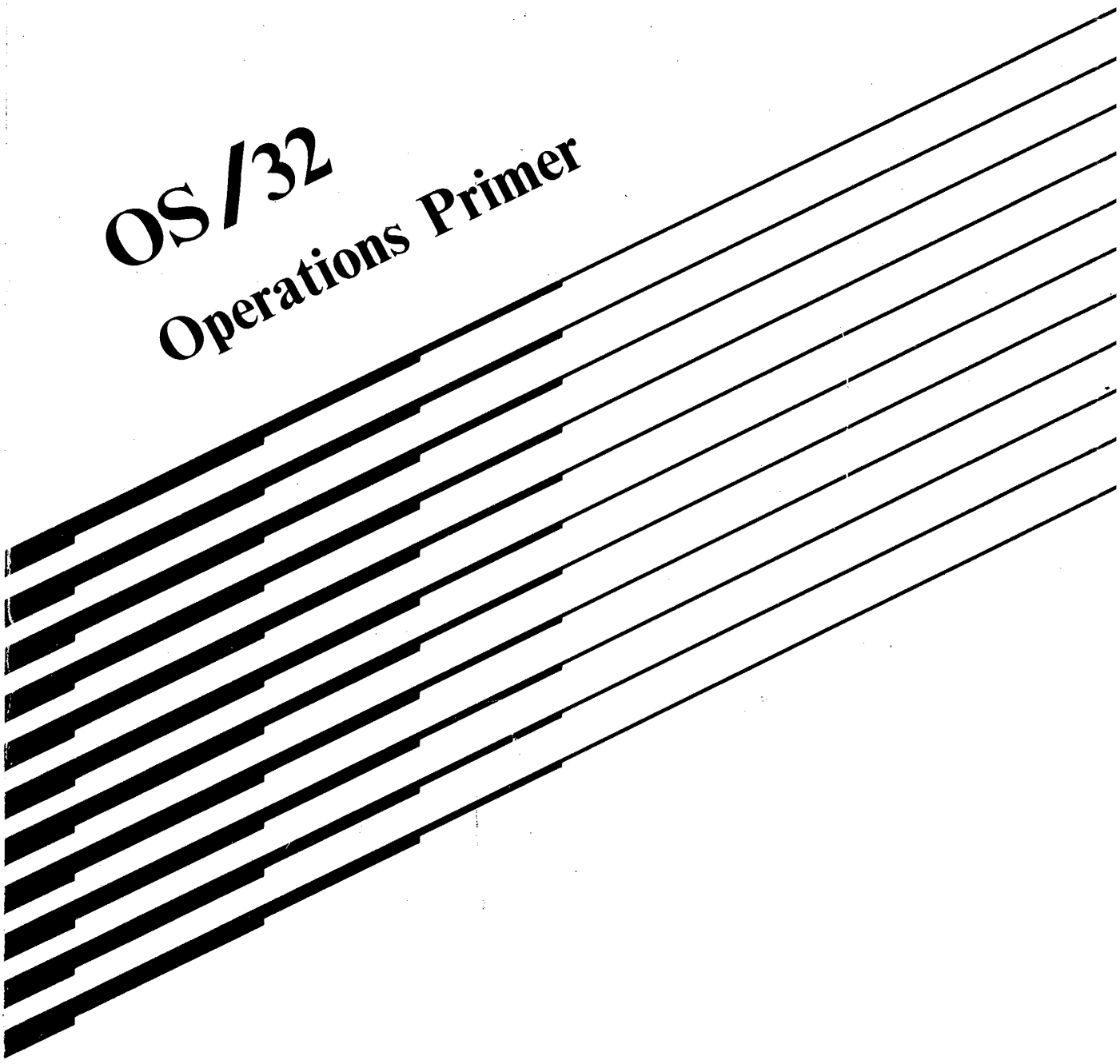
OS/32 Operations Primer



PERKIN-ELMER

PERKIN-ELMER

OS/32 OPERATIONS

Primer

48-076 F00 R00

The information in this document is subject to change without notice and should not be construed as a commitment by The Perkin-Elmer Corporation. The Perkin-Elmer 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 it can be used or copied only in a manner permitted by that license. Any copy of the described software must include the Perkin-Elmer copyright notice. Title to and ownership of the described software and any copies thereof shall remain in The Perkin-Elmer Corporation.

The Perkin-Elmer Corporation assumes no responsibility for the use or reliability of its software on equipment that is not supplied by Perkin-Elmer.

The Perkin-Elmer Corporation, Data Systems Group, 2 Crescent Place,

Oceanport, New Jersey 07757

Printed in the United States of America

TABLE OF CONTENTS

CHAPTERS (Continued)

CHAPTERS (Continued)

CHAPTERS (Continued)

FIGURES

TABLES

PREFACE

The OS/32 Operations Primer is intended for those with little or no practical knowledge of computers. It serves as a guide to walk the novice operator through the concepts of computer organization.

Learning the fundamentals of a real-time operating system is essential to the computer operator. This primer introduces the standard conventions and convenient facilities of Perkin-Elmer's real-time operating system (OS/32).

The main objective of the Operations Primer is to give the operator a reasonably complete overview of OS/32. This guide offers step-by-step procedures, examples and a glossary of technical terms.

For information on the contents of all Perkin-Elmer 32-bit manuals, see the 32-Bit Systems User Documentation Summary.

CHAPTER 1
COMPUTER ORGANIZATION

INTRODUCTION

A general-purpose computer can be used to solve a wide variety of problems. It is a tool used to expand our capabilities. Digital computers are designed to process information in digital (numeric) form. Before you can understand computer operations, you must become familiar with the capabilities of the computing equipment and how, through a system of procedures, this equipment is coordinated with people.

The Glossary in the back of this book should help you become familiar with some of the computer terms we will be discussing.

COMPUTER HARDWARE

Computer hardware refers to the actual physical components of a computer system. Hardware refers to any piece of automatic data processing equipment. A computer is made up of five basic units:

- Control unit
- Arithmetic logic unit (ALU)
- Memory unit
- Input devices
- Output devices

Figure 1-1 illustrates these basic units.

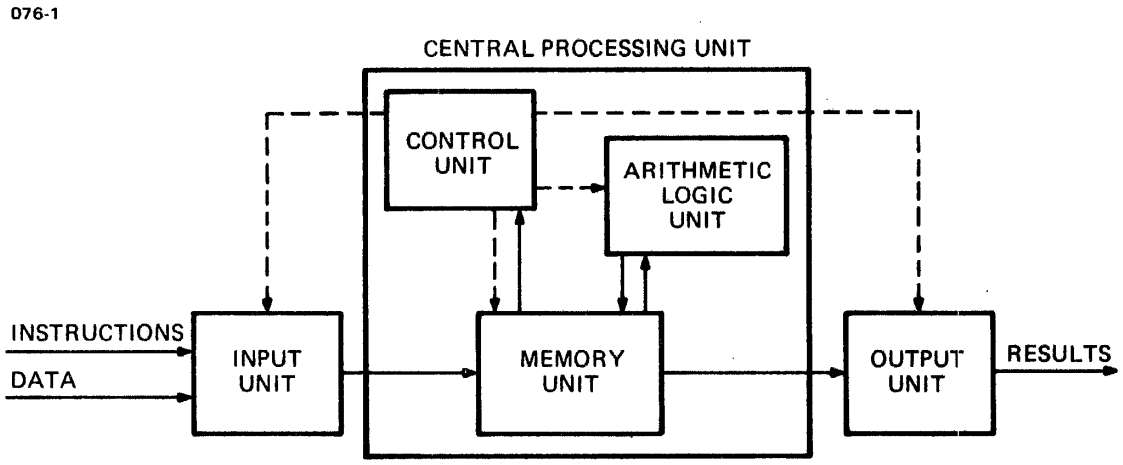


Figure 1-1 The Five Basic Hardware Components of a Computer

CENTRAL PROCESSING UNIT (CPU)

The central processing unit (CPU) consists of three major units:

- The control unit directs the operation of the system. It fetches and interprets instructions from memory and generates signals and commands that cause other system units to perform certain operations. It does not, however, operate on data.

- The arithmetic logic unit (ALU) performs the basic arithmetic operations and the logical and decision-making functions as directed by the control unit.

- The memory unit stores program instructions and data. The memory unit is broken down into individual locations, each of which has a numerical designation called its address. Each location can store only one piece of information at a time. This information can be retrieved by specifying the address of the memory location.

INPUT/OUTPUT (I/O) DEVICES

Input/output (I/O) devices are units used to input data to the computer or to output information from it. Input devices allow data and instructions to be read and transferred to the memory unit. Output devices allow data to be taken from memory and written to some output medium.

Input/output (I/O) devices, such as card readers, line printers, card punches, magnetic tape drives and magnetic disk units are connected to the central processing unit (CPU) via electrical cables. Table 1-1 shows the various input/output (I/O) devices.

TABLE 1-1 INPUT/OUTPUT (I/O) DEVICES

INPUT DEVICES	OUTPUT DEVICES
Card readers	Card punches
Teletype devices	Teletype devices
CRT terminals	CRT terminals
Tape drives	Tape drives
Disk drives	Disk drives
	Printers

These devices provide two ways of handling data:

- Sequential processing
- Direct access processing

Sequential processing is a technique for handling data one item or record after the other. This method generally requires preliminary sorting. The sorted information is then stored sequentially (usually on magnetic tape files).

Direct access processing is a technique for handling data in random order and using files on a direct access storage device (DASD) (usually a magnetic disk).

Magnetic Tapes

Magnetic tapes provide a reliable way to store data and an efficient way to retrieve this data for use. A reflective spot at the beginning of the tape is known as the load point marker or beginning of tape (BOT). This reflective spot indicates where the usable portion of this tape begins.

The reflective spot at the end of the tape is referred to as the end of file marker or end of tape (EOT). This marker signals the tape drive that it has reached the end of the medium.

To prevent accidental destruction of files, a write-protect ring is inserted in the center of the tape reel. The tape can only be written to when this ring is inserted correctly.

Magnetic Tape Units

A magnetic tape unit provides the ability to write to or read from a magnetic tape. It provides a high data transmission rate to and from the computer. The methods of magnetic tape recording are very much like that used in the recording of reel-to-reel tapes on home stereos. A magnetic tape drive has two reels, a supply reel and a take-up reel. The actual threading sequence differs among tape drives; it is generally shown on the front of the tape drive. Perkin-Elmer offers a variety of tape drives. The 6250 tape drive automatically threads the tape around the take-up reel. The dual-density tape unit is more often used where the tape is manually threaded. A typical threading sequence is shown in Figure 1-2.

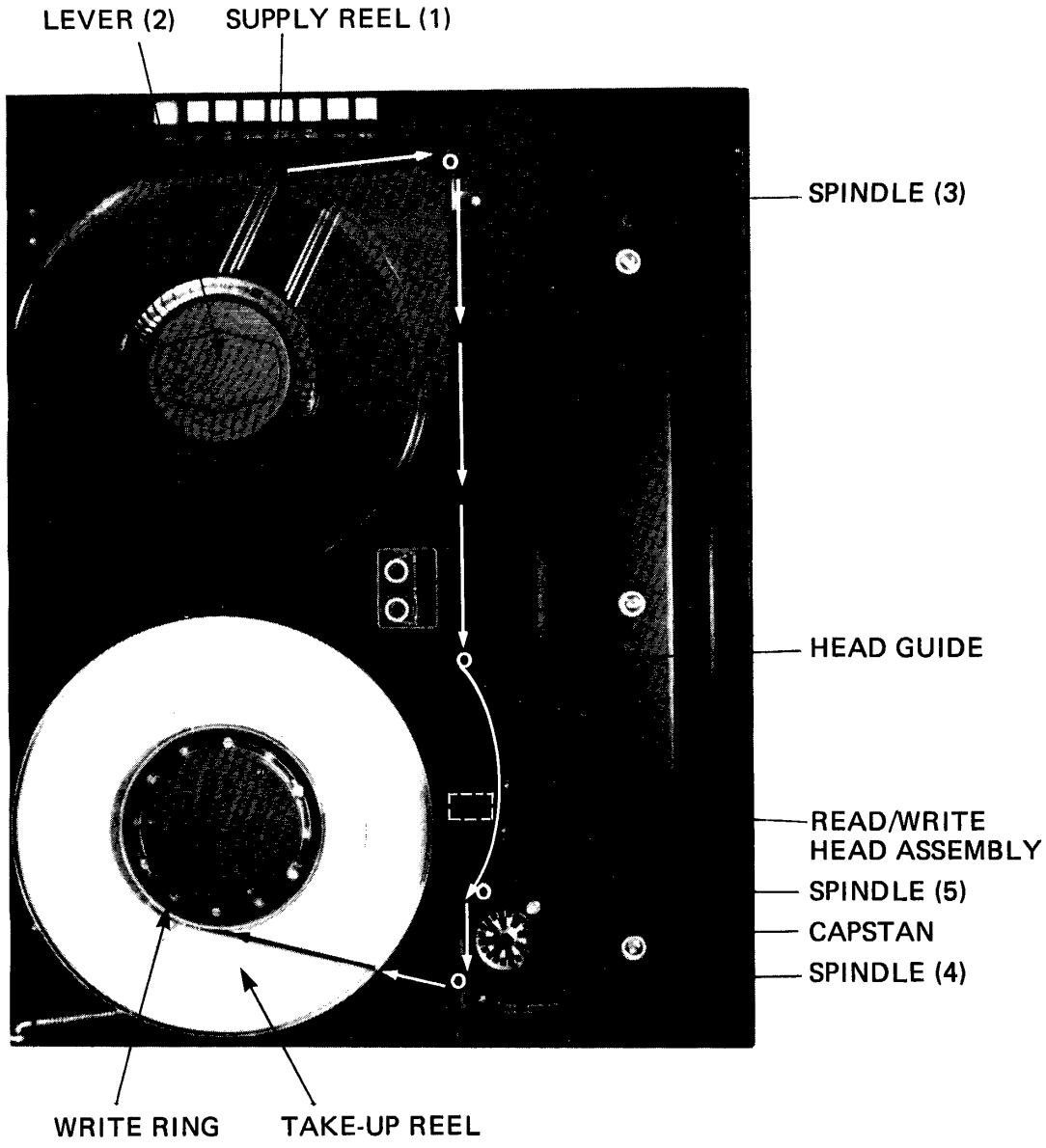


Figure 1-2 Magnetic Tape Unit

To mount a magnetic tape:

1. Open the glass door of the tape drive and place the tape on the supply reel (1).

2. Press the lever (2) in the center of the wheel to lock the tape in place.

3. Thread the tape in front of spindle (3), behind spindle (4) and in front of spindle (5).

4. Manually wrap the tape a few times around the take-up reel.

5. Close the glass door of the tape drive.

6. Press the LOAD button on the top front of the drive.

The magnetic tape is now loaded. To remove the tape:

1. Leave the glass door of the tape unit closed until the next three steps are completed.

2. Press the RESET button to bring the unit off-line. The REWIND button will not work if the unit is still on-line.

3. Press the REWIND button to bring the tape to load point.

4. Press the REWIND button again to remove the rest of the tape from the take-up reel.

5. Press the lever in the middle of the tape and remove the tape.

Read/Write Operations

The recording of data is accomplished by a read/write head. A magnetic tape is divided into tracks running the length of the tape. Data is recorded sequentially onto the tape in block form.

The tape stops each time a block of data is recorded. As the tape stops, blank spaces are left between each block. These blank spaces are known as interblock gaps (IBGs). Interblock gaps (IBGs) allow the tape drive the time it needs to reach the speed required to read or write data properly.

Information can be read from the magnetic tape by passing the tape over the read/write head, but the tape unit must be instructed to read. The tape unit begins reading after it senses the first character following the interblock gap (IBG) and continues reading until the next gap is reached or until a filemark is detected.

Magnetic Disk

A magnetic disk is a direct and random access device. The random access approach allows direct access to a record without having to read sequentially until a record is located. Disk storage is an essential feature of virtually all modern computer systems and the mainstay of data processing. Data is stored on the surface of a circular plate. These circular plates are referred to as platters. Platters are stacked on a spindle or drive to form a disk pack.

The disk's surface is divided into concentric circles called tracks and arc-shaped segments called sectors. A sector is the smallest unit of input/output (I/O) to a disk. Data is recorded as magnetized spots in the tracks. Figure 1-3 illustrates the surface of a disk.

Movable read/write heads are positioned over the desired location to access the desired track. A cylinder is a unit of storage space on a large disk drive. A cylinder might be all the tracks with the same number on the upper and lower surfaces of each platter in the pack, thus conceptually forming a cylinder-shaped unit of storage. Figure 1-4 shows the disk's track and cylinder relation.

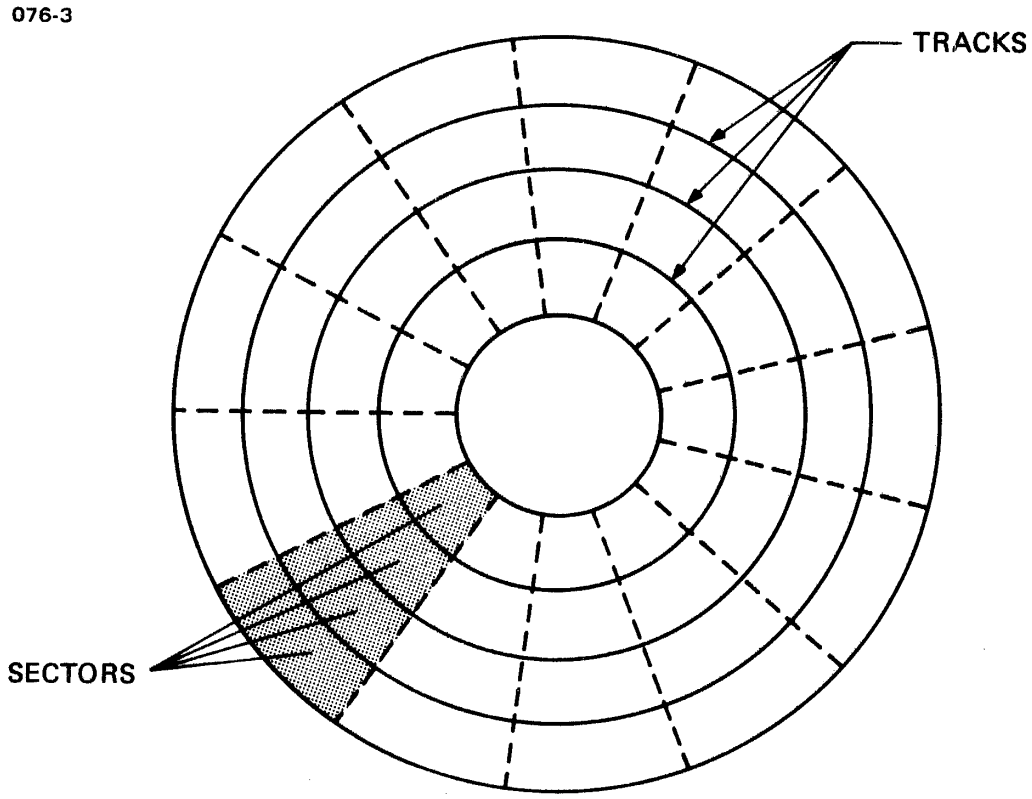


Figure 1-3 Disk Surface

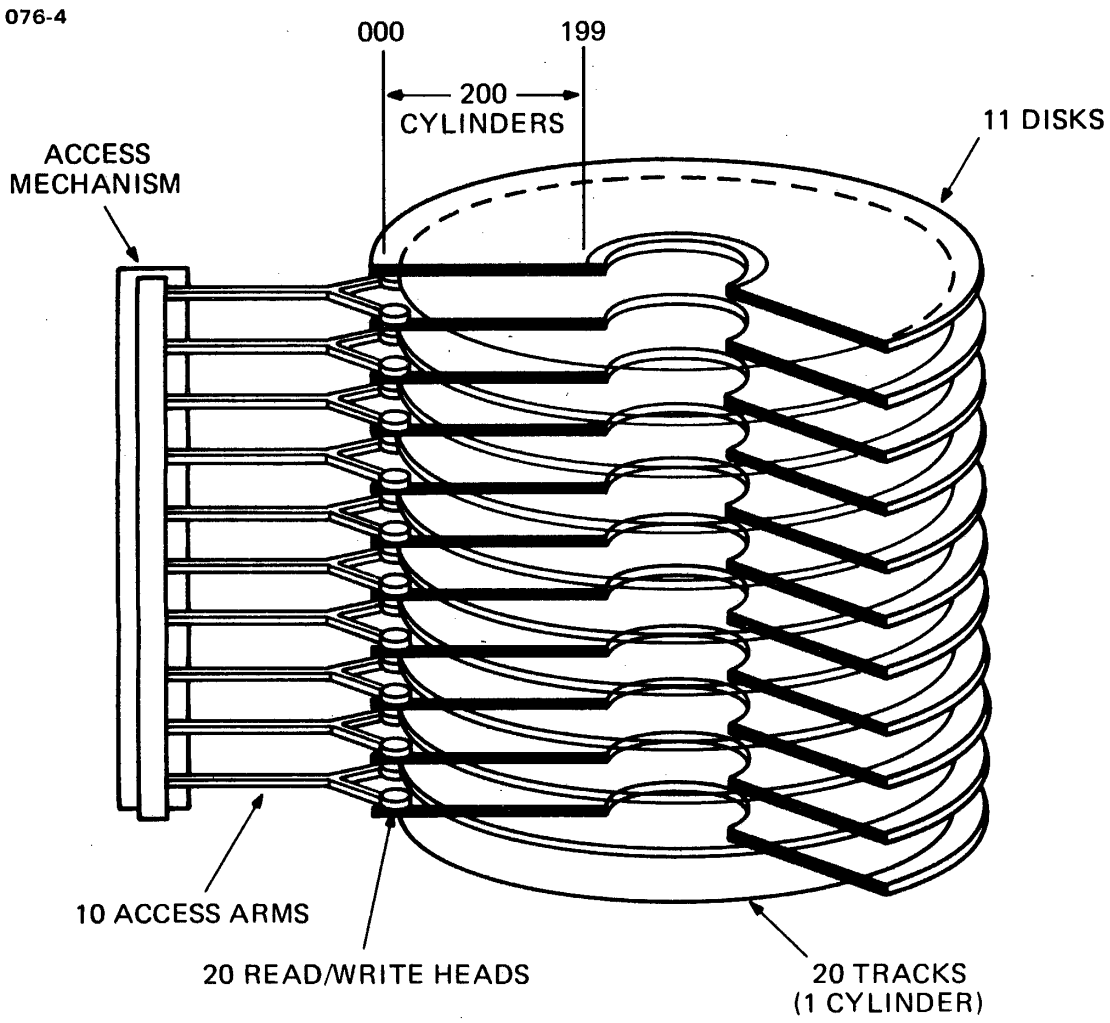


Figure 1-4 Concept of a Magnetic Disk Pack

Removable and Fixed Disks

In a fixed storage device, a disk cannot be removed from the disk drive. In a removable storage device, the disk pack can be removed from the disk drive.

The advantage of having removable disk packs is that secondary storage capacity can be expanded by purchasing other disk packs, rather than purchasing another disk drive.

INSTALLING A DISK PACK

Disk drives and packs are available in a variety of capacities. Each drive supports a particular capacity of disk pack. The capacity of the disk drive is generally printed on it. For example, 80Mb drives are labeled MSM80 and 300Mb drives are labeled MSM300. Disk packs must, of course, be used only on the drives for which they were designed. Figure 1-5 illustrates an MSM80 disk pack and Figure 1-6 shows an MSM80 disk drive.

To install an MSM80 disk pack:

1. Lift the top cover of the disk drive.

2. Ensure that the disk drive is not spinning. Be aware that failures can occur in the disk system circuitry, which will extinguish the ready lamp but allows the disk drive to remain spinning. The high-speed rotation of the disk drive can be deceiving; the drive may appear to be stationary to the unsuspecting eye.

3. Remove the bottom dust cover of the disk pack.

4. Insert the disk pack into the disk drive. Ensure that the disk pack is not angled in any way. Platters can become warped if touched by the storage canister when they are being removed or inserted.

5. Turn the T-handle of the storage canister clockwise until secure.

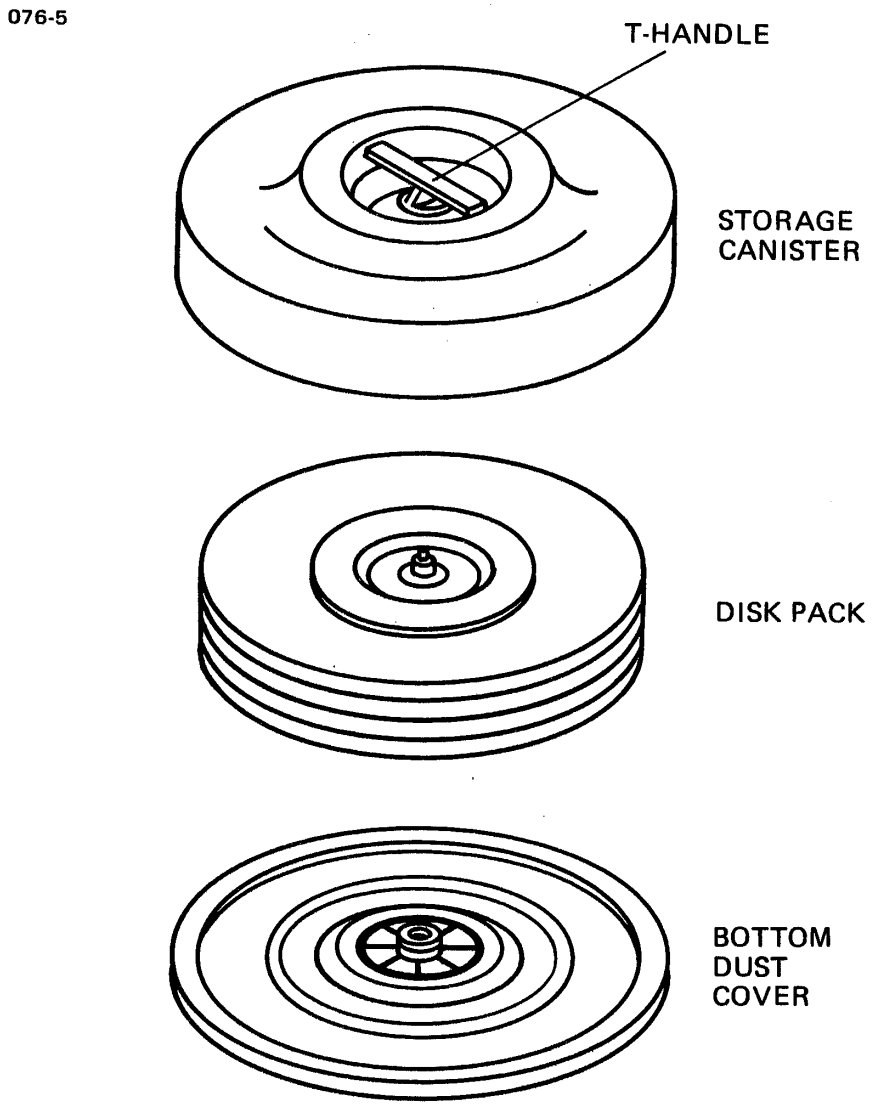


Figure 1-5 MSM80 Disk Pack Storage

6. Remove the storage canister.

7. Close the top cover of the disk drive.

8. Press the start switch.

9. Wait until the ready lamp remains illuminated.

To remove the disk pack:

1. Lift the top cover off the disk drive after the ready indicator lamp goes off. Do not attempt any procedures until the ready indicator lamp of the disk drive remains off. The disk can be damaged if attempts are made while the ready lamp is still blinking.

2. Position the storage canister over the disk.

3. Turn the T-handle counter-clockwise until secure.

4. Remove the disk pack.

5. Fasten the bottom of the dust cover by turning it until it attaches itself to the disk pack.

6. Close the top cover of the disk drive.

COMPUTER SOFTWARE

A general-purpose digital computer operates according to the instructions of a set of routines stored in its memory unit. This sequence of instructions is referred to as a program.

The programs that direct the computer in its operations and in solving applications are commonly known as software. Software is the essential complement to computer hardware. Software is the programs, written in specific computer languages, that the hardware executes.

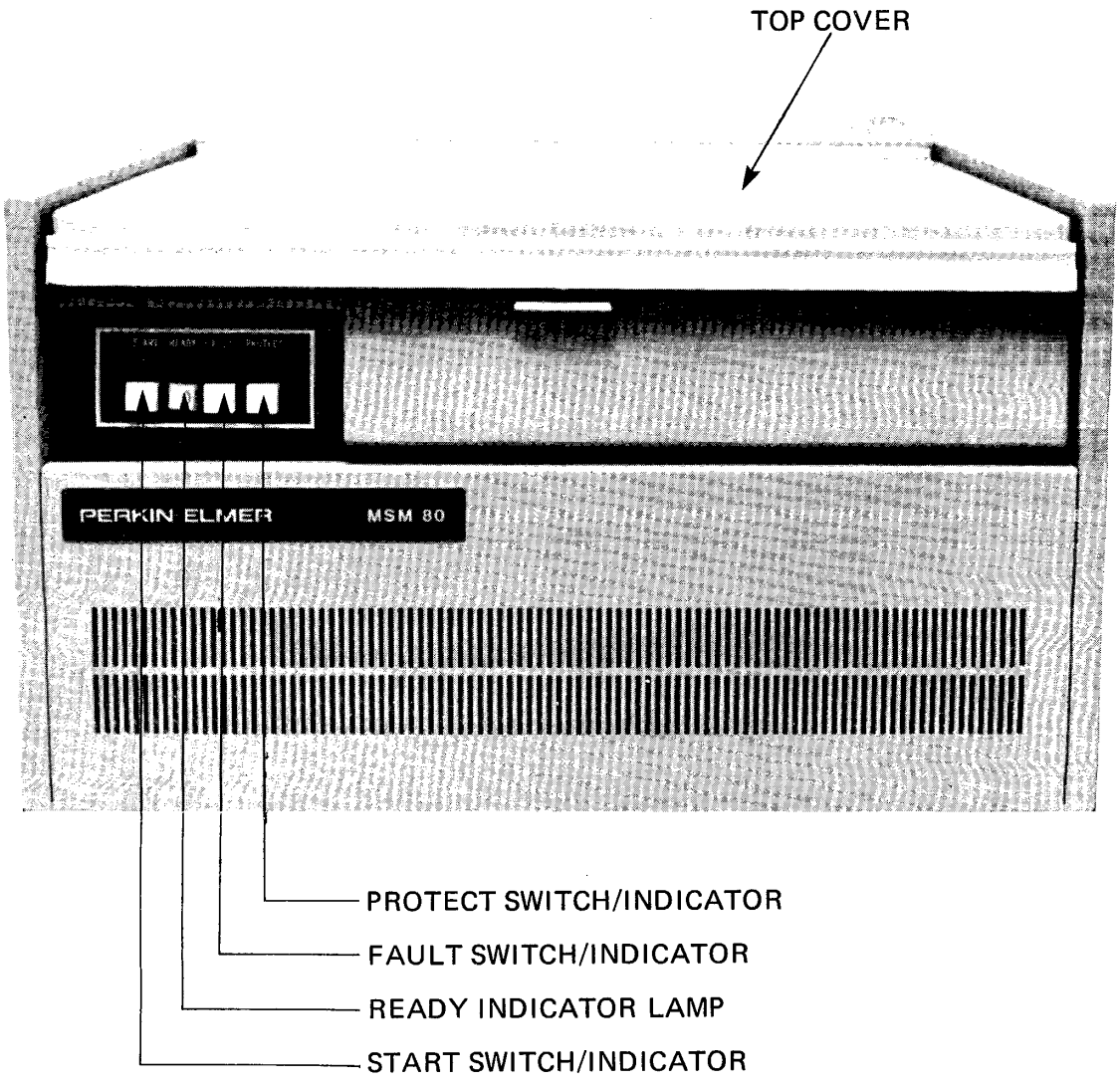


Figure 1-6 MSM80 Disk Drive

SOFTWARE LANGUAGES

The languages that software programs are written in are classified into three basic types:

- Machine languages
- Assembler languages
- High-level languages

Machine Languages

Machine language corresponds to the instructions of the computer. Each machine instruction includes the operation to be performed (opcode) and the locations of the information on which the operation is to be performed. Machine instructions are represented as a hexadecimal shorthand for the binary numbers (1s and 0s) that the computer understands.

Assembler Languages

Due to the inconvenience of machine language, a symbolic way of expressing machine language was developed. This symbolic method is referred to as an assembly language and consists of the representation of machine instructions by mnemonics rather than hexadecimal shorthand. This language enables the programmer to use a combination of letters and numbers to express the opcode.

Since the computer can only understand machine language, an assembly language must be translated into machine language. A program known as the assembler does this translation.

High-Level Languages

High-level languages are computer languages that use symbols and command statements that more closely approximate human thoughts and natural forms of expression. The programmer uses a high-level language, such as FORTRAN, COBOL, Pascal or C to solve the problem at hand.

High-level languages can be considered procedure-oriented; the programmer does not have to be concerned with where and how the operations are to be performed in memory.

A computer cannot directly execute a high-level language. A program called a compiler translates a high-level language into machine-usable code. Each language has its own compiler.

COMPUTER PROGRAMS

Computer software can be classified into three categories:

- System programs
- Utility programs
- Application programs

System Programs

System programs are created to assist and, in part, control the operations of a computer and the execution of other programs. A collection of such programs is known as an operating system. Without these programs, the operator would have to constantly intervene to enter programs, begin executing them and manually record their termination. Thus, the operating system allows the computer to run efficiently with little user intervention. This collection of software improves the efficiency of the computer's resources. We refer to the Perkin-Elmer operating system as OS/32.

Utility Programs

Utility programs are general-purpose routines that perform everyday activities related to administration and housekeeping duties within the system. These programs are most often used by the operator and are controlled by operator commands. The OS/32 Utility programs support features such as input/output (I/O) spooling, disk back-up, and accounting and error reporting. They also perform such tasks as transferring files from tape-to-tape, tape-to-disk, tape-to-printer and preparing direct access storage media for use in subsequent processing. A more detailed description of each utility is presented in a later chapter.

Application Programs

Application programs are created to solve a specific user problem, particularly in business applications. Examples of such programs are payroll, inventory and sales reports. These applications contrast with system programs and utility programs that perform generalized tasks for the benefit of all users of the computer.

CHAPTER 2
THE OS/32 OPERATING SYSTEM

INTRODUCTION

For an operating system to be efficient, it must be actively performing tasks as much of the time as possible. The proprietary operating system for Perkin-Elmer Series 3200 computer systems, OS/32, is a real-time multitasking operating system. This means many independent tasks can run concurrently, as long as enough processing, storage and input/output (I/O) devices are available to accommodate them. In this type of operating system, while one task is waiting for an event, such as an input/output (I/O) operation, to finish or a time interval to end, the computer's facilities are made available to other tasks.

OS/32 AS A TASK

The basic unit of executable software within OS/32 is the task. A task is a sequence of operations to be carried out one at a time. OS/32 itself is a task that must be loaded into memory (made resident) before it can execute. Since OS/32 is a task, it can reside on magnetic tape, disk devices or other media. However, the operating system must be loaded via a bootloader program before it can begin execution.

A MULTITASKING ENVIRONMENT

A real-time multitasking operating system is designed to handle the behavior of tasks communicating with each other and also determine the hierarchy of executable tasks.

The operating system is configured to specify which input/output (I/O) devices will be supported and how tasks requesting such support will share them. The operating system allows for secondary storage devices that will be provided for users.

Scheduling of tasks is handled by OS/32 in a user-specified order of priority, using the same central processing unit (CPU). OS/32 executes lower priority (background) tasks when higher priority (foreground) tasks are not using the system and vice versa. Task processing is temporarily suspended to service interrupt requests from input/output (I/O) devices.

Additional environments provided in conjunction with OS/32 are as follows:

- The multi-terminal monitor (MTM) provides a secure multiuser, time-sharing environment for program development.
- Reliance is a complete on-line transaction processing software package for the commercial user of Perkin-Elmer 32-bit computers.
- PENnet is the Perkin-Elmer networking system that allows terminals and computers at a remote site (in another room or building) to communicate with each other and with the controlling (local) computer.

OS/32 provides many facilities to allow complete application environments to be easily constructed.

TASK STATES

A task must be initially loaded via the OS/32 LOAD command, at which time, the loaded task enters the dormant state. When a task is started either by the OS/32 START command or another task, the task is removed from the dormant state and placed in the ready state. An executing task is in the current state. Other tasks that have been started but are lower in priority than the executing task remain in the ready state. The task scheduler initiates execution of the next ready task with the highest priority.

Tasks of equal priority compete for processor run-time. Two types of scheduling are available to initiate execution of the next ready task. They are as follows:

- Strict priority scheduling allows equal tasks to be scheduled on a first-in/first-out (FIFO) basis. A task remains active until it relinquishes control of the processor.

- Time-slice scheduling allows tasks of equal priority to receive equal shares of processor time. When a task uses its segment of run-time, it returns to the ready state to await its next time slice.

If the executing task becomes suspended, either by OS/32, the system operator or another task, the task enters the wait state. For example, a task becomes suspended when it requests a service from OS/32, such as an input/output (I/O) transfer. The suspended task remains in a wait state until the input/output (I/O) transfer is completed, at which time it returns to the ready state.

Figure 2-1 illustrates task states.

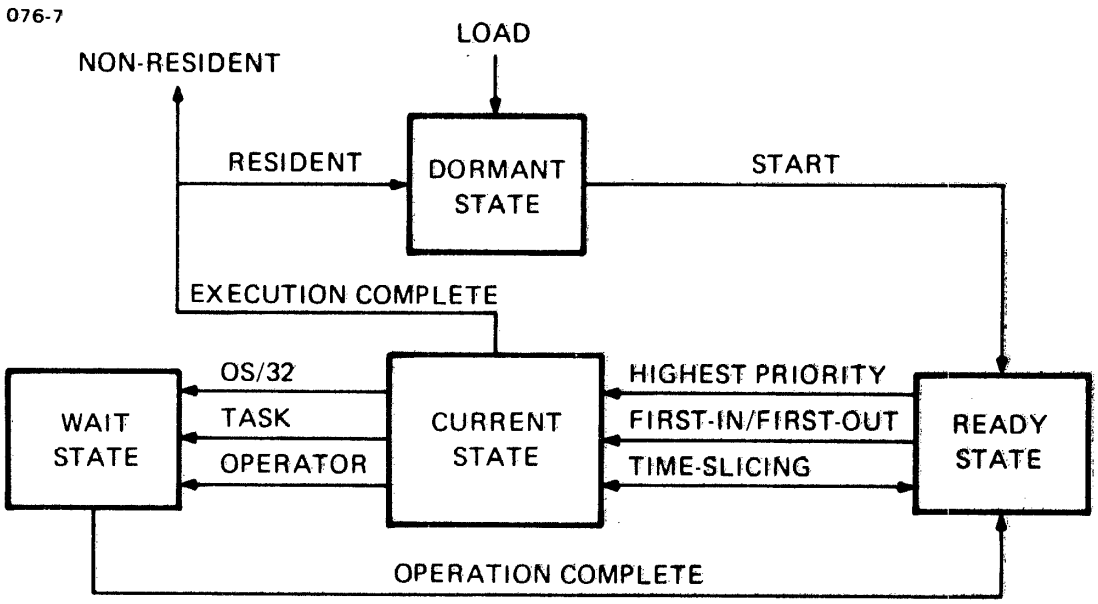


Figure 2-1 Task States

THE OPERATOR IN AN OS/32 ENVIRONMENT

The operating system is controlled by the computer operator through a device called the system console. The operator monitors operations for performance via operator commands and must be able to diagnose errors when malfunctions occur in the software and hardware. Responsibilities also include mounting tapes, loading disks, and cleaning and maintaining these peripherals. A comprehensive command set is provided to allow the operator to control and interact with the various OS/32 environments.

SYSTEM CONSOLE

The computer operator communicates with OS/32 through the system console. The system console is a terminal that has unlimited access to the system. It has a special relationship to the system in that the system receives command inputs from the console and writes system messages to the console. Significant events such as disk failures, power failures and task terminations are logged to the console.

Prompts

There is always a prompt visible at the system console. Prompts are indicators output by interactive software to request instructions from the system console or to ask a question that needs an answer before the software can continue. These prompts take one of the forms listed in Table 2-1.

TABLE 2-1 CONSOLE PROMPTS

PROMPTS	MEANING
*	Command request
taskid>	Data request
.CMDP>	Build request

The command request prompt (*) is output whenever the system is ready to accept another command.

The data request prompt (taskid>) is output whenever a task is attempting to perform a read request to the system console. The task identifier (taskid) of this prompt is the name of the task requesting data. For the background task, the taskid is .BG. This prompt may indicate that a task is waiting for commands or data. The system operator should satisfy the data request as soon as possible, since system messages are held up until the data request is satisfied.

The build request prompt (.CMDP>) is output whenever the command processor task is requesting input. This occurs during the processing of a BUILD command, which is explained in the OS/32 Operator Reference Manual.

BREAK Key

A task can be interrupted in the process of reading from or writing to the system console by depressing the BREAK key (or ESC key on some devices). When the BREAK key is depressed, the operating system is forced into the command mode (* prompt) for the entry of one command line. After the operating system accepts the command line, the input/output (I/O) to the system console proceeds.

FILES

A file is an organized collection of information. A file is analogous to a folder in a filing cabinet in that it is an entity that can be retrieved, opened, processed, closed, moved and placed back into storage as a unit. Every computer file has a unique name by which it is known and referred to by users and by the computer system. A file may contain data or may be a program.

COMMAND SUBSTITUTION SYSTEM (CSS)

The command substitution system (CSS) is an extension to the OS/32 command language. It enables the user to establish files of modifiable commands that can be called from the terminal or other command substitution system (CSS) files. A command substitution system (CSS) file is a series of operator commands created and stored on disks, tapes or other media. In this way, complex operations can be carried out by the terminal user within a few commands.

Most of the tasks that the operator loads will already have the commands to do so set up in a command substitution system (CSS) file. To call a CSS file, enter the name of the CSS file. The operating system treats this as a command. An example of loading a task from a CSS file is shown in Chapter 5.

SYSTEM FAILURE

A system failure occurs when a hardware or software malfunction is detected during execution of the system code. The system cannot proceed without running the risk of destroying information, either on some peripheral device or in memory. The procedures followed after a system failure occurs are discussed in Chapter 6.

This page has been inserted to retain pagination sequence.

CHAPTER 3
POWER-UP/POWER-DOWN PROCEDURE

INTRODUCTION

This chapter describes the basic power-up/power-down procedures for Perkin-Elmer Series 3200 Processors. Depending on the requirement, some installations shut down their system completely at the end of the business day and the operator must "cold start" the system every morning. As the phrase implies, this procedure involves a complete restart of the system, beginning with power-up to booting the operating system.

If the system's hardware is not turned off each night, the operating system is loaded as explained in Chapter 4.

POWER SWITCHES

Each system is furnished with circuit breakers that automatically interrupt electrical current when in the OFF position.

In addition to circuit breakers, system devices and other peripherals have ON, OFF and START switches. Unlike circuit breakers, these switches do not cut primary power; instead, they supply electrical power to a device and enable or disable its ability to function.

LOCATION OF POWER SWITCHES

The following sections describe the location of the circuit breakers and power switches.

Series 3200 Processors

- The circuit breaker is located in the lower right area when viewing the cabinet from the rear.

- The system consolette is located on top of the central processing unit (CPU).

Line Printer

- The circuit breakers are readily visible on the lower right or left side when the front doors of the printer are opened. The location of the circuit breakers varies with the device types.

- The printer is equipped with ON, OFF, STOP and RUN buttons. These buttons are located on the top right side of the printer. Depressing the RUN or STOP buttons, respectively, enables or disables printing.

Cartridge Disk Drive

- When viewing the disk drive from the rear, the circuit breakers become readily visible. Again, the location of the circuit breakers varies with device types.

- Use the START button to spin the disk up and down to speed. When the disk is completely up to speed, the READY light remains lit. To spin the disk down, press the START button. When it is safe to remove the disk, the READY light will go off.

CAUTION

WAIT FOR THE READY LIGHT TO STOP BLINKING BEFORE CONTINUING WITH ANOTHER POWER-UP PROCEDURE.

Magnetic Tape Drive

- The circuit breakers on dual-density units are located on the lower right side viewing the magnetic tape drive from the rear. The circuit breakers on vacuum-type units are available through internal access.

- The ON/OFF button is located on the front left side of the device.

Perkin-Elmer Model 1100, 1200, 1250 and 1251 Terminals

- The ON/OFF rocker button is located on the lower rear panel of the machine. When this button is placed in the ON position, the terminal emits a beep, if equipped with a bell. This indicates the terminal is ready to be used when the operating system is brought up.

Perkin-Elmer Model 550 and 6100 Video Display Units (VDUs)

- The ON/OFF rocker button is located on the front right panel of the display unit. When this button is placed in the ON position, the terminal is ready to be used when the operating system has been brought up.

POWER-UP PROCEDURE

To power-up Series 3200 Systems, use the following procedure:

1. Place the processor circuit breakers in the ON position. For the Model 3200MPS System, make sure the circuit breakers are in the ON position for each of the auxiliary processing units (APUs).

2. Place the printer circuit breakers in the ON position. Press the ON and RUN buttons.

3. Place all terminal ON/OFF rocker switches in the ON position.

4. To prevent initial program load (IPL) on power-up, place the initial program load (IPL) switch in the DISABLE position. See Figure 4-1 to find the location of this switch on the consolette.

5. Turn the key-operated security lock from the STANDBY to the ON position. See Figure 4-1 for the location of this key on the consolette.

6. The power lamp (POWER) and the fault lamp (FAULT) become illuminated. If no errors are detected during the initial microdiagnostic memory test, the fault lamp is turned off. See Figure 4-1 for the location of these lamps.

7. Place the disk and magnetic tape drive circuit breakers in the ON position.

8. Spin the disk up-to-speed by pressing the START button. Wait for the blinking READY light to remain lit.

POWER-DOWN PROCEDURE

To successfully power-down Series 3200 Systems, you must cancel all tasks and mark off all disk drives. The MARK command is detailed in Chapter 4. When powering down a system, the general rule to follow is to power-down devices in their order of significance (least to most). Use the following procedure:

1. Place all terminal ON/OFF rocker switches in the OFF position.

2. Press the STOP button, followed by the OFF button, on the line printer and place the circuit breakers in the OFF position.

3. Spin the disk down by depressing the START switch. Do not continue until the READY lamp goes out.

4. Place the disk and magnetic tape drive circuit breakers in the OFF position.

5. Position the LOCK/ON/STANDBY key in the STANDBY position.

6. Place the processor's circuit breakers in the OFF position.

This page has been inserted to retain pagination sequence.

CHAPTER 4
LOADING THE OPERATING SYSTEM

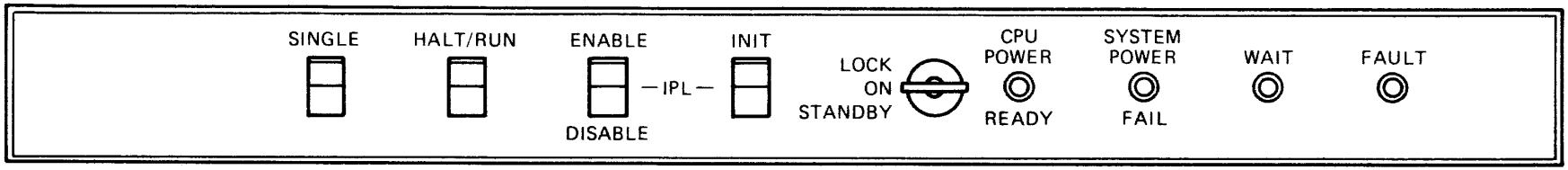
INTRODUCTION

The operating system is stored in object or image format on an auxiliary device and can be loaded into memory via the loader storage unit (LSU) Bootloader program. The following sections describe the procedure used to load the OS/32 image format from disk or magnetic tape.

SYSTEM CONSOLETTE INDICATORS

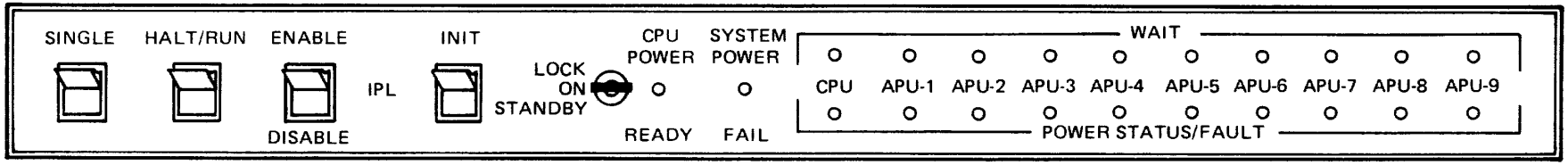
The system consolette controls power to the processor and the initial program loader (IPL). Its light-emitting diodes (LEDs) indicate the current state of the processor. To load the operating system you must understand the functions of the switches on the consolette. The following sections describe the various functions of the key-operated security lock and the control switches. Part A of Figure 4-1 illustrates the consolette for all Perkin-Elmer Series 3200 models except the Model 3200MPS System. Part B of this figure illustrates the consolette for the Model 3200MPS System. This system may contain as many as nine auxiliary processing units (APUs) and a central processing unit (CPU); therefore, the consolette must display this feature.

076-8



PART A

076-9



PART B

Figure 4-1 System Consolette

Key-Operated Security Lock

The key-operated security lock is a three-position (LOCK/ON/STANDBY) switch that allows you to control primary power to the processor. With this lock switch you can also disable (LOCK) the initialize and console switches, preventing any accidental manual input to the processor. If the lock switch is in the ON or LOCK position, the power indicator lamp (POWER) is on.

Control Switches

Primary AC power is applied to the processor when the key-operated lock is in the ON position. All of the control switches are enabled except for the initial program loader (IPL) switch. The four control switches are:

- The SINGLE step control switch. This switch allows you to step through a program one instruction at a time. When this rocker-type switch is in the UP position, the processor relinquishes control to the system console support routine. To execute the next instruction, depress the HALT/RUN switch or type a less than symbol (<) on the system console. To resume normal program execution, place the SINGLE step switch in the DOWN position and depress HALT/RUN or type the less than character (<) on the system console.

- The HALT/RUN switch. This is a momentary contact switch. When this rocker-type switch is depressed, program execution is prevented or resumed, depending on whether or not the processor is running. When program execution is halted, the processor gives control to the system console support routine. This routine examines or modifies memory and restarts program execution. If the processor is already in this routine, program execution is continued.

- The initial program loader (IPL) switch allows you to load a starter system. When you put the security lock in the LOCK position, the initial program loader (IPL) switch is disabled. With the IPL switch in the ENABLE position, the loader storage unit (LSU) or relocating loader (REL) can be loaded after any of the following steps:

  1. Turn the security lock from the STANDBY to the ON position.

  2. Press the INIT switch.

  3. Return AC power to the processor.

- The INIT control switch. This switch initializes the processor. The initialization sequence clears all device controllers and resets certain processor functions. When you press this switch, the fault lamp (FAULT) lights. This lamp goes out at the end of the initialization sequence.

NOTE

If the key-operated security lock is in the LOCK position, HALT/RUN, SINGLE and INIT are disabled.

LOADING OS/32 USING THE 2KB LOADER STORAGE UNIT (LSU)

To load OS/32 image format from disk or magnetic tape using the 2kb loader storage unit (LSU), follow this procedure:

1. Mount the media containing OS/32. If OS/32 is on a disk device, press the START switch and wait for the READY light to come on. If OS/32 is on magnetic tape, position the tape at load point.

2. Turn the LOCK/ON/STANDBY key to the ON position.

3. Press the initial program loader (IPL) ENABLE/DISABLE switch to ENABLE.

4. Initialize and load the 2kb loader storage unit (LSU) BOOTLOADER program by pressing the INIT switch.

5. Place the ENABLE/DISABLE switch to DISABLE and turn the key to the LOCK position. This prevents inadvertent disruption of the booting procedure.

After the loader storage unit (LSU) program is initialized and loaded, the following is displayed on the system console:

```
3200 LSU LOADER Rnn-uu

DEVS
MG85
MGC5
DS5R
DS5F
DS67
D256
FLPY
OTHR
```

Where:

nn	is a decimal number indicating the revision level.
uu	is a decimal number indicating the update level.

The following is referred to as a menu. These mnemonics and the words DEVS (devices) and OTHR (other) are displayed on the system console.

DEVS
MG85
MGC5
DS5R
DS5F
DS67
D256
FLPY
OTHR

After the menu is displayed on the system console, the loader storage unit (LSU) responds with the following examples of prompt. The representation of the examples is such that all responses to prompts are shown in lower-case letters to indicate that they are to be specified by the user. In a realistic setting, all user-specified responses must be entered using upper-case letters for execution by the loader storage unit (LSU) program. If an error is made when entering a response to the prompt examples, press the BREAK or RETURN key to discontinue the response. This action causes the previously referenced menu to be redisplayed on the system console.

After selecting one of the previously referenced menu options, the loader storage unit (LSU) program responds with the following.

Example:

Prompt: DEVICE=

Reponse: xxxx

Where:

xxxx	is a 1- to 4-character string indicating the name of the device containing OS/32.

If the user-specified device mnemonic was not displayed on the system console when the loader storage unit (LSU) was initialized, enter the letters OTHR. See Table 4-1 for a list of device descriptions.

TABLE 4-1 2KB LOADER STORAGE UNIT (LSU) SUPPORTED DEVICE NAMES AND CONFIGURATIONS

DEVICE NAME	DEVICE DESCRIPTION	DEVICE ADDRESS	DEVICE CODE	CON-TROLLER ADDRESS	SELCH ADDRESS
MG85	800 bits per inch (bpi) mag tape	85	40	N/A	F0
MG85	1600 bpi mag tape	C5	41	N/A	F0
DS5F	5Mb disk - fixed	C7	32	B6	F0
DS5R	5Mb disk - removable	C6	33	B6	F0
DS67	67Mb disk	FC	35	FB	F0
D256	256Mb disk	FC	36	FB	F0
FLPY	Floppy disk	C1	37	0	N/A

If a magnetic tape device is specified as the response, the tape is automatically rewound, and the following prompt is displayed on the system console.

Example:

Prompt: FILEMARKS=

Response: xxx

Where:

xxx	is a 1- to 3-digit decimal number from 0 through 255 indicating the number of filemarks to skip to reach the OS/32 file.

When the number of filemarks is entered, followed by a carriage return (CR), the loader storage unit (LSU) loads OS/32 from the file beginning after the last filemark skipped.

If a disk device is specified as the response, the following prompt is displayed on the system console with the volume name of the disk:

Example:

Prompt: VOL=vvvv,FILE=

Where:

vvvv	is a 1- to 4-character volume name of the disk containing OS/32.

The BOOTLOADER program reads and displays the existing volume name to the system console.

Response: filename.ext

Where:

filename.ext	is the OS/32 filename and extension.

After the OS/32 filename and extension are entered, followed by a carriage return (CR), the loader storage unit (LSU) loads OS/32.

If the letters OTHR are specified in response to the DEVICE= prompt, the following prompts are displayed on the system console one at a time:

Example:

Prompt: DEV#=

Response: nnn

Where:

nnn	is a 1- to 3-digit hexadecimal number indicating the device address of the device containing OS/32.

Example:

Prompt: CODE=

Response: cc

Where:

cc	is a 2-digit hexadecimal number indicating the device code of the device containing OS/32.

See Table 4-1 for a list of device codes.

Example:

Prompt: CTLR=

Response: mmm

Where:

mmm	represents a 1- to 3-digit hexadecimal number indicating the controller address of the disk device containing OS/32.

This prompt is not displayed if the device code specifies a magnetic tape device.

Example:

Prompt: SLCH=

Response: sss

Where:

sss	represents a 1- to 3-digit hexadecimal number indicating the selector channel (SELCH) address of the magnetic tape or disk device containing OS/32.

This prompt is not displayed if the device code specifies a floppy disk device.

Example:

Prompt: DRV#=

Response: n

Where:

n	is a decimal number indicating the spindle containing the floppy disk from which OS/32 is to be loaded.

This prompt is displayed only if the device code indicates a floppy disk device.

2KB LOADER STORAGE UNIT (LSU) MESSAGES

The messages listed in Table 4-2 can be generated if errors occur using the 2kb loader storage unit (LSU). These messages will appear on the system console.

TABLE 4-2 LOADER STORAGE UNIT (LSU) LOADER MESSAGES

MESSAGE	MEANING
FILE NOT FOUND	Indicates that the user-specified file descriptor (fd) does not exist or is not contiguous.
IOERROR CNFG	Indicates that an input/output (I/O) error occurred because the hardware configurations do not correspond with that specified by the user.
IOERROR DU	Indicates that the user-specified device is unavailable.

TABLE 4-2 LOADER STORAGE UNIT (LSU)
LOADER MESSAGES (Continued)

MESSAGE	MEANING
IOERROR UNRE	Indicates that the input/output (I/O) error that occurred is unrecoverable.
MEMTST ERR nnnnnn	Indicates that an error occurred during a memory test before OS/32 was loaded. The faulting memory location is at nnnnnn; load OS/32 with a size 2,652 bytes less than location nnnnnn to allow more room for the loader; otherwise, repair memory.

LOADING OS/32 USING THE 8KB LOADER STORAGE UNIT (LSU)

To load OS/32 image format from disk or magnetic tape using the 8kb loader storage unit (LSU) BOOTLOADER program, follow this procedure:

1. Mount the media containing OS/32. If OS/32 is on a disk device, press the START switch and wait for the READY light to come on. If OS/32 is on magnetic tape, position the tape at load point.

2. Turn the LOCK/ON/STANDBY key to the ON position.

3. Place the initial program loader (IPL) ENABLE/DISABLE switch in the ENABLE position.

4. Initialize and load the 8kb loader storage unit (LSU) BOOTLOADER program by pressing the INIT switch.

5. Place the ENABLE/DISABLE switch to DISABLE position and turn the key to the LOCK position.

After the program is initialized and loaded, the following is displayed on the system console:

```
BASIC TEST COMPLETE
3200 8KB LSU BOOTLOADER R04
DEVICE
MG85
MGC5
MG62
DS5R
DS5F
DS67
D256
MM68
MM01
MM67
C13R
C13F
C40F
C67F
FLPY
D300
D19R
D19F
OTHR
DEVICE=
```

After the loader storage unit (LSU) program is loaded, processor capabilities are automatically tested by the basic confidence test. If this test passes, the following message is displayed:

```
BASIC TEST COMPLETE
```

If this test fails, the following messages can be generated by the 8kb loader storage unit (LSU):

CHECKSUM ERROR

CANNOT CLEAR PSW

EPSR ARGUMENT 00000001, L FLAG NOT SET

EPSR ARGUMENT 00000001, ALSO SET C, V, OR G

EPSR DOESN'T PROPERLY UNLOAD PSW

LA---RX1 FAILURE

LA---RX2 FAILURE

THI---CC NOT CLEAR

THI---1ST OP REGISTER CHANGED

THI---CC NOT SET

OR---INCORRECT RESULT

NR---INCORRECT RESULT

SR---INCORRECT RESULT

EXBR---INCORRECT RESULT

SRLS---INCORRECT RESULT

MHR---INCORRECT RESULT

DH---INCORRECT RESULT

8KB LOADER STORAGE UNIT (LSU) MESSAGES

The messages listed in Table 4-3 can be generated if errors occur using the 8kb loader storage unit (LSU).

TABLE 4-3 8KB LOADER STORAGE UNIT (LSU) LOADER MESSAGES

MESSAGE	MEANING
IOERROR	Indicates that the user-specified device is unavailable.
IOERROR UNRE	Indicates that an input/output (I/O) error occurred and is unrecoverable.
IOERROR CONFG	Indicates that the hardware configuration does not correspond with that specified by the user.
FILE NOT FOUND	Indicates that the file descriptor (fd) does not exist or the file is not contiguous.
MEMTST ERR nnnnnn	Indicates that an error occurred during memory test before OS/32 was loaded; the faulting memory location is at nnnnnn.
DISC NOT INITIALIZED	Disk pack is not properly initialized.

After the basic confidence test message is displayed, the following program ID is displayed:

3200 8KB LSU BOOTLOADER Rnn-uu

Where:

nn	is a decimal number indicating the revision level.
uu	is a decimal number indicating the update level.

The loader storage unit (LSU) program now responds with:

Example:

Prompt: DEVICE=

Response: dddd

Where:

dddd	is a 4-character string indicating one of the supported devices listed in the 8kb loader storage unit (LSU) device menu.

Configurations for these supported devices are automatically represented in the system and are listed in Table 4-4. If a unique device configuration that differs from the configurations presented in Table 4-4 is desired, enter OTHR to the DEVICE= prompt.

If an error is made when entering a response, a correction can be made in one of the following three ways:

1. Press the BACKSPACE key or CTRL-H keys to bring the cursor to the character in error.

2. Press the CTRL-X keys to delete the entire line.

3. Press the BREAK key to restart the program.

TABLE 4-4 8KB LOADER STORAGE UNIT (LSU) SUPPORTED DEVICE NAMES AND CONFIGURATIONS

DEVICE NAME	DEVICE DESCRIPTION	DEVICE ADDRESS	DEVICE CODE	CON-TROLLER ADDRESS	SELCH ADDRESS
MG85	800 bits per inch (bpi) mag tape	85	40	N/A	F0
MGC5	1600 bpi mag tape	C5	41	N/A	F0
MG62	6250 bpi mag tape	85	44,45, or 46	N/A	F0
DS5R	5Mb disk - removable	C6	33	B6	F0
DS5F	5Mb disk - fixed	C7	32	B6	F0
DS67	67Mb disk	FC	35	FB	F0
D256	256Mb disk	FC	36	FB	F0
MM68	68.7Mb disk - MMD fixed and head per track (HPT)	FC	38	FB	F0
MM01	1.6Mb disk - MMD HPT	FC	39	FB	F0
MM67	67.2Mb disk - MMD fixed	FC	3A	FB	F0
C13R	13.5Mb disk - CMD removable	FC	3B	FB	F0
C13F	13.5Mb disk - CMD fixed	FC	3C	FB	F0
C40F	40.4Mb disk - CMD fixed	FC	3D	FB	F0
C67F	67.3Mb disk - CMD fixed	FC	3E	FB	F0
FLPY	Floppy disk	C1	37	N/A	N/A
D300	Capricorn 300Mb	FC	2C	FB	F0
D19R	CDD50 - 25Mb removable	FC	2A	FB	F0
D19F	CDD50 - 25Mb fixed	FC	2B	FB	F0

N/A = Not applicable

The type of device entered in the DEVICE= prompt determines which of the following four prompt sequences is issued.

- **MAGNETIC TAPE DEVICE**

  If an 8kb loader storage unit (LSU) supported magnetic tape device is specified as a response, the following prompt is displayed on the system console:

  Example:

  Prompt: FILEMARKS=

  Response: xxx

  Where:

xxx	is a 1- to 3-digit decimal number from 0 through 255.

  This number represents the number of filemarks to skip to reach the OS/32 file. When the number of filemarks is entered, followed by a carriage return (CR), the 8kb loader storage unit (LSU) program loads OS/32 from the file beginning after the last filemark skipped.

- **HARD DISK DEVICE**

  If an 8kb loader storage unit (LSU) supported disk device name is entered as a response, the following prompt is displayed on the system console:

  Example:

  Prompt: VOL=vvvv,FILE=

Where:

vvvv	is a 1- to 4-character volume name of the disk containing OS/32.

The BOOTLOADER program reads and displays the existing volume name to the system console.

Response: filename.ext

Where:

filename.ext	is a 1- to 8-character filename, followed by a 1- to 3-character extension.

After filename.ext is entered, followed by a carriage return (CR), the 8kb loader storage unit (LSU) program loads OS/32.

- **FLOPPY DISK DEVICE**

If FLPY is entered as a response, the following prompt is displayed on the system console:

Example:

Prompt: DRV#

Response: n

Where:

n	is a decimal number from 0 through 3 specifying the spindle containing the floppy disk from which OS/32 is to be loaded.

After the spindle number is specified, the following prompt is displayed:

Example:

```
Prompt:  VOL=vvvv,FILE=
```

Where:

vvvv	is a 1- to 4-character volume name of the floppy disk containing OS/32.

The BOOTLOADER program reads and displays the existing volume to the system console.

```
Response:  filename.ext
```

Where:

filename.ext	is a 1- to 8-character filename, followed by a 1- to 3-character extension.

After filename.ext and a carriage return (CR) is entered, the BOOTLOADER program loads OS/32.

- OTHER

A response of OTHR to the DEVICE= prompt allows the user to specify a device address other than that listed in Table 4-4 for a particular device. The following prompt sequence is displayed on the system console:

Example:

Prompt: DEV ADDR=

Response: aaa

Where:

aaa	is a 1- to 3-digit hexadecimal number indicating the address of the device containing OS/32.

Example:

Prompt: DEV CODE=

Response: cc

Where:

cc	is a 2-digit hexadecimal number indicating the code of the device from which the operating system is to be loaded.

The device codes listed in Table 4-4 are used for 8kb loader storage unit (LSU) supported device names. The device code entered indicates the type of device that contains the operating system. The next prompt displayed in the prompt sequence is dependent upon the type of device indicated by the device code entry.

Example:

Prompt: CTLR ADDR=

Response: mmm

Where:

mmm	is a 1- to 3-digit hexadecimal number indicating the controller address for the device from which the operating system is to be loaded.

If the device is a magnetic tape or a floppy disk, this prompt is not displayed.

Example:

Prompt: SLCH ADDR=

Response: sss

Where:

sss	is a 1- to 3-digit hexadecimal number indicating the selector channel (SELCH) address.

If the device is a floppy disk, this prompt is not displayed.

Example:

Prompt: FILEMARKS=

Response: xxx

Where:

xxx	is a 1- to 3-digit decimal number from 0 through 255.

This number represents the number of filemarks to skip to reach OS/32. This prompt is displayed only if the device code entered indicates a magnetic tape device.

Example:

Prompt: DRV#=

Response: n

Where:

n	is the decimal number specifying the spindle containing the floppy disk from which OS/32 is to be loaded.

This prompt is displayed only if the device code indicates a floppy disk device.

SETTING THE DATE AND TIME

One of your first jobs after the operating system is loaded is to set the date and time. The following will appear on the system console:

```
ENTER DATE AND TIME
```

In response, use the SET TIME command to set the system clock.

```
SET TIME 01/14/83,08:10:02
```

In the above example, the date is set as January 14, 1983, and the time as 10 minutes and 2 seconds after 8:00.

If there is a power failure, the system clock is automatically turned on when the system is restarted. However, the date and time at which the power failure occurred will be displayed. It is your responsibility to correct the time, and, if necessary, the date after a power failure has occurred.

After you set the date and time, use the DISPLAY TIME command to check the date and time. Enter D T at the system console and the date and time will be displayed on the console.

MARKING DEVICES ON

If you have loaded OS/32 from a disk device, you must mark the disk on with the MARK command. Enter the following:

```
MARK device mnemonic:,ON

MA D67A:,ON
```

Now that the operating system is loaded and ready to work, it must have some devices attached to it so that you can give it commands to print information and to send and receive messages. All devices have a device number assigned to them at system generation (sysgen) time. System generation (sysgen) is done by the system administrator; you merely have to tell the operating system which device numbers should be marked on.

This page has been inserted to retain pagination sequence.

CHAPTER 5
SPECIAL TASKS THAT REQUIRE OS/32 MAINTENANCE

INTRODUCTION

In a Perkin-Elmer installation there are other support tasks that can be loaded into memory and run under OS/32 control besides the operating system task. The tasks that require OS/32 intervention and maintenance are:

- Multi-Terminal Monitor (MTM) System
- Environmental Control Monitor (ECM/32)
- PENnet Networking System

Think of these tasks as residing in a hierarchical structure with OS/32 as the main task and the other tasks running under this main task. This concept is shown in Figure 5-1.

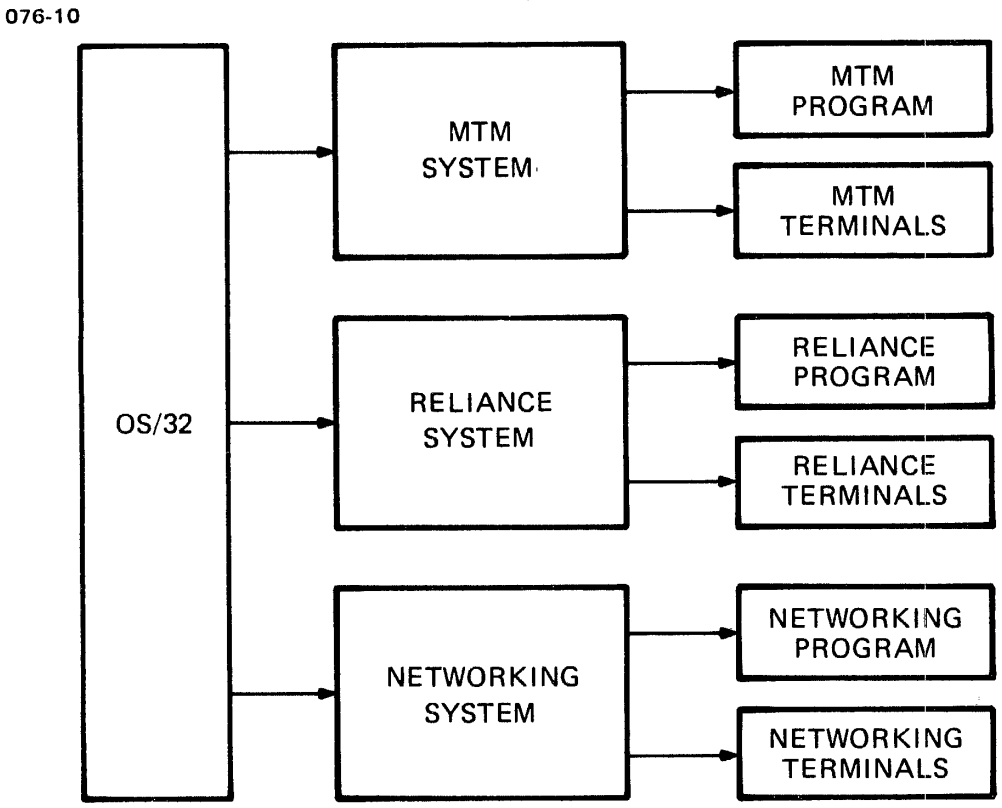


Figure 5-1 Task Hierarchy

MULTI-TERMINAL MONITOR (MTM)

The OS/32 Multi-Terminal Monitor (MTM) adds another dimension to the OS/32 real-time facilities. This dimension is time-sharing. Under the Multi-Terminal Monitor (MTM), up to 64 users can be simultaneously signed on to a Perkin-Elmer system. The Multi-Terminal Monitor (MTM) operates as an executive task (e-task) in an OS/32 environment and can be LOADED, STARTED and CANCELLED without disrupting the execution of other tasks running concurrently under OS/32. Only one Multi-Terminal Monitor (MTM) can be running in the system at a time. One of the main uses of the time-sharing environment is program development. MTM users can create, edit and manipulate their programs, use spooling features and communicate with other users and with the system operator.

The Multi-Terminal Monitor (MTM) requires control from the system console during operation to supervise traffic between the Multi-Terminal Monitor (MTM) and OS/32, and peripheral devices to transmit messages and use spooling features. Multi-Terminal Monitor (MTM) users can also LOAD and run tasks.

Multi-Terminal Monitor (MTM) Mode of Use

Before you LOAD, TASK and START the Multi-Terminal Monitor (MTM), each Multi-Terminal Monitor (MTM) user has to be authorized to use the system and must be assigned a private account number and a password by the operator. Account numbers and passwords are catalogued in a Multi-Terminal Monitor (MTM) file called an authorized user file (AUF). In addition to a number and a password, privileges are also assigned to an account. Once these privileges have been specified, all users signed on to that account are allowed to use those privileges. Privileges include the ability to rename files, change the account password, access the files of other accounts, etc.

To support a diverse community of users, Perkin-Elmer provides the Authorized User Utility. This utility is not for Multi-Terminal Monitor (MTM) users. It is designed as an administrative tool to control and oversee MTM users.

The Authorized User Utility has three modes of operation:

- CREATE
- REPORT
- UPDATE

The CREATE mode allows the system operator to add new accounts. The REPORT mode enables the operator to list an account record and an accounting report for each active account. The UPDATE mode allows an account to be added, deleted, modified or listed.

Loading and Establishing the Multi-Terminal Monitor (MTM)

The Multi-Terminal Monitor (MTM) resides on a secondary storage device. Because of its special nature and relationship to OS/32, MTM must always be given a task identification (taskid) of .MTM. The LOAD and TASK commands load and establish the Multi-Terminal Monitor (MTM) as the current task.

The LOAD command brings a copy of the task file into main memory.

Format:

```
LOAD .MTM,fd
```

Where:

fd	is the file descriptor of the Multi-Terminal Monitor (MTM) task or the name of the device from which MTM is loaded.

The TASK command establishes the Multi-Terminal Monitor (MTM) as the current task.

Format:

```
TASK .MTM
```

The START command starts the Multi-Terminal Monitor (MTM).

Format:

```
START ,AUF=fd
```

Where:

fd	is the file descriptor of the authorized user file (AUF).

The START command offers other parameters that can be issued at the start of MTM. For further discussion of these commands and their parameters, refer to the Multi-Terminal Monitor (MTM) System Planning and Operator Reference Manual or the Multi-Terminal Monitor (MTM) Reference Manual. MTM commands allow the system operator to control MTM, its terminals and its users, from the system console. Most installations have command substitution system (CSS) files set up to facilitate bringing the MTM system up or down. The following illustrates a command substitution system (CSS) to LOAD, TASK and START MTM and to ADD three terminals.

```
LOAD   .MTM,MTMJR
TASK   .MTM
START AUF=MTMJR.AUF
.MTM   ADD CT5A
.MTM   ADD CT7A
.MTM   ADD CT4E
$EXIT
```

This CSS file is named MTMSTART.CSS. To bring up MTM, the operator must type MTMSTART at the system console.

ENVIRONMENTAL CONTROL MONITOR (ECM/32)

Reliance is a system created for commercial users of Perkin-Elmer computers. Reliance offers complete on-line transaction processing and data base management. Reliance operates in a multi-terminal environment under OS/32. The Reliance Environment Control Monitor (ECM/32) provides facilities for users to transfer control of their terminal so they can access MTM, other Reliance environments or remote processors, meaning processors at a location other than that where the user is located. These processors are connected via communication lines and control is transferred without the use of the system console.

Environmental Control Monitor (ECM/32) Mode of Use

ECM/32 is a task that is LOADED, TASKED, STARTED and CANCELLED from the system console. These commands might exist in a command substitution system (CSS) file, in which case it is only necessary to type the name of the CSS file. At that time, it has a group of named terminals assigned to its control. Each terminal displays an ECM/32 identification message and prompts the user to specify which system will be used (for example, MTM or a specified Reliance environment).

It is possible to add terminals to, and remove them, from the group of terminals operating under ECM/32's overall supervisory control while ECM/32 is running. These additions and removals from the ECM/32 terminal group are distinct from the temporary transfers of control achieved by the individual terminal user. It is also possible to obtain a display of the status of all terminals in the group and to send broadcast messages.

Operating Instructions

ECM/32 is loaded from the system console. The LOAD command takes the following form:

```
LOAD task-id,ECM,segment size increment
```

Where:

task-id	is the name to be used for the ECM/32 task.
segment size increment	takes a value determined by the formula (where n is the terminals to be included in the group): $\frac{172*n}{1000}$ (rounded up) kilobytes

Logical unit 3 (lu3) must be assigned before the program is started as follows:

```
ASSIGN 3,ECM.MSG
```

The START command starts ECM/32.

TASK task-id	selects ECM/32 as the current task.
START task-id	starts ECM/32.

Before you can use ECM/32 commands, select ECM/32 as the current task via the TASK command. Each ECM/32 command is prefixed with the OS/32 operator SEND command.

See the ECM/32 Programming and Operations Manual for a complete list and full description of all environment control monitor (ECM) commands.

PENnet (Perkin-Elmer Network)

PENnet allows any two Perkin-Elmer computers to be linked together so that users on one computer can access certain standard facilities on the other. One PENnet system can be linked to any number of other PENnet systems.

Operating Instructions

Before any of the PENnet facilities are available for use, the PENnet software must be loaded and initialized. Terminating PENnet will cause the termination of all network user and network operator facilities at the local system and will unload the PENnet software. Two command substitution system (CSS) files are provided as part of the PENnet package; one for starting and initializing PENnet and one for terminating it. They may only be used at the system console.

Example:

```
PENSTART SYS2
PENSTOP SYS2
```

The PENSTART CSS file ensures that all the required PENnet software is loaded correctly with sufficient memory allocated. Any errors encountered are reported on the system console.

The PENSTOP CSS file ensures an orderly termination of a PENnet system.

For further information on network operator commands and error messages, see the PENnet System Administration Manual.

CHAPTER 6
SYSTEM UTILITIES

INTRODUCTION

Utility programs are general-purpose routines that are used to perform daily system maintenance. Utilities are, in effect, "housekeeping" programs. The following sections discuss the OS/32 System Support Utilities. The utility programs are tasks, and therefore, must be loaded, tasked and started from the system console.

THE SPOOLER

Spooling (simultaneous peripheral operations on-line) offers techniques that permit input to be transcribed from a slow-speed device, like a card reader, to a high-speed data-recording medium, such as a magnetic disk, for subsequent entry into the computer.

Spooling is primarily used for output. Rather than writing records directly to the printer, a relatively slow device, a spooling utility program writes the records to a high-speed disk. From the operator's point of view, spooling provides two major advantages. First, the user need not be concerned with whether or not the printer is busy printing another user's records since the records are written to a disk before they are sent to the printer. Secondly, because a disk can accept records more quickly than a printer, the user perceives that the "printing" of his records was accomplished very rapidly. Later, when the printer is ready and while the user is doing other things, the data is directed from the high-speed device to the printer under the control of the spooler task. In order to accomplish this, the Spooler Utility maintains a spool queue.

The spool queue is a disk file that contains an entry for each file destined for output via the spooler. Each entry contains information relative to the originator and output requirements. The system operator, in turn, exercises a certain amount of control over the spool queue and over the utility itself.

OS/32 Spooler, Perkin-Elmer's first generation spooler, provides basic input/output (I/O) spooling services. For a complete discussion of OS/32 Spooler, see the OS/32 System Support Utilities Reference Manual.

The SPL/32 Spooler offers a more extensive range of features and capabilities than the OS/32 Spooler. See the SPL/32 Spooler System Administration Reference Manual for a complete discussion of SPL/32.

The system administrator determines which spooler will be used on a system at system generation (sysgen) time. Only one spooler can be active on the system at any given time.

BACK-UP

In computer terminology, the word "back-up" refers to a redundant copy of information that is provided by copying the data from on-line storage, such as disk, to off-line storage, such as magnetic tape.

Back-up copies are usually made as a security measure in case the primary file is lost or destroyed due to mechanical or electrical failure. Secondly, back-up is used as an archiving operation by filing old data and releasing the on-line storage space it occupied. The on-line storage can then be used to store data that is currently important rather than data that was important some time in the past. Another use for back-up data is to store information that is important but does not need to be available at a moments notice. This type of data can be stored on an inexpensive medium, such as tape, and restored to a disk (i.e., expensive) device when it is needed.

Perkin-Elmer's OS/32 Backup Utility provides a back-up/restore capability. The Disk Backup Utility provides methods of transferring files from disk-to-disk, disk-to-magnetic tape or tape-to-disk. The procedure to load and start the Disk Backup Utility is detailed in the OS/32 System Support Utilities Reference Manual.

OS/32 Fastback is designed to meet the requirement for a fast disk-to-tape Backup Utility. Fastback copies data directly from the physical input disk device with minimal use of the operating system's overhead. However, tape-to-disk restore operations using Fastback proceed at a somewhat slower rate than back-up operations, because it requires on-line disk work space. OS/32 Fastback is intended to be used for backing up large amounts of data from disk to magnetic tape in situations where the backed-up data rarely needs restoration. The use of Fastback is detailed in the OS/32 Fastback Reference Manual.

ACCOUNTING REPORTING UTILITY

The OS/32 Support Utilities include an accounting function which maintains an accurate and equitable record of machine time usage. The Accounting Reporting Uitility processes accounting data and generates reports or archival files through operator commands. This data pertains to the use of system resources such as disk storage, processor usage and input/output (I/O) on an account basis.

Data collection is performed by OS/32 and the Multi-Terminal Monitor (MTM) through counting, collecting and logging routines, which collect and store accounting information on a disk file. Collected data is stored in subsets according to specific account numbers and time periods.

The Accounting Reporting Utility allows generation of reports and archives through a series of operator commands. These reports summarize the amount of time an individual uses the computer, including the cost factors used to calculate charges for system and disk usage. This utility is documented in the OS/32 System Support Utilities Reference Manual.

ERROR REPORTING UTILITY

The error logger reports memory, input/output (I/O) and system errors to OS/32 and then OS/32 records this information on an error recording file. The Error Reporting Utility produces reports containing diagnostic information for memory errors, input/output (I/O) errors and system errors. The Error Reporting Utility commands allow the operator to:

- specify an error recording file produced by the system error recording routines or a previously created archival file,
- select a subset of the errors to be included in the report by specifying the starting and ending dates of the time period in which those errors occurred,
- build a memory configuration definition file that can be used to interpret memory errors,
- output a summary and an optional itemized list of errors,
- output a memory error report that reports logical addresses as well as physical locations of memory errors, and
- save error logger information on an archival file.

FORMATTING DISKS

When a disk pack is received from a manufacturer, it has no information recorded on it and is referred to as "unformatted". A hardware diagnostic/test program is used to write sector headers on the disk pack and to check the surface of the pack. This process is referred to as "formatting".

Manufacturers do not offer on-line formatting, because the disk pack may be destined for use on any number of machines with an even greater variety of operating systems. Each operating system requires a distinct format for disks. OS/32 assumes disks are already formatted. Formatting a disk on-line may cause overwriting of data; therefore, formatting must be done on a stand-alone system. The term stand-alone describes a computer system operated in an off-line, single user environment.

Formatting is required so that the disk controller hardware can directly address each sector on the disk pack. Thus, loading and executing diagnostic programs to format disks is considered a hardware requirement and is usually performed by engineers and system maintenance personnel.

INITIALIZING DISKS

Preparing a blank disk pack for system use involves two steps: formatting and initializing. Formatting has been discussed as a hardware requirement, whereas initializing is a software requirement. Initialization is required so that OS/32 can allocate, write and read data files on the disk pack. The disk pack is initialized by writing certain control information on the pack. The control information consists of the name of the pack (volume name), a directory that will point to information stored on the pack and a structure used by OS/32 to control allocation of space on the pack - referred to as the bit map.

FASTCHEK UTILITY

The Fastchek Utility is used to perform three different functions: initialize a formatted disk pack, rename a pack and check the integrity of a pack.

During initialization, Fastchek checks and flags any defective sectors on the pack to prevent the operating system from using them.

The rename function is used to change the volume name of a pack that has been previously initialized.

The check function is used to check the integrity of the pack after a system failure. The integrity of the pack is assured, provided all defective sectors are known, the directory contains valid information, all files can be accessed and no files share common sectors.

For details on the Fastchek Utility, see the OS/32 Fastchek Reference Manual.

PANIC DUMP UTILITY

A system failure occurs when a hardware or software malfunction is detected during execution of system code and the system cannot proceed without running the risk of destroying information, either on some peripheral device or in memory. When a system failure occurs, the system displays a crash code on the system console along with the status and location of the failure condition. Also displayed is the name of the module affected by the crash. These messages are used as a diagnostic tool to determine the cause of the system failure.

Panic dumps should be performed immediately following a system failure. The panic dump procedure allows the user to copy the entire contents of memory to a specified magnetic tape or disk device.

When a system crash occurs, the operating system passes control to the Panic Dump Utility and the following prompt is displayed on the system console:

Prompt: DO YOU WANT A SYSTEM DUMP?

Response: Y

The operating system displays the device menu and prompt:

```
Prompt:    DEVICE=
```

If the device is not one of the standard devices listed on the device menu, the response to the prompt is OTHR. If OTHR is the response to the DEVICE= prompt, all necessary data about this device must be given. The following prompts appear on the console and appropriate responses must be supplied:

```
Prompt:  DEV#=

Prompt:  CODE=

Prompt:  CTLR=

Prompt:  SLCH=
```

The operating system writes the information on the desired device, and then the following message appears on the system console:

```
SYSTEM DUMP COMPLETE
```

After a panic dump has been performed, OS/32 must be rebooted and the contents of the dump should be listed to a print device using the Dump Print Utility. See the OS/32 System Support Utilities Reference Manual for instructions regarding the Disk Dump and Disk Print Utilities.

This page is inserted to retain pagination sequence.

GLOSSARY

applications program

A computer program for a given user that solves a specific problem or performs a specific function.

arithmetic logic unit (ALU)

The hardware portion of a computer system where arithmetic and logical operations are performed.

assemble

The act of translating a program written in assembler language into machine language and the assignment of storage for instructions and data.

bug

An error in a software program or in the computer hardware.

capstan

The rotating shaft on a magnetic tape drive that is used to impart uniform motion to the magnetic tape when engaged.

card (punched card)

A data storage medium made of special quality paper stock. It can be processed (read) by a card reader.

central processing unit (CPU)

The component of a computer system that contains the arithmetic, logic and control circuits of the basic system.

COBOL (Common Business Oriented Language)

A common procedural language designed for commercial data processing.

command

An instruction the user inputs to tell the computer system what to do.

compiler

A program that translates high-level source code into machine language code.

console

An interactive device used for operating system control. It is via this device that the computer operator communicates with the total computer system.

crash

The breakdown of a computer operating system.

data

Numbers, symbols or facts that refer to or describe an object, idea, condition or situation.

data processing

The manipulation of data to achieve desired results.

debug

To locate and correct errors in a computer program or to correct the malfunctions in the computer itself.

device

A general name given to any type of electrical machine (terminal, disk drive, tape drive or printer).

diagnostic program

A machine program or routine that does performance checks or discovers potential or actual malfunctions in a computer system.

direct access

The reading and writing of information at any location within a storage device in a constant amount of time.

disk

A high-speed data storage medium operating as a peripheral under the control of a computer.

display unit

A device that provides a visual representation of data.

dump

To print the contents of a storage medium.

end of filemark

A one character indicator on a magnetic tape designating the end of file.

error message

A system-generated, preprogrammed message sent to indicate that an error has been detected.

file

A collection of related records treated as a unit. Files can be stored on tape, disk or within memory.

file protect ring

A plastic ring which, when placed on a magnetic tape, enables the tape to be written on. With this ring off, the tape can be read, but not written on. It is also known as the "write ring".

flag

A bit of information attached to a character or word to indicate the boundary of the field. A flag provides a signal that often indicates good, bad or questionable data.

flowchart

The graphic representation of a problem in terms of data flow, procedures or methods.

FORTRAN (FORmula TRANslation)

An algebraic-type language geared toward solving scientific problems.

hardware

The mechanical, magnetic and electronic components of a computer.

header

A record containing identifying or explanatory information for a group of records which follow.

high-level language

A computer language that allows programmers to specify problem-solving procedures in a notation more familiar than the computer's machine code. Such programs must be fed into a compiler or interpreter for translation into machine executable code.

input

The process by which information is transferred from auxiliary or external storage into internal storage. Contrast with "output".

input/output (I/O)

Equipment or data that is involved with information transferred into and/or out of the computer.

instruction

A coded step that tells the computer what to do for a single operation in a program.

interblock gap

The physical unwritten space on a magnetic tape separating data blocks.

interrecord gap

The unrecorded portion between records on a magnetic tape.

keyboard

A device used for coding data. When the user presses a specific key, the selected character is generated.

line printer

A device that prints an entire line of data at a time and then advances to the next line. Contrast with a character-at-a-time printer, such as a typewriter.

load

To enter various information from auxiliary storage (disks or tapes) into the internal storage (memory) of a computer.

loader

A software program that loads the user's program along with required system routines into the central processing unit (CPU) for execution.

logical unit

The number assigned to specify a peripheral device or direct-access file where the data transfer or command occurs.

machine code

Instructions expressed in binary form that a computer can process. Machine code is commonly used to describe programs written in octal, hexadecimal or decimal notation. These programs can be keyed directly into memory for execution or translation into binary and stored by a loader program.

main storage (local memory)

Usually the fastest storage device of a computer and the one from which instructions are executed.

memory

An area in the computer where information is stored and retrieved.

mnemonic

An abbreviation or acronym that is easy to remember.

multitasking

A method of achieving concurrence by separating a program(s) into two or more interrelated tasks that share code, buffers and files while running.

off-line

The operation of an input/output (I/O) device not under direct control of the central processing unit (CPU).

on-line

The operation of an input/output (I/O) device as a component of the computer under programmed control.

operating system

An integrated collection of supervisory routines (usually user-transparent) responsible for allocating system resources among user tasks. These routines can include memory management, input/output (I/O) handling, logging, storage assignment, operator interaction and job scheduling.

operational code (op code)

Symbols that designate a basic operation to be performed.

output

When information is transferred from internal to external storage. Contrast to "input".

paper tape

A strip of paper on which data is recorded in groups of holes arranged in binary format along the length of the tape. The tape is divided into columns running the length of the tape and channels extending across the width of the tape.

peripheral device

A machine that operates in conjunction with a computer, but is not physically part of the computer (i.e., tape, disk, printer or terminal).

preprocessor

A computer program that affects preliminary computation or organization.

priority

The sequence in which tasks will be processed.

processor

A device capable of receiving data, manipulating data and supplying results, usually of an internally stored program.

program

A set of instructions which, when executed, performs some function. Programs are often referred to as software.

queue

A line or group of items waiting for the processor's attention.

random access

To obtain data from, or place data into, storage when there is no sequential relation governing the access time to successive storage locations. Also, an attribute of a memory device that allows data to be written into or read from memory through directly locating rather than locating through reference to other data in memory. No search is required; the machine can proceed directly to the desired memory location.

read

To copy, usually from one form of storage to another, particularly from external to internal storage. Contrast with "write".

real-time

The actual time during which a physical process transpires.

record

A collection of data treated as a unit.

response time

The amount of time that elapses between an inquiry generated at a terminal and the response received at the terminal.

rewind

To return a magnetic tape to its initial data point.

routine

An ordered set of instructions that have some general or frequent use.

sector

A triangular section of a disk surface. A block of data is addressed by its track and sector numbers.

segment

A self-contained portion of a program that can be executed without the entire program being maintained in internal storage at any one time.

sequential

Of or relating to events or data arranged in sequence.

software

Computer programs, procedures, rules and possible associated documentation concerned with the operation of a data processing system.

spooling

A technique by which output to slow devices is placed into queues on a mass storage device to await transmission. It allows more efficient system use because programs using slow-speed devices can run to completion quickly and free room for others.

storage device

An electronic machine into which data can be entered, retained and retrieved.

supervisory program

A program, usually part of an operating system, which controls the execution of other programs and regulates work flow in a data processing system.

symbolic language

A programming language that expresses addresses and operation codes of instructions in symbols that are convenient to humans, rather than in machine language.

system

A collection of devices organized to accomplish a set of functions.

system programs

A general class of programs whose purpose is to control a computer system.

tape drive

A device that controls the movement of a magnetic tape drive. It moves the magnetic tape past a read head or a write head or rewinds it.

task

A computer program, or portion of a computer program, that can be specified to the operating system as a unit of work. Tasks compete for system resources and space.

time-sharing

Two or more users concurrently using one computer system with different terminals.

track

The portion of a moving storage device that is accessible to a given read or write head position.

transmit

To move data from one location to another.

utility programs

A computer program that supports the processes of the computer.

wait state

A state the processor assumes when it is not actively processing.

write

To transfer information to an output medium; to copy information usually from internal storage (memory) to external storage (disk, tape, printer or terminal). Contrast with "read".

INDEX