conditions, the front end reloads KL10 microcode, reconfigures the cache and main memory, and restarts the KL10 at octal location 407. The code at 407 causes TOPS-10 to dump and reload itself. If the error conditions persist, call Field Service.

Software Errors:

TEN HALTED	The KL10 has executed a HALT instruction due to some other error condition.
------------	-----------------------------------------------------------------------------

SYSTEM RECOVERY PROCEDURES

KEEP ALIVE CEASED

The "keep alive" dialog between the RSX-20F and TOPS-10 monitors senses that the KL10 is looping and not functioning correctly.

For each of the above conditions, a KL10 core dump is taken and the TOPS-10 monitor is reloaded. If the problem persists, call Field Service.

Other Conditions:

TEN REQUESTED RE-BOOT

For some reason, the KL10 monitor wants to be reloaded. RSX-20F calls in the KLINIT routine and takes the BOOT reply to the first question.

POWER FAIL RESTART

RSX-20F has restarted the system when power was restored after a power failure. The microcode is reloaded, cache and memory are reconfigured, and the system is restarted at octal location 70.

The usual sequence of events that takes place when RSX-20F senses an error in the KL10 is to call the KLERR routine to analyze the error and print key system information. The KLINIT routine is then called to reload or restart the system.

When KLINIT attempts to reload after a hardware or software error, it passes control to the crash restart routine at location 407 of the KL10. This routine brings in the bootstrap program BOOTS via the KL10 file system and executes it with the following default command string:

```
/D<CR>SYSTEM.EXE[1,4]<CR>
```

However, if the TOPS-10 monitor had been originally loaded by an explicit command to the bootstrap prompt, this same command will be substituted for SYSTEM.EXE[1,4]. For example, if you had answered the bootstrap prompt with DSKN:RK270A[1,2] instead of merely typing a carriage return, the auto-reload procedure will use the command string:

```
/D<CR>DSKN:RK270A.EXE[1,2]
```

If the 407 start fails, enter KLINIT to read the bootstrap program via the PDP-11 file system.

The following is an example of CTY output for one of the above hardware errors.

DEPOSIT EXAMINE FAILURE

KLERR -- VERSION V001B RUNNING

KLERR -- KL NOT IN HALT LOOP

KLERR -- KL ERROR OTHER THAN CLOCK ERROR STOP

KLERR -- KL VMA: 000000 133051 PC: 000000 133051

KLERR -- PI STATE: ON , PI ON: 175 , PI HLD: 000 , PI GEN: 000

KLERR -- EXIT FROM KLERR

KLI -- VERSION V003D RUNNING

KLI -- MICROCODE VERSION 131 LOADED

LOGICAL MEMORY CONFIGURATION:

CONTROLLER

ADDRESS	SIZE	RQ0	RQ1	RQ2	RQ3	CONTYPE	INT
00000000	128K	04	FOR	ALL		DMA20	4

SYSTEM RECOVERY PROCEDURES

KLI -- KL RESTARTED

BOOTS

DSKN:/D

DSKN:RK270A

6.1.1.3 HUNG Condition - A hung condition exists when the scheduler cannot run any job other than the null job. When it appears to you that the system is not doing any useful work, use the following procedure to examine the program counter:

<u>STEP</u>	<u>FUNCTION</u>
^\ PAR>REPEAT 10;EXAMINE PC	Control backslash gets you into the front-end command language (PARSER). These PARSER commands will cause the contents of the program counter to be printed 10 times. Successive values of 1 usually indicate a hung condition. Other values could indicate a "loop" condition. (See KEEP ALIVE CEASED error message above.)

You can attempt to perform a crash restart by typing:

PAR>ABORT	Halt the KL processor.
PAR%JUMP 407	Take a core dump and reload the monitor using BOOTS from the KL10 file system.

If "BOOTS" is not output, you can try to boot the system via the front end as follows:

<u>STEP</u>	<u>FUNCTION</u>
^\ PAR>SET CONSOLE PROG PAR>RUN KLINIT	Access the PARSER if not already prompted with PAR>. Set console to programmer mode. Load and start the KLINIT dialog.
KLI--ENTER DIALOG [NO, YES, EXIT, BOOT]? KLI>BOOT	Answering "BOOT" to the first question will load the bootstrap program and start it.
KLI--BOOTSTRAP LOADED AND STARTED /D	This will cause a core dump to be written out to disk.
<CR> filename	A carriage return will cause the default monitor from SYSTEM.EXE[1,4] to be loaded. A filename will load a monitor from the specified file.

You may now want to set the console back to operator mode.

SYSTEM RECOVERY PROCEDURES

6.1.1.4 Crash Restart Procedure - Whenever a core dump is taken via the /D switch to BOOTS, the dump data is stored in the system file SYS:CRASH.EXE. If another crash follows, the second dump data overlays the first. Therefore, you should follow a procedure, such as the one shown below, to preserve and augment crash data for the use of the system programmer when debugging the problem. The following is performed on the CTY while logged in under [1,2].

STEP

FUNCTION

.COPY XPN:SERnnn.EXE<055>=SYS:CRASH.EXE

This will create a copy of the crash data just dumped and store it in the crash library, usually area [10,1]. nnn is the number of the SER (Software Error Report) form.

.ASSIGN DSK LPT

Assign the logical name LPT to the DSK.

.SYS XL SYS:CRASH

This causes the SYSTAT program to look at the CRASH.EXE file on SYS: and produce a SYSTAT.TXT file on [1,2].

.DEASSIGN LPT

Deassign LPT from DSK.

.PRINT SYSTAT.TXT

Print the SYSTAT output information.

If you know the value in the PC at the time of the crash, you can use a monitor-specific FILDDT to convert it into the alphanumeric symbolic value. To create a monitor-specific FILDDT, use the following procedure:

.R FILDDT

Run FILDDT

FILE:SYSTEM.EXE/S

Reply to the prompt with the specific monitor to be examined. (/S loads the symbols from the file specified.)

FILE:~C

Exit to the monitor.

.SAVE FDSYS

Save the monitor-specific FILDDT as a file named FDSYS.

Whenever you want to convert a PC value into its symbolic counterpart, use the following procedure:

.R FDSYS

Run FDSYS and reply to the FILE: prompt with a <CR> to indicate that you want to look at the running monitor.

FILE:<CR>

nnnnnnxxxxxx

Type in the octal PC value nnnnnn followed by an underscore (back arrow) and the program will respond with a tab and the symbolic value xxxxxx.

SYSTEM RECOVERY PROCEDURES

6.1.2 Hardware Errors

The most common hardware crash is the result of a memory parity error. This will cause the system to stop. The procedure below should be used to bring the system back up. In other cases of hardware problems, you may have to call Field Service for assistance or merely reload the system.

6.1.2.1 Parity Error - There are two distinct situations that arise as the result of a parity error. If the parity error occurs in the monitor, the system will halt. If the parity error occurs in user core, the system may remain running. In either case, a message will be printed on your operator terminal. The steps to follow are explained below.

Case 1 - Parity Error in the Monitor

When a parity error occurs in the monitor, the CTY will sound five bells, print out ?EXEC PAR ERROR HALT or ?MEM PAR HALT, and then print out one of the following error messages:

?0MEM PAR ERR ON CPU_n FOR JOB IN [prog name]

?1MEM PAR ERR AT yyyyyy ON CPU_n FOR JOB IN [prog name]

?1MEM PAR ERR AT yyyyyy ON CHANNEL 0

?nMEM PAR ERRS FROM yyyyyy TO zzzzzz ON CPU_n FOR JOB IN
[prog name]

followed by the question:

CONTINUE SYSTEM? (TYPE Y OR <CR>)

You can attempt to continue by typing in a Y; however, this is not recommended since the monitor may have been seriously damaged.

If you enter <CR>, the system will load the monitor bootstrap and you will be prompted with "BOOTS."

Type /D to get a crash dump.

Determine in which memory bank the parity error occurred from the location typed out on the CTY and from the memory selection and interleaving switches. (See Section 6.1.2.3.)

Deselect the memory which had the error and reconfigure the remaining memories. (The deselection and reconfiguration procedure is explained below in Section 6.1.2.2.)

Reload the monitor.

Fill out a Memory Parity Error (PAR) form and make an entry in the logbook.

Case 2 - Parity Error in User Core

When a parity error occurs in user core, five bells are sounded and one of the following messages is printed on the CTY:

SYSTEM RECOVERY PROCEDURES

?0MEM PAR ERR ON CPUn FOR JOB IN [prog name]

?1MEM PAR ERR AT yyyyyy ON CPUn FOR JOB IN [prog name]

?1MEM PAR ERR AT yyyyyy ON CHANNEL 0

?nMEM PAR ERRS FROM yyyyyy TO zzzzzz ON CPUn FOR JOB IN
[prog name]

At this point, the system is still running and you should proceed as follows:

1. If the program name that typed out was one of the operator jobs, such as BATCON, OPSER, LPTSPL, PTPSPL, etc., restart that particular job.
2. Fill out a PAR form and make an entry in the logbook.
3. Determine the memory bank in which the parity error occurred. Use the location typed out on the CTY and the setting of the memory selection and interleaving switches. (See Section 6.1.2.3.)
4. If this is the second parity error in the same memory within a 4-hour period, use SEND ALL to inform the users that the system will be reloaded in 15 minutes. If not, let the system continue running.
5. At the specified time, stop the system, deselect the memory, reconfigure the remaining memories (See Section 6.1.2.2), and reload the system.

NOTE

Under certain conditions, portions of memory can be dynamically deselected via the SET MEMORY OFF-LINE command. These conditions are specified in the section describing the command in Appendix A.

6.1.2.2 Altering Your Memory Configuration - There may be times when an error message will indicate that a portion of a memory module or a whole module is unusable due to parity or other problems. Under certain conditions you can merely set a portion of memory off-line and let the system continue processing. If this isn't possible, you will have to stop the system, reconfigure memory without the affected module, and reload the system.

Setting Memory Off-line

The operator-privileged command:

SET MEMORY OFF-LINE FROM n TO m

described in detail in Appendix A, can be used providing that all three of the following conditions are met:

SYSTEM RECOVERY PROCEDURES

1. The memory to be set off-line does not contain any portion of the resident monitor.
2. After the memory is off-line, there will still be a reasonable amount of memory left for user programs. "Reasonable amount" depends on an installation's usage of the system. Consult the system administrator on this.
3. The memory to be set off-line is not interleaved. (If interleaved memory is set off-line, memory that is good and usable will also be placed off-line.)

When the SET MEMORY OFF-LINE command is successful, the system continues. Memory that was set off-line is not referenced.

If the above conditions cannot be met, proceed to reconfiguring memories.

Reconfiguring Memories

ACTION	EXPLANATION
If the bad memory contains any part of the resident monitor, skip the next two steps.	Memory occupied by the resident monitor cannot be set off-line.
Set bad memory off-line. However, if no job can run without that memory, do not try to make the command succeed; skip the next step.	You are going to attempt an orderly shutdown in preparation for reconfiguring memory modules.
Stop timesharing.	See Section 4.1.2.1 for the operator-privileged OPSER command KSYS to stop timesharing at a specified time. Allow enough time for those jobs that can run to complete their processing.
Type ^\ on the CTY.	Invoke the RSX-20F command language PARSER.
PAR>SET CONSOLE MAINTENANCE PAR>RESET	The first command changes the console mode to allow you to issue the second command which in turn causes a master reset of the KL processor.
Reconfigure memories.	Deselect the bad memory, check the lower bound addresses (MADR switches) to get contiguous memory, and correctly set the interleave switches. (See Section 6.1.2.3.)
Reload the monitor	Refer to Section 3.3.1.

6.1.2.3 Memory Selection and Interleaving - Memories are selected or sequentially assigned octal numbers beginning with 0. For example, five memories would be numbered 0,1,2,3 and 4. It is important to note that they must be numbered or selected sequentially.

SYSTEM RECOVERY PROCEDURES

This numbering of memories effectively indicates the beginning memory address or lower bound address for a particular memory. In the example above, if each memory had a capacity of 32K, memory 0 would start with address 0, memory 1 with address 100000 (a 1 in bit 20), memory 4 with address 400000 (a 1 in bit 18).

Memories are normally interleaved because interleaving makes the system run faster.

In memories other than MG10s and MH10s this means that they work in pairs or fours such that consecutive memory addresses alternate or rotate between memories. In the example above, 2-way interleaving would be between units 0 and 1, and between 2 and 3, with memory number 4 operating alone. In the case of 4-way interleaving with nine memories, for example, the sets 0-3 and 4-7 would each be interleaved, with 10 (octal) operating alone. Only memories of the same size and with consecutive addresses starting with an even numbered memory can be interleaved. For example, 2-way interleaving can exist between memories 0 and 1 but not between 1 and 2; 4-way interleaving can exist between 0,1,2, and 3 but not between 1,2,3, and 4.

Interleaving in MG10s and MH10s is somewhat different. For MG10s 2-way interleaving is available within a single cabinet and 4-way interleaving between two cabinets. However, the units or cabinets must still contain equal amounts of storage. For MH10s, either 2-way or 4-way interleaving is available within a single cabinet.

The panel containing the switches that control the numbering, selection, and interleaving of memories is located inside the front door of the memory.

MF10 Memory

In an MF10 there are four groups of identical switches, one group for each port. Each group has a set of seven switches that determine the octal number of the memory to be selected. There is also one switch in each group that controls the interleaving and one switch in each group that selects or deselects the memory. To deselect an entire memory, the operator should put all four select switches into the DES position. The address switch settings of all selected groups must be identical in any one memory.

If an MF10 memory is being used with a version of TOPS-10 earlier than 5.06, and you deselect a bank of memory other than the highest numbered one, you must renumber the memories to eliminate the hole. If after deselecting a memory, in the case of a 2-way interleave, an even number of memories remain, they will all be interleaved. If there is an odd number of memories, all but the last, or high order, memory will be interleaved. The high order memory should be switched to the NORM (no interleaving) position.

Example:

The system configuration consists of 12 (octal) memories numbered 0-11. Memory number 5 is deselected. The operator would now reselect memory number 11 to become number 5. Memory number 10 (octal) must now be switched to the NORM position because it is the ninth memory (odd).

The following table indicates the switch settings for numbering MF10-As (32K each) up to 256K.

SYSTEM RECOVERY PROCEDURES

Table 6-1
MF10-A (32K) Memory Address Switch Settings

Memory Address	MF10 No.	MADR switches							
		14	15	16	17	18	19	20	
000000 - 077777	0	0	0	0	0	0	0	0	
100000 - 177777	1	0	0	0	0	0	0	1	
200000 - 277777	2	0	0	0	0	0	1	0	
300000 - 377777	3	0	0	0	0	0	1	1	
400000 - 477777	4	0	0	0	0	1	0	0	
500000 - 577777	5	0	0	0	0	1	0	1	
600000 - 677777	6	0	0	0	0	1	1	0	
700000 - 777777	7	0	0	0	0	1	1	1	

The following table indicates the switch settings for numbering MF10-Gs (64K each) up to 512K.

Table 6-2
MF10-G (64K) Memory Address Switch Settings

Memory Address	MF10 No.	MADR switches							
		14	15	16	17	IGN	18	19	
0000000 - 0177777	0	0	0	0	0	IGN	0	0	
0200000 - 0377777	1	0	0	0	0	IGN	0	1	
0400000 - 0577777	2	0	0	0	0	IGN	1	0	
0600000 - 0777777	3	0	0	0	0	IGN	1	1	
1000000 - 1177777	4	0	0	0	1	IGN	0	0	
1200000 - 1377777	5	0	0	0	1	IGN	0	1	
1400000 - 1577777	6	0	0	0	1	IGN	1	0	
1600000 - 1777777	7	0	0	0	1	IGN	1	1	

The following table indicates the switch settings on an MF10 for noninterleaved and interleaved operation.

Table 6-3
MF10 Interleave Switch Settings

Mode	MADR switches	
	34	35
Normal	NORM	NORM
2-way	NORM	INTL
4-way	INTL	INTL

SYSTEM RECOVERY PROCEDURES

MG10 and MH10 Memories

In an MG10 or MH10, there is only one group of seven switches used to number the memory, but there are eight switches to determine port selection, because there are eight ports per memory. Each port switch can be set for either the KI/KL or KA processor, or the port can be deselected.

There are also switches that enable the deselection of specific banks of memory. The initial bank switch settings for various memory sizes are given in the table below.

Table 6-4
MG10 and MH10 Memory Bank Selection

Memory Bank Switches ON LINE M0 M1 M2 M3	MG10 Memory Size	MH10 Memory Size
1 0 0 0	32K	64K
1 1 0 0	64K	128K
1 1 1 1	128K	256K
0=OFF LINE 1=ON LINE (deselected) (selected)		

If a hardware problem arises, any bank can be deselected by setting a bank switch OFF LINE. Deselected banks are logically placed into the highest core in the cabinet, while other previously higher portions are pushed down, so that no hole appears in the address base within a cabinet. Also, interleaving switches may have to be reset. For example, a 128K cabinet with 32K deselected cannot be 4-way interleaved with another 128K cabinet.

The following table indicates the memory address switch settings for numbering MG10 and MH10 memories of various sizes. Note that the lower bound address or beginning address of an MG10 memory can only be 0 or a multiple of 32K and that of an MH10 memory can only be 0 or a multiple of 64K.

Table 6-5
MG10-A (32K) Memory Address Switch Settings

Memory Address	Memory No.	MADR Switches						
		14	15	16	17	18	19	20
000000 - 077777	0	0	0	0	0	0	0	0
100000 - 277777	1	0	0	0	0	0	0	1
200000 - 277777	2	0	0	0	0	0	1	0
300000 - 377777	3	0	0	0	0	0	1	1
400000 - 477777	4	0	0	0	0	1	0	0
500000 - 577777	5	0	0	0	0	1	0	1
600000 - 677777	6	0	0	0	0	1	1	0
700000 - 777777	7	0	0	0	0	1	1	1

SYSTEM RECOVERY PROCEDURES

Table 6-5 (Cont.)
MG10-G or MH10-G (64K) Memory Address Switch Settings

Memory Address	Memory No.	MADR Switches						
		14	15	16	17	18	19	20
0000000 - 0177777	0	0	0	0	0	0	0	0
0200000 - 0377777	1	0	0	0	0	0	1	0
0400000 - 0577777	2	0	0	0	0	1	0	0
0600000 - 0777777	3	0	0	0	0	1	1	0
1000000 - 1177777	4	0	0	0	1	0	0	0
1200000 - 1377777	5	0	0	0	1	0	1	0
1400000 - 1577777	6	0	0	0	1	1	0	0
1600000 - 1777777	7	0	0	0	1	1	1	0

MG10-H or MH10-H (128K) Memory Address Switch Settings

Memory Address	Memory No.	MADR Switches						
		14	15	16	17	18	19	20
0000000 - 0377777	0	0	0	0	0	0	0	0
0400000 - 0777777	1	0	0	0	0	1	0	0
1000000 - 1377777	2	0	0	0	1	0	0	0
1400000 - 1777777	3	0	0	0	1	1	0	0
2000000 - 2377777	4	0	0	1	0	0	0	0
2400000 - 2777777	5	0	0	1	0	1	0	0
3000000 - 3377777	6	0	0	1	1	0	0	0
3400000 - 3777777	7	0	0	1	1	1	0	0

MH10-L (256K) Memory Address Switch Settings

Memory Address	Memory No.	MADR Switches						
		14	15	16	17	18	19	20
0000000 - 0777777	0	0	0	0	0	0	0	0
1000000 - 1777777	1	0	0	0	1	0	0	0
2000000 - 2777777	2	0	0	1	0	0	0	0
3000000 - 3777777	3	0	0	1	1	0	0	0
4000000 - 4777777	4	0	1	0	0	0	0	0
5000000 - 5777777	5	0	1	0	1	0	0	0
6000000 - 6777777	6	0	1	1	0	0	0	0
7000000 - 7777777	7	0	1	1	1	0	0	0

Table 6-6 indicates the switch settings on an MG10 or MH10 for interleaved and noninterleaved operation.

Table 6-6
MG10 and MH10 Interleave Switch Settings

Mode	MADR Switches	
	34	35
Normal	NORM	NORM
2-way	NORM	INTL
4-way	INTL	INTL

SYSTEM RECOVERY PROCEDURES

Note that any MG10 or MH10 can be 2-way interleaved, any MH10 can be 4-way interleaved, and any two consecutively numbered MG10's of equal size (word storage) can be 4-way interleaved, with one provision. To 4-way interleave two 96K MG10's, the MADR switch settings of the first 96K MG10 must indicate a lower bound address of 0 or a multiple of 256K.

It is imperative that the operator become proficient in this procedure of numbering, selecting, and interleaving memories. Needless system downtime may result from operator ineptness in this task.

6.1.3 Environmental Errors

The causes of environmental crashes are varied. They include power fluctuations, power outages, temperature extremes, water leaks, accidental spills, mechanical shocks, etc. Some of these are detectable before they cause the system to crash. For example, a failure of the air conditioning system would be noticed by the rise in temperature. Others, like power failures, will occur without warning.

Whenever possible, you should perform an orderly shutdown if any of these conditions are imminent. When the situation is rectified, you can power up and reload the system.

Power failures can be classified into two categories: short fluctuations and prolonged outages. These are discussed separately below. In either situation, a power failure detection circuit senses the impending conditions and initiates a power failure interrupt. This interrupt triggers the operation of a program that saves the PC, flags, mode information, and fast memory registers; it then halts the processor. This allows the system to be restarted in a minimum amount of time. Temperature sensors, strategically placed within the equipment, detect high temperature conditions and cause power shutdown. This, also, initiates the power failure interrupt sequence.

6.1.3.1 Power Fluctuation - Power fluctuations are variations in line voltage that occur over a short space of time. Although the power is not lost completely, the system will probably go through an automatic restart procedure. If the restart is unsuccessful, do the following:

1. Check the disk drives and visually ensure that the heads have retracted. If they have not, notify Field Service and do not attempt to load the system.
2. Check all other equipment to make sure that there is power to the unit. If not, attempt to power up the unit. If this fails notify Field Service.
3. If necessary, reload the monitor. (Refer to Section 3.3.1). If the monitor won't load, check the various controllers; clear them if necessary (see the System Reference Manual in the Software Notebooks), and try to reload the monitor again. If this still fails, notify your supervisor.

6.1.3.2 Power Outage - A power outage means that the power drops out completely. You must now take precautionary steps before the power

SYSTEM RECOVERY PROCEDURES

comes back on. The sudden surge of power may prove harmful to certain pieces of equipment.

Turn the START/STOP switch on all disk pack drives to the STOP position and visually ensure that the heads have retracted completely. If they have not, notify Field Service and do not attempt to power up the drives.

POWERING THE SYSTEM DOWN AND UP

Most of the system hardware can be powered down or up by pressing the POWER switch on the console. However, all pieces of hardware to be controlled in such manner must have their LOCAL/REMOTE switch in the REMOTE position during operation.

6.2 FRONT-END PROCESSOR RECOVERY

The DECsystem-10 front-end processors fall into two categories according to usage. The console/diagnostic front-end takes care of the system operator interface and system diagnostics. The communication front-ends take care of local and remote stations and terminals.

Front-end error recovery is based not only on usage but also on the type of interface to the KL processor. The console/diagnostic and DN87S communications front-ends use the integrated data interface (DTE20) to access the memory and I/O buses. The DC75NP, DN85, and DN87 communications front ends use the DL10 communications channel to connect to the memory and I/O buses.

Front-end error recovery will be discussed under the following headings:

1. Console/Diagnostic Front End
2. DTE20-Interfaced Communications Front Ends
3. DL10-Interfaced Communications Front Ends.

6.2.1 Console/Diagnostic Front End

Whenever the KL processor senses that the console front end is not running ("keep alive" count not being incremented), it checks to see if DTELDR is running and if the automatic reloading of front ends (/AUTO) is in effect. /AUTO is a switch function of DTELDR, the program that dumps and loads front ends connected to the KL processor via a DTE20 interface.

If DTELDR is running and /AUTO is in effect, the KL processor electronically activates the appropriate front-end loader program. Also, unless you include the /NODUMP switch in the DTELDR command, the front-end memory is dumped to a file DSK:DTED00.BIN (.B00, .B01, etc. for successive dumps). The front-end bootstrap is then loaded and in turn, loads the RSX-20F monitor. The KL processor then ensures that the front end is running by checking the "keep alive" count.

To enable the automatic reloading of the console front end as well as any communications front ends, add the following job to your OPR.ATO file:

SYSTEM RECOVERY PROCEDURES

```
:SLOG                ;log in a subjob
:DEF DTE=            ;give it the name DTE
DTE-R DTELDR         ;load the DTELDR program
DTE-/AUTO            ;start it with the auto-reload option
```

If the auto-reload feature is not enabled, or if the DTE subjob has stopped running, you can run DTELDR from any terminal logged in under [1,2] and bring up the console front end as follows:

```
.R DTELDR            ;load DTELDR
*/RELOAD:0           ;reload the console front end (always on DTE
0)
```

A complete description of the DTELDR program switches will be typed at your console if you answer the DTELDR prompt with:

```
*/HELP
```

6.2.2 DTE20-Interfaced Communications Front Ends

Whenever the KL processor senses that a communications front end on a DTE interface is not running, it checks to see if automatic reloading of front ends (/AUTO) is in effect. /AUTO is a switch function of DTELDR, the program that dumps and loads DTE interfaced front ends.

If /AUTO is in effect, the KL processor automatically reloads the front end from a default load file. Also, unless you include the /NODUMP switch in the DTELDR command, the front-end memory is dumped to a default dump file.

Default load files are named DSK:DTEL0n.BIN where n is the DTE number 1, 2, or 3.

Default dump files are named DSK:DTED0n.BIN where n is again the DTE number 1, 2, or 3. Successive dumps of the same front-end are given unique file extensions of .B00, .B01, .B02, etc.

Enabling the /AUTO function for communications front ends is identical to that for the console front end. The OPR.ATO subjob:

```
:SLOG
:DEF DTE=
DTE-R DTELDR
DTE-/AUTO
```

will automatically reload any front end that crashes.

If the auto-load feature is not in effect, you can run DTELDR from any terminal logged in under [1,2] and have a choice of restart procedures.

SYSTEM RECOVERY PROCEDURES

```
.R DTELDR          ;run DTELDR

*/RELOAD:2         ;reload the front end on DTE 2 with the
                   ;program in file DSK:DTEL02:BIN (default
                   ;name)
```

or:

```
*DTA3:COMSYS.BIN/RELOAD:1/NODUMP
                   ;reload the front end on DTE 1 with the file
                   ;COMSYS.BIN on DECTape drive 3. Do not take
                   ;a front-end memory dump.
```

A complete description of the DTELDR program switches will be typed at your console if you answer the DTELDR prompt with:

```
*/HELP
```

6.2.3 DL10-interfaced Communications Front Ends

Whenever a DL10-interfaced front end such as a DC75NP, DN85, or DN87 crashes, an error message such as:

```
%%PDP-11 NUMBER n IS NOT RUNNING
```

or:

```
%%PDP-11 NUMBER n IS HALTED AT pc
```

will appear on your console. n is the port number: 0-3 for the first DL10 and 4-7 for the second, if any. pc is the address of the halt in the PDP-11.

The program to dump and load a DL10 is BOOT11 and your system programmer will usually have made this available on the SYS: area. The following command sequence can be used to dump and reload DL10-interfaced front ends.

```
.R BOOT11          ;load in BOOT11

FILE:/DUMP/PORTNO:p ;dump entire memory of the PDP-11 on port p.

FILE:DN87.BIN/CLEAR/START/PORTNO:p
                   ;clear memory of PDP-11 on port p and load
                   ;and start the program from the file
                   ;DSK:DN87.BIN[1,2].
```

BOOT11 requires that a port number be explicitly entered if there is more than one port in the system.

The default file descriptor for a load operation is:

```
DSK:PDPXIp.BIN[self]
```

and for a dump operation:

```
DSK:PDPXIp.LST[self]
```

where p is the port number.

For additional information on BOOT11 switches refer to the BOOT11 specification in the Software Notebooks and the SYS:BOOT11.HLP file.

BOOT11 reports its progress as follows:

SYSTEM RECOVERY PROCEDURES

"PDP-11 DUMPING	PDP-11 memory is being dumped
"PDP-11 DUMPED	into a DECsystem-10 file.
"CLEARING PDP-11	PDP-11 memory is being zeroed.
"PDP-11 LOADING	PDP-11 is being loaded with formatted binary input file data.
"PDP-11 LOADED	Loading is complete; PDP-11 is
"PDP-11 STARTED	now executing in its own memory.

6.3 DX10 DATA CHANNEL RECOVERY

The DX10 data channel provides direct access to the DECsystem-10 memory and is an integrated part of the TU70 series magnetic tape systems. The DX10 contains a PDP-8 and you must load or reload this processor whenever you first power up the channel or after field service maintenance has been performed. The KL processor will usually reload the DX10 automatically after a crash.

If problems arise and the PDP-8 should halt, you will receive a message:

```
%TAPE CONTROLLER MTx IS OFF-LINE
```

where x specifies the controller attached to the data channel, e.g., MTA, MTB.

First, you should take a dump to record the error conditions. Use the BOOTDX program as follows:

```
.R BOOTDX          ;load BOOTDX
FILE:/D            ;request a dump
```

If a file descriptor is omitted as in the above example, the dump of PDP-8 memory will be placed into a default file, DXMPA.LSD, on your disk area. For additional information on the BOOTDX program, refer to the files BOOTDX.RNO and BOOTDX.HLP.

During the dump operation you will receive the following messages on the CTY:

```
"DX10 dumping      ;PDP-8 memory is being dumped
"DX-10 dumped      ;dumping has completed
```

You can now reload the DX10 data channel using BOOTDX as follows:

```
.R BOOTDX          ;if not already loaded
FILE:SYS:/C/S      ;load the default file DXMPA.BIN from the
                   ;system disk after setting all of PDP-8
                   ;memory to HALT instructions. Then start at
                   ;octal 200.
```

During the clearing and loading, you will receive the following messages at the TTY:

```
"Clearing DX-10    ;PDP-8 memory is being set to all HALTs.
"DX-10 loading     ;PDP-8 is being loaded.
"DX-10 loaded      ;PDP-8 has completed loading.
"DX-10 started     ;PDP-8 is now executing in its own memory.
```

CHAPTER 7

DEVICE OPERATION

It is important that you know how to operate the peripheral devices on your system and how to keep these devices in their best possible condition.

Because most equipment operates more efficiently if it is kept clean, a cleaning procedure is included with each device section. This chapter discusses the operating and cleaning procedures for those devices most commonly found on DECsystem-10s.

Here are some general rules for cleaning:

The computer room should be kept as neat and clean as possible. You should have the following cleaning supplies available.

1. 91% isopropyl alcohol
2. Lint-free wipers
3. Spray cleaner
4. Vacuum cleaner that can blow air as well as intake air and has rubber or plastic attachments
5. Cotton-tipped applicators

You should clean the exterior of all equipment weekly. Vacuum all outside surfaces including cabinet tops. Use spray cleaner on all vertically exposed surfaces except around switches.

In general, do not clean the interior of any equipment unless so directed in the following sections. Your field service representative does that job. Specific instructions for cleaning line printers and magnetic tape drives are given in the sections covering those devices.

Always be careful not to bump or change the position of any switches; this could cause a crash. Likewise, when you are cleaning the exterior of any disk drives, be careful not to jar the equipment; this could cause a serious hardware head crash.

Lastly, if you ever have any problem or doubts concerning the operation of or a cleaning procedure for a device, consult your field service representative.

7.1 LINE PRINTER

Some of the procedures below refer to an LP07 line printer (Figure 7-1) and some refer to an LP10 line printer (Figure 7-2). Be sure to follow the procedures for the appropriate printer.

DEVICE OPERATION

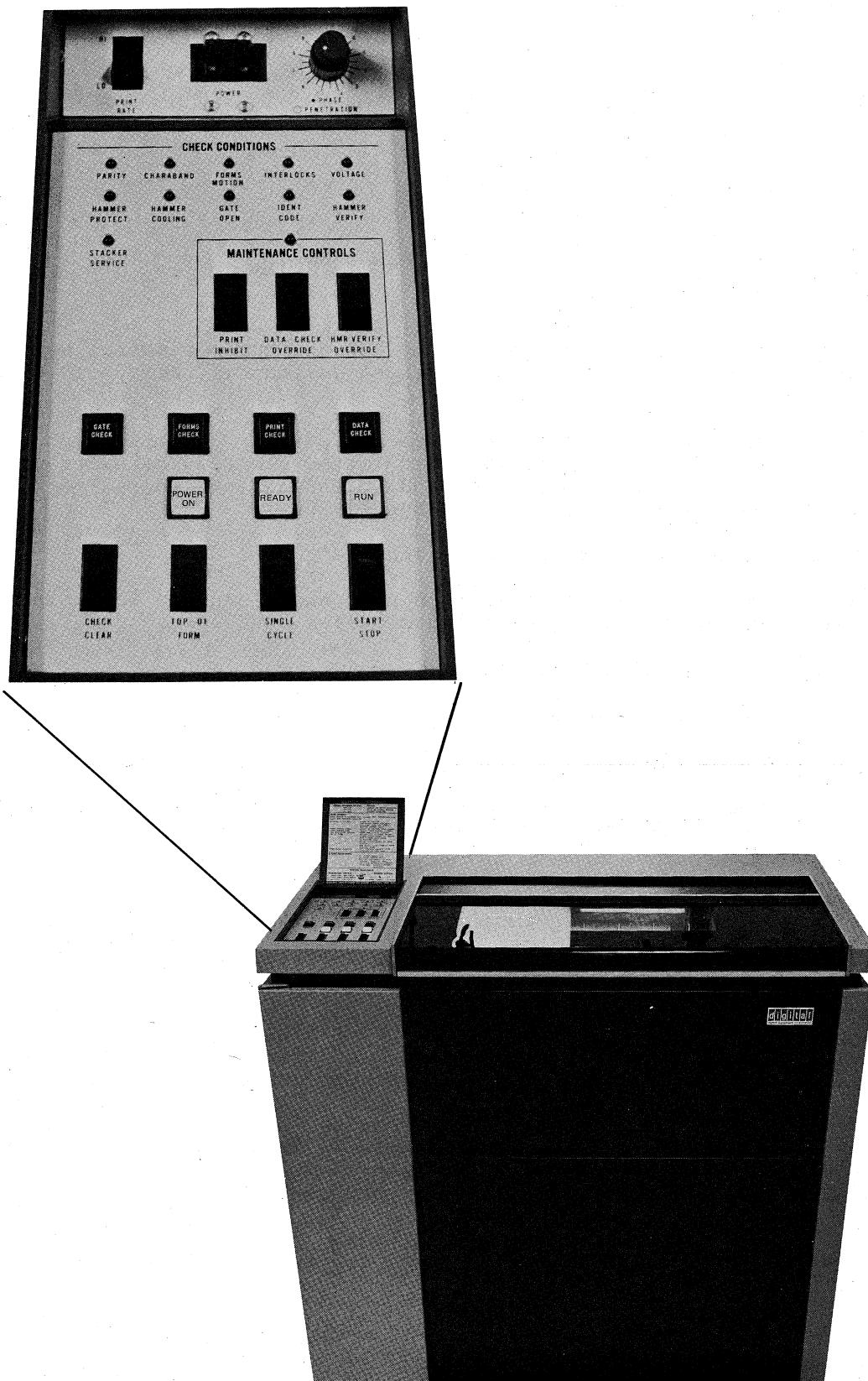


Figure 7-1 LP07 Line Printer

DEVICE OPERATION

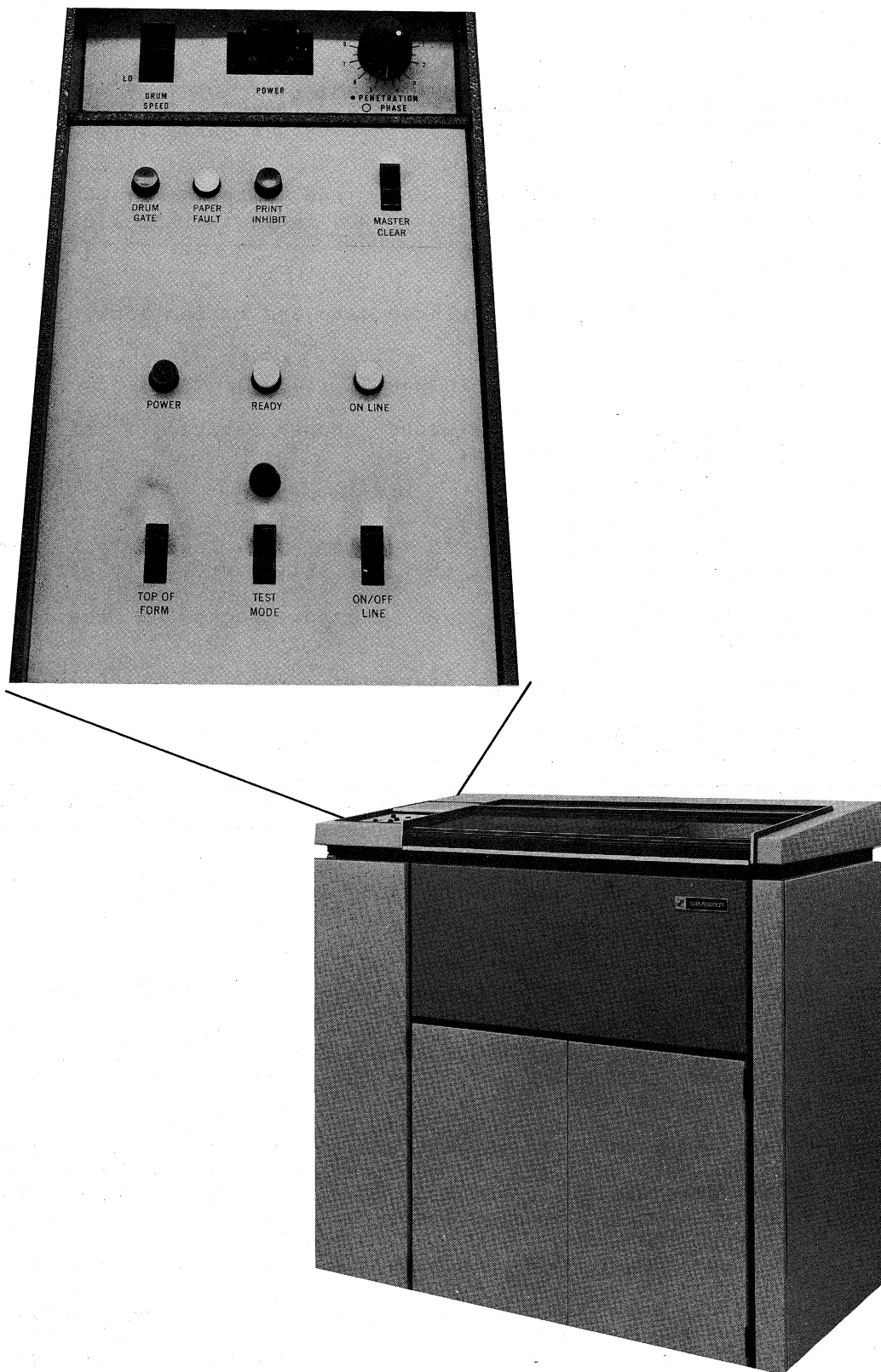


Figure 7-2 LP10 Line Printer

DEVICE OPERATION

7.1.1 Changing Forms

LP07 Procedure

The following procedure describes forms installation, alignment, and adjustment on an LP07 line printer. (Refer to Figures 7-1, 7-3, and 7-4.)

1. If the printer is running, press the START/STOP switch to place the printer in STOP mode (RUN indicator not lit).
2. Lift the printer window.
3. Unlatch and fully open the band gate and ribbon mask.

WARNING

Wait until the Charaband stops rotating before proceeding to the next step.

4. If necessary, reposition the upper and lower left forms tractors. Make sure that they are still vertically aligned.
5. Unlock the upper and lower tractor locks on the right tractors.
6. Open all tractor pressure plates.
7. Insert the left edge of the form in the upper left tractor and close the pressure plate.
8. Slide the upper right tractor to a position beneath the right hand form feed holes and close the pressure plate. Lock the upper right tractor.
9. Loosen the paper tension control lock and set the paper tension control to its extreme counterclockwise position (least tension).
10. Insert form in the lower left tractor and close the pressure plate.
11. Slide the lower right tractor to a position beneath the right hand form feed holes and close the pressure plate. Lock the lower right tractor.
12. Adjust the paper tension control clockwise to achieve proper vertical form tension. Proper tension occurs when there is a slight deformation of the top edge of the form feed holes. Tighten the paper tension lock.
13. Press the TOP OF FORM switch on the operator control panel.
14. Rotate the paper feed clutch clockwise to the ADJ position and use the paper advance adjustment control to position the form at the first line of print. Use the hammer faces as a print-line guide. Return the clutch to the RUN position.

You can use the vertical paper position control to obtain a fine adjustment of vertical position. This is often necessary when using pre-printed forms.

DEVICE OPERATION

15. Close and latch the ribbon mask.
16. Rotate the horizontal paper position control to align the form horizontally; use the horizontal forms alignment scale on the ribbon mask assembly for reference.
17. Set the thickness control to match the form being loaded.
18. Close the band gate and printer window.
19. Press the TOP OF FORM switch several times to ensure that the form is feeding correctly through the tractors.
20. Press the START/STOP switch to place the printer in RUN mode (RUN indicator lit).

Replenishing Forms

If you are merely replenishing the form supply and not changing to a different form, proceed as follows:

1. Lift the printer window.
2. Unlatch and fully open the band gate and ribbon mask.

WARNING

Wait until the Charaband stops rotating before proceeding to the next step.

3. Open all tractor pressure plates.
4. Insert the new form, overlapping the old one by at least one page and aligning the two at the page perforations. Close all tractor pressure plates.
5. Close and latch the ribbon mask.
6. Close the band gate and printer window.
7. Press the START/STOP switch to place the printer in RUN mode (RUN indicator lit).

DEVICE OPERATION

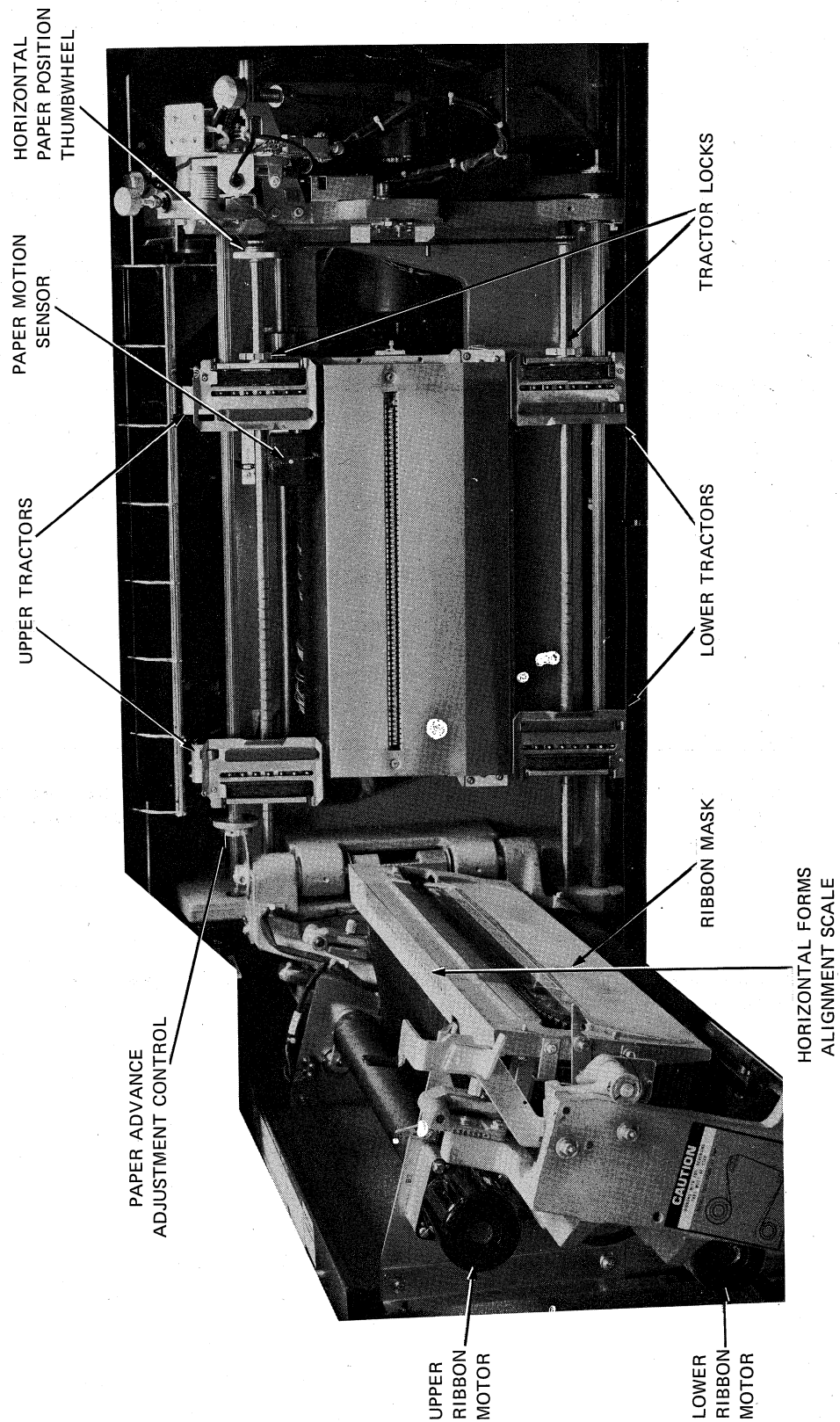


Figure 7-3 LP07 Paper Installation

DEVICE OPERATION

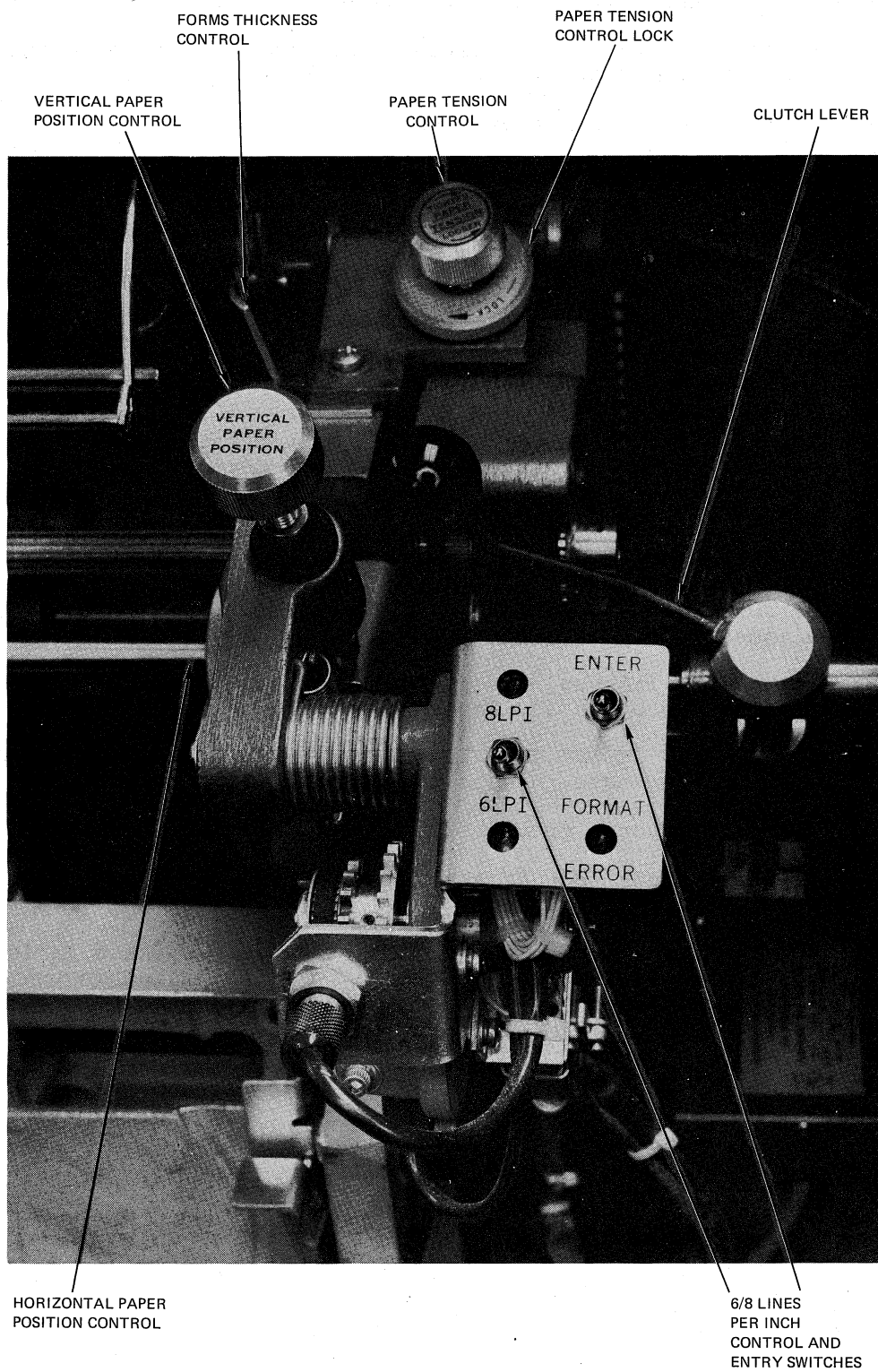


Figure 7-4 LP07 Forms Alignment

DEVICE OPERATION

LP10 Procedure

The following procedure describes forms installation, alignment, and adjustment on an LP10 line printer. (Refer to Figures 7-2 and 7-5.)

1. Press the ON/OFF LINE switch to set the printer off line (ON LINE indicator not lit).
2. Lift the printer window.
3. Using the drum gate latch, unlatch and fully open the drum gate.

WARNING

Wait for the character drum to stop rotating before you proceed.

4. Be sure that the proper vertical format tape has been mounted on the vertical format unit (refer to Section 7.1.2).
5. Press the TOP-OFF-FORM switch.
6. Set the copies control lever to match the form being used.
7. Be sure the upper and lower left tractors are vertically aligned.
8. Rotate the fine adjustment thumbwheels completely clockwise on both the upper and lower right tractors.
9. Unlock the upper and lower tractor locks on the right tractors and move them to the extreme right.
10. Open all tractor pressure plates.
11. Place the left edge of the form in the upper left tractor and close the pressure plate.
12. Position the upper right tractor to allow precise alignment of the tractor feed pins to the form-feed holes in the form. Lock the upper right tractor and close the tractor pressure plate over the form.

VERTICAL PAPER POSITIONER

PAPER TENSION CONTROL KNOB

PAPER TENSION LOCKING KNOB

COPIES CONTROL LEVER

TRACTOR LOCK

FINE ADJUSTMENT THUMBWHEEL

TRACTOR

VERTICAL FORMAT TAPE

HORIZONTAL PAPER POSITION THUMBWHEEL

TRACTOR LOCK

FINE ADJUSTMENT THUMBWHEEL

TRACTOR

FINE ADJUSTMENT THUMBWHEEL

PAPER DRIVE ADJUSTMENT THUMBWHEEL

TRACTOR

7-9

DEVICE OPERATION

13. Loosen the paper tension locking knob and turn the paper tension control knob completely counterclockwise.
14. Place the form in the lower left tractor and close the pressure plate.
15. Position the lower right tractor to allow precise alignment of the tractor feed pins to the form-feed holes in the form. Lock the lower right tractor, and close the tractor pressure plate.
16. Turn the paper tension control knob clockwise until the proper paper tension exists; this occurs when a slight deformation of the top edge of the form-feed holes in the form is noticed. Tighten the paper tension locking knob.
17. Rotate the fine adjustment thumbwheel on both the upper and lower right tractors until proper horizontal paper tension exists. The tension is correct when there is a slight deformation of the right edge of the form-feed holes in the form.
18. Mount the forms alignment scale (located in the paper storage area, right side) across the hammer bank area using the dowel pins located on special castings at both sides of the hammer bank.
19. Using the horizontal paper position thumbwheel, move the form horizontally until the first print column on the form corresponds to the first column on the forms alignment scale.
20. Move the paper feed clutch lever (located to the right of the vertical format unit) to the ADJUST position. Using the paper drive adjustment thumbwheel, adjust the form vertically until the first print line of the form is aligned with the forms alignment scale print line.
21. Remove the forms alignment scale, and replace it in the scabbard (paper storage area, right side).
22. Move the paper feed clutch lever to the RUN position.
23. Close and latch the drum gate.
24. Close the printer window.
25. Check the paper tension by pressing the TOP-OF-FORM switch several times and ensuring that the paper does not pull loose from the tractor feed pins.
26. Press the ON/OFF LINE switch to set the printer on-line (ON LINE indicator lit).

Replenishing Forms

If you are merely replenishing forms and not changing to a different form, proceed as follows:

1. Lift the printer window.

DEVICE OPERATION

2. Using the drum gate latch, unlatch and fully open the drum gate.

WARNING

Wait for the character drum to stop rotating before you proceed.

3. Open all tractor pressure plates.
4. Insert the new form, overlapping the old one by at least one page and aligning the two at the page perforations. Close all tractor pressure plates.
5. Close and latch the drum gate.
6. Close the printer window.
7. Press the ON/OFF LINE switch to set the printer on-line (ON LINE indicator lit).

7.1.2 Controlling the Vertical Format Unit

LP07 Procedure

The LP07 line printer has a direct access vertical format unit (DAVFU) that is the software counterpart of a carriage control tape. The direct access vertical format unit is loaded with a file containing an 8-channel by 143-line bit matrix. The ON bits in this matrix represent the punched holes in the corresponding paper carriage control tape.

The LP07 printer also provides for the software control of the print density (6 or 8 lines per inch) as well as for operator control via a switch.

Information on the generating, storing, and loading of DAVFU matrices is contained in the files: MAKVFU.DOC and MAKVFU.HLP.

To generate a default matrix that corresponds to channels 1-8 of the standard line printer carriage tape for the LP10, proceed as follows:

.R MAKVFU	Load the MAKVFU program.
*OUTPUT	If this is the only command, the default matrix is generated and is stored as DSK:NORMAL.VFU on your disk area.
*EXIT	Return to the monitor.
.	

To generate a custom matrix, load MAKVFU and give the appropriate MAKVFU commands as described below.

DEVICE OPERATION

<u>Command</u>	<u>Function</u>
CHANNEL n ALL	Set all lines in logical page for channel n. (n = 1,2,...8)
CHANNEL n BOTTOM	Set last line in logical page for channel n. (n = 1,2,...8)
CHANNEL n EVERY m	Set every m-th line in physical page for channel n. (n = 1,2,...8) (m = 1,2,...length)
CHANNEL n LINES m1 m2 ...	Set lines m1, m2, etc. in physical page for channel n. (n = 1,2,...8) (m1 m2... = 1,2,...length)
CHANNEL n TOP	Set line 1 in physical page for channel n. (n = 1,2,...8)
EXIT	Exit from MAKVFU.
FORMS	Clear all forms break channels.
FORMS n	Define forms break for channel n.
HELP	Display this file on the terminal.
INFORMATION	Display a brief description of the VFU file, including length, page size, forms break, and LPI setting.
INFORMATION n	Display a description of the lines defined for channel n.
LENGTH x	Define the physical page size to be x. (x = 1 - 143). Length must be greater than or equal to page size.
LPI y	Define the print density (lines per inch). (y = 0, 6, or 8 where 0 means manual switch setting.)
OUTPUT dev:file.ext[p,pn]	Generate a binary VFU file with the specified name. If the argument is omitted, the default is DSK:NORMAL.VFU on user area.
PAGE z	Define the logical page size to be z. (z = 1 - 143.) Page size must be less than or equal to length.

The automatic loading of an appropriate DAVFU file is discussed under LPTSPL in Section 4.5.1.

LP10 Procedure

To make a standard carriage control tape for vertical format control of 11 inch paper with six lines of printing per inch, do the following:

1. Obtain a manual- or machine-operated punch made to punch 12 channel carriage control tapes.

DEVICE OPERATION

2. Obtain a carriage control tape with feed holes, channels numbered 1 through 12, and lines numbered 0 to at least 135. In this case, the tape loop accommodates two forms.
3. Align the tape in the punch, so that you begin punching in line 0.
4. Using Table 7-1 as a guide, proceed line by line on the tape, and punch a hole for each of the channels indicated on a line.

Table 7-1
Standard LP10 Line Printer Carriage Tape

Form 1 Line	Form 2 Line	Channels Punched
00	66	1-2-3-4-5-6-7-8-9-10-11-12
01	67	5-8
02	68	3-5-8
03	69	4-5-8
04	70	3-5-8
05	71	5-8-9
06	72	3-4-5-8
07	73	5-8
08	74	3-5-8
09	75	4-5-8
10	76	3-5-6-8-9
11	77	5-8
12	78	3-4-5-8
13	79	5-8
14	80	3-5-8
15	81	4-5-8-9-10
16	82	3-5-8
17	83	5-8
18	84	3-4-5-8
19	85	5-8
20	86	3-5-6-7-8-9
21	87	4-5-8
22	88	3-5-8
23	89	5-8
24	90	3-4-5-8
25	91	5-8-9-11
26	92	3-5-8
27	93	4-5-8
28	94	3-5-8
29	95	5-8
30	96	2-3-4-5-6-8-9-10
31	97	5-8
32	98	3-5-8

DEVICE OPERATION

Table 7-1 (Cont.)
Standard LPl0 Line Printer Carriage Tape

Form 1 Line	Form 2 Line	Channels Punched
33	99	4-5-8
34	100	3-5-8
35	101	5-8-9
36	102	3-4-5-8
37	103	5-8
38	104	3-5-8
39	105	4-5-8
40	106	3-5-6-7-8-9-12
41	107	5-8
42	108	3-4-5-8
43	109	5-8
44	110	3-5-8
45	111	4-5-8-9-10
46	112	3-5-8
47	113	5-8
48	114	3-4-5-8
49	115	5-8
50	116	3-5-6-8-9-11
51	117	4-5-8
52	118	3-5-8
53	119	5-8
54	120	3-4-5-8
55	121	5-8-9
56	122	3-5-8
57	123	4-5-8
58	124	3-5-8
59	125	5-8
60	126	5
61	127	5
62	128	5
63	129	5
64	130	5
65	131	5

DEVICE OPERATION

5. After you have punched all the lines through 131, cut the tape at line 135. Spread some rubber cement between lines 129 and 135. Place line 0 over line 132. Keeping the feed holes aligned, press the ends of the tape together.
6. Repunch the holes covered by the overlapped tape.
7. Set the printer off-line, and open the printer cover.
8. Lift the tape loop reader handle so that the sprocket shoe clears the sprocket teeth.
9. Place the tape over the tape loop reader capstan so that the feed holes are over the sprocket teeth. Be sure that channel 12 of the tape is toward your left as you are facing the front of the line printer.
10. Close the tape loop reader. Be sure the tape stays attached to the drive sprocket while you clamp the drive sprocket shoe in place.
11. Press TOP-OF-FORM. The tape should halt with the channel 1 star wheel just beyond the TOP-OF-FORM hole.
12. Align forms vertically. (Refer to Section 7.1.1.)
13. Close the printer cover.

The standard LP10 carriage tape causes the vertical actions shown in Table 7-2. Channels 1-8 are used by system software, and channels 9-12 are used by diagnostic programs.

DEVICE OPERATION

Table 7-2
Standard LP10 Line Printer Vertical Action

Channel	Vertical Action
1	Top of form
2	Space half a page (30 lines)
3	Double space with automatic top of form after 30 impressions
4	Triple space with automatic top of form after 20 impressions
5	Single space (Channel 5 is the only channel that can be specified for the last 6 lines of the page.)
6	Space one sixth of a page (10 lines)
7	Space one third of a page (20 lines)
8	Single space with automatic top of form after 60 impressions
9	Space one twelfth of a page (5 lines)
10	Space one fourth of a page (15 lines)
11	Space 25 lines
12	Space 40 lines

A programmer wanting a nonstandard carriage control tape must tell you what channels need to be punched on each line. Be sure that you punch a tape to accommodate whatever number of forms are needed to make the tape loop at least ten inches in circumference.

7.1.3 Changing and Reversing the Ribbon

If the print quality isn't as good as it should be, you may need to change or reverse the ribbon.

LP07 Procedure

The following procedure describes how to change the ribbon on an LP07 line printer. (Refer to Figures 7-1, 7-3, and 7-6.)

1. Lift the operator panel access cover and set the POWER circuit breaker to the OFF position.
2. Lift the printer window; unlatch and fully open the band gate leaving the ribbon mask against the forms.

DEVICE OPERATION

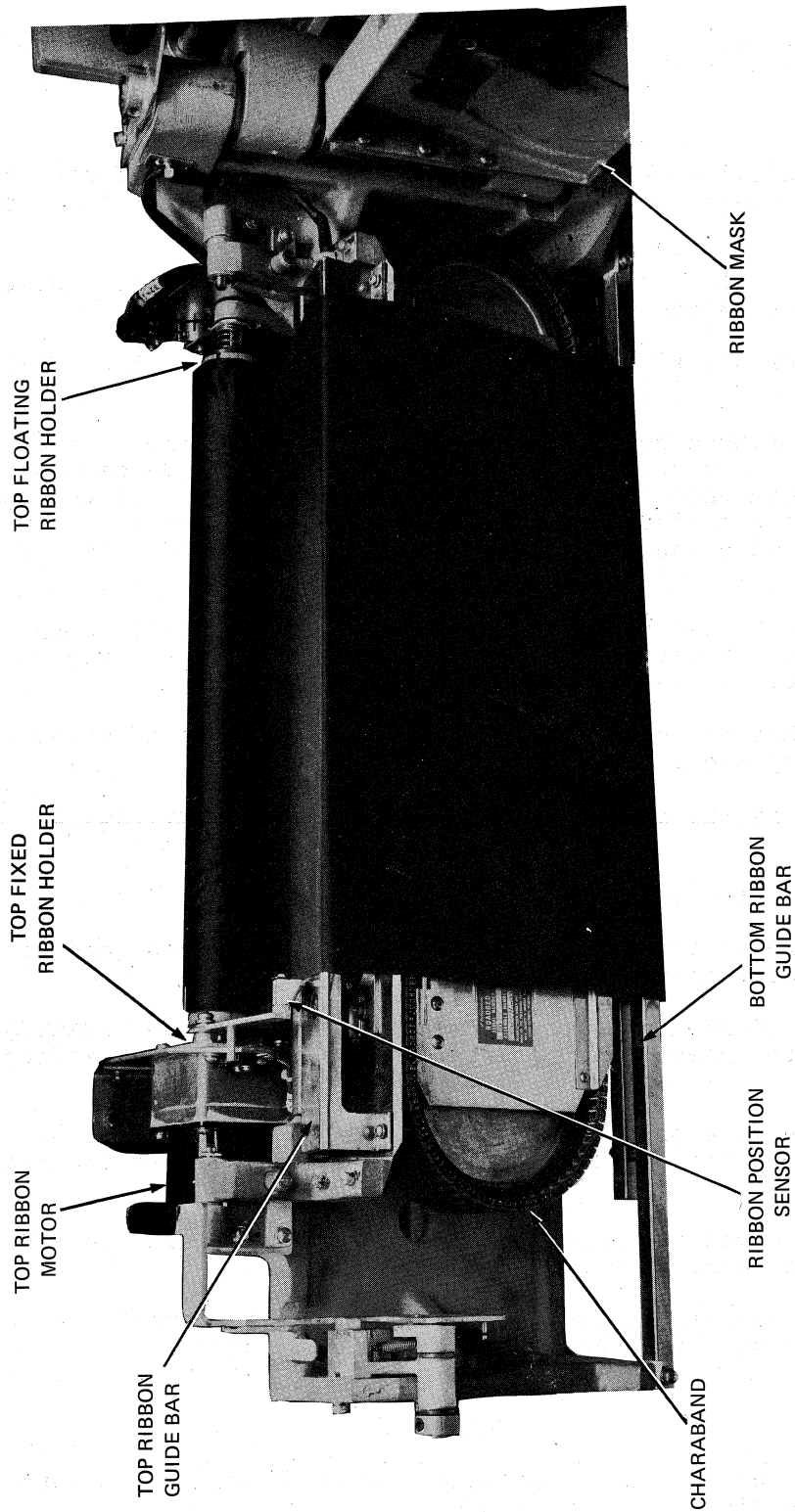


Figure 7-6 LP07 Ribbon Installation

DEVICE OPERATION

WARNING

Wait until the Charaband stops rotating before proceeding to the next step.

3. Using plastic gloves, grasp the ribbon rolls and force them to the right until the ribbon cores are free of the ribbon motors.
4. Pull the ribbon completely free of the band gate. Discard the ribbon unless you are reversing it.
5. Remove the new ribbon from its box and remove the plastic wrapping and any protective cardboard collars.
6. Hold the ribbon cores with the fully wound core above the other so the ribbon will unwind from the underside of the ribbon core when installed. Force the right end of the upper ribbon core against the top floating ribbon holder, align the left end with the fixed ribbon holder, and allow the spring tension to lock the core into position.
7. Unwind the other ribbon core over the ribbon guide bars and Charaband. Position the lower ribbon core between the bottom ribbon holders in the same manner.
8. Ensure that the slots in the ends of the ribbon cores are engaged by the ribbon holder guide pins.
9. Ensure that the ribbon position sensor is not covered by the ribbon.
10. Close the band gate and printer window.
11. Set the POWER circuit breaker to the ON position and close the operator panel access cover.
12. When the READY indicator comes on, press the START/STOP switch to place the printer in RUN mode. (RUN indicator lit).

Some listings have more print on the left half of the forms. To prevent the ribbon from wearing unevenly, you can reverse the ribbon as follows:

1. Lift the operator panel access cover and set the POWER circuit breaker to the OFF position.
2. Lift the printer window; unlatch and fully open the band gate leaving the ribbon mask against the forms.

WARNING

Wait until the Charaband stops rotating before proceeding to the next step.

DEVICE OPERATION

3. Using plastic gloves, grasp the ribbon rolls and force them to the right until the ribbon cores are free of the ribbon motors.
4. Pull the ribbon completely free of the band gate.
5. Holding the two ribbon cores as you took them off the printer, rotate them so that the top and bottom cores are reversed.
6. Force the right end of the upper ribbon core against the top floating ribbon holder, align the left end with the fixed ribbon holder, and allow the spring tension to lock the core into position.
7. Unwind the other ribbon core over the ribbon guide bars and Charaband. Position the lower ribbon core between the bottom ribbon holders in the same manner.
8. Ensure that the slots in the ends of the ribbon cores are engaged by the ribbon holder guide pins.
9. Ensure that the ribbon position sensor is not covered by the ribbon.
10. Close the band gate and printer window.
11. Set the POWER circuit breaker to the ON position and close the operator panel access cover.
12. When the READY indicator comes on, press the START/STOP switch to place the printer in RUN mode. (RUN indicator lit).

LP10 Procedure

The following procedure describes how to change the ribbon on an LP10 line printer. (Refer to Figures 7-2 and 7-7.)

1. Use the plastic gloves supplied with the ribbon.
2. Press the ON/OFF LINE switch to set the printer off-line (ON LINE indicator not lit).
3. Lift the printer window.
4. Move the drum gate latch left and pull forward to fully open the drum gate.

WARNING

Wait for character drum to stop rotating before you proceed.

5. Grasp the ribbon cores and force them to the left toward the drum gate latch until the floating ribbon-holder springs are completely compressed.
6. Remove the ribbon by pulling the right end of the ribbon cores away from the drum gate. Discard the ribbon unless you are reversing it.

DEVICE OPERATION

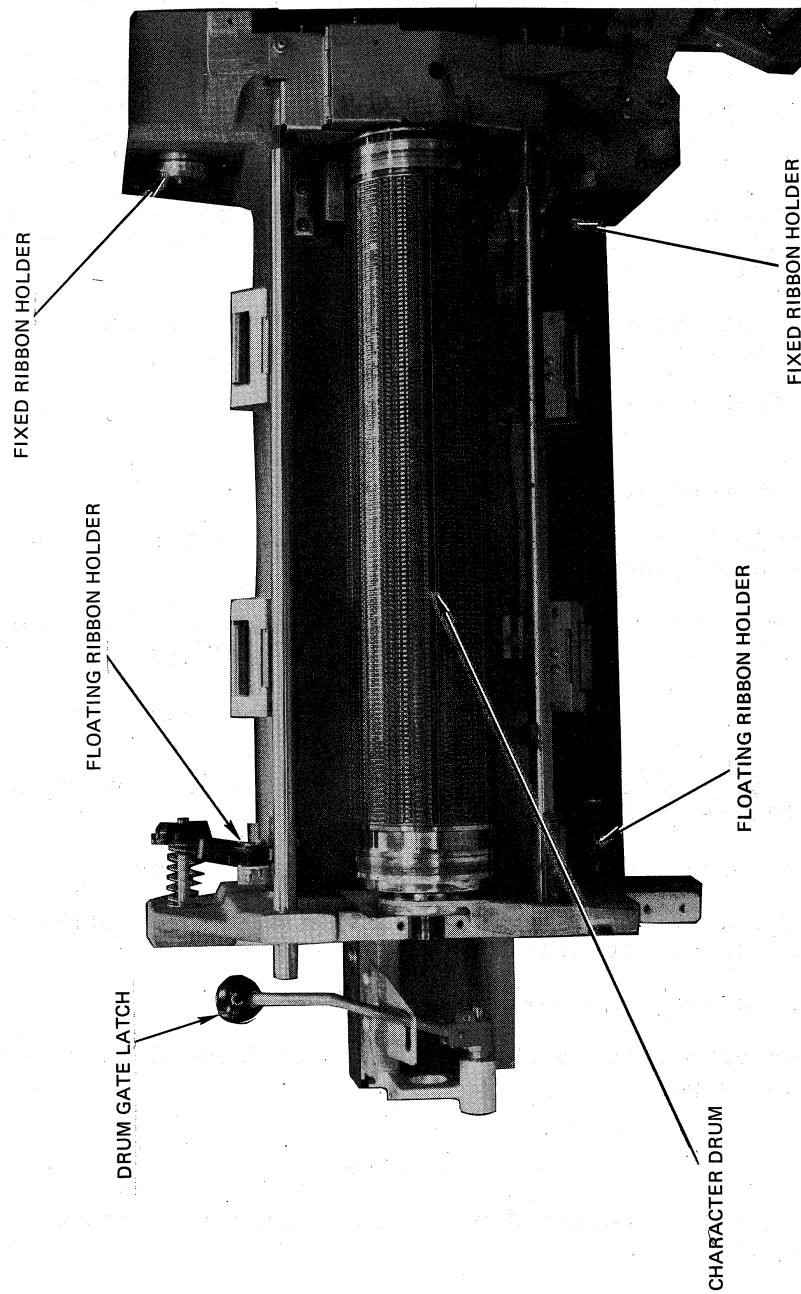


Figure 7-7 LP10 Ribbon Installation

DEVICE OPERATION

7. Remove the new ribbon from the box and hold the ribbon cores together. Remove any protective cardboard collars.
8. Place the fully wound ribbon core over the bottom floating ribbon holder.
9. Push the core to the left and place the right end over the bottom fixed ribbon holder. Be sure that the holder guidepin slips into the slot on the core end.
10. Unwind enough of the fully wound ribbon core to bring the ribbon up and over the character drum.
11. Slip the ribbon between the ribbon guide clips and the box sensor.
12. Place the ribbon core against the top floating ribbon holder.
13. Push the core to the left and place the right core end over the top fixed ribbon holder. Be sure that the holder guidepin slips into the slot on the core end.
14. Close and latch the drum gate.
15. Close the printer window.
16. Press the ON/OFF LINE switch to set the printer on-line (ON LINE indicator lit).

Most listings have more print on the left half of the paper. Therefore, to prevent the ribbon from wearing unevenly, you should reverse the ribbon. Do the following:

1. Use the plastic gloves supplied with the ribbon.
2. Press the ON/OFF LINE switch to set the printer off-line (ON LINE indicator not lit).
3. Lift the printer window.
4. Move the drum gate latch left and pull forward to fully open the drum gate.

WARNING

Wait for character drum to stop rotating before you proceed.

5. Grasp the ribbon cores and force them to the left toward the drum gate latch until the floating ribbon-holder springs are completely compressed.
6. Remove the ribbon by pulling the right end of the ribbon cores away from the drum gate.
7. Holding the two ribbon cores as you took them off the printer, rotate them so that the top core is at the bottom.
8. Push the core to the left and place the right end over the bottom fixed ribbon holder. Be sure that the holder guide pin slips into the slot on the core end.

DEVICE OPERATION

9. Unwind enough of the fully wound ribbon core to bring the ribbon up and over the character drum.
10. Slip the ribbon between the ribbon guide clips and the box sensor.
11. Place the ribbon core against the top floating ribbon holder.
12. Push the core to the left and place the right core end over the top fixed ribbon holder. Be sure that the holder guide-pin slips into the slot on the core end.
13. Close and latch the drum gate.
14. Close the printer window.
15. Press the ON/OFF LINE switch to set the printer on-line (ON LINE indicator lit).

7.1.4 Cleaning the Line Printer

In addition to the general cleaning procedures given at the beginning of the chapter, you should pay particular attention to the printing elements.

You should inspect the Charaband assembly on the LP07 daily for excessive ink and paper residue collection. Use isopropyl alcohol to clean away ink accumulation from the Charaband slugs. Whenever the Charaband is removed or reversed, the roadbed and top ribbon guide should be vacuumed.

You should clean the character drum on the LP10 every 100 hours, or sooner if necessary, using a soft suede brush and a vacuum cleaner. Every 500 hours, or sooner if necessary, clean the character drum with isopropyl alcohol.

7.2 CARD READER

The operator control panels for the low-speed reader CR10-F and the high-speed reader CR10-E are shown in Figures 7-8 and 7-9, respectively.

7.2.1 Loading and Unloading Cards

Use the following procedures to load the input hopper with punched cards to be read.

1. Be sure the reader is powered ON (POWER indicator lit), and check that the MODE switch on the rear control panel of the reader is set to REMOTE.
2. Pull the hopper follower back with one hand and begin loading card decks into the hopper. Be sure to place the first card to be read at the front with the 9 edge down, column 1 to the left.

DEVICE OPERATION

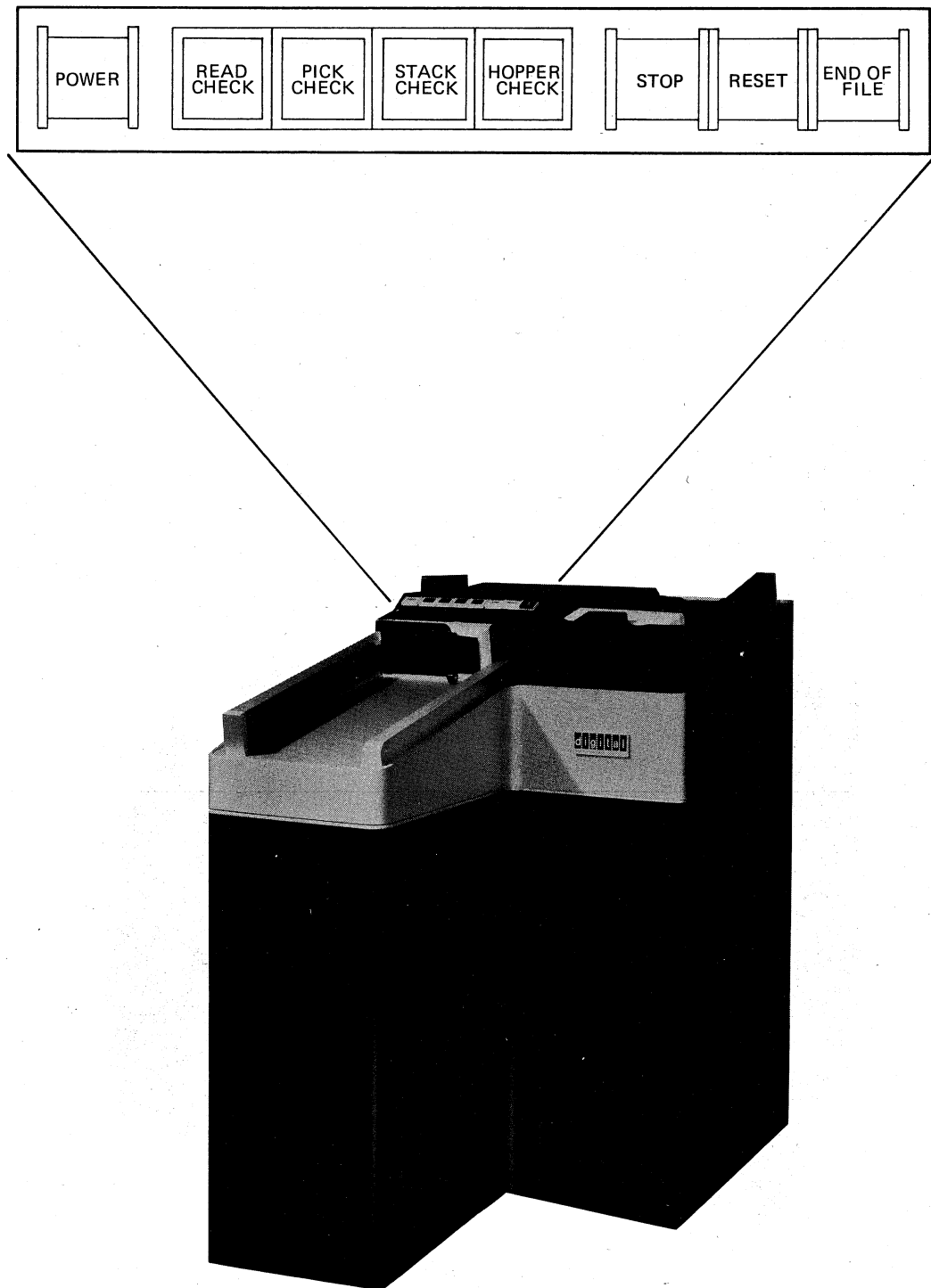


Figure 7-8 CR10-F Card Reader

DEVICE OPERATION

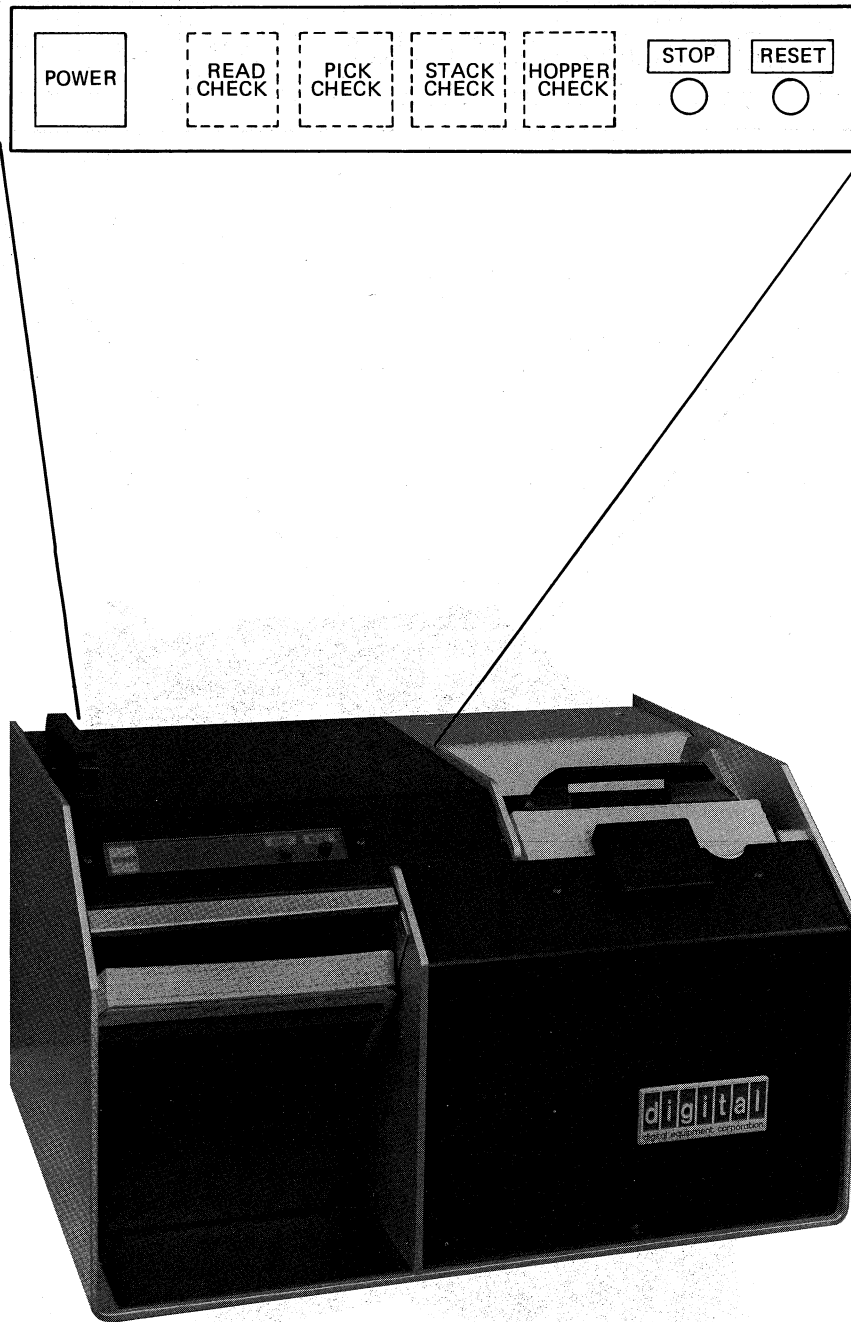


Figure 7-9 CR10-E Card Reader

DEVICE OPERATION

3. Continue placing cards into the input hopper until it is loosely filled.

CAUTION

Do not pack the input hopper so full that the air from the blower cannot riffle the cards properly. If the cards are packed too tightly, the vacuum picker cannot work properly.

4. Press RESET to allow cards to be read.
5. You may continue loading cards while the reader is operating provided you maintain tension in the front portion of the deck as you add cards to the rear. Do not add more cards until the hopper is at least half empty.

CAUTION

When you are trying to maintain pressure on the card deck, use just enough pressure to maintain the riffle action to prevent damaging cards or jamming the reader.

6. Usually, all cards are moved through the reader into the stacker. However, if you need to remove cards from the input hopper, simply pull back the follower and remove the card deck.
7. To unload cards from the output stacker, pull the stacker follower back with one hand and remove the card deck from the stacker. Be careful to maintain the order of the deck. The stacker may be unloaded while cards are being read.

7.2.2 Cleaning the Card Reader

The important cleaning area for a card reader is the card path, which accumulates card dust. If this builds up, it causes data errors, card motion errors, and possibly even card jams. The card path should be cleaned at least weekly. With the power OFF, the path from the hopper to the stacker should first be blown out and then well vacuumed.

7.3 DECTAPE

The following procedures describe the actions necessary to mount and dismount a DECTape on the TU56 transport. To assign a drive to your job, enter:

.ASSIGN DTAn (where n = 0-7, the number on the drive selector indicator of some unused drive. Note, however, that on some drives 0 is replaced by 8; you must still enter DTA0.

DEVICE OPERATION

7.3.1 Mounting and Dismounting DECTapes

Refer to Figure 3-1 for a view of the TU56 transport.

Mounting

1. Place the REMOTE/LOCAL switch to OFF (middle position). This switch is located immediately to the right of the drive selector indicator.
2. Mount the DECTape onto the left hand hub so the tape feeds upward at the left perimeter of the reel.
3. Thread the tape to the right, over the top of the tape guides and tape head, and clockwise onto the take-up reel.
4. Wind several turns of tape clockwise onto the take-up reel.
5. Press the REMOTE/LOCAL switch into the LOCAL (bottom pressed) position.
6. Advance the tape about ten turns onto the take-up reel by depressing the top end of the rightmost rocker switch.
7. Press the REMOTE/LOCAL switch into the REMOTE (top pressed) position.
8. To write on the tape, set the WRITE ENABLE/WRITE LOCK switch to the WRITE ENABLE (top pressed) position.

Dismounting

If the software has rewound and unloaded the tape, skip steps 1 and 2.

1. Set the REMOTE/LOCAL switch to the LOCAL position.
2. Wind the tape completely onto the left reel by pressing the rewind switch (top end of the leftmost rocker switch) and holding it pressed until all the tape has wound onto the left reel.
3. Place the REMOTE/LOCAL switch to OFF (middle) position and remove the tape.

7.3.2 Cleaning DECTape Drives

DECTapes do not have a complicated tape path; however, the brown oxide powder that flakes off when a tape is in motion will accumulate on the tape guides and tape head. Once each shift, or at least once daily, you should clean these areas with a cotton-tipped applicator and 91% isopropyl alcohol.

7.4 MAGNETIC TAPE

The DECsystem-10 supports a variety of magnetic tape drives. They cover both 7 track and 9 track operation, tape speeds of 45 to 200 inches per second, and recording densities of 200 to 6250 bits per inch. Various size reels can be accommodated and some drives will accept industry-standard cartridges.

DEVICE OPERATION

7.4.1 Mounting and Dismounting

The following procedures describe the action necessary to mount and dismount magnetic tape reels on the various drives supported by the DECsystem-10. Some drives provide for the automatic threading of the tape onto the take-up reel while others require you to thread the tape manually. For all drives, you must insert a write enable ring in the groove on the back of the tape reel in order to allow writing. If this ring is not present, an indicator light (FILE PROTECT on some drives, WRITE DISABLE on others) will warn you that writing is inhibited.

7.4.1.1 TU10 and TU16 Drives (Refer to Figure 7-10)

Mounting

1. Rotate the reel holddown knob of the lower hub counterclockwise as far as it will go, and place the supply reel over the holddown knob with the groove toward the back. Hold the reel firmly against the hub flange and turn the holddown knob clockwise until it is tight.
2. Press BR REL and pull the tape from the supply reel. The tape should unwind from the left with the oxide (dull) side toward the hub. If the tape unwinds from the top, check that the reel is mounted with the groove toward the back.
3. Grasp the tape in both hands with the left hand at the end and the right hand back about a foot. Place the tape against the bottom and left side of the head cover over the threading slot. Pull the tape taut and slide it upwards and into the slot. Lead the tape around the upper right corner of the vacuum column cover, slide it into the slot between the cover and the upper tape guide, and lead it around the guide. Place the end of the tape over the top of the take-up reel hub, and wind about six turns of the tape clockwise on the reel.
4. Press LOAD to draw tape into the vacuum columns. Press FWD and push START to advance the tape to loadpoint. When the loadpoint marker is sensed, the tape stops, the FWD light goes out, and LD PT comes on. Note that if tape motion continues for more than ten seconds, the tape is beyond loadpoint. Press STOP, REV and then START. The tape should move back and stop at loadpoint.
5. Press ONLINE.

Dismounting

1. Rewind to loadpoint by pressing OFF LINE, setting the rocker switch to REWIND, and pressing START.

DEVICE OPERATION

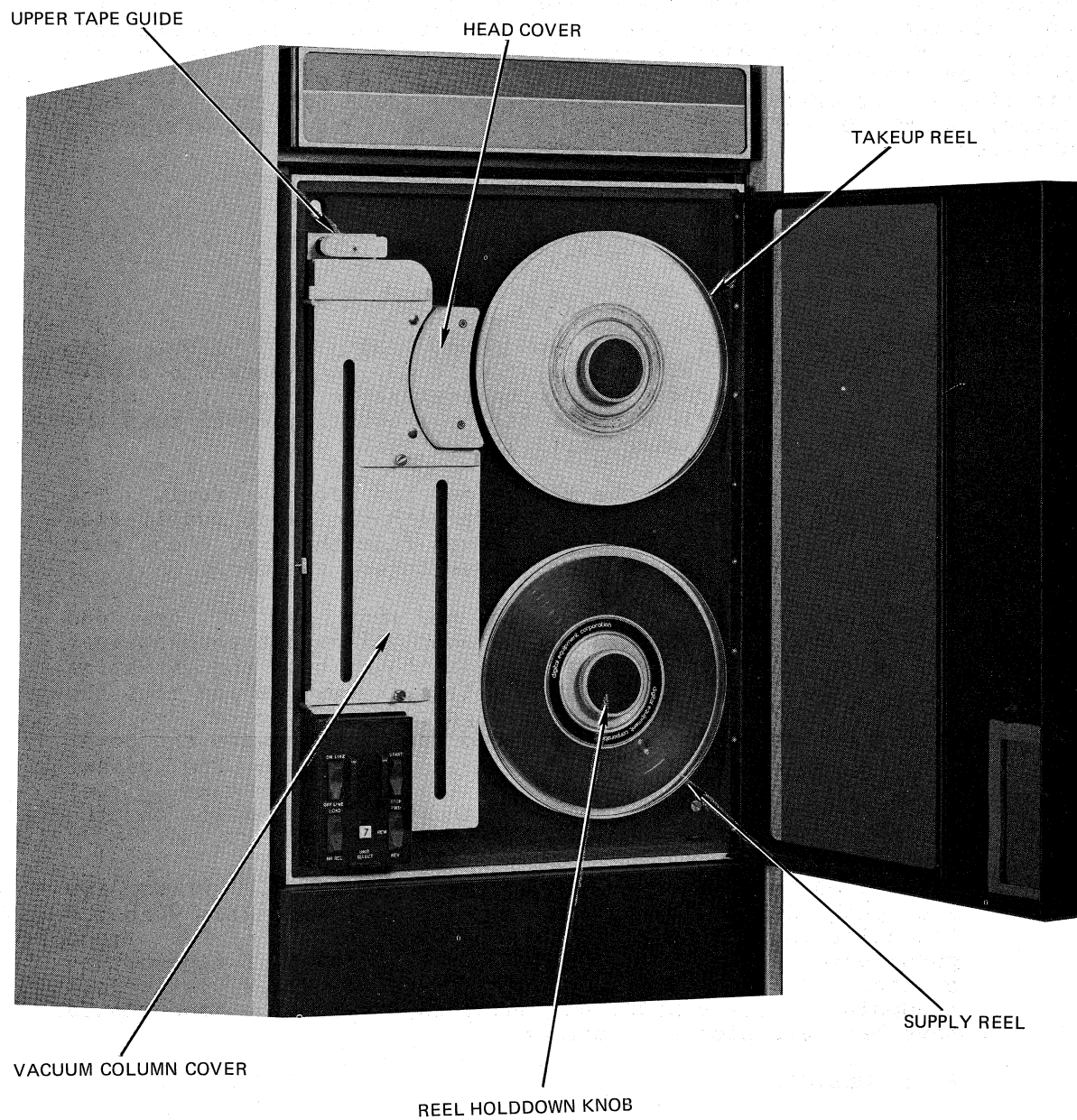


Figure 7-10 TU16 Magnetic Tape Drive

DEVICE OPERATION

2. Press BR REL and wind the remaining tape onto the supply reel.
3. Turn the holddown knob counterclockwise and remove the reel.

7.4.1.2 TU40 and TU41 Drives (Refer to Figure 7-11)

Mounting

1. If the transport is closed, press UNLOAD REWIND to lower the window. Pull one end of the reel lock release on the right hub and place the supply reel over the hub. If the reel does not have a cartridge, slip the end of the tape into the shoe at the lower right of the reel. If the reel has a cartridge, place it on the hub so that the projections from the cartridge fit into the two reel-positioning guides. The pin on the motor face plate should go into the opening on the tape cartridge toggle.
2. Press LOAD REWIND to load the tape automatically. Press START to place the unit on line.

Dismounting

1. Press RESET and then UNLOAD REWIND.
2. When the window has opened, remove the tape by opening the reel lock release and pulling the reel from the hub.

7.4.1.3 TU70, TU71, and TU72 Drives (Refer to Figure 7-12)

Mounting

1. Press the HUB/WINDOW UP button to open the transport window.
2. Push the tape onto right-hand hub so that the tape unwinds clockwise. (The tape snaps onto the hub and is automatically secured.)
3. Ensure that the end of tape is not twisted and lies flush on the reel.
4. Press the RESET button, causing the window to close.
5. Press LOAD REWIND button, putting the tape at load point.
6. Press START. READY light goes on.

If the tape did not load correctly, the MACHINE CHECK button will flash, and the window will open. The operator should then reposition the tape and repeat steps 4, 5, and 6 described above.

Dismounting

1. Press RESET.

DEVICE OPERATION

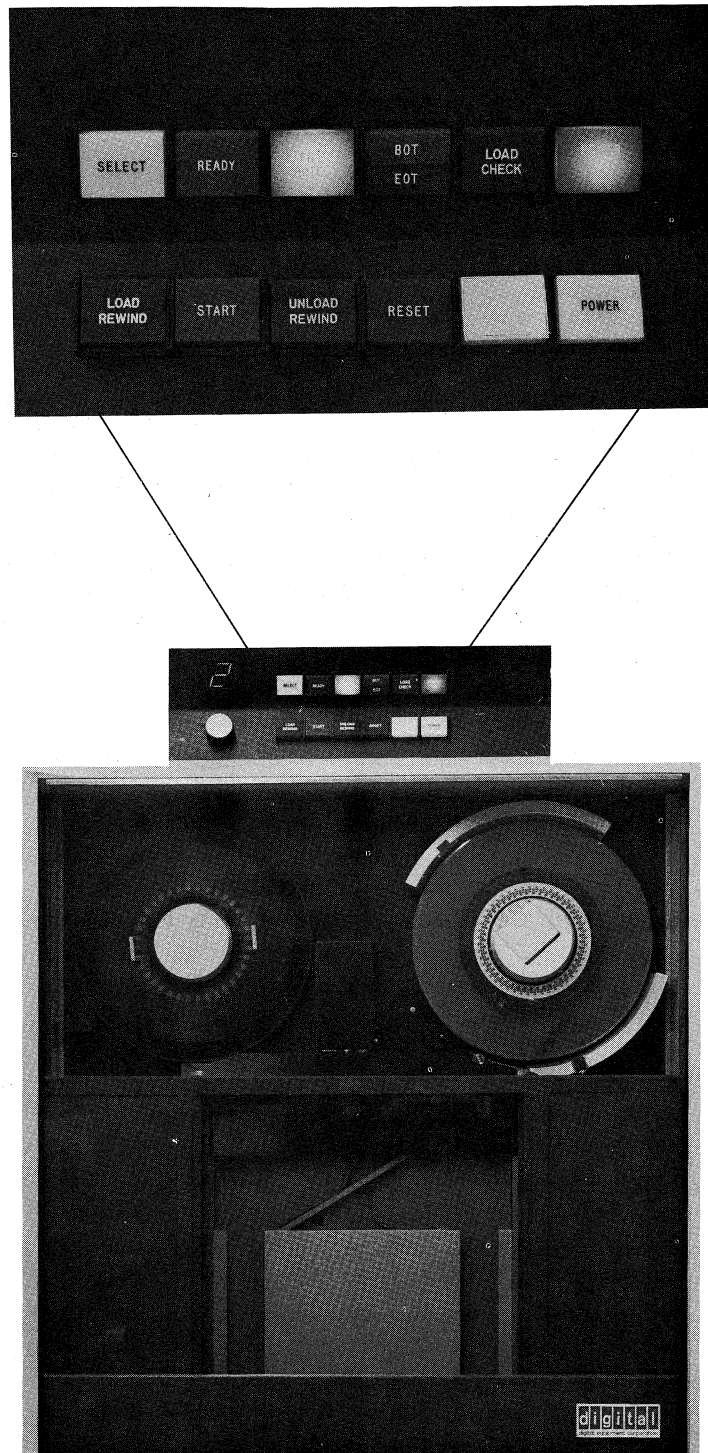


Figure 7-11 TU40 Series Magnetic Tape Drive

DEVICE OPERATION

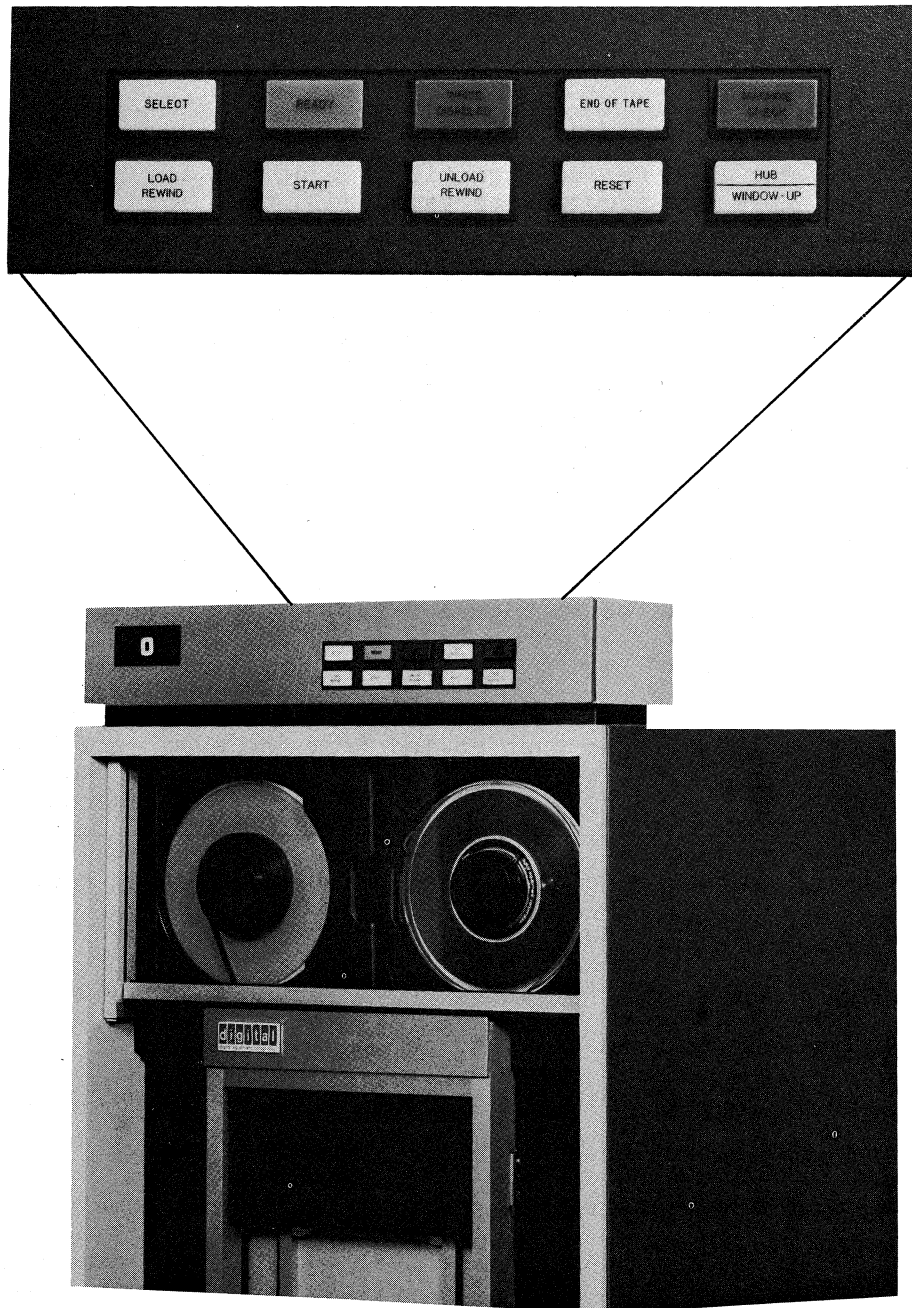


Figure 7-12 TU70 Series Magnetic Tape Drive

DEVICE OPERATION

2. Press UNLOAD REWIND button causing the tape to rewind completely and the transport window to open.
3. Pull the tape off the hub.

NOTE

The TU70 series magnetic tape drives are able to accommodate tape reels that are enclosed in cartridges. When mounting, the cartridge must be positioned on the hub so that the two tabs on either side of the cartridge toggle fit into the two slots on the left side of the right-hand hub. (The cartridge toggle will automatically be opened in the loading process and closed in the unloading process.)

7.4.1.4 TU45 Drives (Refer to Figure 7-13)

Mounting

The supply reel (reel to be read or written) is located at the top of the transport. The tape must unwind from the supply reel when the reel is turned in a clockwise direction. Position the reel on the quick-release hub and depress the hub actuator. Thread the tape as follows:

1. Toward the arrow to the upper left of the supply reel
2. Over the two top guide wheels from right to left
3. Down and to the left of the upper white guide wheel
4. Slightly right and down, to go under the head
5. Down and to the left of the lower white guide wheel
6. Down under the capstan
7. Up and right, toward the arrow to the left of the takeup reel

Wrap the tape leader onto the take-up reel so that the tape will be wound onto the reel when it is rotated clockwise. Wind at least three turns onto the take-up reel. Remove all slack in the tape path.

After the tape has been loaded and checked for correct seating in the guides, position the tape at the Load Point by pressing the LOAD switch. Recheck tape tracking in the guides and close the dust cover door.

NOTE

The transport door should remain closed during normal operation to achieve maximum data reliability.

DEVICE OPERATION

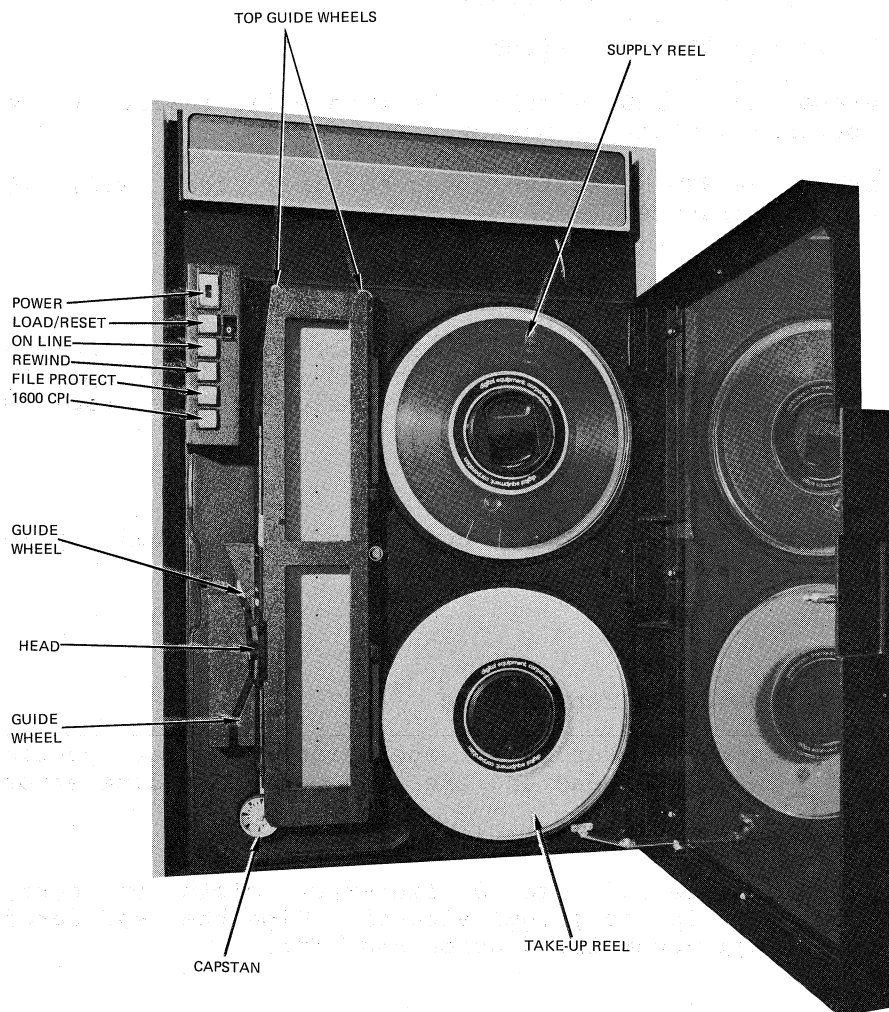


Figure 7-13 TU45 Magnetic Tape Drive

DEVICE OPERATION

When the load cycle is complete, the tape will stop at the Load Point. Then, if the ONLINE indicator is not lit, press ONLINE. The transport is now ready to receive additional commands.

Dismounting

To unload a tape, do the following:

1. Press the OFFLINE switch.
2. Press the REWIND switch; the tape will rewind to the BOT (beginning of tape) marker.
3. Press the REWIND switch a second time; this will start an unload sequence.

NOTE

It is not necessary to wait for the tape to rewind to the BOT marker if an unload is desired; press the REWIND switch twice in succession and the tape will rewind and unload.

4. Open the transport door and remove the reel. Close the transport door.

7.4.2 Cleaning Magnetic Tape Drives

Cleanliness is essential for proper operation. Minute particles of dirt trapped between the head and the tape can cause data errors.

Each day you should do the following:

1. Clean the head. Use a lint-free cloth or cotton swab moistened in isopropyl alcohol. Wipe the head carefully to remove all accumulated oxide and dirt.

NOTE

Do not use rough or abrasive cloths to clean the head.

2. Clean the fixed guides, capstan, and vacuum chamber with a cotton swab moistened with isopropyl alcohol to remove accumulated oxide and dirt. The vacuum column door may be opened to gain access to the chamber surfaces.
3. Clean the rotating tape guides and rubber punch rollers with a lint-free cloth or cotton swab moistened in isopropyl alcohol. Wipe the surfaces carefully to remove all accumulated oxide and dirt.

DEVICE OPERATION

NOTE

Do not soak the guides with excessive solvent; it can seep into the guide bearings, causing contamination or a breakdown of the bearing lubricant.

7.5 DISK DRIVES

Three types of disk drives with removable disk packs, along with their operator control panels, are shown in Figures 7-14, 7-15, and 7-16. The RHS04 fixed-head swapping disk system does not require operator action.

7.5.1 Mounting and Dismounting Disk Packs

The following steps describe how to mount and dismount disk packs on the RP02, RP03, RP04, and RP06 disk drives. Extreme care must be taken to avoid physical damage to the pack and drive. Do not put anything on top of a drive except empty disk pack covers. (Refer to Figures 7-14, 7-15, and 7-16 for the operator controls mentioned in the following instructions.)

Mounting

1. Slide back or lift the drive cover.
2. Remove the bottom plate from the disk pack container by means of the locking handle located on the underside of the bottom plate.
3. Be sure that you have the pack with the correct visual information on it; slide the pack straight down onto the disk drive hub.
4. Turn the pack cover clockwise until the pack is locked into position. DO NOT FORCE THE COVER ONCE IT STOPS TURNING.
5. Pull the pack cover straight up and off the pack.
6. Close the drive cover and put the empty disk pack cover on top of the drive.
7. Push or toggle the START/STOP switch to the START position.
8. When the READY light is on, the drive is ready.

Dismounting

1. Push or toggle the START/STOP switch to the STOP position and wait until the pack comes to a complete stop.
2. Slide back or lift the drive cover.
3. Slip the pack cover straight down over the pack.

DEVICE OPERATION

4. Turn the cover counterclockwise until the pack is free.
5. Lift the pack straight up until it clears the drive.
6. Secure the bottom plate to the pack cover by means of the locking handle located on the underside of the bottom plate.
7. Close the drive cover.

7.5.2 Cleaning Disk Drives

Other than keeping the outside of the drive and the cover of the packs clean, you should leave this task to your field service representative.

7.5.3 Formatting Disk Packs

Refer to the section called Formatting Disk Packs in Chapter 5.

DEVICE OPERATION

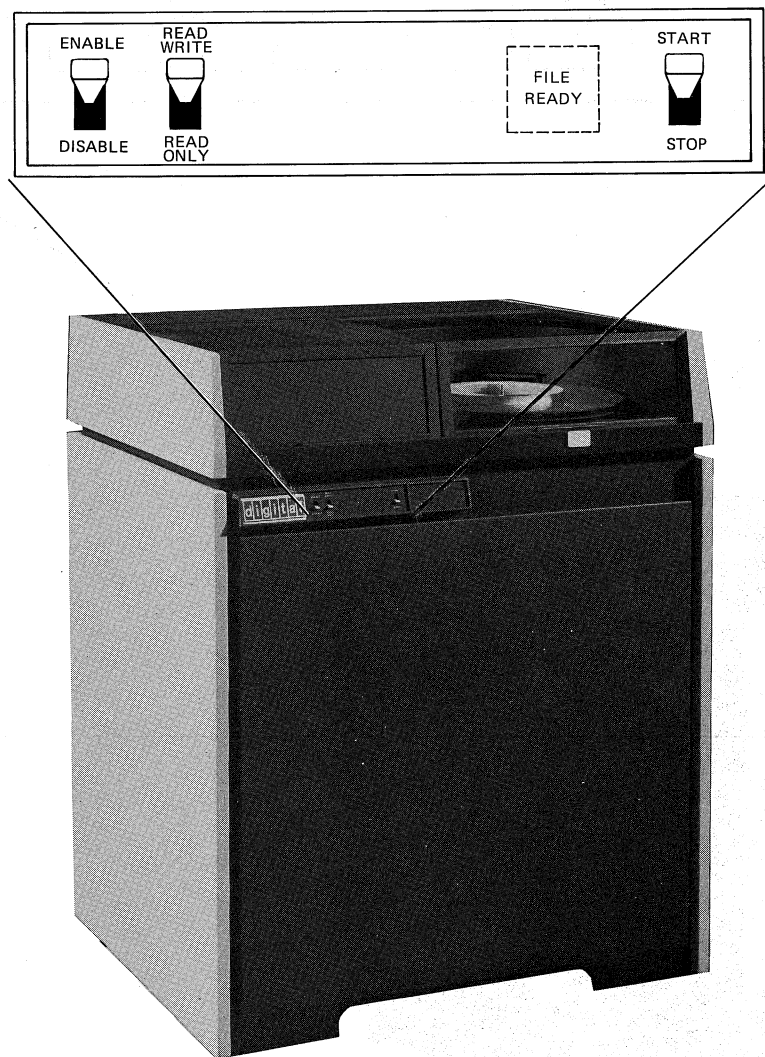


Figure 7-14 RP03 Disk Drive

DEVICE OPERATION

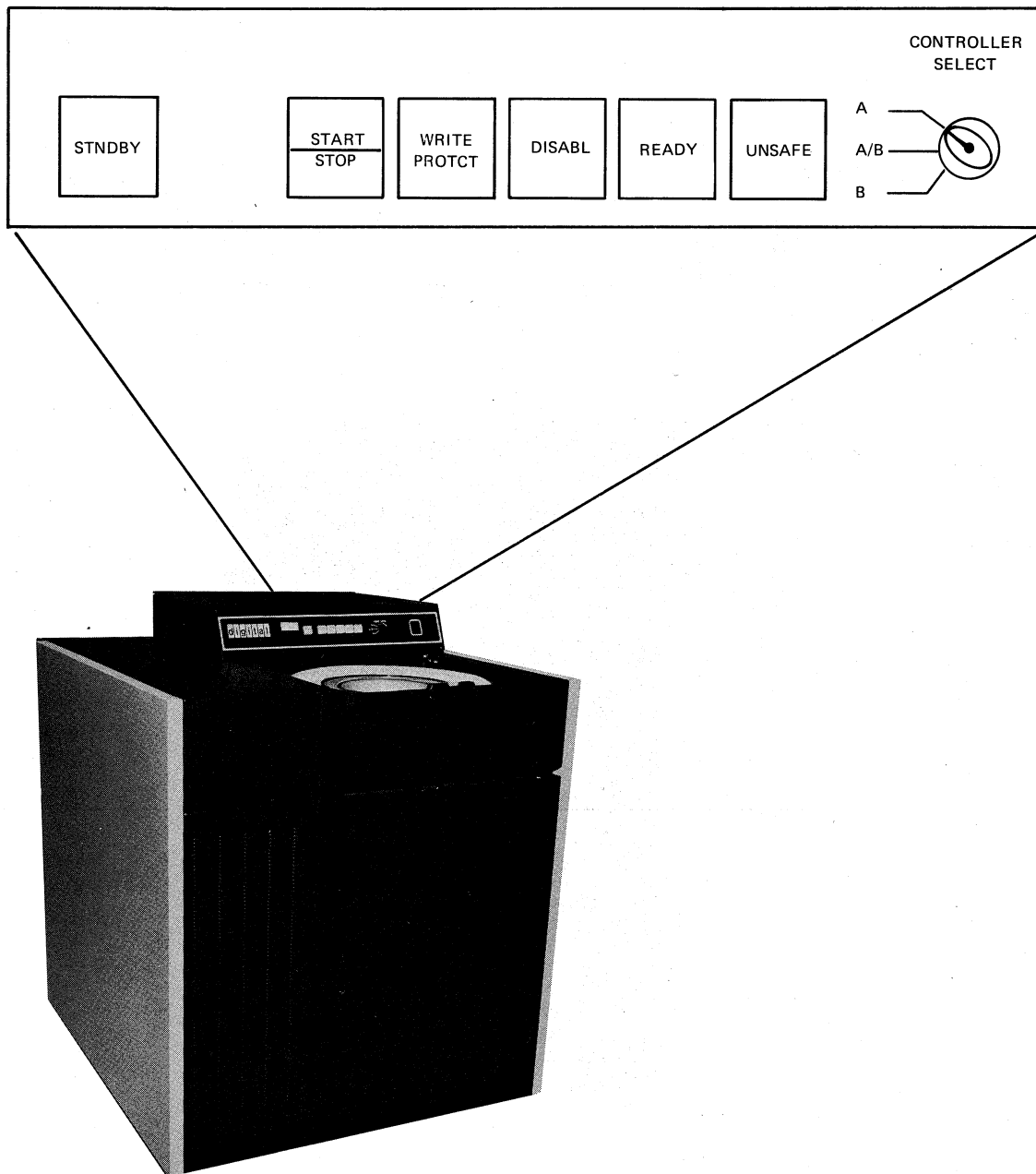


Figure 7-15 RP04 Disk Drive

DEVICE OPERATION

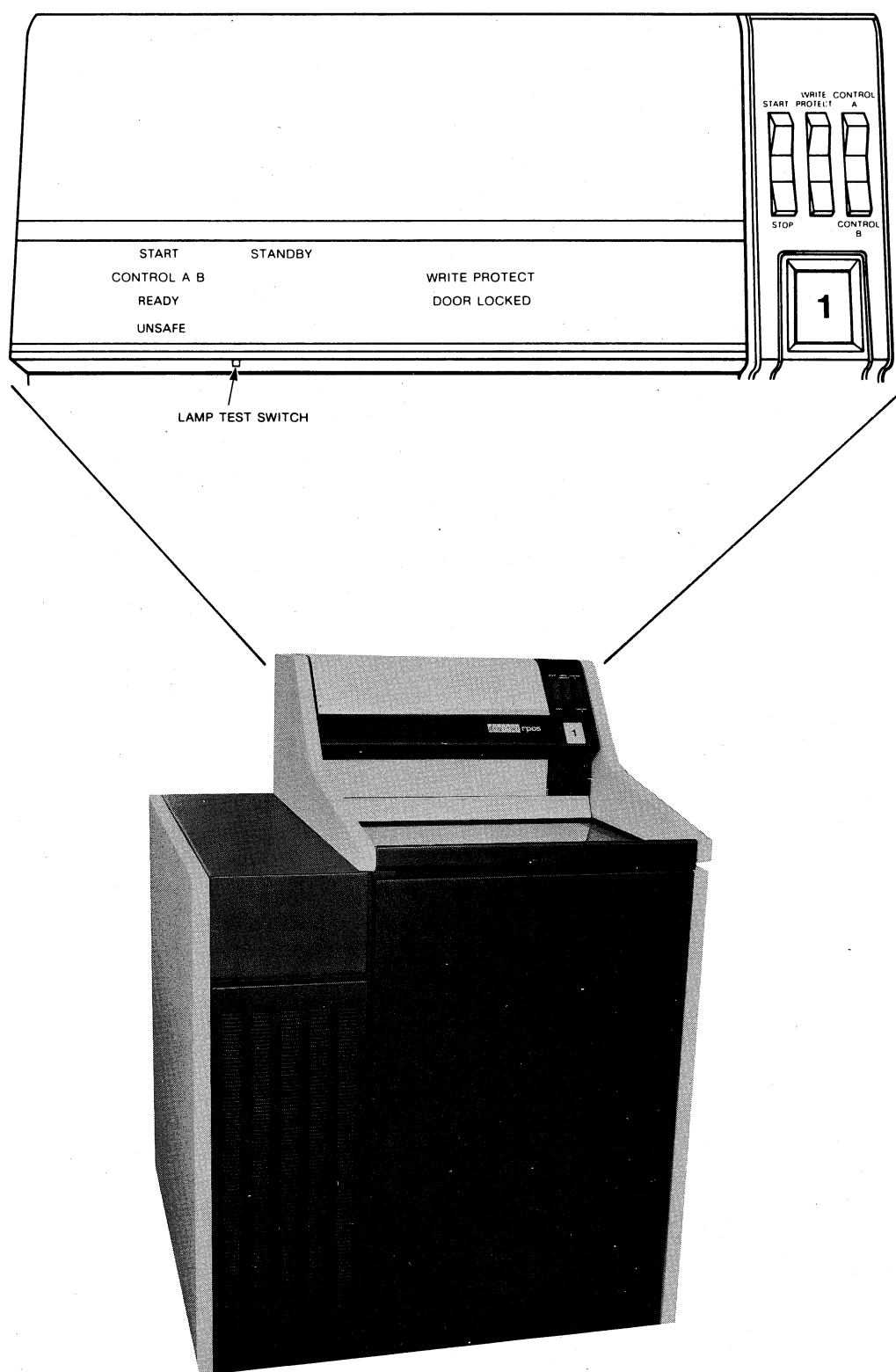


Figure 7-16 RP06 Disk Drive

APPENDIX A
TOPS-10 OPERATOR-PRIVILEGED COMMANDS

A.1 INTRODUCTION

The commands in this appendix are used to change various system parameters and, therefore, are restricted to users logged in under [1,2] or using device OPR. The commands discussed do not require LOGIN, unless otherwise stated; at completion they leave the terminal in monitor mode. (Note that operator-privileged commands for the OPSER system program are described in Section 4.1.2.1.)

The remainder of the operating system commands (commands available to the general user) are described in detail in the DECsystem-10 Operating System Commands manual.

A.2 COMMAND DESCRIPTIONS

ATTACH device

Function

The ATTACH device command returns a previously DETACHED device to the operator. The operator must then DEASSIGN the device in order to place it in the monitor pool of available resources.

Command Format

ATTACH dev

dev=the device to which the operator is attaching.

Restriction

The device must have been previously DETACHED.

Associated Messages

?NO SUCH DEVICE

The specified device does not exist in this monitor configuration.

?WASN'T DET

The specified device is not detached.

TOPS-10 OPERATOR-PRIVILEGED COMMANDS

ATTACH diskunit

Function

The ATTACH disk unit command designates that the specified disk unit is now up and the operator can then MOUNT packs on it.

Command Format

ATTACH diskunit/comment

diskunit=the disk unit that the operator wants to declare up, e.g., DPA2, RPA3.

comment=text, whose first two characters go into ERROR.SYS along with the command as configuration changes (PM=preventive maintenance, CM=corrective maintenance).

Restrictions

The unit must have been DETACHED or declared DOWN at system initialization.

Associated Messages

?NO SUCH DEVICE

The disk unit specified does not exist.

ATTACH job

Function

The ATTACH job command detaches the current job, if any, and connects the terminal to any job specified.

Command Format

ATTACH n[proj,prog]

n=the job number of the job to which the terminal is to be attached.

[proj,prog]=the project-programmer number of the originator of the desired job. This argument may be omitted if it is the same as the project-programmer number of the job to which the terminal is currently attached. The operator (device OPR) or a user logged-in under [1,2] may always attach to a job provided the proper [proj, prog] is specified.

Any other user, even with operator privileges, will be interrogated for the appropriate password.

Associated Messages

?NOT A JOB

The job number specified does not exist.

TOPS-10 OPERATOR-PRIVILEGED COMMANDS

?CAN'T ATT TO JOB

The [proj,prog] was not correct for the job number.

DETACH device

Function

The DETACH device command assigns the specified device to JOB 0, thus making it unavailable.

Command Format

DETACH dev

dev=the name of the device to be detached.

Restrictions

DSK and TTY cannot be detached.

Associated Messages

?ALREADY ASSIGNED TO JOB n

The device specified is already in use.

?CAN'T DET DEV

The user is attempting to detach DSK or TTY.

?NO SUCH DEVICE

The specified device does not exist in this monitor configuration.

DETACH diskunit

Function

The DETACH disk unit command designates the specified disk unit as being DOWN. This is similar to declaring it DOWN at system initialization.

Command Format

DETACH diskunit/comment

diskunit=the disk unit the operator wants to declare DOWN, e.g., DPA0, RPA4.

comment=text, whose first two characters go into ERROR.SYS along with the command as configuration changes (PM=preventive maintenance, CM=corrective maintenance).

Restrictions

The unit must not be part of a mounted file structure. If it is, a DISMOUNT must be issued for the structure; then, the disk unit may be DETACHED.

TOPS-10 OPERATOR-PRIVILEGED COMMANDS

Associated Messages

?NO SUCH DEVICE

The disk unit specified does not exist.

SET BREAK

Function

The address break facility of the hardware makes available to users of the system, a powerful debugging facility. To make this feature available the monitor must set the address which the user is breaking on into the address switches and the break conditions into the break conditions register. The SET BREAK command allows the operator to make the address break facility available to users; it also allows the operator to prevent it from being available to users if he wishes to make the address break facility available to system programmers for their exclusive use. The default setting when the monitor is loaded is SET BREAK USERS.

Command Format

SET BREAK USERS

gives users the address break facility. The system will change the address switches and break conditions for programs being debugged using address break.

SET BREAK NO USERS

causes the message ?NOT AVAILABLE to be typed on a user's console if the user attempts to use the address break facility.

Restrictions

The SET BREAK command depends on the value of FTSET as specified in the FGEN dialog during system generation. The SET BREAK command also requires LOGIN.

Associated Messages

?NOT AVAILABLE

The operator tried to SET BREAK NO USERS but the address break facility was already in use by one or more users.

SET CORMAX

Function

The SET CORMAX command is used to change the system parameter CORMAX. The largest size that any job (sum of low and high segments) can be is CORMAX-1.

TOPS-10 OPERATOR-PRIVILEGED COMMANDS

Command Format

SET CORMAX n

n=decimal number representing nK. The minimum value of n is ten unless changed at system generation or system initialization time.

Associated Messages

None

SET CORMIN

Function

The SET CORMIN command is used to change the system parameter CORMIN. CORMIN is the guaranteed amount of contiguous core that a single unlocked job can have. This command is used only with a monitor that has the capability of locking jobs in core (usually a real-time monitor).

Command Format

SET CORMIN n

n=decimal number representing nK. This argument is required.

Associated Messages

None

SET DATE

Function

The SET DATE command is used to change the date. Note, however, that this command does not check the validity of the argument as is the case at system initialization.

Command Format

SET DATE mm dd yy

mm=1- or 2-character decimal number of month.
dd=1- or 2-digit decimal day of month.
yy=2-digit decimal year.

Associated Messages

None

TOPS-10 OPERATOR-PRIVILEGED COMMANDS

SET DAYTIME

Function

The SET DAYTIME command changes the time of day.
Command Format

SET DAYTIME n

n=decimal number 0 through 2359, representing 24-hour time, i.e., hours * 100 + minutes. This argument is required.

Associated Messages

None

SET MEMORY

Function

The SET MEMORY command allows the operator to dynamically alter the system memory configuration without necessarily affecting system operation. The command is most useful for removing faulty memory from the system, returning repaired memory to the system, or reconfiguring systems that contain multiple CPU's. If the command is successfully executed, the specified memory is set on or off line.

Command Format

SET MEMORY ON-LINE FROM n TO m
SET MEMORY OFF-LINE FROM n TO m

where FROM and TO are optional portions of the command line.

n and m are numbers (n less than or equal to m) that specify the range of addresses that are to be set ON or OFF line. The numbers are treated as decimal unless preceded by # implying octal. If a number is followed by the letter P or K, the range is in pages (P) or blocks (K).

Restrictions

Available only on systems that have the LOCK facility.

Associated Messages

?ATTEMPT TO SET MONITOR MEMORY OFF-LINE

The range of addresses specified in a SET MEMORY OFF-LINE command overlaps the resident monitor. The memory occupied by the resident monitor cannot be set off-line.

?JOB(S) TOO BIG TO CONTINUE TO RUN

?PROBLEM WITH JOB(S) n1[prog1],n2[prog2],...nn[progn]

If the range of addresses specified in a SET MEMORY OFF-LINE command were taken off-line, the jobs listed would be too big to continue to run. The job numbers and program names are listed so that the operator can ATTACH to the jobs, KJOB them, and type the command again if he wishes the command to succeed.

TOPS-10 OPERATOR-PRIVILEGED COMMANDS

?ATTEMPT TO SET MEMORY CONTAINING LOCKED JOBS OFF-LINE
?PROBLEM WITH JOB(S) n1[progl],n2[prog2],...nn[progn]

The range of addresses specified in a SET MEMORY OFF-LINE command overlaps memory containing locked jobs. The job numbers and program names are listed so that the operator can ATTACH to the jobs, KJOB them, and type the command again, if he wishes the command to succeed.

?p NXM ERRS FROM n TO m

When a SET MEMORY ON-LINE command is executed, the system references the memory in the range specified in the command to ensure that it actually exists and is on-line. Should any nonexistent memory be encountered during this procedure, the range of nonexistent locations (n,m) is reported on the OPR terminal along with the number of errors (p) and the memory in that range is treated as off-line.

SET NOMESSAGE

Function

The SET NOMESSAGE command is used by the operator to turn off the once-a-minute message that is output when a peripheral device such as a disk, magtape, or PDP11 has dropped off-line. The message normally output is as follows:

```
UNIT xxx WENT OFF-LINE
PLEASE POWER DOWN AND TURN IT ON AGAIN
```

where xxx is the unit name.

When the unit is operating properly (that is, back on-line), the effect of this command is cancelled. The once-a-minute message will be restored for subsequent off-line conditions on the unit.

Command Format

SET NOMESSAGE unit

unit=a peripheral unit name, e.g., DPA3, RPA2, MTB, MTA3,
PDP11 0 or PDP11 * for all front-end -11's.

Associated Messages

None

SET OPR

Function

The SET OPR command is used to change the terminal designated as OPR to another terminal while the system is running.

Command Format

```
SET OPR TTYn
or
SET OPR CTY
```

TOPS-10 OPERATOR-PRIVILEGED COMMANDS

TTYn=the name of the terminal to be designated as OPR.
CTY=the terminal on the console front end.

Associated Messages

None

SET SCHED

Function

The SET SCHED command changes the scheduled use of the system.

Command Format

SET SCHED n

n is octal and is stored in the right half of the STATES word in COMMON.

n=0 regular timesharing and batch operations.

The following bits have independent meanings and several can be specified concurrently.

n=1 no further LOGINs allowed except from CTY.

n=2 no further LOGINs from remote terminals, and do not answer data sets.

n=4 batch only.

n=10 stand-alone

n=100 device mounts can be done without operator intervention.

n=200 unspooling allowed.

n=400 no operator coverage.

Associated Messages

None

SET TTY

Function

When used with a device argument, the SET TTY command declares to the scanner service special properties of a terminal line other than the one on which the command is typed. The device argument may be used to modify the characteristics of any terminal line on the system.

TOPS-10 OPERATOR-PRIVILEGED COMMANDS

Command Formats

1. SET TTY TTYn: word
2. SET TTY TTYn: NO word
3. TTY TTYn: word
4. TTY TTYn: NO word

TTYn: = the terminal whose line is to be controlled. This argument is optional; if omitted, the command is identical to the SET TTY command described in the DECsystem-10 Operating System Commands manual.

NO = the argument that determines whether a bit is to be set or cleared. This argument is optional.

word = the various words that represent bits that may be modified. The words are as follows:

SET TTY TTYn: ALTMODE	Converts the ALTMODE codes of 175 and 176 to the standard ASCII ESCape character 033 (initial state if, at monitor generation, the TTYALT symbol is assigned a non-zero value).
SET TTY TTYn: NO ALTMODE	Restores the individual identity of the codes 175 and 176.
SET TTY TTYn: BLANKS	Restores multiple carriage return/line feeds and form feeds (initial state).
SET TTY TTYn: NO BLANKS	Suppresses blank lines (consecutive carriage return/line feeds after the first) and outputs form feeds and vertical tabs as two carriage return line feeds. This is useful for a video display terminal in order to increase the amount of output which fits on the screen.
SET TTY TTYn: CRLF	Restores the free carriage return (initial state).
SET TTY TTYn: NO CRLF	The free carriage return normally output at the end of a line exceeding the carriage width is suppressed.
SET TTY TTYn: DEBREAK	Notifies the system that the terminal has a feature that allows the computer to lock the keyboard. (Model 2741 Terminals)
SET TTY TTYn: NO DEBREAK	Turns off the SET TTY TTYn: DEBREAK feature.
SET TTY TTYn: ECHO	Restores the normal echoing of each character typed in. Most terminals on the DECsystem-10 are used in full-duplex mode which means that the terminal does not print each character as the user types it. Instead the monitor echoes

TOPS-10 OPERATOR-PRIVILEGED COMMANDS

each character typed so that the character will be printed. If a terminal prints ^C when it is typed but does not print any of the noncontrol characters typed by the user, then this command can be used to establish normal echoing. (Initial setting determined by each installation when constructing its monitor at system generation time.)

SET TTY TTYn: NO ECHO

This command suppresses monitor echoing of input characters. Local copy terminals i.e., terminals which automatically print each character as the user types it) do not require normal monitor echoing, because the echoing would cause each character typed to be printed twice. This command is used to stop double printing of each character on a local copy terminal. (Initial setting determined by each installation when constructing its monitor at system generation time.)

SET TTY TTYn: ELEMENT n

Changes the typing element number for Model 2741-type terminals. The element numbers available are:

Element	Meaning
987	APL correspondence
029	Standard correspondence
087	Call 360 BASIC
963	Extended Binary
938	BCD
988	APL (EBCD)

Note that the SET TTY TTYn: ELEMENT n command also sets the terminal as if the SET TTY TTYn: NO LC command were used.

SET TTY TTYn: NO ELEMENT

Turns off the SET TTY TTYn: ELEMENT n command.

SET TTY TTYn: FILL n

The filler class n is assigned to this terminal. Many different types of terminals are supported by the DECsystem-10 operating system. Some terminals require one or more filler characters to be sent following certain control characters such as line feed (LF) and horizontal tab (HT). Table A-1 illustrates the number of fillers sent for each character and filler class. The filler characters are CR (215 octal for even parity) for carriage return characters (CR, 215 octal) and DEL (RUBOUT, 377 octal) for all other characters. No fillers are supplied for image mode output. (Initial setting determined by each installation when constructing its monitor at system generation time.)

TOPS-10 OPERATOR-PRIVILEGED COMMANDS

SET TTY TTYn: NO FILL Equivalent to TTY FILL 0. (Initial setting is determined by each installation when constructing its monitor at system generation time.) Fillers for output and echoing are determined from Table A-1.

Table A-1
Number of Fillers by Character and Filler Class

Character Name	Octal	Number of Fillers for Filler class			
		0	1	2	3
BS	010	0	2	6	6
HT	011	0	1 or 2	0	1 or 2+
LF	012	0	2	6	6
VT	013	0	2	6	6
FF .	014	0	12	21	21
CR on					
output	015	0	1	3	3++
free CR+++		0	2	4	4
CRLF on					
output	015-012	0	3	9	9++++
XON	021	1	1	1	1
TAPE	022	1	1	1	1
XOFF	023	1	1	1	1
NTAP	024	1	1	1	1
+1 if 0-3 spaces to tab stop; 2 if 4-7 spaces to tab stop. ++ Output only; no fillers on input. +++ Refer to the SET TTY TTYn CRLF command. ++++ Sum of the fillers output for a CR and LF.					

SET TTY TTYn: FORM The terminal has hardware FORM (page) and VT (vertical tab) characters (initial setting determined by each installation when constructing its monitor at system generation time).

SET TTY TTYn: NO FORM The monitor sends eight line feeds for a FORM and four line feeds for a VT (initial setting determined by each installation when constructing its monitor at system generation time).

SET TTY TTYn: GAG Messages transmitted by the SEND command cannot be received at this terminal unless the terminal is at command level (initial state).

SET TTY TTYn: NO GAG Messages transmitted by the SEND command can be received at this terminal even though it is not at command level.

SET TTY TTYn: IGNORE Allows the operator to disable a line. The line will be ignored by the PDP-11.

SET TTY TTYn: NO IGNORE Turns off the SET TTY TTYn: IGNORE command.

TOPS-10 OPERATOR-PRIVILEGED COMMANDS

SET TTY TTyn: LC	The translation of lowercase character input to uppercase is suppressed.
SET TTY TTyn: NO LC	This command causes the monitor to translate lowercase characters to uppercase as they are received. Frequently, it is convenient to have a terminal with both uppercase and lowercase to simulate the behavior of one with uppercase only. This command causes the monitor to perform this simulation. The echo sent back by the monitor always matches the case of the characters after translation. By looking at the printout, the user can determine what translation was performed by the monitor (initial state).
SET TTY TTyn: PAGE	This command gives the user the ability to temporarily suspend system typeout without losing it. The XOFF key (^S) suspends the typeout, and the XON key (^Q) restores it. The XOFF and XON keys are not echoed and are not sent to the user's program. This command is useful for video display terminals where the user may want to read a page of text before it disappears from the screen. Note that this preempts the use of ^S and ^Q for reading paper tape. (See SET TTY TAPE.)
SET TTY TTyn: NO PAGE	The typeout control ability of the XOFF and XON keys is disabled. The current interpretation of these keys depends on the last SET TTY TAPE command (initial state).
SET TTY TTyn: RTCOMP	Disables the ^R (reprint current line after making corrections) and ^T (print job status without interrupting execution) features allowing ^R and ^T to be passed to the user program.
SET TTY TTyn: NO RTCOMP	Turns on the ^R and the ^T features.
SET TTY TTyn: SLAVE	The specified terminal becomes slaved; that is, no commands may be typed on the terminal. The terminal may then be ASSIGNED by another user. The operator can slave any terminal including his own. (Initial setting is determined by each installation when constructing its monitor at system generation time.)
SET TTY TTyn: NO SLAVE	The specified terminal becomes unslaved; that is, commands may now be typed on the console. If a user terminal is slaved, the operator may unslave it (since the operator is the only person that can give the command with a device argument). If OPR is slaved, the operator can LOGIN under [1,2] on another terminal to unslave OPR.

TOPS-10 OPERATOR-PRIVILEGED COMMANDS

	(Initial setting is determined by each installation when constructing its monitor at system generation time.)
SET TTY TTYn: SPEED n	Sets the speed at which the system sends and receives characters to n baud (DC76 only). (The user adjusts the speed at the terminal by setting a switch.)
SET TTY TTYn: SPEED m,n	Sets the terminal transmitting speed to n baud and the receiving speed to m baud (DC76 only). (This is especially useful for display terminals as they allow transmitting and receiving at different speeds.)
SET TTY TTYn: TAB	The terminal has hardware TAB stops set every eight columns. (Initial setting is determined by each installation when constructing its monitor at system generation time.)
SET TTY TTYn: NO TAB	The monitor simulates TAB output from programs by sending the necessary number of SPACE characters.
SET TTY TTYn: TAPE	The XON key (^Q) causes the terminal to read paper tape. The XOFF key (^S) causes the terminal to stop reading paper tape. Rubouts will be ignored and free CRLF's will not be sent.
SET TTY TTYn: NO TAPE	The XON key (^Q) and the XOFF key (^S) have no special paper tape function. They may, however, have a PAGE function. (See SET TTY TTYn:PAGE command.)
SET TTY TTYn: TIDY	Specifies (for Model 2741 terminals) that every character occupy one print space. The terminal normally types out characters so that they appear the same as what the user typed in. For example, [prints out as ^<. In TIDY mode, [prints out as <.
SET TTY TTYn: NO TIDY	Turns off the SET TTY TTYn: TIDY command.
SET TTY TTYn: UC	Equivalent to SET TTY TTYn: NO LC.
SET TTY TTYn: NO UC	Equivalent to SET TTY TTYn: LC.
SET TTY TTYn: WIDTH n	The carriage width (the point at which a free carriage return is inserted) is set to n. The range of n is 17 (two TAB stops) to 200 decimal. The initial state is 72.

Restrictions

The SET TTY command:

Requires LOGIN if unslaving OPR.

TOPS-10 OPERATOR-PRIVILEGED COMMANDS

Associated Messages

None

SET VMMA

Function

The SET VMMA command specifies the maximum amount of virtual memory available to all users of the system.

Command Format

SET VMMA nP
 nK

where nP and nK are alternative arguments.
1K equals 1024 words, and 1P equals a page of 512 words.
If K and P are omitted, K is assumed.

After the execution of this command, the new value of VMMA will be one of the following:

1. The total amount of allocated swapping space, if n is greater than the total amount allocated.
2. The amount currently in use by all virtual memory users, if n is less than the total amount in use.
3. The value of n, if neither of the above is true.

Associated Messages

None

XCHNGE

Function

The XCHNGE command enables a system with a bad disk unit to continue without undue user disruption by logically exchanging disk drives and then having the operator physically exchange disk packs.

Command Format

XCHNGE diskunit1 diskunit2/comment

diskunitn=the disk unit the operator wants to exchange with another disk unit, e.g., DPA0 DPA2.

comment=text, whose first two characters go into ERROR.SYS along with the command as configuration changes
(PM=preventive maintenance, CM=corrective maintenance).

TOPS-10 OPERATOR-PRIVILEGED COMMANDS

CAUTION

Injudicious use of this command can destroy information on disk packs because the monitor assumes that the physical disk packs have been switched by the operator. For that reason, both disk drives should be powered down before issuing the command. One precaution the monitor does take whenever this command is issued is to automatically place both disk drives OFFLINE. It is up to the operator to then switch the disk packs before placing the units back ONLINE.

Restrictions

diskunit1 and diskunit2 must be on the same controller.

Associated Messages

?XCHNGE diskunitn?

The disk unit specified does not exist.

?XCHNGE diskunit1 diskunit2?

The two disk drives are on different controllers.

APPENDIX B

RSX-20F CONSOLE PROCESSOR COMMANDS

The PARSER, an RSX-20F program, accepts commands entered on the console terminal (CTY) and performs various console functions on the KL and front-end processors.

B.1 ENTERING AND EXITING THE PARSER

You can enter the PARSER from the TOPS-10 monitor mode by typing a CTRL/\ (control-backslash) on the CTY.

If you are currently communicating with a TOPS-10 job, enter a CTRL/\ to get to the PARSER.

If you are currently communicating with another RSX-20F task such as KLINIK, enter a CTRL/Z to exit the task and then a CTRL/\ to enter the PARSER.

To exit the PARSER, type the QUIT command or use the CTRL/Z character.

B.2 PARSER CONSOLE MODES

There are four PARSER console modes that you can set via the SET CONSOLE command. (See Section B.4.) These modes determine which subset of the PARSER commands you will be allowed to use.

OPERATOR console mode allows only those commands that will not crash the TOPS-10 monitor.

PROGRAMMER mode allows all PARSER commands except diagnostic functions.

MAINTENANCE mode allows the full set of PARSER commands.

USER mode has the effect of exiting the PARSER (equivalent to a QUIT command).

B.3 PARSER PROMPTS

While you are in the PARSER task, you will be prompted by one of the following character sequences:

PAR> to indicate that the KL processor is running a program.

RSX-20F CONSOLE PROCESSOR COMMANDS

PAR% to indicate that the KL processor is not running a program,
 but the KL clock is on and EXAMINES are allowed.

PAR# to indicate that the KL clock is off.

NOTE

If you should ever see the PAR# prompt displayed during timesharing, reload the system.

If the PARSER encounters an error during its initialization, an error message will be typed preceding the prompt.

B.4 PARSER COMMANDS FOR THE OPERATOR

When you give commands to the console processor command language, you must use one of the following forms.

1. A single command followed by <CR>
2. Several commands separated by semicolons and ending with <CR>

If you need to continue a command line beyond a physical line on the terminal, end the line with a hyphen followed by <CR>. The PARSER will prompt you for the next line by typing a hyphen. The command line is complete when there is a <CR> that is not preceded by a hyphen. (The maximum number of characters in a command line is 280.)

When you press the final <CR> of a command line, the PARSER scans the command line buffer for invalid characters. If the PARSER finds an invalid character (LINE FEED or ESCAPE) in the command line, no part of the command line is executed. If the characters are all valid, the PARSER begins executing the individual commands. If the PARSER encounters an invalid command, it stops and that command as well as any others left in the command line are not executed.

There is a help facility built into the PARSER. If you type ? after the prompt, the PARSER will output a list of available commands. This list will conform to the console mode in effect at the time. In the example below, the console was in OPERATOR mode.

----- Example -----

PAR>?

PARSER COMMANDS ARE:

ABORT
CLEAR
EXAMINE
JUMP
MCR
REPEAT
RUN
SET
SHUTDOWN
QUIT
WHAT

PAR>

RSX-20F CONSOLE PROCESSOR COMMANDS

You can also type ? in place of the argument of a command to find out what arguments are available for that command. Again, the list will reflect only those arguments available to you based on the console mode in effect at the time.

----- Example -----

PAR>EXAMINE ?

EXAMINE COMMANDS ARE:

PC
DTE-20
ELEVEN
KL
TEN
DECREMENT
INCREMENT
NEXT
PREVIOUS
THIS

PAR>

The following commands comprise the subset of the console processor command language allowable in OPERATOR console mode. They are the commands relevant to your functions as an operator.

In the subsequent command formats, the following notational conventions are used:

1. Any single argument must be specified.
2. A multiple choice list, enclosed in special brackets `[[]]`, means that an entry is optional. A default entry will be specified.
3. A multiple choice list, enclosed in braces `{ }`, means that one of the entries must be specified.

ABORT

This command stops the KL processor by forcing it into a HALT loop. Use this command only when the SHUT DOWN command fails to come up with the DECSYSTEM-10 NOT RUNNING or the **HALTED** message.

CLEAR CONSOLE

This command forces the console into OPERATOR mode.

CLEAR INCREMENT

This command sets the increment counter to zero. (Refer to the EXAMINE INCREMENT/DECREMENT commands.)

CLEAR MEMORY

This command forces all subsequent EXAMINE commands to reference PDP-10 memory.

RSX-20F CONSOLE PROCESSOR COMMANDS

CLEAR NOT

This combination is the equivalent of a SET command and requires an argument. (Refer to the SET commands.)

CLEAR REPEAT

This command sets the command line repeat counter to zero.

EXAMINE PC

This command types out the contents of the KL program counter in octal.

EXAMINE KL

This command types out the contents of the KL program counter (PC) and virtual memory address register (VMA), the state of the priority interrupt system (PI), and the PC flags; all in octal.

EXAMINE $\left[\begin{array}{c} \text{TEN} \\ \text{ELEVEN} \end{array} \right] \left\{ \begin{array}{c} \text{addr} \\ \text{INCREMENT} \\ \text{DECREMENT} \\ \text{PREVIOUS} \\ \text{THIS} \\ \text{NEXT} \end{array} \right\}$

This command types out the contents of the specified core location in octal.

TEN specifies that this command will reference physical addresses in KL10 memory.

ELEVEN specifies that this command will reference word (even) addresses in PDP-11 memory.

If neither TEN nor ELEVEN are specified, the memory to be referenced will be determined by the most recent SET MEMORY command. If no SET MEMORY command has been issued, PDP-10 memory will be referenced.

There are six arguments to determine the specific memory location; one of them must be entered.

addr is the location address in octal. This must be an even number when referencing the PDP-11.

INCREMENT means add the contents of the increment counter to the address of the location last examined to get the current location to be examined. When referencing PDP-11 memory, the contents of the increment counter are ignored and both EXAMINE INCREMENT and EXAMINE NEXT give the same result.

DECREMENT means subtract the contents of the increment counter from the address of the location last examined to get the current location to be examined. When referencing PDP-11 memory, the contents of the increment counter are ignored and both EXAMINE DECREMENT and EXAMINE PREVIOUS give the same result.

RSX-20F CONSOLE PROCESSOR COMMANDS

PREVIOUS means examine the word previous to the one last examined.

THIS means re-examine the word that was last examined.

NEXT means examine the word after the last one examined.

JUMP addr

This command starts the KL processor at the location specified and then exits the PARSER. addr is an octal, positive, non-zero address with a maximum value of 17777777 octal.

MCR filename

This command loads and starts a PDP-11 program. filename must be the file name of a PDP-11 task.

QUIT

This command exits the console processor command language.

REPEAT nnn; [[command1; command2; ...]]

This command will execute the rest of the command line a specified number of times.

nnn is the repeat factor and is a decimal, positive, non-zero number.

The command line can contain as many commands as will fit within the 280 character buffer limitation. You can nest REPEATs within the command line. Also, if a SET REPEAT command is in effect, the two repeat factors are multiplied to arrive at the actual number of times commands are repeated.

For example, the command:

```
REPEAT 10; EXAMINE PC
```

will examine the PC ten times.

The command:

```
REPEAT 3; EXAMINE PC; REPEAT 2; EXAMINE NEXT
```

will execute the following sequence:

```
EXAMINE PC
EXAMINE NEXT
EXAMINE NEXT
EXAMINE PC
EXAMINE NEXT
EXAMINE NEXT
EXAMINE PC
EXAMINE NEXT
EXAMINE NEXT
```

RSX-20F CONSOLE PROCESSOR COMMANDS

If SET REPEAT 4 had been previously entered, the above sequence would be repeated four times.

If no commands are specified, the effect is that of a null command.

RUN filename

This command performs the same function as the MCR command.

SET CONSOLE

MAINTENANCE
OPERATOR
PROGRAMMER
USER

You must use this command to change the subset of PARSER commands that you can subsequently execute. Entry and exit to the PARSER does not change console mode.

MAINTENANCE allows all PARSER commands.

OPERATOR allows only those commands that can not crash the TOPS-10 monitor.

PROGRAMMER allows all commands except diagnostic functions. PROGRAMMER is the default argument.

USER allows no PARSER commands; its effect is to leave the command language as in the QUIT command.

SET INCREMENT nnn

This command sets the increment counter to the value specified.

nnn is any octal number (36-bit maximum) and must be specified. Note that if the value is zero, EXAMINE INCREMENT and EXAMINE DECREMENT both function as EXAMINE THIS. Also, if the value is negative, the functions of EXAMINE INCREMENT and EXAMINE DECREMENT are reversed. The initial value of the increment counter is zero.

SET MEMORY {ELEVEN}
{TEN}

This command forces all subsequent EXAMINE commands to reference the memory specified.

ELEVEN means reference PDP-11 memory.

TEN means reference PDP-10 memory.

The command has no default. When the system is first initialized, PDP-10 memory references are in effect.

SET NOT

This combination is the equivalent of a CLEAR command and requires an argument. (Refer to the CLEAR commands.)

RSX-20F CONSOLE PROCESSOR COMMANDS

SET REPEAT nnn

This command sets the repeat counter to the value specified. All subsequent command lines will be repeated that number of times.

nnn is a decimal, positive, non-zero number and must be specified.

(See the REPEAT command for an example of its use.) The initial value of the repeat counter is one.

SHUTDOWN

This command stops the KL timesharing system without causing a reload. This command is used to stop the TOPS-10 monitor after the TOPS-10 OPSER command, KSYS, has logged off all user and system jobs. To reenter the PARSER after issuing this command, type a control backslash.

WHAT $\left\{ \begin{array}{l} \text{CONSOLE} \\ \text{INCREMENT} \\ \text{MEMORY} \\ \text{REPEAT} \\ \text{VERSION} \end{array} \right\}$

This command types the requested information on the console terminal.

CONSOLE types out the current console mode.

INCREMENT types out the contents of the increment counter in octal.

MEMORY types out the type of memory that is currently being referenced by EXAMINE commands.

REPEAT types out the contents of the repeat counter in decimal.

VERSION types out the version number of the PARSER and the RSX-20F monitor.

B.5 PARSER ERROR MESSAGES

The following list contains all the error messages that may be output by the PARSER while in any of the three console modes. The format of each message is:

PAR -- [command name] code - message

The command name is the name of the command that caused the error. However, this command name can be PARSER if you typed a string that caused an error in the command parser, rather than in a specific command. For example, if you type an invalid command:

PAR>KLEAR CONSOLE

you will receive the error message:

PAR -- [PARSER] NSK - NO SUCH KEYWORD "KLEAR"

RSX-20F CONSOLE PROCESSOR COMMANDS

However, if you type in an invalid argument:

```
PAR>CLEAR KONSOLE
```

you will receive the error message:

```
PAR -- [CLEAR] NSK - NO SUCH KEYWORD "KONSOLE"
```

A short explanation of each error message is included along with operator action, where necessary.

CODE	MESSAGE AND EXPLANATION
------	-------------------------

AMB	AMBIGUOUS KEYWORD "xxx"
-----	-------------------------

where xxx is the ambiguous keyword. The PARSER found more than one keyword matching the abbreviation you typed.

NOTE

The PARSER matches your abbreviation against the complete set of commands and arguments regardless of the subset allowable by the console mode you are in.

APE	KL APR ERROR
-----	--------------

The PARSER encountered a CPU error (nonexistent memory, parity error, etc.). Call your field service representative.

BAE	BURST ARGUMENT ERROR
-----	----------------------

There was an internal programming failure. Notify your software contact.

CAE	KL CRAM ADDRESS ERROR
-----	-----------------------

There was an internal programming failure. Notify your software contact.

CBO	COMMAND BUFFER OVERFLOW
-----	-------------------------

You typed a command line that was over 280 characters in length. Reenter as two or more command lines.

CES	CLOCK ERROR STOP - <code> ERROR STOP
-----	--------------------------------------

<code> is either CRAM, DRAM, FM, or FS-STOP. This happens when the CPU encounters a fatal internal hardware error. Note the code received, and call your field service representative. Also try to reload the system via DISK or DECTape, or via the switch register, provided that you reload the microcode.

CFH	CAN'T FIND KL HALT LOOP
-----	-------------------------

The PARSER tried to halt the KL and failed to do so. Notify your field service representative.

CNR	COMMAND IS NOT REPEATABLE
-----	---------------------------

You tried to repeat a command that cannot be repeated. However, the command has been executed once.

RSX-20F CONSOLE PROCESSOR COMMANDS

CPE CAN'T PHASE EBOX

There was an internal hardware error. Call your field service representative.

CSK CAN'T START KL

The PARSEr tried to start the KL and failed to do so. Call your field service representative.

DCK DIVIDE CHECK

There was an internal programming error. Call your software contact.

DMF DEPOSIT KL MEMORY FAILED

There was an internal programming failure. RSX-20F didn't accept a deposit directive. Call your software contact.

DNP DTE-20 IS NOT PRIVILEGED

This is a fatal error. The DTE-20 mode switch is in the wrong position. Call your field service representative and/or your software contact.

DSF DTE-20 STATUS FAILED

A read or write to one of the DTE status registers failed. Call your software contact.

DTC DTE-20 CONFUSED - RUN AND HALT LOOP

This is a fatal error. The run and halt loop flags were set simultaneously, an impossible situation. Call your field service representative.

ECT EBOX CLOCK TIMEOUT

While the PARSEr was doing an execute function, the KL failed to reenter the halt loop within the allotted time. Call your software contact.

EMF EXAMINE KL MEMORY FAILED

There was an internal programming failure. RSX-20F didn't accept an examine directive. Call your software contact.

EOC END OF COMMAND REQUIRED

The command was ended with a ? and no further arguments exist. Retype the command and press carriage return.

EPE EBUS PARITY ERROR

This a fatal error. The PARSEr encountered an EBUS parity error. Call your field service representative.

ESD EBOX STOPPED - DEPOSIT

The PARSEr executed a deposit directive and found that the KL clock was stopped.

RSX-20F CONSOLE PROCESSOR COMMANDS

ESE EBOX STOPPED - EXAMINE

The PARSER executed an examine directive and found that the KL clock was stopped.

FRF FUNCTION READ n FAILED

n is the code of the function that failed. This is a fatal error. Call your field service representative and your software contact. If the system crashes, try to reload it.

FWF FUNCTION WRITE n FAILED

n is the code of the function that failed. This is a fatal error. Call your field service representative and your software contact. If the system crashes, try to reload it.

FXF FUNCTION XCT n FAILED

n is the code of the function that failed. This is a fatal error. Call your field service representative and your software contact. If the system crashes, try to reload it.

IFC ILLEGAL FUNCTION CODE

There was an internal programming error. The PARSER received an illegal function read, write or execute code. Call your software contact.

ILC ILLEGAL CHARACTER "c"

c is the printer's translation of an illegal character. Non-printing characters are preceded by an up-arrow (^) and converted to their printing equivalent for output. (Refer to Sections B.1 through B.4.)

ILS ILLEGAL SEPARATOR CHARACTER "s"

s is the printer's translation of an illegal separator character. Control characters are preceded by an up-arrow (^). (Refer to Sections B.1 through B.4.) Note that each tab is converted to a single space.

IOC ILLEGAL KL OPCODE

The PARSER tried to execute a KL instruction with an illegal format. Call your software contact.

IRC ILLEGAL REPEAT COUNT

You typed a zero or negative argument to either the REPEAT or SET REPEAT command.

ITN ILLEGAL TASK NAME

The RUN or MCR command was entered with no argument.

KCN KL CLOCK IS OFF

You tried to execute a command that requires the clock to be on and the clock was off.

RSX-20F CONSOLE PROCESSOR COMMANDS

KLA	KL ADDRESS ERROR
	You specified an EXAMINE or DEPOSIT command with a KL address that was out of range (over 22 bits), negative, or not in octal format.
KLR	ILLEGAL COMMAND WHILE KL RUNNING
	You tried to execute a command that is illegal while the KL is running.
KNC	KL IS NOT CONTINUABLE
	You tried to resume processing with the CONTINUE command but the KL was not in a continuable state. i.e., after a RESET command.
MRA	MISSING REQUIRED ARGUMENT
	You didn't specify all the arguments necessary for the command, or there is an error in syntax.
NER	NUMERIC EXPRESSION REQUIRED
	The command expects a numeric expression as an argument and something else was entered. This error message can also occur if the command syntax is in error.
NOR	INPUT NUMBER OUT OF RANGE
	You specified a number that was out of range or in the wrong number base. (Refer to Section B.4.)
NSK	NO SUCH KEYWORD "xxx"
	xxx is an incorrect keyword. (Refer to Section B.4.)
NST	NO SUCH TASK
	You specified a nonexistent task as an argument to the MCR or RUN command.
OAI	ODD ADDRESS ILLEGAL
	You tried to examine an odd-numbered PDP-11 address. This is illegal.
OFC	ODD FUNCTION CODE
	There was an internal programming error. Call your software contact.
RPM	RIGHT PARENTHESIS MISSING
	You specified a numeric expression with a right parenthesis missing.
SCF	SET CLOCK FAILED
	The command parser cannot validate the parameter it has just set. This is a hardware error; call your field service representative.

RSX-20F CONSOLE PROCESSOR COMMANDS

SPF SET PARITY FAILED

The command parser cannot validate the parameter it has just set. This is a hardware error; call your field service representative.

SZI START AT ZERO ILLEGAL

You tried to start the KL at location zero; this is illegal.

TAA TASK IS ALREADY ACTIVE

You issued a RUN or MCR command while another task was active.

UEC UNMATCHED ERROR CODE - "xxx"

xxx is an unmatched error code. There was an internal programming error. Notify your software contact.

UNL KL MICROCODE NOT LOADED

Reload the system via DISK or DEctape, or via the switch register, provided that you reload the microcode.

WRM COMMAND NOT AVAILABLE IN THIS CONSOLE MODE

The command just issued is not available. Type a ? to the PAR> prompt to obtain a listing of the commands available in the current console mode.

XTO KL EXECUTE TIMED OUT

While performing an execute function, the KL failed to reenter the halt loop within the allotted time. Call your field service representative.

APPENDIX C

KL INITIALIZATION PROGRAM (KLINIT)

The KL initialization program, KLINIT, has both a default and a dialog method of operation.

When you load the system via the DISK or DECTAPE load switch, a default path is taken through the KL initialization operator dialog. It does the following without asking you any questions:

1. Loads the KL processor microcode from the appropriate microcode file on the front-end load device. (UA.MCB for the KL10A, UB.MCB for the KL10B.)
2. Configures and enables cache memory according to the KLINIT configuration file, KL.CFG. If this file is not present on the front-end load device, all cache is enabled.
3. Configures and interleaves KL memory according to the KLINIT configuration file, KL.CFG. If this file is not present on the front-end load device, all available memory is configured with the highest possible interleaving.
4. If the KL.CFG file did not exist, KLINIT creates a file by that name and stores it on the front-end load device. The file contains the cache and memory configuration in effect at the time.
5. Loads and starts the default KL bootstrap program located on the disk or DECTape file, BOOT.EXB.
6. The bootstrap program then loads and starts the default TOPS-10 monitor located in the file SYS:SYSTEM.EXE.

If you do not want all these defaults, you must enter the dialog mode of KLINIT. Then, you can do the following:

1. Verify the microcode.
2. Configure cache memory as you want it.
3. Configure KL memory as you want it.
4. Load and start any bootstrap program.
5. Specify switches to the bootstrap program.
6. Load and start any monitor from disk or magnetic tape.

To enter the dialog, load the system using the switch register load switch with at least switches 0, 1, and 2 set. (See Section 3.2.3.)

KL INITIALIZATION PROGRAM (KLINIT)

During the dialog, the following rules apply:

1. All answers to a question must end with <CR> (carriage return).
2. RUBOUT or DELETE deletes a character.
3. Typing only a <CR> selects the default answer to the question. The default answer is the first answer listed.
4. CTRL/Z terminates the operator dialog and exits to the RSX-20F monitor without rewriting the KL.CFG file.
5. CTRL/U deletes the current line.
6. NO to the ENTER DIALOG question skips the rest of the dialog and assumes all the default answers.
7. BACK to any question returns you to the previous question unless stated otherwise.
8. RESTART to the EXIT question goes back to the first question in the dialog.
9. Pressing ESCape at any point in a reply, before <CR>, restarts the dialog. Note that ESCape does not echo on your terminal.
10. An unacceptable answer results in an error message and causes the question to be repeated.
11. The dialog continues with the next question listed unless stated otherwise.
12. The minimum truncation for any answer other than filename is the first two characters.

C.1 KLINIT OPERATOR DIALOG

The following are all the possible questions from KLINIT and the acceptable answers. However, depending on the answers that you give, some questions may be omitted in an actual dialog. When a question from KLINIT appears on your console terminal, it is preceded by KLI --. KLINIT then prompts for an answer with KLI>. Therefore, during the actual dialog the format is:

```
KLI -- question [ans1, ans2,...]?  
KLI>
```

The brackets [], used in the following explanation of KLINIT questions, are for clarity only; do not type brackets around your answers.

ENTER DIALOG [NO,YES,EXIT,BOOT]?

Ans. 1 [NO] means assume the default answers for all the remaining questions. This is the last chance to bypass the dialog and take the default path.

Ans. 2 [YES] continues the dialog and asks the next question.

KL INITIALIZATION PROGRAM (KLINIT)

Ans. 3 [EXIT] means discontinue the dialog and return to the RSX-20F monitor.

Ans. 4 [BOOT] skips the rest of the dialog, enables cache memory as directed by KL.CFG, and immediately loads and starts the standard KL bootstrap program found in BOOT.EXB. No defaults are taken when this option is selected.

RELOAD MICROCODE [YES,VERIFY,NO]?

Ans. 1 [YES] loads the KL10 microcode from the appropriate microcode file on the bootstrapping device into the KL10 processor.

Ans. 2 [VERIFY] verifies that the microcode in the KL10 processor matches the microcode in the appropriate microcode file on the bootstrapping device.

Ans. 3 [NO] neither loads nor verifies the microcode.

RECONFIGURE CACHE [FILE,ALL,YES,NO]?

Ans. 1 [FILE] means configure cache memory as specified in the configuration file, KL.CFG. If this file does not exist, all cache memory is enabled. The dialog continues with the CONFIGURE KL MEMORY question.

Ans. 2 [ALL] means enable all cache memory. The dialog continues with the CONFIGURE KL MEMORY question.

Ans. 3 [YES] means configure cache memory under dialog control.

Ans. 4 [NO] means do not reconfigure cache memory; the existing configuration is left unchanged. The dialog continues with the CONFIGURE KL MEMORY question.

ENABLE WHICH CACHES [ALL,NONE,0-3]

Ans. 1 [ALL] means enable all cache memory.

Ans. 2 [NONE] means disable all cache memory.

Ans. 3 [n,n,...] (where n is 0, 1, 2, or 3) means enable only the caches specified.

CONFIGURE KL MEMORY [FILE,ALL,YES,NO]?

NOTE

A reply of BACK to this question will return you to the RECONFIGURE CACHE question.

Ans. 1 [FILE] means configure KL memory as specified in the configuration file, KL.CFG. If this file does not exist, [ALL] is assumed. The dialog continues with the LOAD KL BOOTSTRAP question. If the configuration in the KL.CFG file is not consistent with the setting of the memory interleaving switches, an error message is issued and the dialog restarts with the first question.

KL INITIALIZATION PROGRAM (KLINIT)

- Ans. 2 [ALL] configures KL memory with as much memory as possible and with the highest interleaving possible consistent with the setting of the memory interleaving switches on the memory units. The dialog continues with the LOAD KL BOOTSTRAP question.
- Ans. 3 [YES] means configure memory under dialog control. Before the next question is asked, KLINIT examines memory and prints out a physical memory map.
- Ans. 4 [NO] means do not configure memory at all. The previous memory configuration remains. The dialog continues with the LOAD KL BOOTSTRAP question.

INTERLEAVE KL MEMORY [OPTIMAL,4,2,1]?

- Ans. 1 [OPTIMAL] means configure KL memory with the highest interleaving possible consistent with the setting of the memory interleaving switches on the memory units.
- Ans. 2 [4] means 4-way interleaving. A test is performed on the available memory to see if 4-way interleaving is possible. If it is, the available memory is configured with 4-way interleaving. Otherwise, an error message is issued and the question is repeated.
- Ans. 3 [2] means 2-way interleaving. A test is made on the available memory to see if 2-way interleaving is possible. If it is, the available memory is configured with 2-way interleaving. If not, an error message is issued and the question is repeated.
- Ans. 4 [1] means configure all available memory with 1-way or no interleaving.

LOAD KL BOOTSTRAP [YES,NO,filename]?

NOTE

A reply of BACK to this question will return you to the CONFIGURE KL MEMORY question.

- Ans. 1 [YES] means the default KL bootstrap program is to be loaded into KL memory from the bootstrap device. The file from which the program is read is BOOT.EXB (.EXB denotes an RSX-Binary version of a .SAV file). The KLINIT dialog is over; you will now be communicating with the bootstrap program.

When the bootstrap program has been loaded and started, you will be prompted with BTS>. Enter the filename of the TOPS-10 monitor to be loaded or press <CR> to get the default monitor, SYS:SYSTEM.EXE.

- Ans. 2 [NO] means no KL bootstrap program is to be loaded into KL memory. (See the next question.)
- Ans. 3 [filename] is the name of the alternative bootstrap file (xxxxxxxxx.yyy) to be loaded. xxxxxxxxx is the 1- to 9-alphanumeric character filename of the KL bootstrap and

KL INITIALIZATION PROGRAM (KLINIT)

yyy is the 1- to 3-alphanumeric character filename extension. If only a filename is specified, a default extension of .EXB is assumed. This file must be in RSX-Binary format. The KLINIT dialog is over; you will now be communicating with the bootstrap program.

NOTE

The following question is asked only when NO is answered to the previous question.

EXIT [YES,RESTART]?

Ans. 1 [YES] exits to the RSX-20F monitor.

Ans. 2 [RESTART] means restart the dialog.

Before KLINIT loads the bootstrap program, it checks to see if the cache memory was configured. If the reply to the RECONFIGURE CACHE question was NO, KLINIT configures the cache according to the KL.CFG file. If the KL.CFG file does not exist, KLINIT configures all cache. In the latter situation, a KL.CFG file is created and stored on the front-end load device.

When the bootstrap program has been loaded and started you will be prompted with BTS> (or BTM> if the magnetic tape bootstrap was specified). Enter the filename of the TOPS-10 monitor to be loaded or press <CR> to get the default monitor from SYS:SYSTEM.EXE.

C.2 KLINIT MESSAGES

There are four classes of messages issued by the KL initialization program: informational, warning, dialog error, and system error messages.

C.2.1 Informational Messages

Informational messages indicate the normal completion of a KLINIT function. The message text is preceded by "KLI --".

The informational messages are:

KLI -- ALL CACHES ENABLED

All four of the KL processor caches have been enabled.

KLI -- BOOTSTRAP LOADED AND STARTED

A KL bootstrap program has been loaded into KL memory and started. Any messages that follow will be a function of the particular bootstrap program being used.

KLI -- CACHES DISABLED

All cache memory has been disabled.

KL INITIALIZATION PROGRAM (KLINIT)

KLI -- CACHES n,n... ENABLED

The specified caches have been enabled.

KLI -- CONFIGURATION FILE ALTERED

The KL.CFG file was updated with a new cache and/or memory configuration. This message is issued whenever you set up a non-default configuration or if the KL.CFG file did not previously exist.

KLI -- KL RESTARTED

The KL processor has been restarted following a power failure or a hardware or software crash.

KLI -- MICROCODE VERSION xxx LOADED

The KL microcode, version xxx, has been loaded into the KL system from the appropriate microcode file on the front-end bootstrap device.

KLI -- MICROCODE VERSION xxx VERIFIED

The KL microcode, version xxx, currently residing in the system has been compared correctly with the code in the appropriate microcode file on the front-end bootstrap device.

C.2.2 Warning Messages

Warning messages inform the operator of some unusual condition. The KLINIT dialog continues. These messages are preceded by "KLI -- %".

The warning messages are:

KLI -- % DMA20 PARITY ERROR AT loc=data BEING REWRITTEN

KLINIT encountered a parity error while trying to read data at location "loc". "loc" is expressed as eight octal digits and the data, as KLINIT read it, is represented as two groups of six octal digits.

SYSTEM ACTION:

KLINIT will rewrite the data at the specified location and attempt to read it again.

KLI -- % NO FILE - ALL CACHE BEING CONFIGURED

The default to the RECONFIGURE CACHE question was taken and KLINIT could not find the KL.CFG file in the directory.

SYSTEM ACTION:

KLINIT enables all caches.

KLI -- % NO FILE - ALL MEMORY BEING CONFIGURED

The default to the CONFIGURE KL MEMORY question was taken and KLINIT could not find the KL.CFG file in the directory.

KL INITIALIZATION PROGRAM (KLINIT)

SYSTEM ACTION:

KLINIT configures all available memory and sets the interleaving at the highest level consistent with the setting of the interleave switches on the memory units.

C.2.3 Dialog Error Messages

Dialog error messages indicate that your answer to the current KLINIT question is unacceptable. KLINIT repeats the question and waits for an appropriate reply. The message text is preceded by "KLI --".

Unless noted otherwise, the system action for all dialog error messages is to repeat the question and re-prompt.

The dialog error messages are:

KLI -- COMMAND SYNTAX ERROR

Your reply was not one of the possible answers as specified in the question.

OPERATOR ACTION:

Reply with one of the acceptable answers, correctly spelled, or use carriage return to take the default answer.

KLI -- INTERLEAVE TOO HIGH FOR SWITCH SETTINGS

Your reply to the INTERLEAVE KL MEMORY expressed a value higher than possible according to the physical setting of the interleave switches on the memory units.

OPERATOR ACTION:

Reply with a lower value of interleaving.

--or--

Change the switches on the memory unit. Type BACK to get to the previous question. Retype the two previous answers.

KLI -- NO EXTERNAL MEMORY AVAILABLE

Physical switches on the memory units are set such that no memory is available (all modules deselected).

OPERATOR ACTION:

Change the switch settings on the memory boxes and repeat the reply.

C.2.4 System Error Messages

System error messages indicate conditions in which KLINIT cannot continue. These conditions can be brought about by software, hardware, or environmental failures. Sometimes a retry will be successful; other times you may require the assistance of your field service engineer or software support specialist. For any system

KL INITIALIZATION PROGRAM (KLINIT)

error, it is very important to save all console log data and memory dump listings; this material is of prime importance when troubleshooting a problem. System error messages are preceded by "KLI -- ?".

Unless noted otherwise, the system action for all system error messages is to restart the KLINIT dialog and re-prompt. Whenever a file is specified in a message text, it will be identified in the following format.

'dev:filename.ext;ver'

The system messages are:

KLI -- ? BOOTSTRAP LOAD FAILED

A software or hardware error occurred while loading the KL bootstrap program. (See accompanying messages for additional information.)

OPERATOR ACTION:

Reload the bootstrap program by replying:

KLI>BOOT

If the trouble persists, call your field service engineer.

KLI -- ? C-RAM DIFFERS AT xxxxxx

During the microcode verify operation, the contents of octal location xxxxxx in the KL10 Control RAM did not match the corresponding code in the appropriate microcode file.

OPERATOR ACTION:

Reload the KL microcode and reverify it via the KLINIT dialog. If the trouble persists, call your field service engineer.

KLI -- ? CACHE ENABLE FAILED

This is probably a hardware error while trying to configure the cache memory. (See accompanying messages for additional information.)

OPERATOR ACTION:

Retry the operation; if the trouble persists, call your field service engineer. You can also temporarily reconfigure with no cache memory.

KLI -- ? CANNOT DETERMINE DMA20 INTERLEAVE

When configuring memory, KLINIT cannot determine the setting of the interleave switches on the memory boxes. This can be a hardware or a software failure.

OPERATOR ACTION:

Check the setting of the interleave switches and retry the operation. If the trouble persists, call your field service engineer.

KL INITIALIZATION PROGRAM (KLINIT)

KLI -- ? CANNOT DETERMINE PHYSICAL CONFIGURATION

When configuring memory, KLINIT cannot determine the physical configuration of the memory units. (See accompanying messages for additional information.)

OPERATOR ACTION:

Reload the system; if trouble persists, call your field service engineer.

KLI -- ? CANNOT FIND [5,5] DIRECTORY

KLINIT cannot locate the PDP-11 system file directory; a software error may have overlaid it.

SYSTEM ACTION:

The system exits the KLINIT program.

OPERATOR ACTION:

Reload the system; if the trouble persists, call your software support specialist.

KLI -- ? CANNOT FIND HALT LOOP

KLINIT tried to start the microcode and it failed to run properly.

OPERATOR ACTION:

Reload the microcode; if the trouble persists, call your field service engineer.

KLI -- ? CANNOT GET DEVICES

KLINIT cannot open a system device for communications. This is probably a software error in RSX-20F.

SYSTEM ACTION:

The system exits the KLINIT program.

OPERATOR ACTION:

Reload the system; if the trouble persists, call your software support specialist.

KLI -- ? CANNOT RUN KLINIT WHILE KL IS IN PRIMARY PROTOCOL

An attempt was made to run the KLINIT program while the KL processor was running. This condition can arise only if KLINIT is loaded via the PARSER command language instruction:

PAR>RUN KLINIT

SYSTEM ACTION:

The system exits to the TOPS-10 monitor.

OPERATOR ACTION:

If the intent was to rerun KLINIT, follow the appropriate procedures to shut down TOPS-10; then reload the system and

KL INITIALIZATION PROGRAM (KLINIT)

enter the KLINIT program. If TOPS-10 will not shut down properly, set console to PROGRAMMER mode and reload KLINIT.

KL I -- ? CANNOT START KL

A hardware or software failure has occurred while trying to restart from a power failure or system crash during memory determination. (See accompanying messages for additional information.)

OPERATOR ACTION:

Reload the microcode and retry the operation. If the trouble persists, call your field service engineer.

KL I -- ? CLOCK ERROR STOP DURING KL RESTART

The KL processor clock has stopped while KLINIT was monitoring a restart operation. (See accompanying messages for additional information.)

OPERATOR ACTION:

Retry loading the KL bootstrap and monitor. If the trouble persists, call your software support specialist.

KL I -- ? CONFIGURATION FILE NOT CHANGED

The KL.CFG configuration file cannot be updated because the old file cannot be read, the new file cannot be written, or some other error has occurred. (See accompanying messages for additional information.)

OPERATOR ACTION:

If the old file cannot be read, you can exit KLINIT, rename the old file so no attempt will be made to read it, and then retry the operation. A sequence of commands to do this would be:

```
KL I -- ENTER DIALOG [NO,YES,EXIT,BOOT]?
KL I>^Z
PAR%M PIP
PIP>XKL.CFG/RE=KL.CFG
PIP>^Z
PAR%M KL I
KL I -- ENTER DIALOG [NO,YES,EXIT,BOOT]?
```

If the trouble persists, call your software support specialist.

KL I -- ? D-RAM DIFFERS AT xxxxxx

During the microcode verify operation, the contents of octal location xxxxxx in the KL10 Dispatch RAM did not match the corresponding code in the appropriate microcode file.

OPERATOR ACTION:

Reload the KL microcode and reverify it via the KLINIT dialog. If the trouble persists, call your field service engineer.

KL I -- ? DEPOSIT FAILED

KLINIT could not store information into KL10 memory.

KL INITIALIZATION PROGRAM (KLINIT)

OPERATOR ACTION:

Reload the system and retry the operation. If the trouble persists, call your field service engineer.

KLI -- ? DEVICE 'device' FULL

KLINIT cannot find room on the specified front-end load device for an updated copy of the configuration file KL.CFG.

OPERATOR ACTION:

Exit from KLINIT and use a front-end system program such as PIP to delete files and make room for the updated KL.CFG file. Then reenter KLINIT and retry the operation.

KLI -- ? DF EXECUTE FAILED

A diagnostic function execute failed while KLINIT was initializing the KL10 processor.

OPERATOR ACTION:

Reload the system and retry the operation. If the trouble persists, call your field service engineer.

KLI -- ? DF READ FAILED

A diagnostic function read failed while KLINIT was initializing the KL10 processor.

OPERATOR ACTION:

Reload the system and retry the operation. If the trouble persists, call your field service engineer.

KLI -- ? DF WRITE FAILED

A diagnostic function write failed while KLINIT was initializing the KL10 processor.

OPERATOR ACTION:

Reload the system and retry the operation. If the trouble persists, call your field service engineer.

KLI -- ? DIRECTIVE ERROR -n ON FILE 'filename'

A system error occurred while KLINIT was trying to access the file 'filename.' The "n" is an octal error code for use by software support.

OPERATOR ACTION:

Reload the system and retry the operation. If the trouble persists, call your software support specialist.

KLI -- ? DMA20 HAS TOO MANY GAPS

Over 16 non-contiguous blocks of memory were found on a DMA20.

OPERATOR ACTION:

Call your field service engineer.

KL INITIALIZATION PROGRAM (KLINIT)

KLI -- ? DMA20 PARITY ERROR RETRY FAILED AT loc=data

A previous parity error, signalled by a warning message, was retried and failed again. "loc" is the failing address (eight octal digits) and "data" (12 octal digits) represents the contents of that address.

OPERATOR ACTION:

If possible, deselect that portion of memory that is failing and reload the system. Otherwise, call your field service engineer.

KLI -- ? EXAMINE FAILED

KLINIT could not examine contents of KL10 memory.

OPERATOR ACTION:

Reload the system and retry the operation. If the trouble persists, call your field service engineer.

KLI -- ? FILE 'filename' NOT FOUND

KLINIT cannot find BOOT.EXB, the appropriate microcode file, or the alternate KL bootstrap file in the PDP-11 file directory [5,5] on SY0:.

OPERATOR ACTION:

Ensure that the file being requested resides on the front-end load device and retry the operation.

KLI -- ? FILE DOES NOT FIT PHYSICAL CONFIGURATION

The configuration file, KL.CFG, specifies an interleaving mode higher than that allowed by the interleaving switches on the memory units.

OPERATOR ACTION:

Reconfigure memory via the KLINIT dialog or change the switch settings; then retry the operation.

KLI -- ? I/O ERROR -n ON FILE 'filename'

An I/O error occurred while KLINIT was trying to access the file 'filename.' The "n" is an RSX-11 octal error code for use by software support.

OPERATOR ACTION:

Reload the system and retry the operation. If the trouble persists, call your software support specialist.

KLI -- ? ILLEGAL DMA20 IN CONFIGURATION

A hardware error occurred while KLINIT was trying to determine the physical memory configuration.

OPERATOR ACTION:

Reload the system and retry the operation. If the trouble persists, call your field service engineer.

KL INITIALIZATION PROGRAM (KLINIT)

KLI -- ? INPUT RECORD LENGTH ERROR

Error occurred while KLINIT was trying to read KL.CFG, the appropriate microcode file, or the KL bootstrap file. This error could be caused by software or hardware failure.

OPERATOR ACTION:

If possible, try other copies of the files. If the trouble persists, call your software support specialist. If the file in question is KL.CFG, you can get around the error by renaming or deleting the file. KLINIT will then write a new KL.CFG file by default.

KLI -- ? INSUFFICIENT MEMORY FOR BOOTSTRAP

Insufficient or no memory is available where KLINIT is trying to load the bootstrap program. (See any accompanying messages for additional information.) Memory selection switches on the memory units may be set in error.

OPERATOR ACTION:

Check memory selection switches on the memory units and retry the operation. If trouble persists, call your field service engineer.

KLI -- ? KL HALT DURING RESTART

The KL processor stopped on a HALT instruction while KLINIT was monitoring a restart operation.

OPERATOR ACTION:

Re-boot and load the KL monitor; if the trouble persists, call your software support specialist.

KLI -- ? MASTER RESET FAILED

A master reset function to the KL failed. This is a hardware error.

OPERATOR ACTION:

Reload the system and retry the operation. If the trouble persists, call your field service engineer.

KLI -- ? MEMORY CONFIGURATION FAILED

A hardware or software error occurred while KLINIT was configuring memory. (See accompanying messages for additional information.)

OPERATOR ACTION:

Reload the system and retry the operation. If the trouble persists, call your field service engineer.

KL INITIALIZATION PROGRAM (KLINIT)

KLI -- ? MICROCODE LOAD FAILED

A hardware or software error occurred while KLINIT was loading the KL microcode. (See accompanying messages for additional information.)

OPERATOR ACTION:

Retry loading the microcode; if the trouble persists, call your field service engineer.

KLI -- ? MICROCODE VERIFY FAILED

The verification of the KL microcode discovered errors which are itemized in preceding error messages.

OPERATOR ACTION:

Reload the microcode and verify it. If the trouble persists, call your field service engineer.

KLI -- ? NO MEMORY AT LOCATION ZERO

When KLINIT was configuring memory it could not locate any memory unit with address switches set at zero.

OPERATOR ACTION:

Check the memory units and ensure that one has its address switches set at zero; then retry loading.

KLI -- ? OUTPUT RECORD LENGTH ERROR

Error occurred while KLINIT was trying to write an updated configuration file, KL.CFG.

OPERATOR ACTION:

Retry the operation and if the problem persists, call your software support specialist.

KLI -- ? POWER-FAIL RESTART FAILED

KLINIT could not restart the KL processor during a power-fail recovery. (See accompanying message for additional information.)

OPERATOR ACTION:

Reload the system using one of the load switch procedures. If the system still will not come up, call your field service engineer.

KLI -- ? READ ERROR

A hardware or software error occurred while KLINIT was accessing KL.CFG, the appropriate microcode file, or the KL bootstrap file. (See any accompanying messages for additional information.)

OPERATOR ACTION:

Retry the operation; if the trouble persists, call your software support specialist. If the file in question is

KL INITIALIZATION PROGRAM (KLINIT)

KL.CFG, you can get around the read operation by renaming or deleting the file. A new file will be written by default.

KLI -- ? SYSTEM ERROR DURING KL RESTART

This is a KLINIT software error during a KL restart operation.

OPERATOR ACTION:

Reload the system using one of the load switch procedures. If you still have problems bringing up the system, call your software support specialist.

KLI -- ? TIMEOUT DURING KL RESTART

While KLINIT was monitoring a KL restart, the 30-second allowable time limit was exceeded.

OPERATOR ACTION:

Try reloading the KL processor via the dialog default. If you still cannot bring the system up, call your software support specialist.

KLI -- ? WRITE ERROR

A hardware or software error occurred while KLINIT was writing an updated copy of the configuration file, KL.CFG. (See any accompanying messages for additional information.)

OPERATOR ACTION:

Retry the write operation; if the trouble persists, call your software support specialist.

C.3 REPORTS RELATING TO KL MEMORY CONFIGURATION

If you answer YES to the CONFIGURE KL MEMORY question in the KLINIT dialog, you will get a physical memory configuration map printed as follows:

PHYSICAL MEMORY:

CONTROLLER ADDRESS	TYPE	STORAGE
4	DMA20	512K

EXTERNAL MEMORY RESPONSE

ADDRESS	SIZE
00000000	512K

This map represents the physical memory allocation, where:

CONTROLLER ADDRESS = memory controller number; this is always 4
for a DMA20

TYPE = memory controller type

STORAGE = total storage available

Under EXTERNAL MEMORY RESPONSE, the total storage is broken down by contiguous blocks and their beginning addresses, where:

ADDRESS = beginning address of memory block

SIZE = size of the memory block

KL INITIALIZATION PROGRAM (KLINIT)

Whenever KLINIT configures KL memory, either by default or through the dialog, it prints a logical memory configuration map on your console terminal. If you answer NO to the CONFIGURE KL MEMORY question, the map is not printed. The format of the map is as follows:

LOGICAL MEMORY CONFIGURATION:
CONTROLLER
ADDRESS SIZE RQ0 RQ1 RQ2 RQ3 CONTYPE INT
000000 512K 04 FOR ALL DMA20 4

This map tells you how KL memory has been configured, where:

ADDRESS = KL memory address

SIZE = KL memory size in K words

RQ0 = SBUS request bit 0

RQ1 = SBUS request bit 1

RQ2 = SBUS request bit 2

RQ3 = SBUS request bit 3

CONTYPE = memory controller type; DMA20 for external memory

INT = interleave mode

The following memory maps provide an example of non-contiguous memory.

Assume 2 MG10-A memory cabinets, each with 4 banks of 32K memory, for a total of 256K. The lower bound address in each cabinet is 00000000 and 00400000, respectively. If you deselect the high order memory bank in each cabinet and answer YES to the CONFIGURE KL MEMORY question, the physical memory map will look like the following:

PHYSICAL MEMORY:
CONTROLLER ADDRESS TYPE STORAGE
4 DMA20 192.K
EXTERNAL MEMORY RESPONSE
ADDRESS SIZE
00000000 96.K
00400000 96.K

The logical memory configuration, after an appropriate reply to the INTERLEAVE KL MEMORY question, will look like the following:

LOGICAL MEMORY CONFIGURATION:
CONTROLLER
ADDRESS SIZE RQ0 RQ1 RQ2 RQ3 CONTYPE INT
00000000 96.K 04 FOR ALL DMA20 1
00300000 32.K NONEXISTANT MEMORY
00400000 96.K 04 FOR ALL DMA20 1

C.4 KLINIT DIALOG EXAMPLES

1. This example shows the output at your console terminal when you load the system using the DISK load switch. KLINIT automatically takes the default values without asking you any questions. However, it does tell you that the RAMs (random access memories) have been loaded with the microcode. It prints out the logical memory map and then loads and starts the KL bootstrap program.

KL INITIALIZATION PROGRAM (KLINIT)

RSX-20F V006A 0:02 21-JUN-76

```
[SY0:REDIRECTED TO DB0:]
[DB0:MOUNTED]
KLI -- VERSION V002Q RUNNING
KLI -- MICROCODE VERSION 131 LOADED
KLI -- ALL CACHES ENABLED
LOGICAL MEMORY CONFIGURATION:
      CONTROLLER
ADDRESS    SIZE  RQ0  RQ1  RQ2  RQ3  CONTYPE  INT
00000000   256K  04   FOR  ALL           DMA20   4
KLI -- BOOTSTRAP LOADED AND STARTED
BOOTS V22(103)
```

BTS>

2. This example shows the output at your console terminal when you load the system via the switch register with switches 0, 1 and 2 set. The KLINIT dialog is entered only to load and start the KL bootstrap. This allows you to leave the microcode and memory configuration as they were.

RSX-20F V006A 0:02 21-JUN-76

```
[SY0:REDIRECTED TO DB0:]
[DB0:MOUNTED]
KLI -- VERSION V002Q RUNNING
KLI -- ENTER DIALOG [NO,YES,EXIT,BOOT]?
KLI>BOOT
KLI -- ALL CACHES ENABLED
KLI -- BOOTSTRAP LOADED AND STARTED
BOOTS V22(103)
```

BTS>

3. This example shows the KLINIT dialog being used to reconfigure KL memory. KLINIT prints out both the physical memory configuration and the logical memory map indicating that the 256K of memory is now 2-way interleaved.

RSX-20F V006A 0:02 21-JUN-76

```
[SY0:REDIRECTED TO DB0:]
[DB0:MOUNTED]
KLI -- VERSION V002Q RUNNING
KLI -- ENTER DIALOG [NO,YES,EXIT,BOOT]?
KLI>YES
KLI -- RELOAD MICROCODE [YES,VERIFY,NO]?
KLI>YES
KLI -- MICROCODE VERSION 131 LOADED
KLI -- RECONFIGURE CACHE [FILE,ALL,YES,NO]?
KLI>ALL
KLI -- ALL CACHES ENABLED
KLI -- CONFIGURE KL MEMORY [FILE,ALL,YES,NO]?
KLI>YES
PHYSICAL MEMORY:
CONTROLLER ADDRESS  TYPE      STORAGE
      4             DMA20     256K
EXTERNAL MEMORY RESPONSE
ADDRESS  SIZE
00000000 256K
KLI -- INTERLEAVE KL MEMORY [OPTIONAL,4,2,1]?
KLI>2
```

KL INITIALIZATION PROGRAM (KLINIT)

LOGICAL MEMORY CONFIGURATION:

```
                CONTROLLER
ADDRESS    SIZE  RQ0  RQ1  RQ2  RQ3  CONTYPE  INT
00000000   256K  04   FOR  ALL           DMA20   2
KLI -- LOAD KL BOOTSTRAP [YES,NO,FILENAME]?
KLI>YES
KLI -- CONFIGURATION FILE ALTERED
KLI -- BOOTSTRAP LOADED AND STARTED
BOOTS V22(103)
```

BTS>

4. This example shows the console terminal output when the system is loaded via the DECTape load switch. KLINIT did not find a KL.CFG file on the DECTape (%NO FILE messages); therefore, it configured all cache and all available memory. Note that KLINIT informs you whenever it writes a new KL.CFG file; it does so whenever you answer ALL or YES to the CONFIGURE KL MEMORY question or if there was no previous KL.CFG file.

RSX-20F V006A 0:05 21-JUN-76

```
[SY0:REDIRECTED TO DT0:]
[DT0:MOUNTED]
KLI -- VERSION V002Q RUNNING
KLI -- MICROCODE VERSION 131 LOADED
KLI -- % NO FILE - ALL CACHE BEING CONFIGURED
KLI -- ALL CACHES ENABLED
KLI -- % NO FILE - ALL MEMORY BEING CONFIGURED
LOGICAL MEMORY CONFIGURATION:
                CONTROLLER
ADDRESS    SIZE  RQ0  RQ1  RQ2  RQ3  CONTYPE  INT
00000000   256K  04   FOR  ALL           DMA20   4
KLI -- CONFIGURATION FILE ALTERED
KLI -- BOOTSTRAP LOADED AND STARTED
BOOTS V22(103)
```

BTS>

5. This example shows that the specified bootstrap file XXBOOT.EXB was not found. Therefore, after the fatal error messages, the KLINIT dialog restarts.

RSX-20F V006A 0:02 21-JUN-76

```
[SY0:REDIRECTED TO DB0:]
[DB0:MOUNTED]
KLI -- VERSION V002Q RUNNING
KLI -- ENTER DIALOG [NO,YES,EXIT,BOOT]?
KLI>YES
KLI -- RELOAD MICROCODE [YES,VERIFY,NO]?
KLI>NO
KLI -- RECONFIGURE CACHE [FILE,ALL,YES,NO]?
KLI>NO
KLI -- CONFIGURE KL MEMORY [FILE,ALL,YES,NO]?
KLI>NO
KLI -- LOAD KL BOOTSTRAP [YES,NO,FILENAME]?
KLI>XXBOOT
KLI -- ? FILE 'DB0:XXBOOT.EXB;0' NOT FOUND
KLI -- ? BOOTSTRAP LOAD FAILED
KLI -- ENTER DIALOG [NO,YES,EXIT,BOOT]?
KLI>
```

KL INITIALIZATION PROGRAM (KLINIT)

6. This example shows that a <CR> defaults to the first reply listed. (NO to ENTER DIALOG) In this case the default signals KLINIT to bypass any further dialog and assume the default answers to all the remaining questions. However, it found only 96K of core available and when it tried to load the bootstrap program, it failed. After the two error messages, KLINIT restarts the dialog.

RSX-20F V006A 0:02 21-JUN-76

```
[SY0:REDIRECTED TO DB0:]
[DB0:MOUNTED]
KLI -- VERSION V002Q RUNNING
KLI -- ENTER DIALOG [NO,YES,EXIT,BOOT]?
KLI> (carriage return was pressed here)
KLI -- MICROCODE VERSION 131 LOADED
KLI -- ALL CACHES ENABLED
LOGICAL MEMORY CONFIGURATION:
      CONTROLLER
ADDRESS    SIZE  RQ0  RQ1  RQ2  RQ3  CONTYPE  INT
00000000   96K   04   FOR  ALL          DMA20   4
KLI -- ? INSUFFICIENT MEMORY FOR BOOTSTRAP
KLI -- ? BOOTSTRAP LOAD FAILED
KLI -- ENTER DIALOG [NO,YES,EXIT,BOOT]?
KLI>
```

7. This example shows the dialog first being used to load and verify the microcode. Then it shows the cache memory being configured. The TOPS-10 monitor is to be loaded from a magnetic tape so the program BOOTM must be loaded in place of the default program, BOOTS. BOOTM is contained in the file MTBOOT.EXB. KLINIT accepts the file name, appends the default file type of .EXB, and loads and starts the magnetic tape bootstrap program.

RSX-20F V006A 0:02 21-JUN-76

```
[SY0:REDIRECTED TO DB0:]
[DB0:MOUNTED]
KLI -- VERSION V002Q RUNNING
KLI -- ENTER DIALOG [NO,YES,EXIT,BOOT]?
KLI>YES
KLI -- RELOAD MICROCODE [YES,VERIFY,NO]?
KLI>YES
KLI -- MICROCODE VERSION 131 LOADED
KLI -- RECONFIGURE CACHE [FILE,ALL,YES,NO]?
KLI>BACK
KLI -- RELOAD MICROCODE [YES,VERIFY,NO]?
KLI>VERIFY
KLI -- MICROCODE VERSION 131 VERIFIED
KLI -- RECONFIGURE CACHE [FILE,ALL,YES,NO]?
KLI>YES
KLI -- ENABLE WHICH CACHES [ALL,NONE,0-3]?
KLI>0,1,3
KLI -- CACHES 0,1,3 ENABLED
KLI -- CONFIGURE KL MEMORY [FILE,ALL,YES,NO]?
KLI>YES
PHYSICAL MEMORY:
CONTROLLER ADDRESS  TYPE  STORAGE
      4             DMA20  256K
EXTERNAL MEMORY RESPONSE
ADDRESS    SIZE
00000000   256K
KLI -- INTERLEAVE KL MEMORY [OPTIMAL,4,2,1]?
KLI>2
```

KL INITIALIZATION PROGRAM (KLINIT)

LOGICAL MEMORY CONFIGURATION:

```

                                CONTROLLER
ADDRESS    SIZE  RQ0  RQ1  RQ2  RQ3  CONTYPE  INT
00000000   256K  04   FOR  ALL          DMA20   2
KLI -- LOAD KL BOOTSTRAP [YES,NO,FILENAME]?
KLI>MTBOOT
KLI -- CONFIGURATION FILE ALTERED
KLI -- BOOTSTRAP LOADED AND STARTED
BOOTM V1(10)
BTM>

```

8. This example shows that an error occurred in verifying the existing microcode. Because the dialog is restarted after a fatal error, the solution to try is to answer YES to the RELOAD MICROCODE question the next time.

RSX-20F V006A 0:02 21-JUN-76

```

[SY0:REDIRECTED TO DB0:]
[DB0:MOUNTED]
KLI -- VERSION V002Q RUNNING
KLI -- ENTER DIALOG [NO,YES,EXIT,BOOT]?
KLI>YES
KLI -- RELOAD MICROCODE [YES,VERIFY,NO]?
KLI>VERIFY
KLI -- ? C-RAM DIFFERS AT 1
KLI -- ? MICROCODE VERIFY FAILED
KLI -- ENTER DIALOG [NO,YES,EXIT,BOOT]?
KLI>

```

9. This example shows ESCape being used during the reply to the RELOAD MICROCODE question in order to restart the dialog. It also shows unacceptable answers causing questions to be repeated. Finally, a CTRL/Z causes the dialog to exit to the console processor command language (the PARSER).

RSX-20F V006A 0:02 21-JUN-76

```

[SY0:REDIRECTED TO DB0:]
[DB0:MOUNTED]
KLI -- VERSION V002Q RUNNING
KLI -- ENTER DIALOG [NO,YES,EXIT,BOOT]?
KLI>YES
KLI -- RELOAD MICROCODE [YES,VERIFY,NO]?
KLI>VER<ESC>
KLI -- ENTER DIALOG [NO,YES,EXIT,BOOT]?
KLI>YNOT
KLI -- COMMAND SYNTAX ERROR
KLI -- ENTER DIALOG [NO,YES,EXIT,BOOT]?
KLI>YES
KLI -- RELOAD MICROCODE [YES,VERIFY,NO]?
KLI>NO
KLI -- RECONFIGURE CACHE [FILE,ALL,YES,NO]?
KLI>MAYBE
KLI -- COMMAND SYNTAX ERROR
KLI -- RECONFIGURE CACHE [FILE,ALL,YES,NO]?
KLI>NO
KLI -- CONFIGURE KL MEMORY [FILE,ALL,YES,NO]?
KLI>NO
KLI -- LOAD KL BOOTSTRAP [YES,NO,FILENAME]?
KLI>^Z
PAR%

```

INDEX

- Abnormal termination, 1-2
- ABORT command in PARSE, B-3
- Account control,
 - weekly, 5-15
- ACCT.SYS file, 5-4
- Addressing memories, 6-10
- /AFTER switch in QUEUE, 4-15
- AID interpreter, 2-6
- /ALCNT switch in LPFORM.INI, 4-37
- ALGOL compiler, 2-6
- ALIGN output spooler command, 4-29
- /ALIGN switch in LPFORM.INI, 4-38
- /ALSLP switch in LPFORM.INI, 4-38
- Altering of memory configuration, 6-8
- Analysis of file structures, 5-14
- APL interpreter, 2-6
- Assigning a DECTape drive, 7-25
- Asynchronous front ends, 2-2
- ATTACH operator command,
 - device, A-1
 - disk unit, A-2
 - job, A-2
- :AUTO command in OPSE, 4-5
- /AUTO switch in DTELDR, 6-15, 6-16
- Auto-load with DTELDR, 6-15, 6-16
- Auto-reload,
 - DECSYSTEM-10, 6-1
 - default monitor, 6-4
 - KLINIT, 6-4
- AUXACC.SYS file, 5-4

- BACKSPACE output spooler command, 4-29
- BACKUP program,
 - daily, 5-10
 - multi-reel save, 5-18
 - tape overflow, 5-17
 - weekly, 5-16
- Backup system disk, 5-10
- BACKUP utility, 2-7
- /BANNER switch in LPFORM.INI, 4-38
- BASIC interpreter, 2-6
- BAT blocks, 5-19
- /BATCH switch in SPACE, 5-11

- BATCON, 4-19
 - commands to, 4-19
 - commands to control subjobs, 4-22
 - commands to set parameters, 4-21
 - display parameters of, 4-22
 - LOG file, 4-25
 - monitoring, 4-26
 - operation of, 4-24
 - operator messages, 4-26
 - starting up, 4-19
- BATCON commands,
 - CORE, 4-21
 - CURRENT, 4-22
 - EXAMINE, 4-22
 - EXIT, 4-21
 - GO, 4-22
 - HELP, 4-22
 - KILL, 4-22
 - MCORE, 4-21
 - MJOB, 4-21
 - MTIME, 4-21
 - NEXT, 4-21
 - NONJOB, 4-22
 - OPERATOR, 4-23
 - REQUEUE, 4-23
 - RESET, 4-21
 - ROUTE, 4-24
 - START, 4-21
 - STOP, 4-24
 - TELL, 4-24
 - TIME, 4-21
 - WHAT, 4-22
- BOOT11 program, 3-12, 6-17
- BOOTDX program, 3-14, 6-18

- Capabilities KLINIT, C-1
- Card punches, 2-3
- Card reader, 2-3
 - cleaning of, 7-25
 - CR10-E, 7-24
 - CR10-F, 7-23
 - loading, 7-22
 - operation of, 7-22
 - unloading, 7-22
- Card reader input to SPRINT, 4-47
- CARDS output spooler command, 4-29
- Carriage control tape,
 - channels for, 7-16
 - LP10, 7-12
 - standard LP10, 7-13

INDEX (CONT.)

- Central processor, 2-1
 - error recovery, 6-1
- /CHAIN switch in LPFORM.INI, 4-38
- Changing forms,
 - LP07 printer, 7-4
 - LP10 printer, 7-8
- Changing MLIMIT in LPTSPL, 4-35
- CHANNEL command in MAKVFO, 7-12
- Channels for carriage control tape, 7-16
- CHKPNT output spooler command, 4-29
- Cleaning,
 - card reader, 7-25
 - DECTape drive, 7-26
 - devices, 7-1
 - disk drives, 7-36
 - line printer, 7-22
 - magnetic tape, 7-34
- CLEAR CONSOLE command in PARSEr, B-3
- CLEAR INCREMENT command in PARSEr, B-3
- CLEAR MEMORY command in PARSEr, B-3
- CLEAR NOT command in PARSEr, B-4
- CLEAR REPEAT command in PARSEr, B-4
- :CLOSE command in OPSER, 4-6
- COBOL compiler, 2-6
- Command switches in QUEUE, 4-15
- Commands,
 - operator-privileged, A-1
 - output spooler, 4-29
 - SPRINT, 4-48
 - to BATCON, 4-19
 - to control BATCON subjobs, 4-22
 - to set BATCON parameters, 4-21
- Communication,
 - one-way, 4-54
 - two-way, 4-53
 - user to operator, 4-53
- Communications front ends, 2-2
- Compiler,
 - ALGOL, 2-6
 - COBOL, 2-6
 - FORTRAN, 2-6
- Configuration maps in KLINIT, C-15
- Configure memory,
 - KLINIT, C-3
 - operator command, A-6
- Console log documentation, 1-3
- Console mode,
 - MAINTENANCE, B-1
 - OPERATOR, B-1
 - PROGRAMMER, B-1
 - USER, B-1
- Console terminal, 2-1
- Console/Diagnostic processor, 2-1
 - :CONTINUE command in OPSER, 4-6
- Conventions for KLINIT, C-2
- Convert PC value to symbolic, 6-6
- /COPIES switch in QUEUE, 4-15
- CORE command in BATCON, 4-21
- /CORE switch in QUEUE, 4-15
- CP10-D, 2-3
- CR10-E card reader, 2-3, 7-24
- CR10-F card reader, 2-3, 7-23
- Crash data,
 - preserve, 6-6
- Crash restart procedure, 6-5
- CRT display terminals, 2-5
- :CTRL/C command in OPSER, 4-7
- CURRENT command in,
 - BATCON, 4-22
 - output spooler, 4-29
- :CURRENT OPSER command, 4-6
- Damage assessment of file structures, 5-14
- DAS98 remote station, 2-5
- DAVFU on LP07, 7-11
- :DAYTIME command in OPSER, 4-6
- DC72 remote station, 2-5
- DCRPE program, 5-19
- DDRPC program, 5-22
- DDRPI program, 5-23
- DDT utility, 2-7
- /DEADLINE switch in QUEUE, 4-15
- DEBUG STOPCD, 6-2
- DECsystem-10,
 - auto-reload of, 6-1
- DECTape drive,
 - assigning a, 7-25
 - cleaning of, 7-26
 - operation of, 7-25
- DECTape load switch, 3-1
- DECTapes, 2-3
 - dismounting, 7-26
 - mounting, 7-26
- Default monitor auto-reload, 6-4
- Defaults in KLINIT, C-1
- :DEFINE command in OPSER, 4-6
- Definitions,
 - switch register bit, 3-6
- DETACH device operator command, A-3
- DETACH disk unit operator command, A-3
- :DEVICE command in OPSER, 4-7
- Devices,
 - cleaning of, 7-1
 - operation of, 7-1

INDEX (CONT.)

- Dialog error messages in
 KLINIT, C-7
- DIGITAL documentation, 1-2
- Direct access vertical format
 unit, 7-11
- Disk drives,
 cleaning of, 7-36
 dual-ported, 2-3
 operation of, 7-35
- Disk formatting,
 DCRPE program, 5-19
 DDRPC program, 5-22
 DDRPI program, 5-23
- Disk input to SPRINT, 4-47
- DISK load switch, 3-1
- Disk packs,
 formatting, 5-19
 mapping, 5-19
- Disk systems,
 RHP02, 2-3
 RHP03, 2-3
 RHP04, 2-3
 RHP06, 2-3
 RHS04, 2-3
- Display parameters of BATCON,
 4-22
- /DISPOSE switch in QUEUE, 4-15
- DL10,
 dumping a, 6-17
 reloading a, 6-17
- DL10-type front-end error
 recovery, 6-17
- DN80 remote station, 2-5
- DN81 remote station, 2-5
- DN82 remote station, 2-5
- Documentation,
 console log, 1-3
 DIGITAL, 1-2
 installation-supplied, 1-2
 operator, 1-2
- Documentation on tape, 1-2
- Documenting complaints, 5-12
- /DRUM switch in LPFORM.INI,
 4-38
- DSKLST program, 5-14
- DSKRAT program, 5-14
- DTE20, 2-1
- DTE20-type front-end error
 recovery, 6-15, 6-16
- DTELDLDR,
 action switches, 3-13
 /AUTO switch in, 6-15, 6-16
 auto-load with, 6-15, 6-16
 /IMAGE switch, 3-13
 modification switches, 3-13
 /NODUMP switch, 3-13, 6-15, 6-16
 /NOLOG switch, 3-13
 OPR.ATO file for, 6-15, 6-16
 /RELOAD switch, 3-13
 reload with, 6-15, 6-16
- Dual processor,
 loading, 3-11
- Dual-ported disk drives, 2-3
- Dumping a DL10, 6-17
- Dumping a DX10, 6-18
- Duties,
 hardware-related, 1-1
 operator, 1-1
 software-related, 1-1
- DX10 data channel,
 dumping, 6-18
 error recovery, 6-18
 reloading, 6-18
- EMERGENCY POWER OFF switch,
 3-1
- ENABLE load switch, 3-1
- Entering complaints, 5-12
- Entry format LPFORM.INI file,
 4-37
- Environmental error conditions,
 6-14
- Error messages,
 OPSER, 4-9
 PARSER, B-7
 QUASAR, 4-17
- :ERROR command in OPSER, 4-7
- Error recovery, 6-1
 central processor, 6-1
 DL10-type front end, 6-17
 DTE20-type front end, 6-15,
 6-16
 DX10 data channel, 6-18
 front-end processor, 6-15
 HUNG condition, 6-5
 parity error, 6-7
 power failure, 6-14
 user core failure, 6-7
- EXAMINE command in BATCON, 4-22
- EXAMINE ELEVEN command in PARSER,
 B-4
- EXAMINE KL command in PARSER,
 B-4
- EXAMINE PC command in PARSER,
 B-4
- EXAMINE TEN command in PARSER,
 B-4
- Examples of KLINIT dialog,
 C-16
- Examples of LPFORM.INI file,
 4-40
- :EXIT OPSER command, 4-7
- EXIT command in,
 BATCON, 4-21
 MAKVFU, 7-12
 output spooler, 4-29
 SPRINT, 4-48

INDEX (CONT.)

FACT files,
 MONEY listings, 5-13
 saving, 5-12
 /FACT switch in SPACE, 5-2
 FACT.SYS file, 5-3
 Faulty memory module, 6-8
 FILCOM program, 5-7
 FILDDT program, 6-6
 File compare program, 5-7
 File structures,
 analysis of, 5-14
 damage assessment of, 5-14
 Fillers for output and echoing,
 A-11
 Formatting,
 disk packs, 5-19
 partial disk pack, 5-23
 Forms change,
 line printer, 7-4
 LP07, 7-4
 LP10, 7-8
 Forms control,
 LP07, 7-11
 LP10, 7-12
 FORMS command in MAKVFU, 7-12
 FORMS output spooler command,
 4-30
 FORTRAN compiler, 2-6
 FORWARD output spooler command,
 4-30
 :FREE command in OPSER, 4-7
 FREEZE output spooler command,
 4-30
 Front-end loader,
 BOOT11, 3-12
 DTELDL, 3-12
 Front-end processor error
 recovery, 6-15
 Front ends,
 asynchronous, 2-2
 synchronous, 2-2

 GALAXY batch system, 4-16
 GALAXY components,
 BATCON, 4-19
 LPTSPL, 4-34
 output spoolers, 4-28
 QUASAR, 4-16
 QUEUE, 4-14
 SPRINT, 4-46
 SPROUT, 4-43
 GALAXY utility, 2-7
 GO command in,
 BATCON, 4-22
 output spooler, 4-30
 SPRINT, 4-48
 GRIPE program, 5-12
 GRIPE.CTL file, 5-12

 HALT STOPCD, 6-3
 Hardcopy terminals, 2-4
 Hardware control of print den-
 sity, 7-11
 Hardware errors, 6-7
 Hardware-related duties, 1-1
 /HEADER switch in LPFORM.INI,
 4-38
 HELP command in,
 BATCON, 4-22
 MAKVFU, 7-12
 output spooler, 4-30
 SPRINT, 4-48
 :HELP OPSER command, 4-7
 /HELP switch in SPACE, 5-3
 HUNG condition error recovery,
 6-5

 INFORMATION command in MAKVFU,
 7-12
 Informational messages KLINIT,
 C-5
 Initialization dialog,
 DATE:, 3-10
 KLINIT, C-2
 STARTUP OPTION:, 3-10
 TIME:, 3-10
 TOPS-10, 3-9
 WHY RELOAD:, 3-10
 Input device characteristics
 SPRINT, 4-48
 Input devices for SPRINT, 4-47
 Installation-supplied docu-
 mentation, 1-2
 Interleave memory in KLINIT,
 C-4
 Interleave switch settings,
 MF10, 6-11
 MG10, 6-13
 MH10, 6-13
 Interleaving memory, 6-9
 Interpreter,
 AID, 2-6
 APL, 2-6
 BASIC, 2-6

 :JCONT command in OPSER, 4-4
 JOB STOPCD, 6-2
 JUMP command in PARSER, B-5

 "keep-alive" count, 6-4
 KILL command in,
 BATCON, 4-22
 output spooler, 4-30
 SPRINT, 4-48

INDEX (CONT.)

- :KILL OPSEER command, 4-7
- /KILL switch in QUEUE, 4-15
- :KJOB command in OPSEER, 4-7
- KLERR routine, 6-4
- KLINIT,
 - capabilities, C-1
 - configuration maps, C-15
 - configure memory, C-3
 - conventions for, C-2
 - defaults, C-1
 - dialog error messages, C-7
 - dialog rules, C-2
 - exit, C-5
 - how to enter, C-1
 - informational messages, C-5
 - interleave memory, C-4
 - load bootstrap, C-4
 - memory maps, C-15
 - messages, C-5
 - reconfigure cache, C-3
 - reload microcode, C-3
 - system error messages, C-7
 - verify microcode, C-3
 - warning messages, C-6
- KLINIT auto-reload, 6-4
- KLINIT dialog, C-2
 - examples of, C-16
- :KSYS command in OPSEER, 4-4, 5-13

- LA180 terminal, 2-4
- LA35RO terminal, 2-4
- LA36 terminal, 2-4
- LA37APL terminal, 2-4
- Languages,
 - assemblers, 2-6
 - compilers, 2-6
 - interpreters, 2-6
- LENGTH command in MAKVFU, 7-12
- LIMIT output spooler command, 4-30
- /LIMIT switch in QUEUE, 4-15
- Line printer,
 - cleaning of, 7-22
 - operation of, 7-1
- Line printer forms change, 7-4
- Line printer ribbon change, 7-16
- Line printers,
 - LP07, 2-2
 - LP10, 2-2
- /LINES switch in LPFORM.INI, 4-38
- LINK-10 utility, 2-7
- /LIST switch in QUEUE, 4-15
- Load bootstrap in KLINIT, C-4
- Load switches, 3-1
- Loader,
 - BOOTDX, 3-14
 - Loading card reader, 7-22
 - Loading communications front ends, 3-11
 - Loading dual processor, 3-11
 - Loading RSX-20F,
 - from DECTape, 3-4
 - from disk, 3-3
 - via SW/REG, 3-5
 - Loading the central processor, 3-8
 - Loading the DX10 data channel, 3-14
 - Loading the front end, 3-3
 - Loading TOPS-10,
 - from disk, 3-8
 - from tape, 3-8
 - Loading via the DL10, 3-12
 - Loading via the DTE20, 3-12
 - LOCK output spooler command, 4-30
 - LOG file in BATCON, 4-25
 - :LOGIN command in OPSEER, 4-7
 - LP07 line printer,
 - forms change, 7-4
 - forms control, 7-11
 - ribbon change, 7-16
 - ribbon reversal, 7-18
 - vertical forms control, 7-11
 - LP10 line printer,
 - carriage control tape, 7-12
 - forms change, 7-8
 - forms control, 7-12
 - ribbon change, 7-19
 - ribbon reversal, 7-21
 - LPFORM.INI file, 4-37
 - entry format, 4-37
 - examples of, 4-40
 - LPFORM.INI switches,
 - /ALCNT, 4-37
 - /ALIGN, 4-38
 - /ALSLP, 4-38
 - /BANNER, 4-38
 - /CHAIN, 4-38
 - /DRUM, 4-38
 - /HEADER, 4-38
 - /LINES, 4-38
 - /NOTE, 4-38
 - /PAUSE, 4-38
 - /RIBBON, 4-38
 - /TAPE, 4-38
 - /TRAILER, 4-39
 - /VFU, 4-39
 - /WHAT, 4-39
 - /WIDTH, 4-39
 - LPI command in MAKVFU, 7-12
 - LPTSPL, 4-34
 - abort current print request, 4-36
 - changing MLIMIT in, 4-35
 - divert output from, 4-36

INDEX (CONT.)

LPTSPL (cont.),
 operator messages, 4-40
 special forms handler, 4-37
 starting, 4-28
 starting and stopping, 4-35

MACRO assembler, 2-6

Magnetic tape drives, 2-3
 cleaning of, 7-34
 operation of, 7-26

Magnetic tape input to SPRINT,
 4-47

Magnetic tapes,
 dismounting, 7-27
 mounting, 7-27

MAINTENANCE console mode, B-1

MAKVFU commands,
 CHANNEL, 7-12
 EXIT, 7-12
 FORMS, 7-12
 HELP, 7-12
 INFORMATION, 7-12
 LENGTH, 7-12
 LPI, 7-12
 OUTPUT, 7-12
 PAGE, 7-12

MAKVFU program, 7-11

Mapping disk packs, 5-19

MCORE,
 BATCON command, 4-21
 SPRINT command, 4-48

MCR PARSE command, B-5

Memory,
 addressing, 6-10
 interleaving, 6-9
 maps, C-15
 modules, 2-1, 6-10, 6-12
 multiplexer, 2-2
 parity errors, 6-7
 port switches, 6-12
 selection switches, 6-12

Memory address switches,
 MF10, 6-11
 MG10, 6-12, 6-13
 MH10, 6-13

Memory configuration,
 altering of, 6-8

Message format,
 STOPCD, 6-2

Message of the day, 5-1

MESSAGE output spooler command,
 4-30

MF10 memory, 6-10
 interleave switch settings,
 6-11
 memory address switch settings,
 6-11

MG10 memory, 6-12
 interleave switch settings,
 6-13
 memory address switch settings,
 6-12, 6-13
 memory selection switches,
 6-12

MH10 memory, 6-12
 interleave switch settings,
 6-13
 memory address switch settings,
 6-13
 memory selection switches,
 6-12

MJOB command in BATCON, 4-21

MLIMIT output spooler command,
 4-31

/MODIFY switch in QUEUE, 4-15

Modules,
 memory, 2-1

MONEY listings FACT files, 5-13

MONEY program, 5-3

Monitor error stop, 6-2

:MONITOR command in OPSER, 4-7

Monitor parity error, 6-7

Monitoring BATCON, 4-26

Mounting DECTapes, 7-26

Mounting disk packs, 7-35

Mounting magnetic tapes, 7-27

:MSGLVL command in OPSER, 4-7

MSGLVL command in SPRINT, 4-48

MTIME command in BATCON, 4-21

Multi-reel save in BACKUP, 5-18

Multiplexer, 2-2

NEXT command,
 BATCON, 4-21
 output spooler, 4-31

/NODUMP switch in DTELDL, 6-15,
 6-16

Non-daily operator tasks, 5-15

Non-scheduled operator tasks,
 5-19

NONJOB command in BATCON, 4-22

NOSUPPRESS output spooler
 command, 4-31

/NOTE switch in LPFORM.INI, 4-38

NOTICE.TXT file, 5-1

Numbering of memories, 6-10

OMOUNT,
 FILE command, 4-11
 MOUNT/DISMOUNT command, 4-11
 PROCEED? prompt, 4-12
 PROCEED? replies, 4-12

INDEX (CONT.)

- OMOUNT system program, 4-11
- One-way communication, 4-54
- Operating systems, 2-5
- Operation of,
 - card reader, 7-22
 - DEctape drives, 7-25
 - devices, 7-1
 - disk drives, 7-35
 - line printer, 7-1
 - magnetic tape drives, 7-26
- OPERATOR command in BATCON, 4-23
- Operator commands,
 - ATTACH device, A-1
 - ATTACH diskunit, A-2
 - ATTACH job, A-2
 - DETACH device, A-3
 - DETACH diskunit, A-3
 - SET BREAK, A-4
 - SET CORMAX, A-4
 - SET CORMIN, A-5
 - SET DATE, A-5
 - SET DAYTIME, A-6
 - SET MEMORY, A-6
 - SET NOMESSAGE, A-7
 - SET OPR, A-7
 - SET SCHED, A-8
 - SET TTY, A-8
 - SET VMMAx, A-14
 - to configure memory, A-6
 - XCHNGE, A-14
- Operator commands,
 - OPSER, 4-4
 - PARSER, B-2
- OPERATOR console mode, B-1
- Operator dialog,
 - KLINIT, C-2
- Operator documentation, 1-2
- Operator duties, 1-1
- Operator messages,
 - BATCON, 4-26
 - LPTSPL, 4-40
 - SPRINT, 4-49
 - SPROUT, 4-44
- Operator responsibilities, 1-1
- Operator tasks, 5-1
 - daily, 5-1
 - non-daily, 5-15
 - non-scheduled, 5-19
- Operator's notebook, 1-3
- Operator-privileged commands, A-1
- OPR.ATO file, 4-4
 - for DTELD, 6-15, 6-16
- OPSER,
 - AUTO feature, 4-4
 - commands to OPSER, 4-2
 - commands to subjobs, 4-3
 - error messages, 4-9
 - operation, 4-1
 - OPSER (cont.),
 - operator commands for, 4-4
 - specifying subjobs to, 4-2
 - starting, 4-1
 - timestamp, 4-3
 - user commands for, 4-5
 - OPSER commands,
 - :AUTO, 4-5
 - :CLOSE, 4-6
 - :CONTINUE, 4-6
 - :CTRL/C, 4-7
 - :CURRENT, 4-6
 - :DAYTIME, 4-6
 - :DEFINE, 4-6
 - :DEVICE, 4-7
 - :ERROR, 4-7
 - :EXIT, 4-7
 - :FREE, 4-7
 - :HELP, 4-7
 - :JCONT, 4-4
 - :KILL, 4-7
 - :KJOB, 4-7
 - :KSYS, 4-4
 - :LOGIN, 4-7
 - :MONITOR, 4-7
 - :MSGVLV, 4-7
 - :QUEUE, 4-8
 - :RESOURCES, 4-8
 - :RESTRICT, 4-5
 - :REVIVE, 4-8
 - :SCHED, 4-8
 - :SEND, 4-8
 - :SET BATMAX, 4-5
 - :SET BATMIN, 4-5
 - :SET BREAK, 4-5
 - :SET CORMAX, 4-5
 - :SET CORMIN, 4-5
 - :SET DATE, 4-5
 - :SET DAYTIME, 4-5
 - :SET LOGMAX, 4-5
 - :SET MEMORY, 4-5
 - :SET NOMESSAGE, 4-5
 - :SET OPR, 4-5
 - :SET RUN, 4-5
 - :SET SCHED, 4-5
 - :SET TTY, 4-5
 - :SET VMMAx, 4-5
 - :SILENCE, 4-8
 - :SLOGIN, 4-8
 - :STOP, 4-8
 - :SYSTAT, 4-8
 - :TIME, 4-8
 - :TLDG, 4-8
 - :TSILENCE, 4-8
 - :TTYTST, 4-9
 - :UNRESTRICT, 4-5
 - :WHAT, 4-9
 - :WHERE, 4-9
 - OPSER system program, 4-1
 - OUTPUT command in MAKVFU, 7-12

INDEX (CONT.)

- Output spooler commands,
 - ALIGN, 4-29
 - BACKSPACE, 4-29
 - CARDS, 4-29
 - CHKPNT, 4-29
 - CURRENT, 4-29
 - EXIT, 4-29
 - FORMS, 4-30
 - FORWARD, 4-30
 - FREEZE, 4-30
 - GO, 4-30
 - HELP, 4-30
 - KILL, 4-30
 - LIMIT, 4-30
 - LOCK, 4-30
 - MESSAGE, 4-30
 - MLIMIT, 4-31
 - NEXT, 4-31
 - NOSUPPRESS, 4-31
 - PAUSE, 4-31
 - PLOT, 4-31
 - REPRINT, 4-31
 - REQUEUE, 4-32
 - RESET, 4-32
 - SKPCOPY, 4-33
 - SKPFILE, 4-33
 - START, 4-33
 - STOP, 4-33
 - SUPPRESS, 4-33
 - TAPE, 4-34
 - UNFREEZE, 4-34
 - UNLOCK, 4-34
 - WHAT, 4-34
- Output spoolers, 4-28
 - starting, 4-28
- PAGE command in MAKVFU, 7-12
- Paper installation,
 - LP07, 7-4
 - LP10, 7-8
- Parity error,
 - monitor, 6-7
 - recovery, 6-7
 - user core, 6-7
- PARSER,
 - console modes, B-1
 - entering, B-1
 - error messages, B-7
 - exiting, B-1
 - operator commands, B-2
 - prompts, B-1
- PARSER commands, B-1
 - ABORT, B-3
 - CLEAR CONSOLE, B-3
 - CLEAR INCREMENT, B-3
 - CLEAR MEMORY, B-3
 - CLEAR NOT, B-4
- PARSER commands (cont.),
 - CLEAR REPEAT, B-4
 - EXAMINE ELEVEN, B-4
 - EXAMINE KL, B-4
 - EXAMINE PC, B-4
 - EXAMINE TEN, B-4
 - JUMP, B-5
 - MCR, B-5
 - QUIT, B-5
 - REPEAT, B-5
 - RUN, B-6
 - SET CONSOLE, B-6
 - SET INCREMENT, B-6
 - SET MEMORY, B-6
 - SET NOT, B-6
 - SET REPEAT, B-7
 - SHUTDOWN, B-7
 - WHAT CONSOLE, B-7
 - WHAT INCREMENT, B-7
 - WHAT MEMORY, B-7
 - WHAT REPEAT, B-7
 - WHAT VERSION, B-7
- Partial save of system, 5-11
 - /PAUSE switch in LPFORM.INI, 4-38
- PAUSE command in,
 - output spooler, 4-31
 - SPRINT, 4-49
- Peripherals, 2-2
- PIP utility, 2-7
- PLEASE command, 4-53
 - in BATCON, 4-25
- PLOT output spooler command, 4-31
- Plotter XY10-A, 2-4
- Plotter XY10-B, 2-4
- Ports,
 - memory, 2-2
- Power fail restart, 6-14
- Power fluctuations, 6-14
- POWER OFF switch, 3-1
- POWER ON switch, 3-1
- Power outages, 6-14
- Powering up the system, 3-3, 6-15
- Preserve crash data, 6-6
- Print density,
 - hardware control of, 7-11
 - software control of, 7-11
- /PRIORITY switch in QUEUE, 4-15
- PROCEED?,
 - OMOUNT prompt, 4-12
 - OMOUNT reply, 4-12
- Processor,
 - central, 2-1
 - Console/Diagnostic, 2-1
 - front-end, 2-2
- PROGRAMMER console mode, B-1

INDEX (CONT.)

- QUASAR, 4-16
 - error messages, 4-17
- QUEUE command,
 - format, 4-14
 - queues, 4-14
 - switches, 4-15
- :QUEUE command in OPSER, 4-8
- QUEUE switches,
 - /AFTER, 4-15
 - /COPIES, 4-15
 - /CORE, 4-15
 - /DEADLINE, 4-15
 - /DISPOSE, 4-15
 - /KILL, 4-15
 - /LIMIT, 4-15
 - /LIST, 4-15
 - /MODIFY, 4-15
 - /PRIORITY, 4-15
 - /RESTART, 4-15
 - /TAPE, 4-15
 - /TIME, 4-15
- QUEUE system program, 4-14
- QUIT command in PARSE, B-5

- REACT program, 5-5, 5-6
- Reconfigure cache with KLINIT, C-3
- Reconfiguring memories, 6-9
- Recovery procedures, 6-1
- Reloading,
 - DL10, 6-17
 - DX10, 6-18
 - microcode, C-3
 - with DTELDR, 6-15
- Remote stations,
 - DAS98, 2-5
 - DC72, 2-5
 - DN80, 2-5
 - DN81, 2-5
 - DN82, 2-5
- Removing inactive jobs, 5-8
- REPEAT command in PARSE, B-5
- Reply to PLEASE command, 4-54
- REPRINT output spooler command, 4-31
- REQUEUE command in,
 - BATCON, 4-23
 - Output spooler, 4-32
- RESET command in,
 - BATCON, 4-21
 - Output spooler, 4-32
 - SPRINT, 4-49
- :RESOURCES command in OPSER, 4-8
- Restart procedure,
 - crash, 6-5
- /RESTART switch in QUEUE, 4-15
- :RESTRICT command in OPSER, 4-5
- :REVIVE command in OPSER, 4-8
- RH20, 2-1
- RHP02 disk system, 2-3
- RHP03 disk system, 2-3
- RHP04 disk system, 2-3
- RHP06 disk system, 2-3
- RHS04 disk system, 2-3
- Ribbon change,
 - line printer, 7-16
 - LP07, 7-16
 - LP10, 7-19
- Ribbon reversal,
 - LP07, 7-18
 - LP10, 7-21
- /RIBBON switch in LPFORM.INI, 4-38
- ROUTE command in BATCON, 4-24
- RP02 disk packs,
 - dismounting of, 7-35
 - formatting, 5-19, 5-22
 - mounting of, 7-35
- RP03 disk drive, 7-37
- RP03 disk packs,
 - dismounting of, 7-35
 - formatting, 5-19, 5-22
 - mounting of, 7-35
- RP04 disk drive, 7-38
- RP04 disk packs,
 - dismounting of, 7-35
 - formatting, 5-23
 - mounting of, 7-35
- RP06 disk drive, 7-39
- RP06 disk packs,
 - dismounting of, 7-35
 - formatting, 5-23
 - mounting of, 7-35
- RSX-20F,
 - command parser, B-1
 - detected errors, 6-3
 - monitor, 2-5
- RUN command in PARSE, B-6
- RUNOFF program, 2-7

- Saving disk contents, 5-10
- Saving FACT files, 5-12
- :SCHED command in OPSER, 4-8
- Selection of memory, 6-9
- SEND command, 4-54
- :SEND command in OPSER, 4-8
- :SET BATMAX command in OPSER, 4-5
- :SET BATMIN command in OPSER, 4-5
- :SET BREAK command in OPSER, 4-5
- SET BREAK operator command, A-4

INDEX (CONT.)

SET CONSOLE command in PARSEr, B-6
 :SET CORMAX command in OPSER, 4-5
 SET CORMAX operator command, A-4
 :SET CORMIN command in OPSER, 4-5
 SET CORMIN operator command, A-5
 :SET DATE command in OPSER, 4-5
 SET DATE operator command, A-5
 :SET DAYTIME command in OPSER, 4-5
 SET DAYTIME operator command, A-6
 SET INCREMENT command in PARSEr, B-6
 :SET LOGMAX command in OPSER, 4-5
 SET MEMORY, operator command, A-6
 PARSER command, B-6
 :SET MEMORY command in OPSER, 4-5
 :SET NOMEsSAGE command in OPSER, 4-5
 SET NOMEsSAGE operator command, A-7
 SET NOT command in PARSEr, B-6
 :SET OPR command in OPSER, 4-5
 SET OPR operator command, A-7
 SET REPEAT command in PARSEr, B-7
 :SET RUN command in OPSER, 4-5
 :SET SCHED command in OPSER, 4-5
 SET SCHED operator command, A-8
 :SET TTY command in OPSER, 4-5
 SET TTY operator command, A-8
 :SET VMMAX command in OPSER, 4-5
 SET VMMAX operator command, A-14
 Setting memory off-line, 6-8
 Shutdown of system, 5-13
 SHUTDOWN command in PARSEr, B-7
 :SILENCE command in OPSER, 4-8
 SKPCOPY output spooler command, 4-33
 SKPFILE output spooler command, 4-33
 :SLOGIN command in OPSER, 4-8
 Software control of print density, 7-11
 Software-related duties, 1-1
 SOUP program, 2-7
 SPACE program with,
 /BATCH switch, 5-11
 /FACT switch, 5-2
 /HELP switch, 5-3
 Special forms handler LPTSPL, 4-37
 Specifying subjobs to OPSER, 4-2
 Spoolers,
 input, 4-46
 output, 4-28
 SPRINT,
 card reader input to, 4-47
 disk input to, 4-47
 input device characteristics, 4-48
 input devices, 4-47
 magnetic tape input to, 4-47
 operator messages, 4-49
 starting, 4-47
 SPRINT commands,
 EXIT, 4-48
 GO, 4-48
 HELP, 4-48
 KILL, 4-48
 MCORE, 4-48
 MSGVLV, 4-48
 PAUSE, 4-49
 RESET, 4-49
 START, 4-49
 STOP, 4-49
 TELL, 4-49
 WHAT, 4-49
 SPRINT system program 4-46
 SPROUT,
 divert output from, 4-44
 operator messages, 4-44
 starting, 4-28, 4-44
 SPROUT system program, 4-43
 Standard LP10 carriage control tape, 7-13
 START command in,
 BATCON, 4-21
 output spooler, 4-33
 SPRINT, 4-49
 Starting,
 BATCON, 4-19
 LPTSPL, 4-28
 OPSER, 4-1
 output spooler, 4-28
 SPRINT, 4-47
 SPROUT, 4-28, 4-44
 the system, 3-1
 STOP command in,
 BATCON, 4-24
 output spooler, 4-33
 SPRINT, 4-49
 :STOP command in OPSER, 4-8
 STOPCD,
 DEBUG, 6-2
 HALT, 6-3
 JOB, 6-2
 STOP, 6-3
 STOPCD message format, 6-2
 SUPPRESS command in,
 output spooler, 4-33

INDEX (CONT.)

- SW/REG load switch, 3-1
- Switches,
 - DECTape load, 3-1
 - DISK load, 3-1
 - EMERGENCY POWER OFF, 3-1
 - ENABLE load, 3-1
 - POWER OFF, 3-1
 - POWER ON, 3-1
 - SW/REG load, 3-1
- Switch register, 3-1
 - bit definitions, 3-6
- Synchronous front-ends, 2-2
- SYSCHK, 3-9
- SYSTAT command, 5-8
- :SYSTAT command for OPSER, 4-8
- System,
 - hardware, 2-1
 - logbook, 1-3
 - partial save of, 5-11
 - powering up, 3-3
 - recovery procedures, 6-1
 - services, 4-1
 - shutdown, 5-13
 - software, 2-5
 - starting the, 3-1
 - total save of, 5-16
 - utilities, 2-7
- System components, 2-1
- System program,
 - LPTSPL, 4-34
 - OMOUNT, 4-11
 - OPSER, 4-1
 - QUEUE, 4-14
 - SPRINT, 4-46
 - SPROUT, 4-43
- Tape dismounting,
 - TU10, 7-27
 - TU16, 7-27
 - TU40, 7-29
 - TU41, 7-29
 - TU45, 7-34
 - TU70, 7-29
 - TU71, 7-29
 - TU72, 7-29
- Tape mounting,
 - TU10, 7-27
 - TU16, 7-27
 - TU40, 7-29
 - TU41, 7-29
 - TU45, 7-32
 - TU70, 7-29
 - TU71, 7-29
 - TU72, 7-29
- TAPE output spooler command, 4-34
- /TAPE switch in LPFORM.INI, 4-38
- /TAPE switch in QUEUE, 4-15
- TECO program, 2-7
- TELL command in,
 - BATCON, 4-25
 - SPRINT, 4-49
- Terminals,
 - console, 2-1
 - CRT display, 2-5
 - hardcopy, 2-4
 - LA180, 2-4
 - LA35RO, 2-4
 - LA36, 2-4
 - LA37APL, 2-4
 - VT50, 2-5
 - VT52, 2-5
 - VT61, 2-5
- Terminating timesharing, 5-13
- Termination,
 - abnormal, 1-2
- Text files on tape, 1-2
- TIME command in BATCON, 4-21
- :TIME command in OPSER, 4-8
- /TIME switch in QUEUE, 4-15
- Timesharing,
 - terminating, 5-13
- Timestamp in OPSER, 4-3
- :TLOG command in OPSER, 4-8
- TOPS-10 batch system, 4-16
- TOPS-10 initial dialog, 3-9
- TOPS-10 monitor, 2-5
- Total save of system, 5-16
- /TRAILER switch in LPFORM.INI, 4-39
- TRPSET, 3-14
- :TSILENCE command in OPSER, 4-8
- :TTYTST command in OPSER, 4-9
- TU10 magnetic tape drive, 2-3
- TU10 tape,
 - dismounting, 7-27
 - mounting, 7-27
- TU16 magnetic tape drive, 2-4, 7-28
- TU16 tape,
 - dismounting, 7-27
 - mounting, 7-27
- TU40 magnetic tape drive, 2-4, 7-30
- TU40 tape,
 - dismounting, 7-29
 - mounting, 7-29
- TU41 magnetic tape drive, 2-4
- TU41 tape,
 - dismounting, 7-29
 - mounting, 7-29
- TU45 magnetic tape drive, 2-4, 7-33
- TU45 tape,
 - dismounting, 7-34
 - mounting, 7-32
- TU56 DECTape, 2-3
- TU70 magnetic tape drive, 2-4, 7-31

INDEX (CONT.)

TU70 tape, dismounting, 7-29 mounting, 7-29	/VFU switch in LPFORM.INI, 4-39 VT50 terminal, 2-5 VT52 terminal, 2-5 VT61 terminal, 2-5
TU71 magnetic tape drive, 2-4	
TU71 tape, dismounting, 7-29 mounting, 7-29	
TU72 magnetic tape drive, 2-4	
TU72 tape, dismounting, 7-29 mounting, 7-29	
Two-way communication, 4-53	
	Warning messages in KLINIT, C-6
	Weekly, account control, 5-15 BACKUP, 5-16
	:WHAT command in OPSER, 4-9
	WHAT command in, BATCON, 4-22 output spooler, 4-34 SPRINT, 4-49
UNFREEZE output spooler command, 4-34	WHAT CONSOLE command in PARSER, B-7
Unloading card reader, 7-22	WHAT INCREMENT command in PARSER, B-7
UNLOCK output spooler command, 4-34	WHAT MEMORY command in PARSER, B-7
:UNRESTRICT command in OPSER, 4-5	WHAT REPEAT command in PARSER, B-7
Updating account files, 5-4	/WHAT switch in LPFORM.INI, 4-39
User commands for OPSER, 4-5	WHAT VERSION command in PARSER, B-7
USER console mode, B-1	:WHERE command in OPSER, 4-9
User core failure error recovery, 6-7	/WIDTH switch in LPFORM.INI, 4-39
User core parity error, 6-7	
User to operator communication, 4-53	
Utility programs, 2-7	
	XCHNGE operator command, A-14
	XY10-A plotter, 2-4
	XY10-B plotter, 2-4
Verify microcode in KLINIT, C-3	
Vertical forms control, LP07, 7-11 LP10, 7-12	

READER'S COMMENTS

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

Did you find errors in this manual? If so, specify by page.

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

Is there sufficient documentation on associated system programs required for use of the software described in this manual? If not, what material is missing and where should it be placed?

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
- ☐ Non-programmer interested in computer concepts and capabilities

Name _____ Date _____

Organization _____

Street _____

City _____ State _____ Zip Code _____

or
Country

----- **Fold Here** -----

----- **Do Not Tear - Fold Here and Staple** -----

FIRST CLASS
PERMIT NO. 33
MAYNARD, MASS.

[REDACTED]

Postage will be paid by:

digital

Software Documentation
146 Main Street ML5-5/E39
Maynard, Massachusetts 01754

