

The user may find the following booklet, prepared by the FCC, to be helpful:

"How to Identify and Resolve Radio-TV Interference Problems".

This booklet, Stock No. 004-000-00345-4, is available from the US Government Printing Office, Washington, D.C. 20402.

*** NOTE ***

Recommended shielded cables must be used to connect between RS-232C devices or parallel printer in order that FCC Class B emission limits are still being satisfied.

Operations/DOS

Reference Guide

Exec. Partner™

Portable Computer

1. THE EXEC. PARTNER SYSTEM FEATURES	1-1
2. OPTIONS AVAILABLE FOR THE EXEC. PARTNER	2-1
3. HOW TO ORDER	3-1
4. PRECAUTIONS	4-1
5. PLANS	5-1
CHAPTER 1	
SETTING UP THE SYSTEM	1-1
1.1. A PRELIMINARY CHECK	1-1
1.2. UNPACKING	1-2
CANADA	
1.3. INSTALLATION	1-3
1.4. CHECKING THE SYSTEM	1-4
OTHERS	
1.5. DISPLAY	1-5
1.6. PRINTING	1-6
1.7. TERMINATION	1-7
1.8. TOP SECRET	1-8

Panasonic

©Copyright Matsushita Electric Industrial Co., Ltd. 1985.

©Copyright Microsoft Corporation 1985.

Exec. Partner™ is a Trademark of Matsushita Electric Industrial Co., Ltd.

USA

Panasonic Industrial Company
Division of Matsushita Electric Corporation of America
One Panasonic Way,
Secaucus, New Jersey 07094

Panasonic Hawaii Inc.
91-238 Kauhi St. Ewa Beach
P.O. Box 774
Honolulu, Hawaii 96808-0774

Panasonic Sales Company
Division of Matsushita Electric of Puerto Rico, Inc.
Ave. 65 De Infanteria, KM 9.7
Victoria Industrial Park
Carolina, Puerto Rico 00630

CANADA

Matsushita Electric of Canada Limited
5770 Ambler Drive, Mississauga,
Ontario L4W 2T3

OTHERS

Matsushita Electric Trading Co., Ltd.
32nd floor, World Trade Center Bldg.,
No. 4-1, Hamamatsu-Cho 2-Chome,
Minato-Ku, Tokyo 105, Japan
Tokyo Branch P.O. Box 18 Trade Center

CONTENTS

CHAPTER 1	
INTRODUCTION	1-1
1. THE EXEC, PARTNER SYSTEM FEATURES ..	1-2
2. OPTIONS AVAILABLE FOR THE EXEC. PARTNER	1-2
3. HOW TO USE THIS MANUAL	1-3
4. PRECAUTIONS	1-6
5. PLASMA DISPLAY	1-7
CHAPTER 2	
SETTING UP THE SYSTEM	2-1
1. A PROCEDURE OF SETTING UP THE UNIT ..	2-3
2. UNPACKING	2-4
Inside the Box	2-4
Setting Up the Exec. Partner	2-5
3. THE EXEC. PARTNER	2-11
Display Panel	2-11
Side Panels	2-12
Top Panel	2-14
Rear Panel	2-14
Connecting the AC Power Cord	2-15
4. THE KEYBOARD	2-16

INTRO

SET UP

5. THE INTERNAL PRINTER	2-17
Thermal Printing	2-20
Thermal Transfer Printing	2-28
6. TRANSPORTING THE EXEC. PARTNER	2-35
7. MEMORY EXPANSION	2-39
8. DIP SWITCH SETTING	2-42
9. OPTION BOARD INSTALLATION	2-45

CHAPTER 3

OPERATING THE SYSTEM

1. KEYBOARD USAGE	3-5
Indicators	3-5
Alphanumeric Keys	3-7
Keys That Can Confuse You	3-8
Control Keys	3-9
Ten Key Pad	3-12
Cursor Control	3-13
Function Keys	3-16
System Restart	3-16
2. PURCHASING DISKS FOR YOUR EXEC. PARTNER	3-17
Disks	3-17
Before Using Disks	3-18
Inserting and Removing Disks	3-20
The Write-Protect Notch	3-24
3. BRINGING UP THE SYSTEM	3-25
The System Disk	3-25
Turning on the Power	3-26
Inserting the System Disk into Drive A	3-27
The Flashing Cursor	3-28

The In Use Indicator	3-28
The Opening Screen	3-29
Prompts	3-30
If the Opening Screen Doesn't Appear	3-30
The Real Time Clock	3-31
Entering the Date	3-31
Entering the Time	3-32
4. USING THE FT-70/FT-80	3-33

FOR USING FT-70

5. BACKING UP THE SYSTEM	3-34
What is a Backup?	3-34
Device Designations	3-35
Default Drive	3-36
The Procedure of making the Backup System Disk with FT-70	3-37
6. CREATING A FILE	3-40
Files	3-40
A File's Name	3-41
Filenames and Extensions	3-42
Extensions	3-44
Drive Specifiers	3-45
Creating a File	3-46
Before Using DOS Commands	3-50
7. GETTING ACQUAINTED WITH DOS:FT-70	3-52
How to use a new disk (The FORMAT Command)	3-52
How to see the list of files (The DIR Command)	3-56
How to copy a file (The COPY Command)	3-61
How to display the contents of a file (The TYPE Command)	3-65
How to change the name of a file (The RENAME Command)	3-67
How to delete a file (The ERASE Command)	3-69
Using DOS	3-71

FOR USING FT-80

8. BACKING UP THE SYSTEM	3-72
What is a Backup?	3-72
Device Designations	3-74
Default Drive	3-74
The Procedure of making the Backup System Disk with FT-80	3-76
9. CREATING A FILE	3-80
Files	3-80
A File's Name	3-81
Filenames and Extensions	3-82
Extensions	3-84
Drive Specifiers	3-85
Creating a File	3-86
Before Using DOS Commands	3-90
10. GETTING ACQUAINTED WITH DOS:FT-80	3-92
How to use a new disk (The FORMAT Command)	3-92
How to see the list of Files (The DIR Command)	3-96
How to copy a File (The COPY Command) ..	3-101
How to display the contents of a File (The TYPE Command)	3-106
How to change the name of a File (The RENAME Command)	3-107
How to delete a File (The ERASE Command)	3-109
Using DOS	3-109
11. SWITCHING THE SYSTEM STATUS	3-110
How to Display 40 Char. or 80 Char.	3-110
How to Switch to FAST Processing Speed ..	3-111
Using the Real Time Clock	3-114
Switching Between the Internal Printer and External Parallel Printer	3-116
12. BASIC	3-118

CHAPTER 4
PRINTER

1. SWITCHING BETWEEN THE INTERNAL PRINTER AND EXTERNAL PRINTER	4-2
The Internal Printer	4-2
Using an External Printer by Connecting it to the Parallel Port	4-3
Using an External Printer by Connecting it to the Serial Port	4-4
2. SPECIAL KEYS FOR THE PRINTER	4-6
Indicator	4-6
Special Keys for the Printer	4-6
3. SETTING THE PRINTER DIP SWITCHES	4-8
How to Set the Printer DIP Switches	4-9
4. PRINTER CONTROL CODES	4-10
5. PRINTER CHARACTER SETS	4-38
Printer Character Set 1	4-38
Printer Character Set 2	4-40

CHAPTER 5
HARD DISK (FT-80 only).....5-1

1. HARD DISK	5-2
Hard Disk Setup Operation	5-2
Formatting	5-7
FDISK Command Function	5-10
Making the DOS Partition (Option 1)	5-12
Changing the Partition Status (Option 2) ..	5-15
Removing the DOS Partition (Option 3) ..	5-17
Displaying Partition Information (Option 4) ..	5-19

CHAPTER 6

USING DOS

DOS COMMAND

1. INTRODUCTION	6-1
2. THE KEYBOARD	6-2
3. FILES	6-6
Naming Files	6-6
Reserved Device Names	6-6
Wildcard Characters	6-7
4. DIRECTORIES	6-8
Directory Structure	6-8
Current Directory	6-10
Paths to Files	6-12
Setting a Path	6-13
Table of Directory Commands	6-13
Directory Commands	6-14

CHAPTER 7

COMMAND REFERENCE

USING DOS

1. INTRODUCTION	7-3
2. TYPES OF DOS COMMANDS	7-4
3. SYNTAX NOTATION	7-5
4. PARAMETERS	7-6
Reserved Device Names	7-7
Wildcard Characters	7-8
5. TABLE OF COMMANDS	7-10

6. COMMAND DESCRIPTIONS

ASSIGN	7-14
BACKUP	7-16
BREAK	7-18
CHDIR	7-20
CHKDSK	7-21
CLS	7-22
COPY	7-24
CTTY	7-25
DATE	7-33
DEL	7-34
DIR	7-35
DISKCOMP	7-36
DISKCOPY	7-40
ERASE	7-43
EXE2BIN	7-46
EXTPRT	7-48
FAST	7-51
FC	7-52
FDISK	7-53
FIND	7-57
FORMAT	7-58
GRAPHICS	7-59
MKDIR	7-63
MODE	7-64
MORE	7-66
PATH	7-72
PRINT	7-73
PROMPT	7-75
RECOVER	7-79
RENAME	7-82
RESTORE	7-84
RMDIR	7-85
SET	7-87
SORT	7-88
SYS	7-91
TIME	7-92
TREE	7-93
TYPE	7-95
VER	7-96
VERIFY	7-97
VOL	7-98

DOS COMMAND

CHAPTER 8

BATCH	BATCH PROCESSING	8-1
	1. INTRODUCTION	8-2
	2. THE AUTOEXEC.BAT FILE	8-4
	3. CREATING A .BAT FILE	8-5
	4. EXECUTING A .BAT FILE	8-7
EDLIN	5. TABLE OF BATCH COMMANDS	8-8
	ECHO	8-9
	FOR	8-11
	GOTO	8-12
	IF	8-13
	PAUSE	8-16
	REM	8-17
	SHIFT	8-18

CHAPTER 9

EDLIN	9-1
1. INTRODUCTION	9-2
2. HOW TO START USING EDLIN	9-3
3. SPECIAL EDITING KEYS	9-5
4. EDLIN COMMAND INFORMATION	9-16
5. PARAMETERS	9-18
6. TABLE OF EDLIN COMMANDS	9-20

7. COMMAND DESCRIPTIONS	9-21
Append	9-21
Copy	9-21
Delete	9-22
Edit	9-24
End	9-26
Insert	9-28
List	9-29
Move	9-32
Page	9-35
Quit	9-36
Replace	9-37
Search	9-38
Transfer	9-41
Write	9-44

CHAPTER 10

LINK	10-1
1. INTRODUCTION	10-2
2. FILES	10-4
Input File Extensions	10-4
Output file Extensions	10-5
VM. TMP (Temporary) File	10-5
3. DEFINITIONS	10-6
Segment	10-6
Class	10-6
Group	10-6
4. COMMANDS	10-7
Prompts	10-7
Switches	10-9
Characters	10-12
5. HOW TO START THE LINKER	10-14
6. SAMPLE LINKER SESSION	10-19

CHAPTER 11

DEBUG	11-1
1. INTRODUCTION	11-2
2. STARTING THE DEBUG PROGRAM	11-3
3. COMMAND PARAMETERS	11-4
4. TABLE OF DEBUG COMMANDS	11-7
5. DEBUG COMMANDS	11-9
Assemble	11-10
Compare	11-13
Dump	11-14
Enter	11-16
Fill	11-18
Go	11-19
Hex	11-21
Input	11-22
Load	11-23
Move	11-25
Name	11-26
Output	11-28
Quit	11-29
Register	11-30
Search	11-32
Trace	11-33
Unassemble	11-34
Write	11-36

DEBUG

CHAPTER 12

INPUT AND OUTPUT OPTIONS	12-1
1. REDIRECTION OF INPUT AND OUTPUT DEVICES	12-2
2. PIPING INPUT AND OUTPUT	12-4
3. FILTERS	12-5

IN OUT

APPENDIX A—User Diagnostics A-1

APPENDIX B—Error Messages B-1

APPENDIX C—Pin Configurations C-1

APPENDIX D—Interrupts and Function Calls D-1

APPENDIX E—Control Blocks E-1

APPENDIX F—Esc. Sequence F-1

APPENDIX G—Installing Popular
Application Software G-1

APPENDIX H—Character Set/Plasma Display H-1

APPENDIX I—Demonstration Software I-1

APPENDIX J—Specifications J-1

APPENDIX K—Index K-1

CHAPTER 1

INTRODUCTION

1. THE EXEC. PARTNER SYSTEM FEATURES 1-2
2. OPTIONS AVAILABLE FOR THE EXEC.
PARTNER 1-2
3. HOW TO USE THIS MANUAL 1-3
4. PRECAUTIONS 1-6
5. PLASMA DISPLAY 1-7

THE EXEC. PARTNER SYSTEM FEATURES

The following features of the Exec. Partner help to make the system more responsive to your needs.

- High Resolution Gas Plasma Display (See page 1-7)
- CPU Clock Frequency is switchable for faster operation from the standard 4.77 MHz to 7.16 MHz. (See page 3-111)
- Internal dual mode printer. (60 cps thermal/40 cps thermal transfer correspondence quality) IBM graphics printer compatible. (See page 2-17, 4-2)
- Built in options: Parallel Printer Port (See page 4-3)
 RS-232C Serial Port (See page C-4)
 Realtime Clock (See page 3-114)
 RAM Expansion (See page 2-39)
 BUS Expansion (See page C-6)

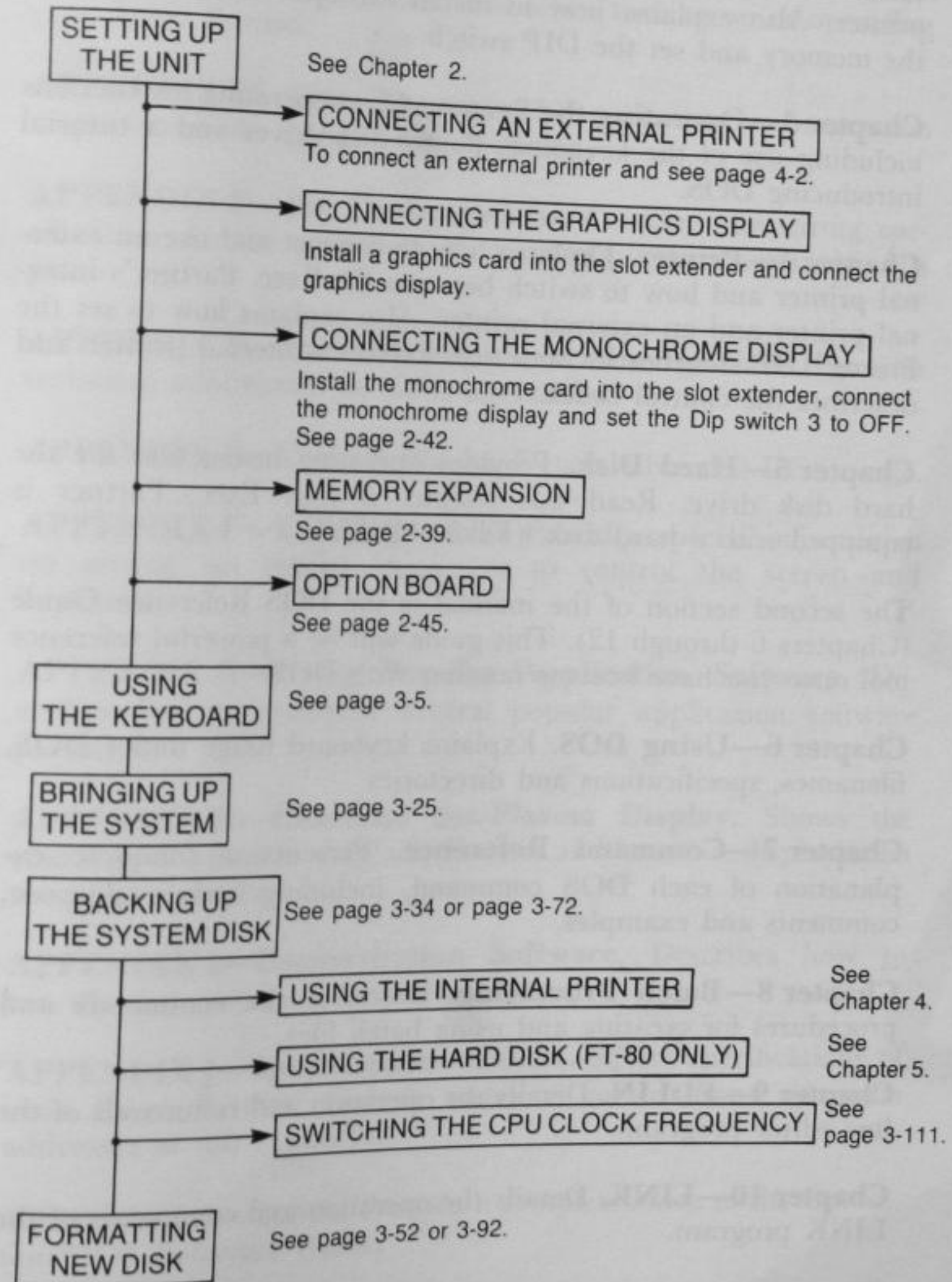
OPTIONS AVAILABLE FOR THE EXEC. PARTNER

Your Exec. Partner is powerful and complete computer system just as you purchased it from the dealer. Panasonic is offering the following options for reorder supplies and further system expansion:

OPTION	MODEL NUMBER
Thermal Paper	RD-9671
Printer Ribbon	CF-PR110
Thermal Transfer Paper (Cut Sheet 8½×11)	RD-9442
Slot Extender (For Bus Expansion)	RD-9655
Replacement DOS/Operations Guide	
Replacement BASIC Manual	

HOW TO USE THIS MANUAL

Basic Exec. Partner operation is discussed in Chapter 1 through 5. An outline is shown below. Follow this flow chart to set up and start your Exec. Partner.



Chapter 1—Introduction. Lists precautions to note when using the Exec. Partner and explains manual organization. Also highlight the major system features.

Chapter 2—Setting Up the System. Explains how to set up the Exec. Partner with printer ribbon and paper for the internal printer. Also explains how to install the option board, expand the memory and set the DIP switch.

Chapter 3—Operating the System. Gives operating instructions including use of the keyboard, floppy disk drives and a tutorial introducing DOS.

Chapter 4—Printer. Explains how to connect and use an external printer and how to switch between the Exec. Partner's internal printer and an external printer. Also explains how to set the Printer DIP switches in the Exec. Partner's internal printer and describes the control codes.

Chapter 5—Hard Disk. Provides operating instruction for the hard disk drive. Read this chapter if your Exec. Partner is equipped with a hard disk. (FT-80 ONLY)

The second section of the manual is the DOS Reference Guide (Chapters 6 through 12). This guide will be a powerful reference tool once you have become familiar with DOS.

Chapter 6—Using DOS. Explains keyboard usage under DOS, filenames, specifications and directories.

Chapter 7—Command Reference. Presents a complete explanation of each DOS command, including syntax, purpose, comments and examples.

Chapter 8—Batch Processing. Describes the commands and procedures for creating and using batch files.

Chapter 9—EDLIN. Details the operation and commands of the line editor program.

Chapter 10—LINK. Details the operation and commands of the LINK program.

Chapter 11—DEBUG. Details the operation and commands of the DEBUG program.

Chapter 12—Input and Output Options. Gives instructions for modifying standard input and output procedures.

Several useful Appendices are located at the back of this manual:

APPENDIX A—User Diagnostics. Helps you troubleshoot your system should problems arise when turning on or operating your Exec. Partner.

APPENDIX B—Error Messages. Lists the error messages you might encounter during DOS operation.

APPENDIX C—Pin Configurations. Contains the wiring specifications for the serial port, the parallel port and expansion bus port.

APPENDIX D—DOS Interrupts and Function Calls. Provides technical information for DOS operations.

APPENDIX E—Control Blocks. Details advanced DOS usage.

APPENDIX F—ESC Sequences. Provides special information on setting up special characters to control the screen and keyboard.

APPENDIX G—Using Popular Application Software. Describes how to configure several popular application software packages.

APPENDIX H—Character Set/Plasma Display. Shows the character set used in the Exec. Partner and how it is displayed on the plasma display.

APPENDIX I—Demonstration Software. Describes how to run the demonstration software.

APPENDIX J—Specifications. Gives complete specifications of your Exec. Partner. Include technical informations on ports addresses in this appendix.

APPENDIX K—Index. Presents a complete index to the Operation/DOS Reference Guide.

PRECAUTIONS

1. Be sure to select the proper power line voltage (120 or 220 VAC) on the power voltage selector.
2. Do not turn off the power switch of the Exec. Partner while the hard disk's red "In Use" indicator is on. Switching off the power switch while the light is on may destroy data on the disk and damage the disk itself. (FT-80 ONLY)
3. Keep the Exec. Partner away from extreme heat such as direct sunlight, heat radiators, or closed vehicles.
4. Never subject the Exec. Partner to sudden shock or extreme vibration. Do not drop it or hit it with other equipment.
5. Keep away from moisture. If the Exec. Partner is suddenly moved from a cold place to a warm place, undesirable moisture may collect inside the Exec. Partner. Avoid sudden temperature changes. If unavoidable, wait a while till the moisture evaporates before using the system.
6. Avoid places where there is high humidity or dust, which may cause damage to the Exec. Partner.
7. When not in use, set the computer down flat on its rubber feet on a flat and stable surface.
8. Do not insert a floppy disk into the floppy disk drive before the Exec. Partner is turned on.

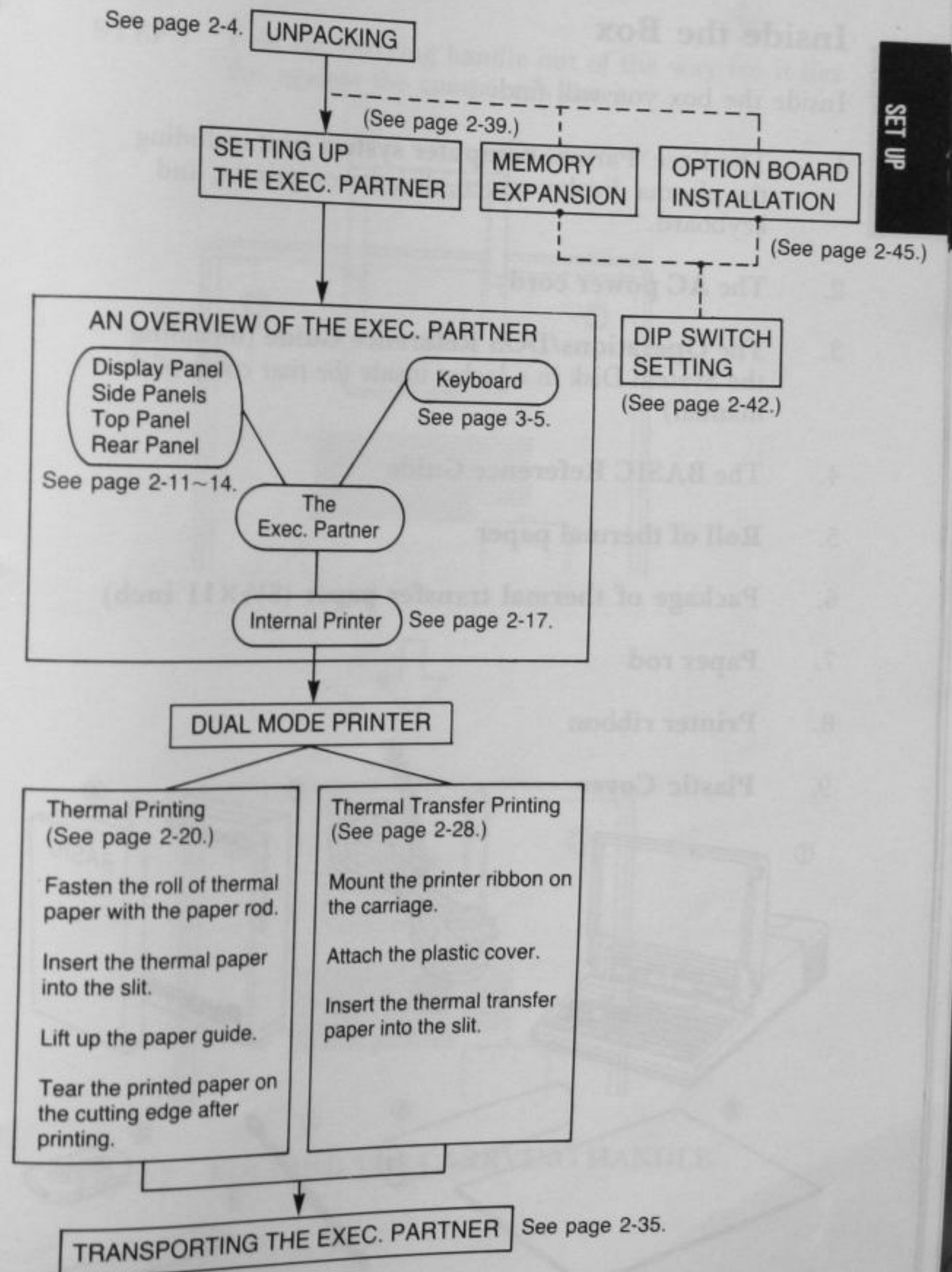
PLASMA DISPLAY

The Exec. Partner's high resolution plasma display is the state of the art in flat panel display technology. The computer's sleek design is made possible through the use of this display. Its neon-orange color is designed to be easy on the eyes, even for extended viewing periods. Some of the plasma displays features include:

- High Resolution 640×400 pixels
- Size is equivalent to 12 inch CRT
- Faster response than CRT & LCD
- Non glare screen
- Flickerless scrolling
- Eliminates CRT ghosting
- Long life (6 times CRTs, 5 times LCDs)
- Light weight

The intensity of the gas plasma display has been set for optional viewing under various lighting conditions and can not be adjusted. Because the plasma screen only displays neon orange characters on a dark background the Exec. Partner will display color text by generating a different character font based on the color. The following table shows the character font selected based on the four components of color (Red, Green, Blue & Intensity).

A PROCEDURE OF SETTING UP THE UNIT



UNPACKING

Inside the Box

Inside the box you will find:

1. The Exec. Partner **computer system unit** including the plasma display screen, disk drives, printer, and keyboard.
2. The **AC power cord**
3. The **Operations/DOS Reference Guide** (including the System Disk in a jacket inside the rear cover of this manual)
4. The **BASIC Reference Guide**
5. **Roll of thermal paper**
6. **Package of thermal transfer paper (8½×11 inch)**
7. **Paper rod**
8. **Printer ribbon**
9. **Plastic Cover**

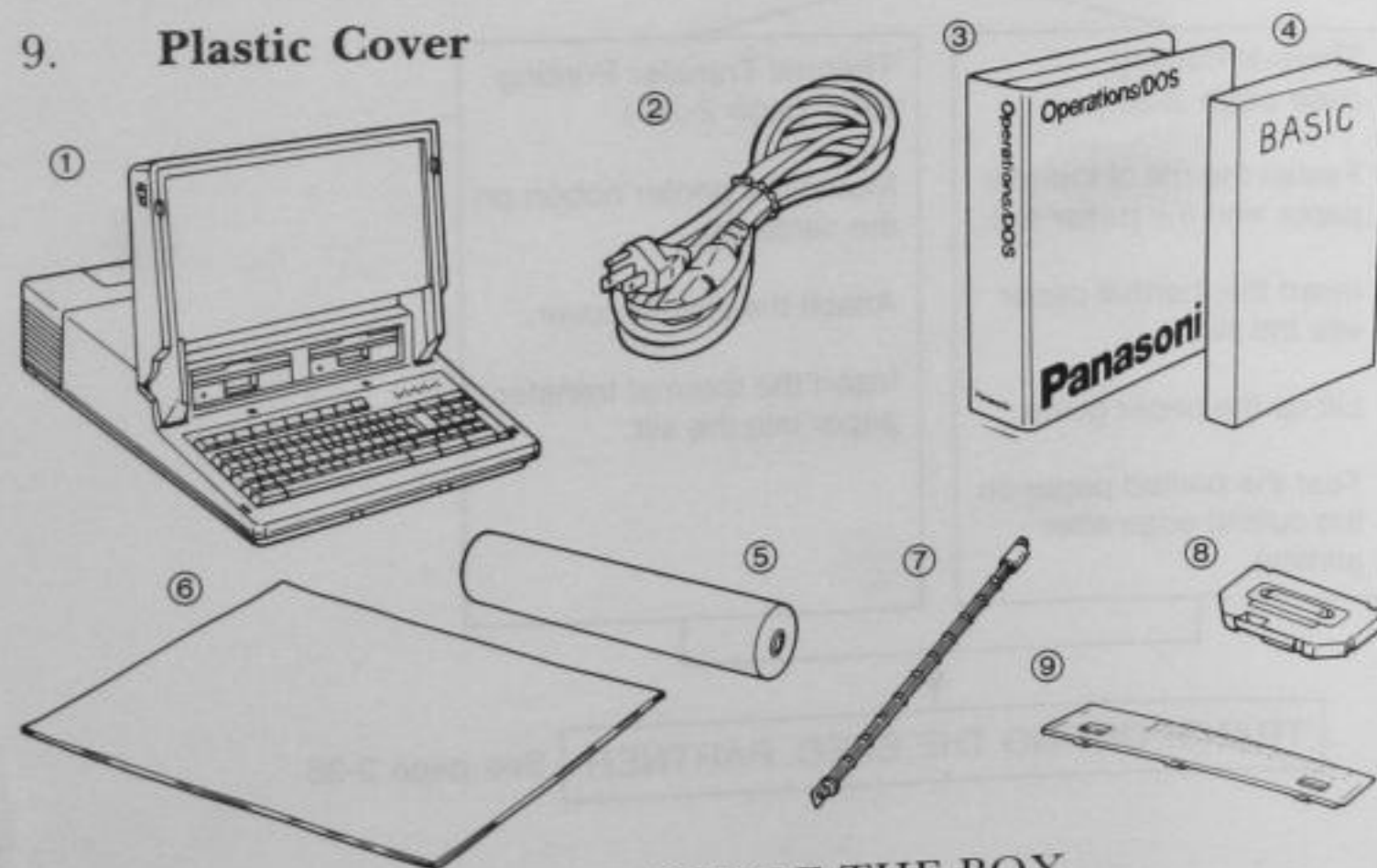


Fig. 2-1. INSIDE THE BOX

Setting Up the Exec. Partner

NOTE: For your convenience, set up the computer within easy reach of an AC outlet.

- STEP 1** Fold the carrying handle out of the way (so it lies flat against the computer's case).

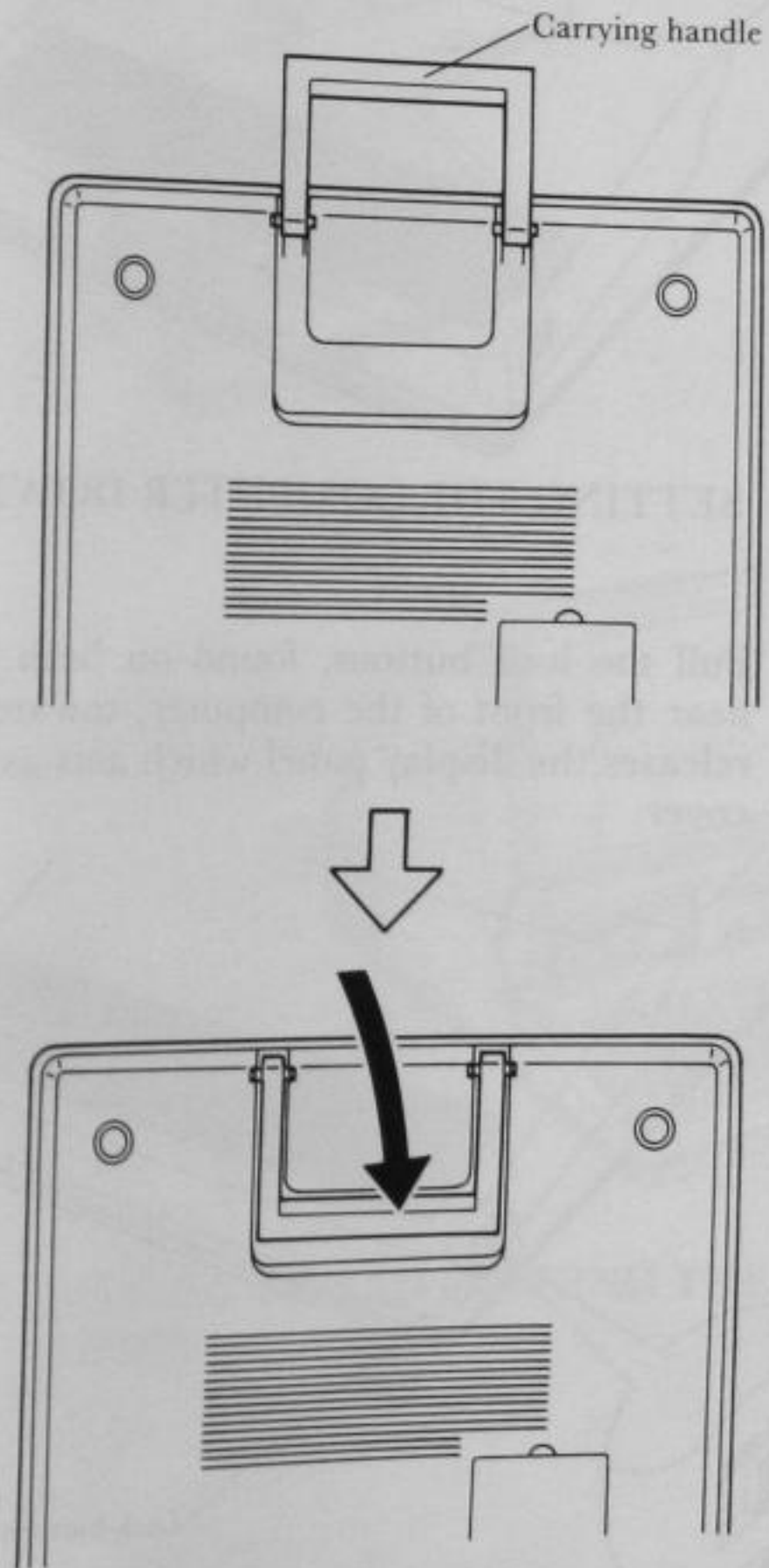


Fig. 2-2. FOLDING THE CARRYING HANDLE

STEP 2 Set the computer down flat on its rubber feet on a flat and stable surface.

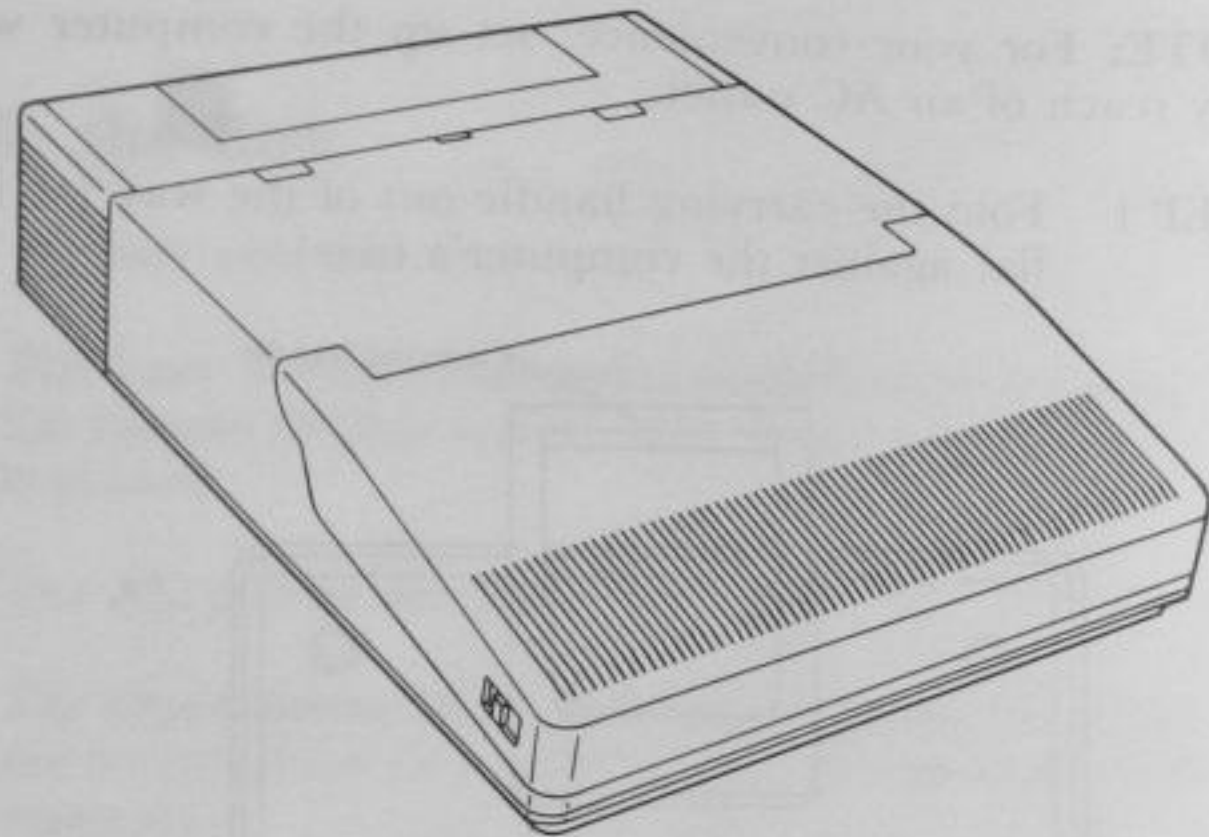


Fig. 2-3. SETTING THE COMPUTER DOWN FLAT

STEP 3 Pull the lock buttons, found on both side panels near the front of the computer, toward you. This releases the display panel which acts as a keyboard cover.

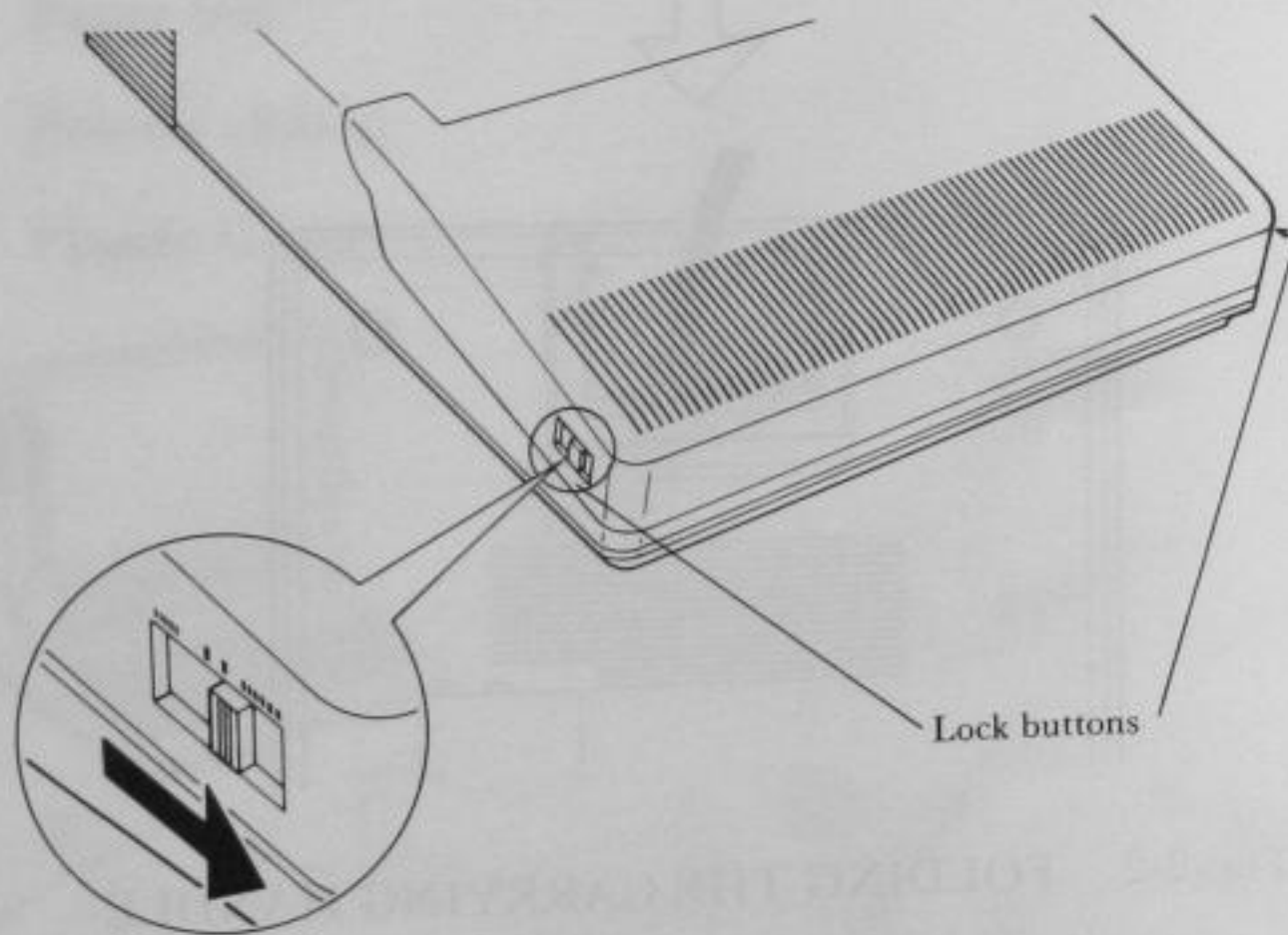


Fig. 2-4. PULLING THE LOCK BUTTONS

STEP 4 Lift up and open the display panel to expose the plasma display screen, disk drives and keyboard. The specially designed hinges allow you to vary the angle of the display panel for optimum viewing.

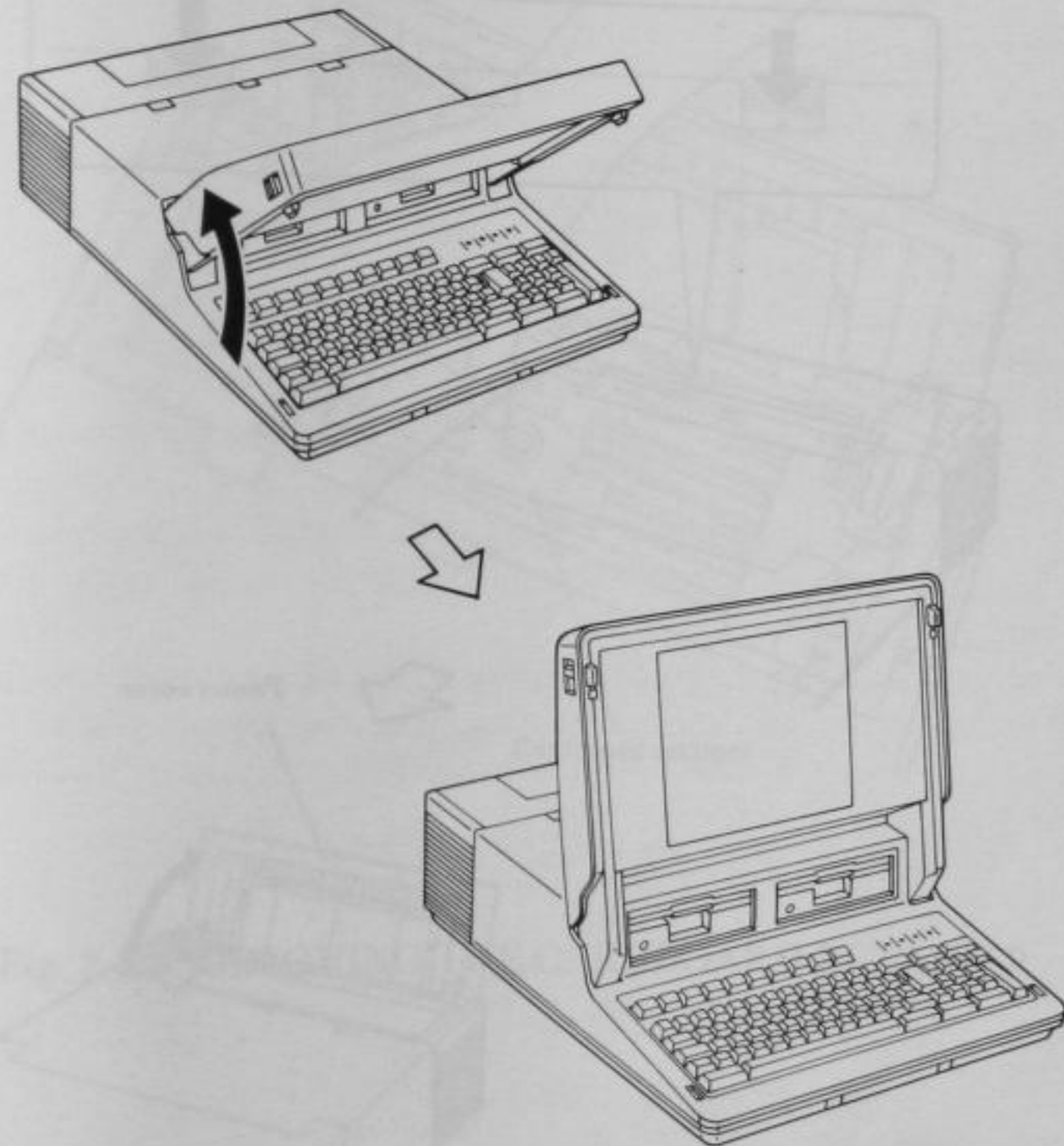


Fig. 2-5. LIFTING UP AND OPENING THE DISPLAY PANEL

STEP 5 The two ribbed-tabs on the top panel are used to lock the printer cover. Push these tabs toward the front of the computer to unlock the printer cover. Flip up and open the printer cover to expose the printer mechanism.

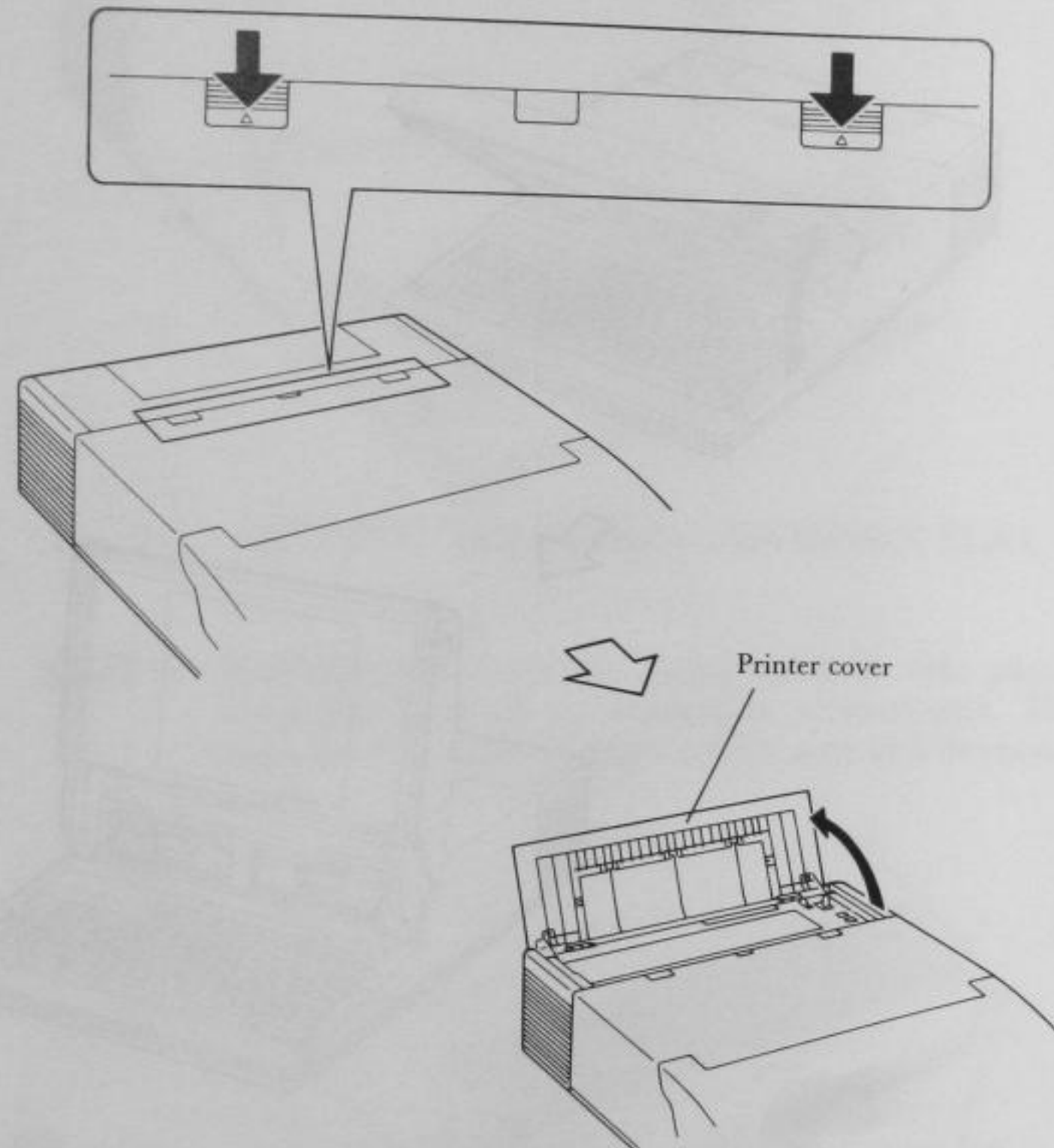


Fig. 2-6. UNLOCKING THE PRINTER COVER

STEP 6 Remove the cardboard retainer from the printer mechanism.

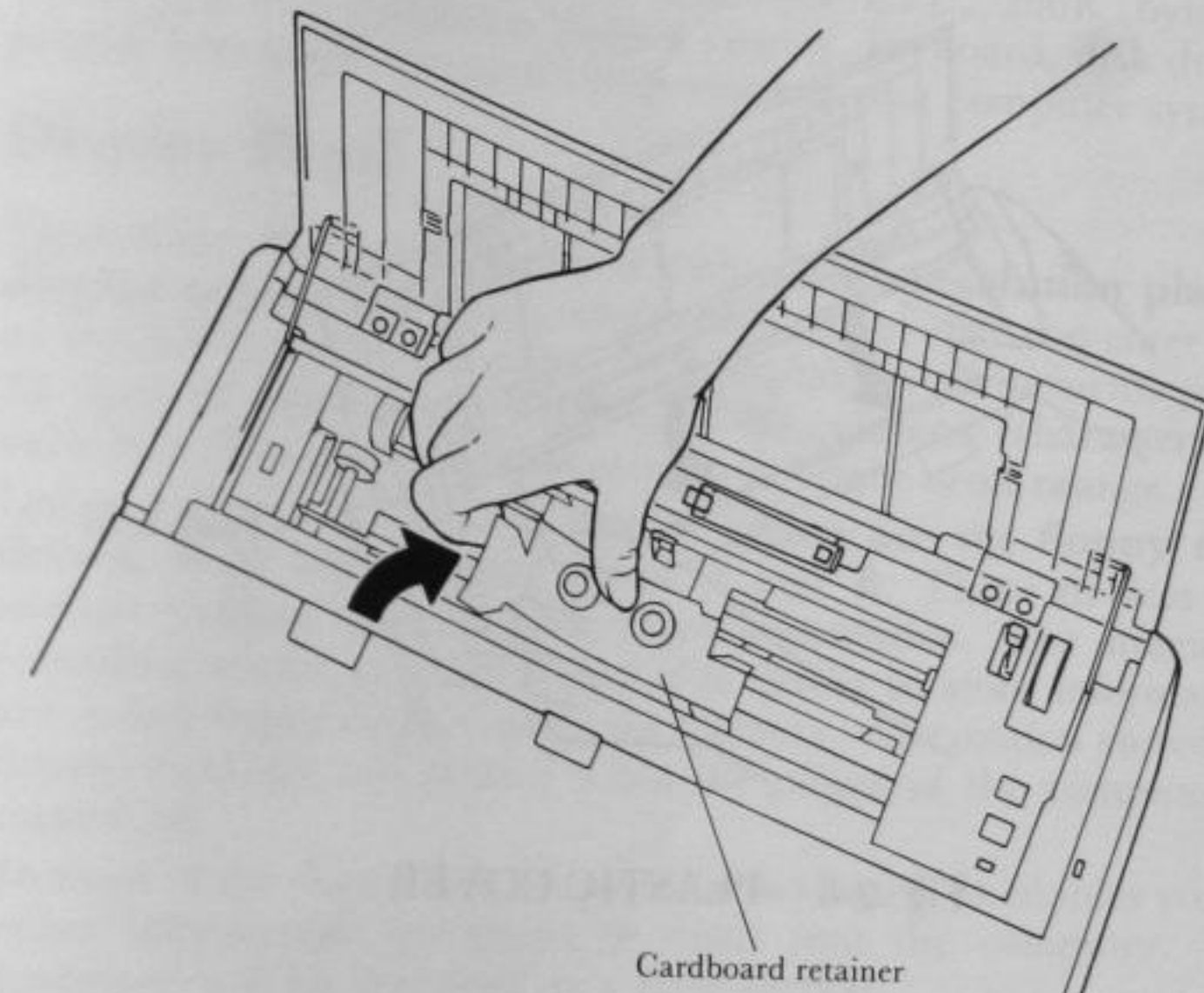


Fig. 2-7. REMOVING THE CARDBOARD RETAINER

STEP 7 Attach the plastic cover.

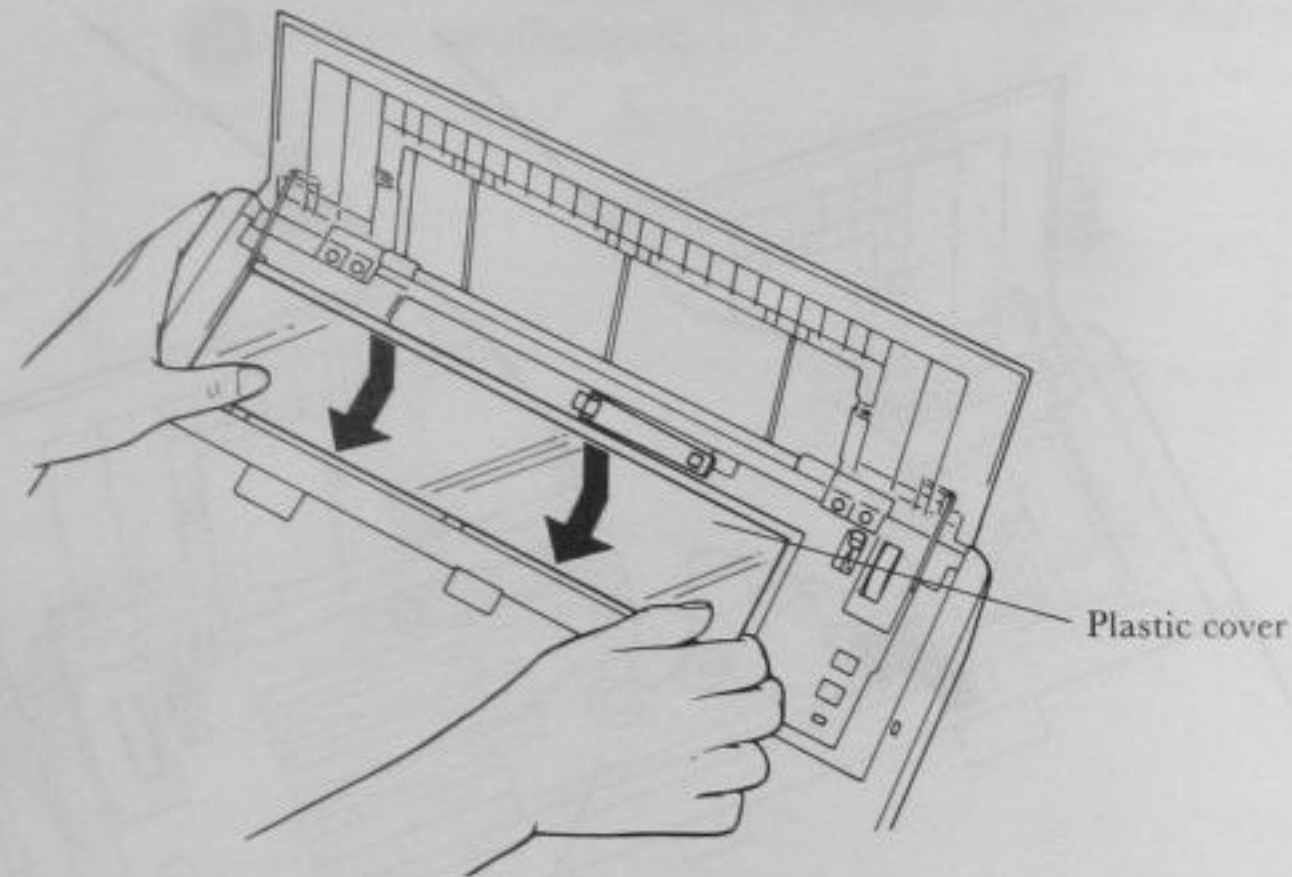


Fig. 2-8. PLASTIC COVER

THE EXEC. PARTNER

Your Exec. Partner contains an 8086-2 CPU, 256K bytes of memory, a high resolution plasma display, keyboard, disk drives, printer which are the major components of your computer system.

Display Panel

Your Exec. Partner is equipped with a **high-resolution plasma display screen**. It displays information (data) that you enter into or receive from the computer. The display screen presents up to 25 lines of information, with up to 80 columns (characters) in each line. The displayed characters are bright neon orange.

Located below the plasma display screen are the **floppy disk drives**, from left to right, drives A and B. These devices are storage devices, that is, they store information. The magnetic recording media used by floppy disk drives to store information are called floppy disks. Unlike the memory, information stored on floppy disks are not cleared when the power of the computer is turned off.

In front of the floppy disk drives is the keyboard. It enables you to enter information (programs or data) into the computer. The keyboard will be described in a later section.

NOTE: If your Exec. Partner has the hard disk, next to the drive A is the **hard disk drive unit**.

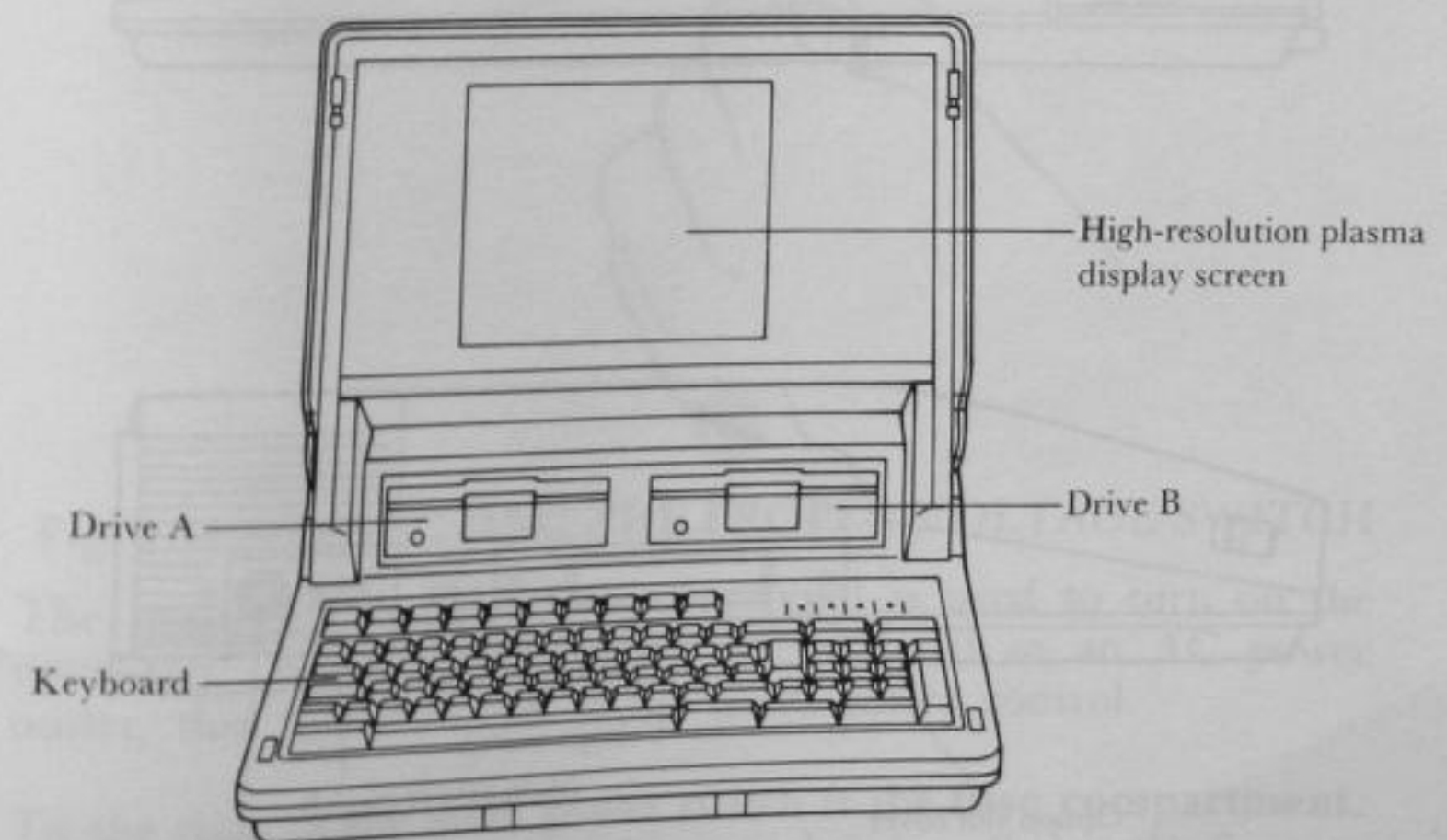


Fig. 2-9. DISPLAY PANEL (FT-70)

Side Panels

On the left side panel, near the rear, is the **ventilation** for the **cooling fan**. Since the fan keeps the internal parts from overheating, the ventilation must not be covered.

On the right side panel is the **option slot cover** which covers the option slot. You can install an option board in the option slot to expand the functions of the Exec. Partner.

Near the rear of the panel are the **voltage switch, main power switch and fuse**.

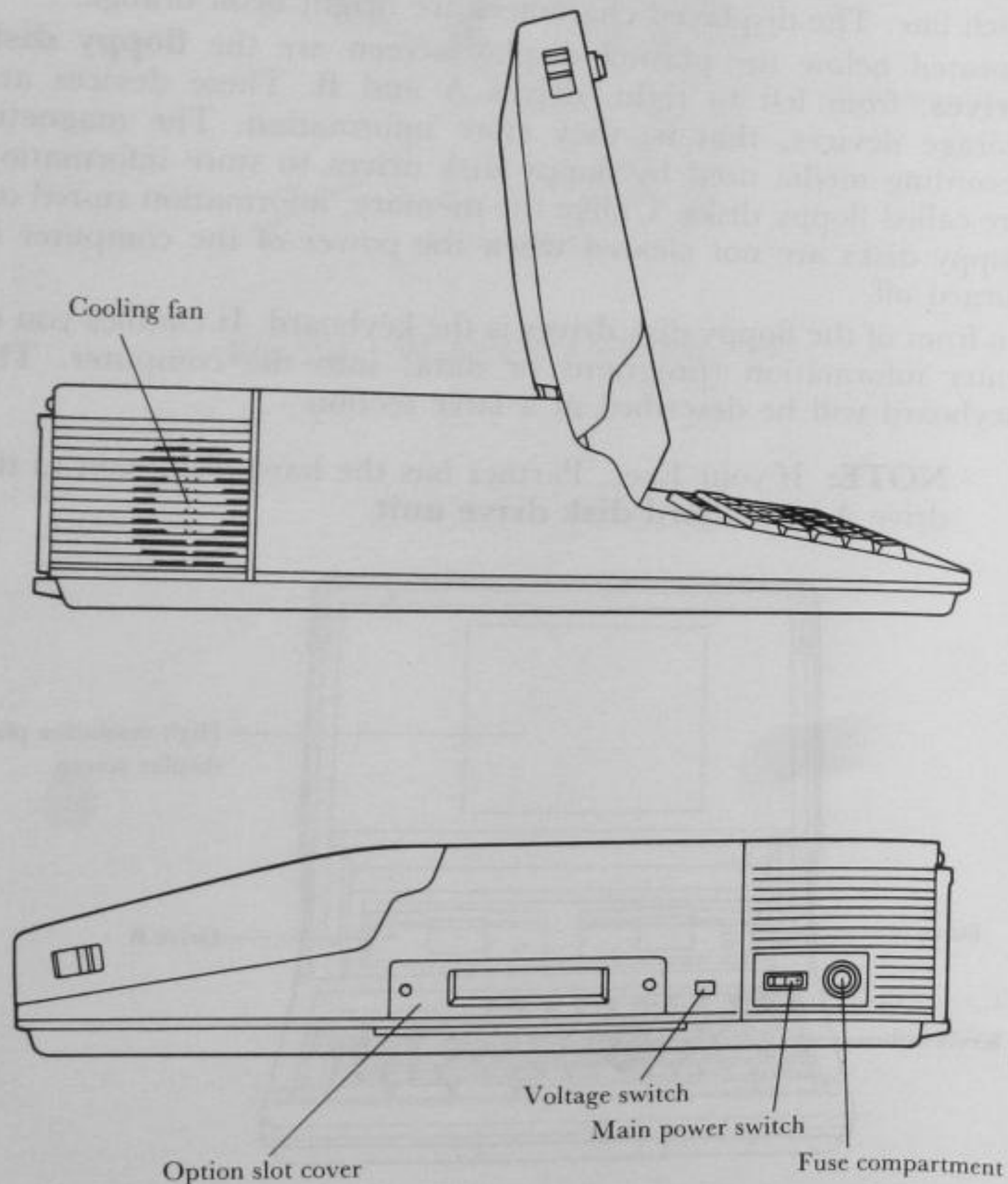


Fig. 2-10. SIDE PANELS

The **voltage switch** (slide type) allows you to set the Exec. Partner for use at either 120 VAC or 220 VAC. The switch is recessed so that it cannot be changed accidentally. Using a screwdriver, slide the switch to 120 V or 220 V, to match the AC power line voltage used in your area.

NOTE: The power cord that came with your Exec. Partner cannot be used at a power voltage of 240 V. Use the special 240 V power cord when operating the Exec. Partner at 240 V (after setting the voltage selector).

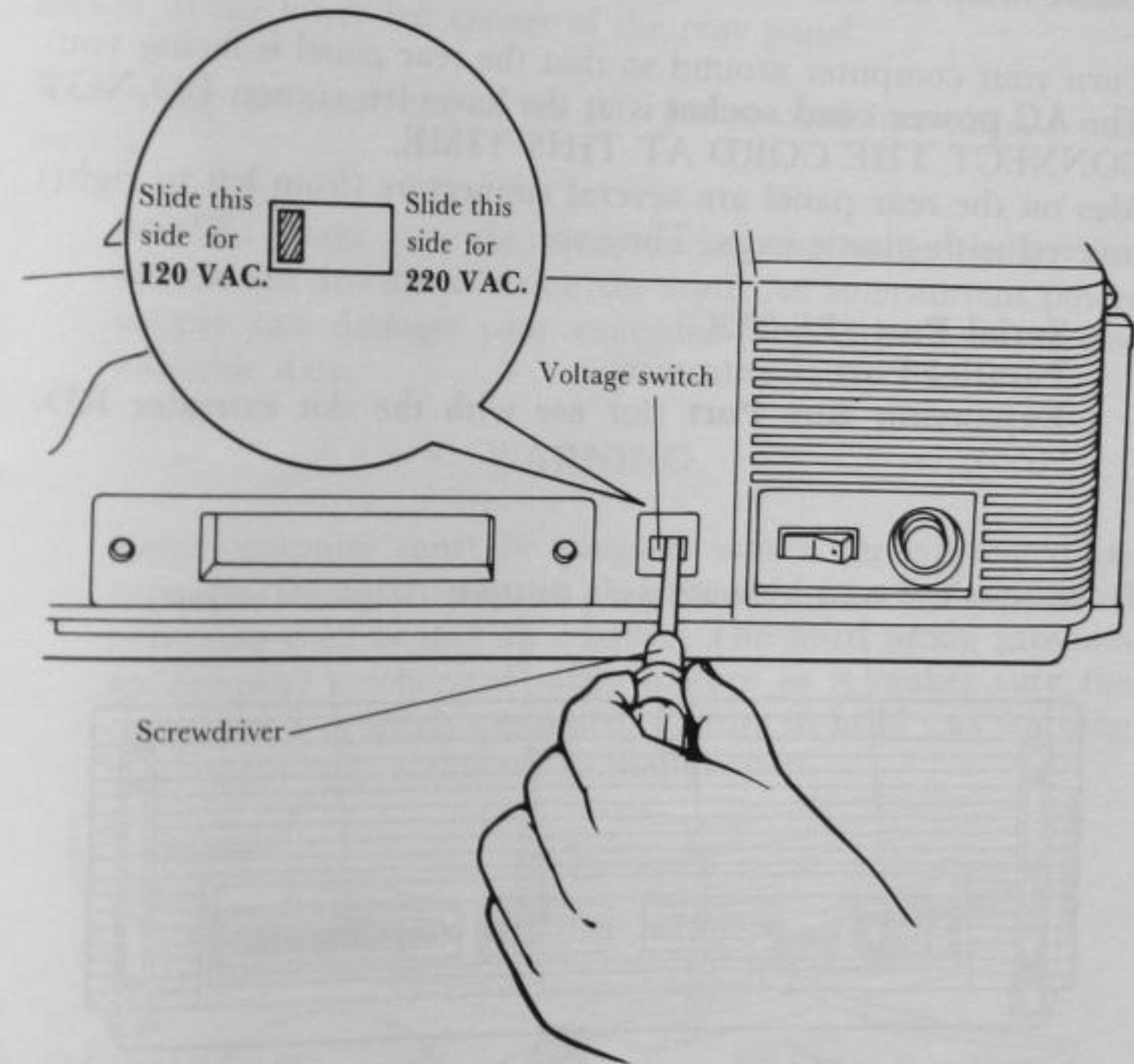


Fig. 2-11. SELECTING THE PROPER VOLTAGE SWITCH

The **main power switch** (rocker type) is used to turn on the computer. When the computer is connected to an AC power outlet, this rocker switch is the main power control.

To the right of the main power switch is the **fuse compartment**. It contains one 250 VAC 5A fuse which is installed at the factory. To change the fuse, see the instructions in Appendix A.

Top Panel

At the rear of the top panel is the cover for the **internal printer**. The setup of thermal printer is described in a later section.

To the right of this cover is the **power-on indicator**. It illuminates (red) when the power is turned on and goes out when the power is turned off.

Rear Panel

Turn your computer around so that the rear panel is facing you. The **AC power cord socket** is at the lower left corner. **DO NOT CONNECT THE CORD AT THIS TIME.**

Also on the rear panel are several connectors (from left to right) covered with plastic caps. They are:

Serial Port (RS-232C)

Parallel Port (Centronics)

Expansion Bus Port (for use with the slot extender RD-9655).

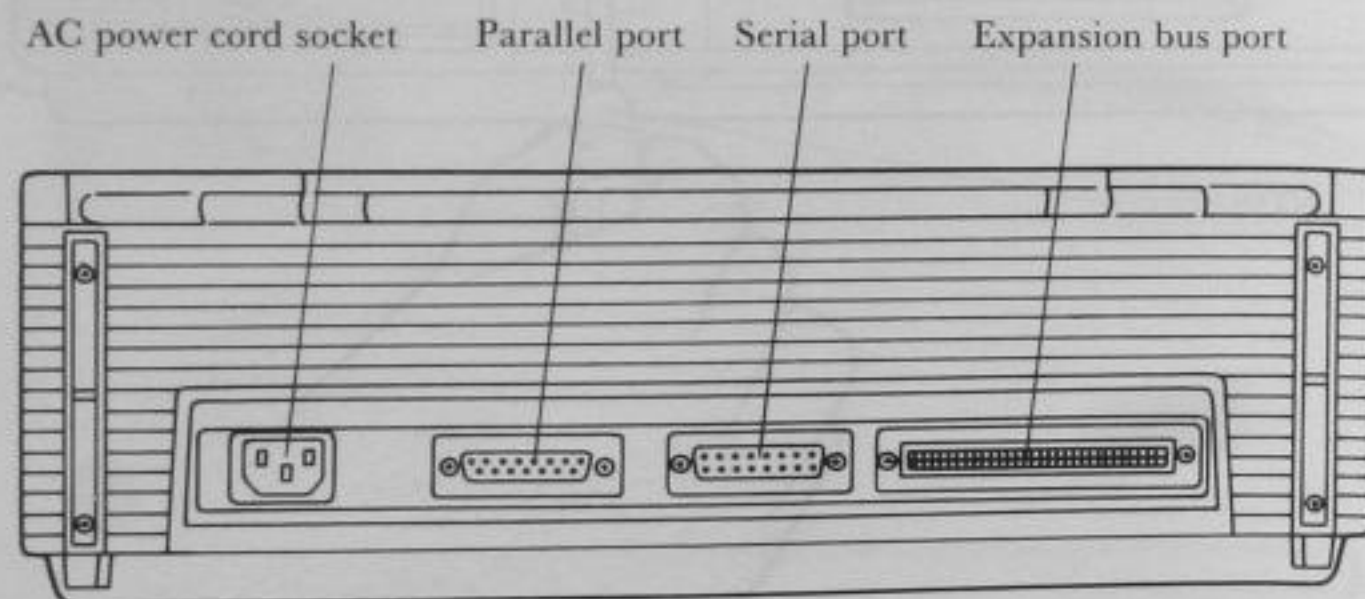


Fig. 2-12. REAR PANEL

Connecting the AC Power Cord

... WARNING ...

BE SURE THE MAIN POWER SWITCH IS IN THE OFF POSITION BEFORE PLUGGING IN THE AC POWER CORD.

One end of the AC power cord has 3 holes arranged in a triangle. Hold this end so that the two holes are at the top and the single hole is at the bottom. Insert this end into the AC power cord socket in the lower left corner of the rear panel.

Now plug the other end of the AC power cord into an AC wall outlet.

NOTE: Make sure the cord reaches the outlet easily. You should not stretch or strain the cord. An intermittent power supply can damage your computer and cause you to lose valuable data.

... WARNING ...

Your computer must be plugged into a three-pronged receptacle. Do not attempt to plug the cord into a two-pronged extension cord or into an adapter. The third prong provides an essential mechanical safety feature as it makes sure the computer is properly grounded. Failure to heed this warning may cause your computer to malfunction.

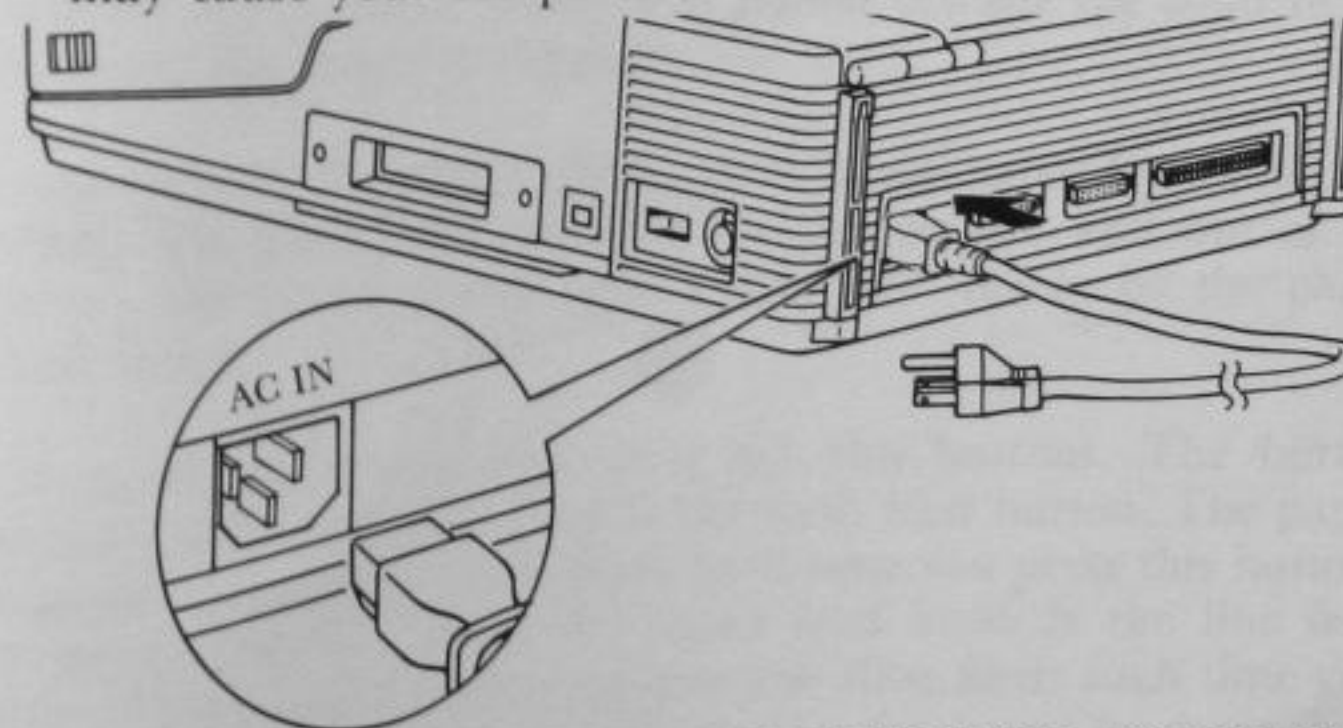


Fig. 2-13. INSERTION OF PLUG INTO THE AC POWER CORD

THE KEYBOARD

Before you begin to enter data into your computer, take a moment to acquaint yourself with the main features of the keyboard. The keys and indicators on the keyboard will be described in detail in Chapter 3.



Fig. 2-14. KEYBOARD

THE INTERNAL PRINTER

The Exec. Partner's internal printer is located on the top near the back. It is a thermal/thermal transfer type printer capable of printing on both thermal paper (RD-9671) and thermal transfer paper (RD-9442). A printer ribbon is supplied for thermal transfer printing on thermal transfer paper.

For thermal transfer printing on thermal transfer paper, the printer ribbon must be mounted on the printer carriage. For thermal printing on thermal paper, the printer ribbon must be removed. We will describe how to set up your internal printer for both printing methods in a later section.

Now let's take a look at the internal printer.

On the left side is the density selector. It is used to select the print density (darkness of the characters printed on paper) or printed character font. Three print densities can be selected. To access the switch, first remove the plastic cover by sliding it back and lifting it up if you haven't done so already. Position 1 is for normal density. Position 2 is for dark density. Position 3 is for bold character font.

On the right side, near the rear, is the paper release lever. When it is pushed toward the back of the computer, the paper is held firmly by the printer. When it is pulled toward the front of the computer, the paper is released.

To the right of the paper release lever is the paper feed knob. By turning this knob, you can feed the paper into and out of the printer. The paper moves only if it is held firmly by the paper release lever.

In front of the paper feed knob are two buttons. The button closest to the paper feed knob is the form feed button. The paper advances by one page (form feed) each time you press this button. The button furthest from the paper feed knob is the line feed button. The paper advances by one line (line feed) each time you press this button. If you want to advance the paper by more than one line and less than a page, hold down this button until the paper has advanced the desired number of lines.

Directly in front of the line feed button is the error indicator. This indicator illuminates when a printing error has occurred which prevents the printer from operating (e.g. when the printer is out of paper).

Below the printer mechanism on the right side are four Printer DIP switches. They are provided so that you can set the various features of the printer. An explanation of the functions and how to set them are given in Chapter 4.

Now that you know the basics on the internal printer, we will describe how to set it up for thermal printing and thermal transfer printing.

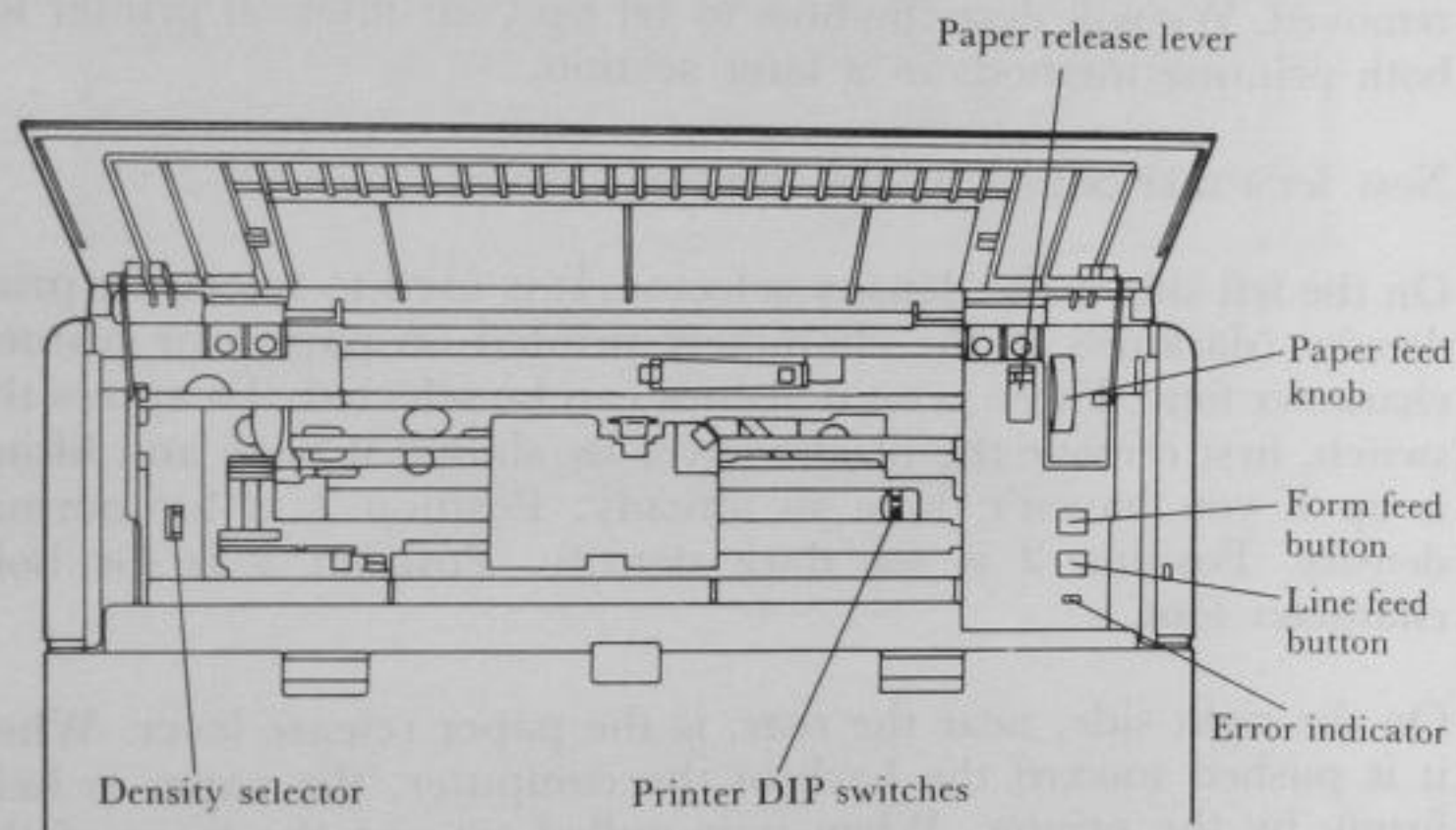


Fig. 2-15. THE INTERNAL PRINTER

NOTE: When the error indicator illuminates, make sure the paper is properly inserted. However, if the indicator lights even after the paper has been correctly inserted, see your Panasonic dealer.

If the Density selector switch is set to the Position 3 (bold character), Exec. Partner does not print at 132 characters per line.

Dual Mode Printer

The internal printer of the Exec. Partner is capable of printing using either of two methods: thermal printing or thermal transfer printing.

To print using thermal printing, remove the printer ribbon from the carriage, and fasten the roll of thermal paper with the paper rod.

Then insert the thermal paper into the slit and you are ready to print. See page 2-20 for the procedures on how to print.

To print using thermal transfer printing, mount the printer ribbon onto the carriage, and insert the thermal transfer paper into the slit. See page 2-28 for the procedures on how to print using thermal transfer printing.

Thermal Printing

Inserting the thermal paper

A roll of thermal paper is included as an accessory. Although there are many kinds of thermal paper rolls available, you must use only Panasonic thermal paper (RD-9671). Other types of thermal paper must NOT be used. You will also use the paper rod.

STEP 1 When the printer cover is open all the way it will stop. Note the smaller panel, the inside cover, in the printer cover. Push out the inside cover so that it falls toward the rear thus creating a horizontal rectangular opening in the printer cover. Push the plastic cover toward the rear side of the printer and lift up it.

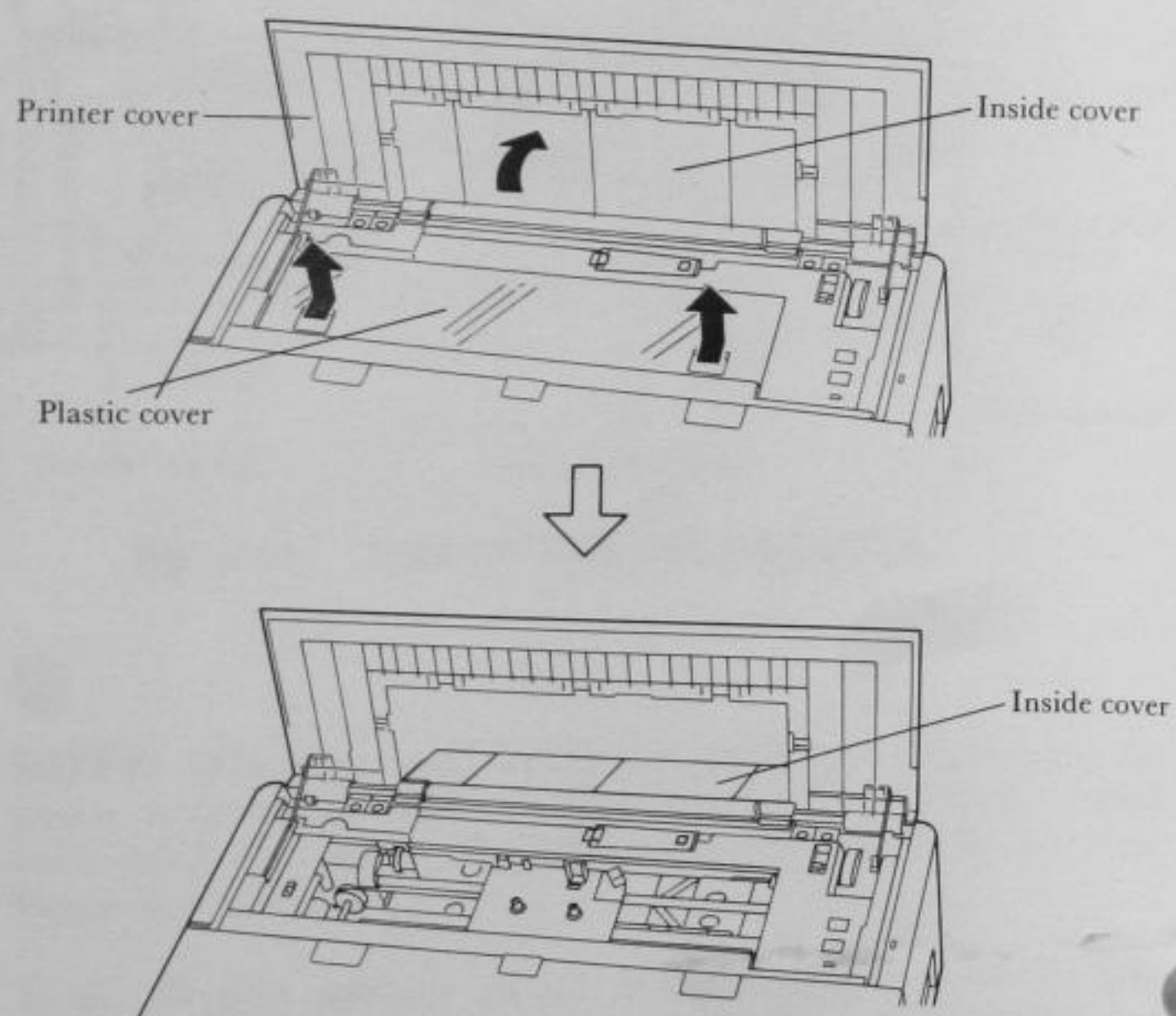


Fig. 2-16. OPENING THE PRINTER COVER AND THE INSIDE COVER

STEP 2 Hold the thermal paper horizontally with the end of the paper pointing at you from below. Insert the paper rod into the roll of thermal paper so that the large end is on the left. Mount the paper rod (with paper roll) onto the notches at the left and right of the rectangular opening in the printer cover.

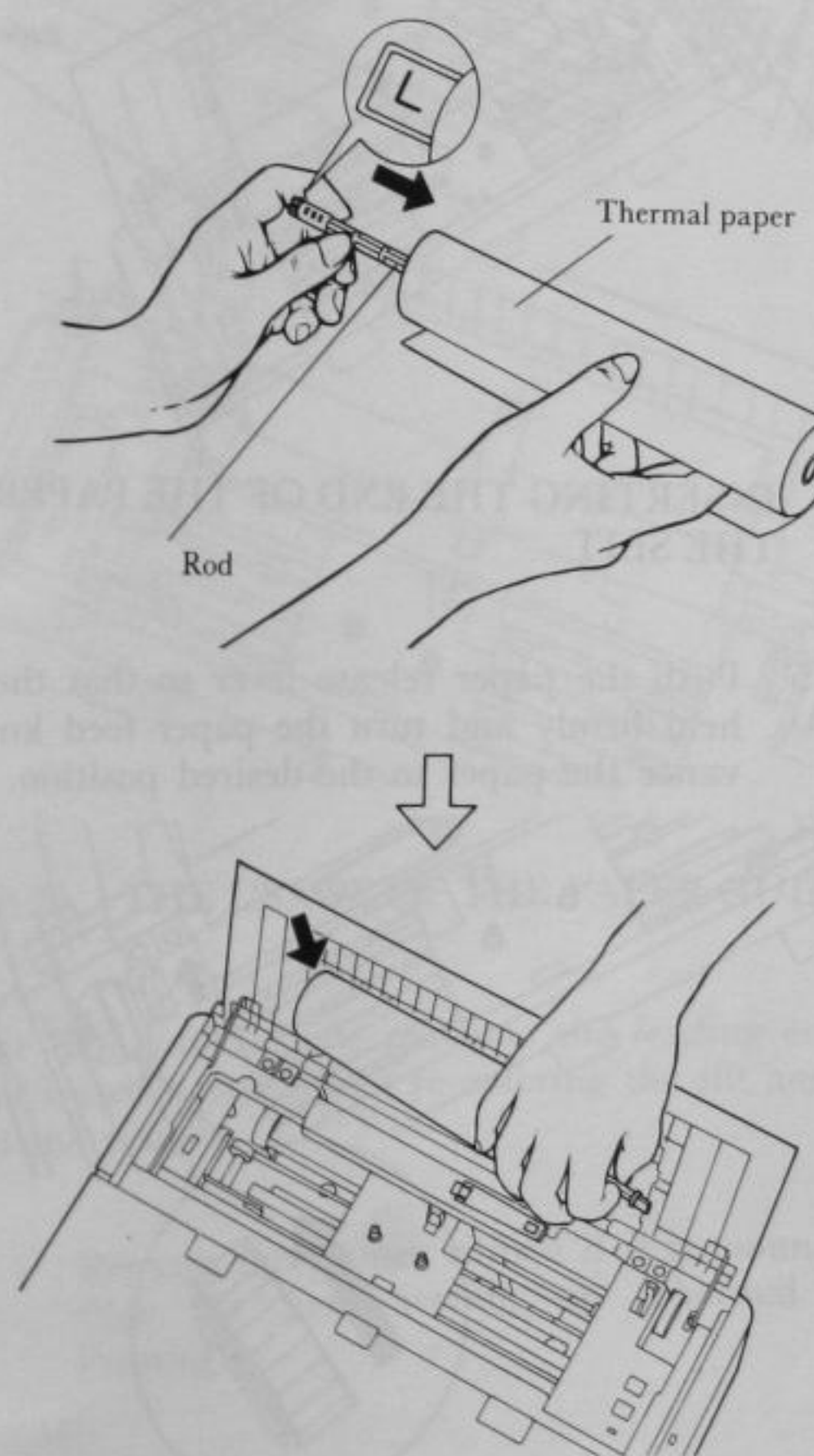


Fig. 2-17. MOUNTING THE PAPER

STEP 3 Pull the paper release lever toward you.

STEP 4 Insert the end of the paper into the slit in front of the lower edge of the cover as far as it will go.

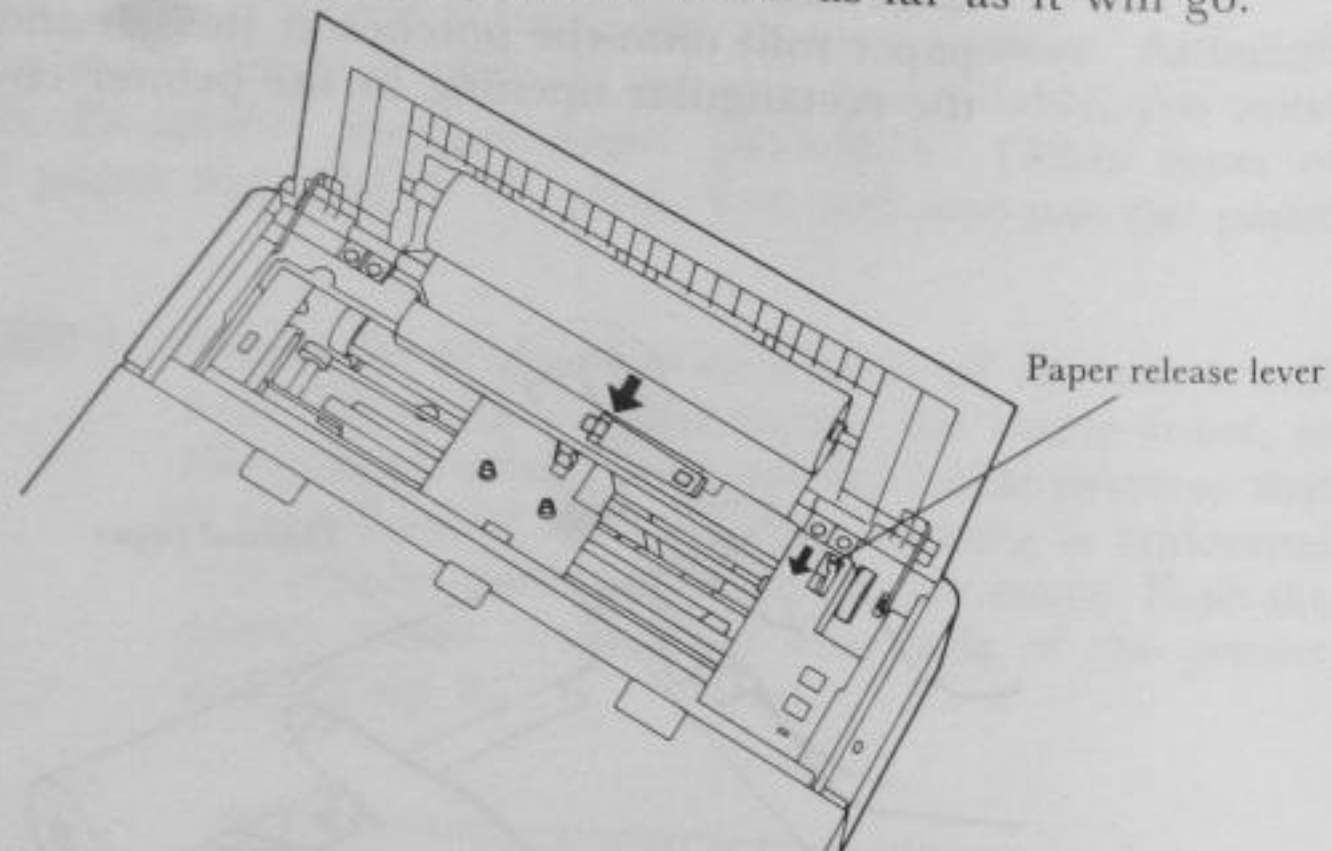


Fig. 2-18. INSERTING THE END OF THE PAPER INTO THE SLIT

STEP 5 Push the paper release lever so that the paper is held firmly and turn the paper feed knob to advance the paper to the desired position.

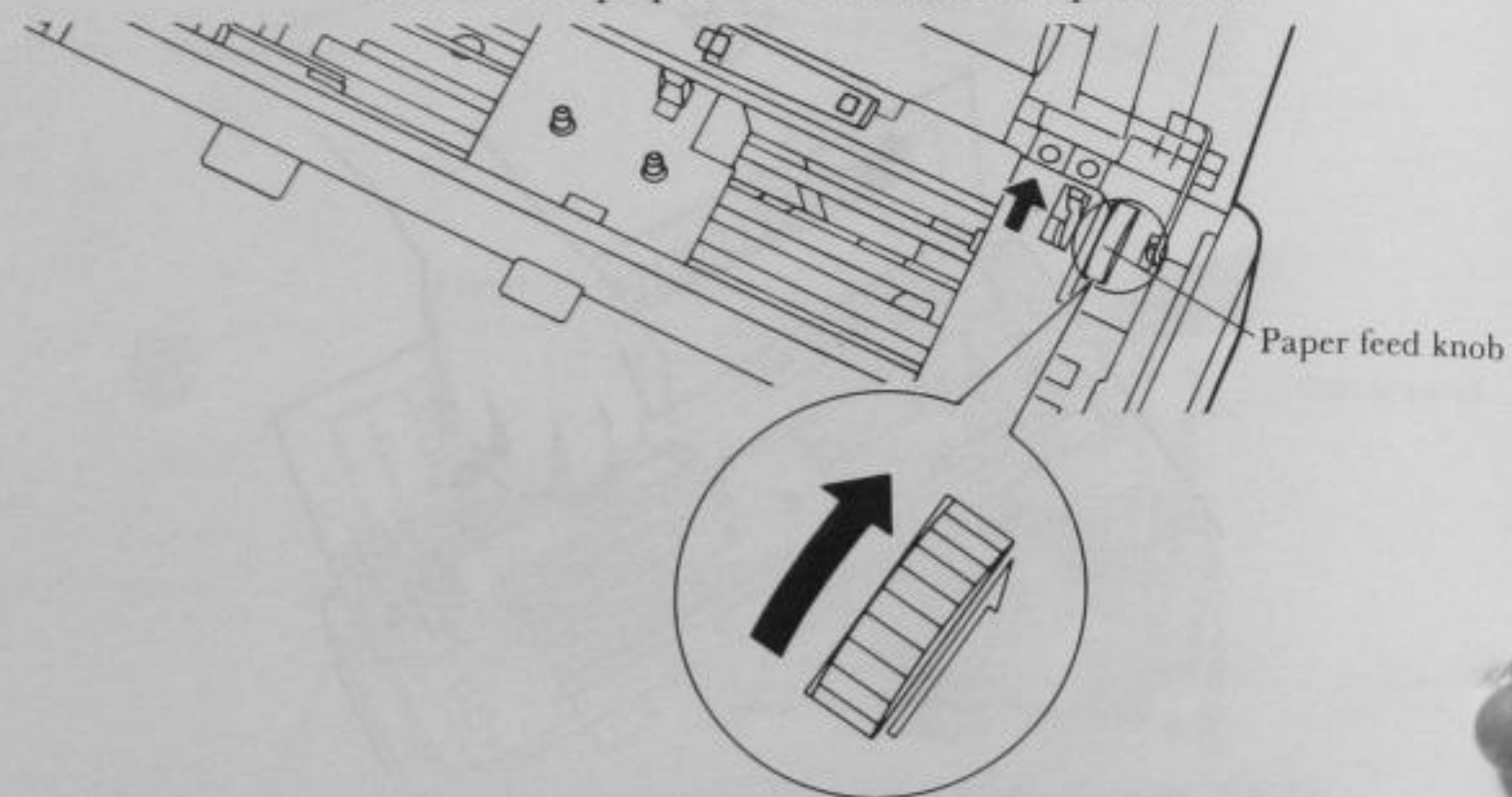


Fig. 2-19. ADVANCING THE PAPER TO THE DESIRED POSITION

STEP 6 Lift up the paper guide, turn it a quarter-turn counterclockwise and let it rest against the roll of paper.

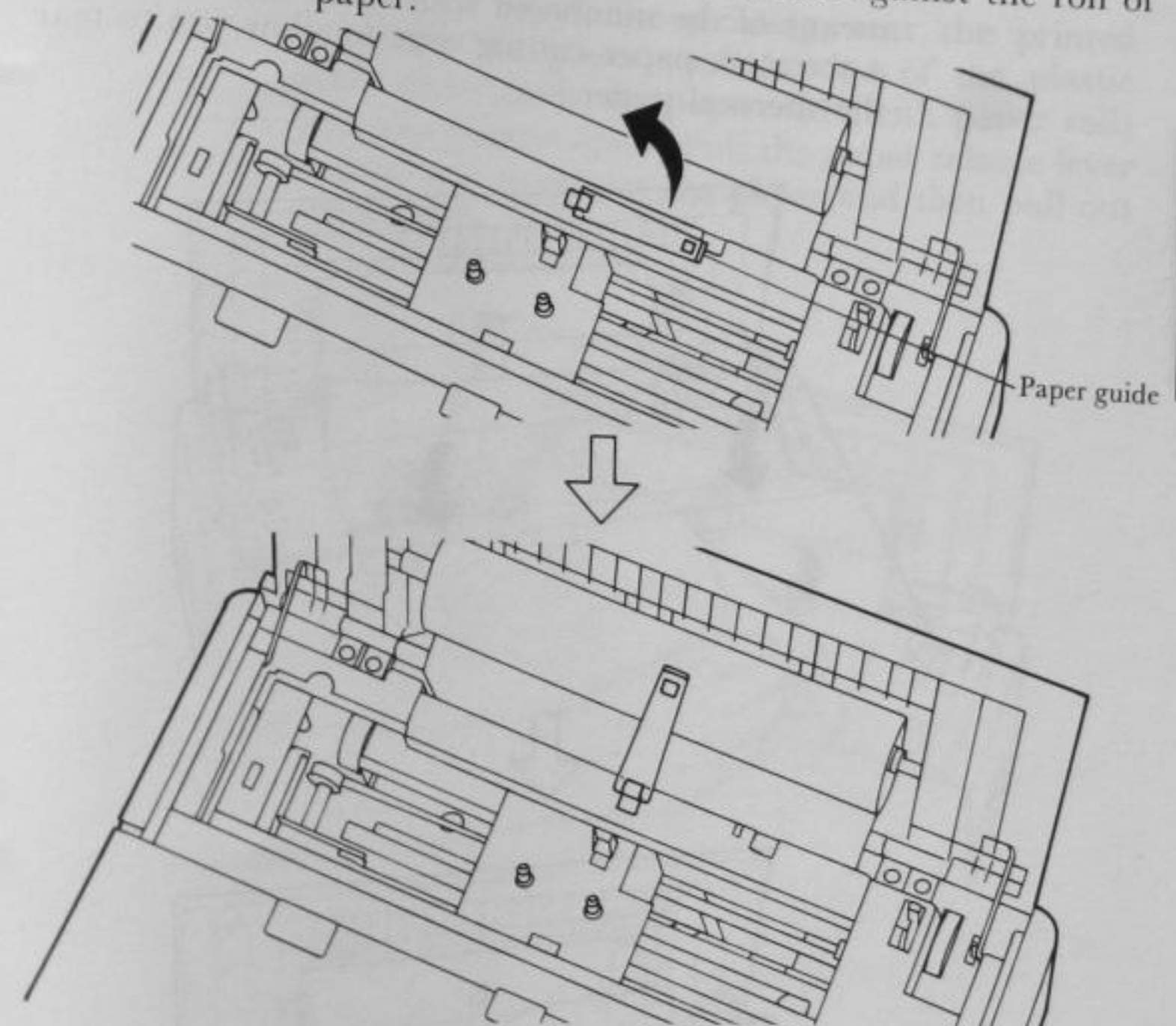


Fig. 2-20. THE USAGE OF THE PAPER GUIDE

NOTE: The paper guide prevents the leading edge of the thermal transfer paper from re-entering the slit and winding around the printer.

STEP 7 Remove the printer ribbon if it is mounted. (See the next section on Thermal Transfer Printing.)

STEP 8 Attach the plastic cover. (This cover allows you to determine the position of the printed characters by means of the numbered scale. This cover also has a serrated, paper cutting edge to allow you to tear the thermal paper.)

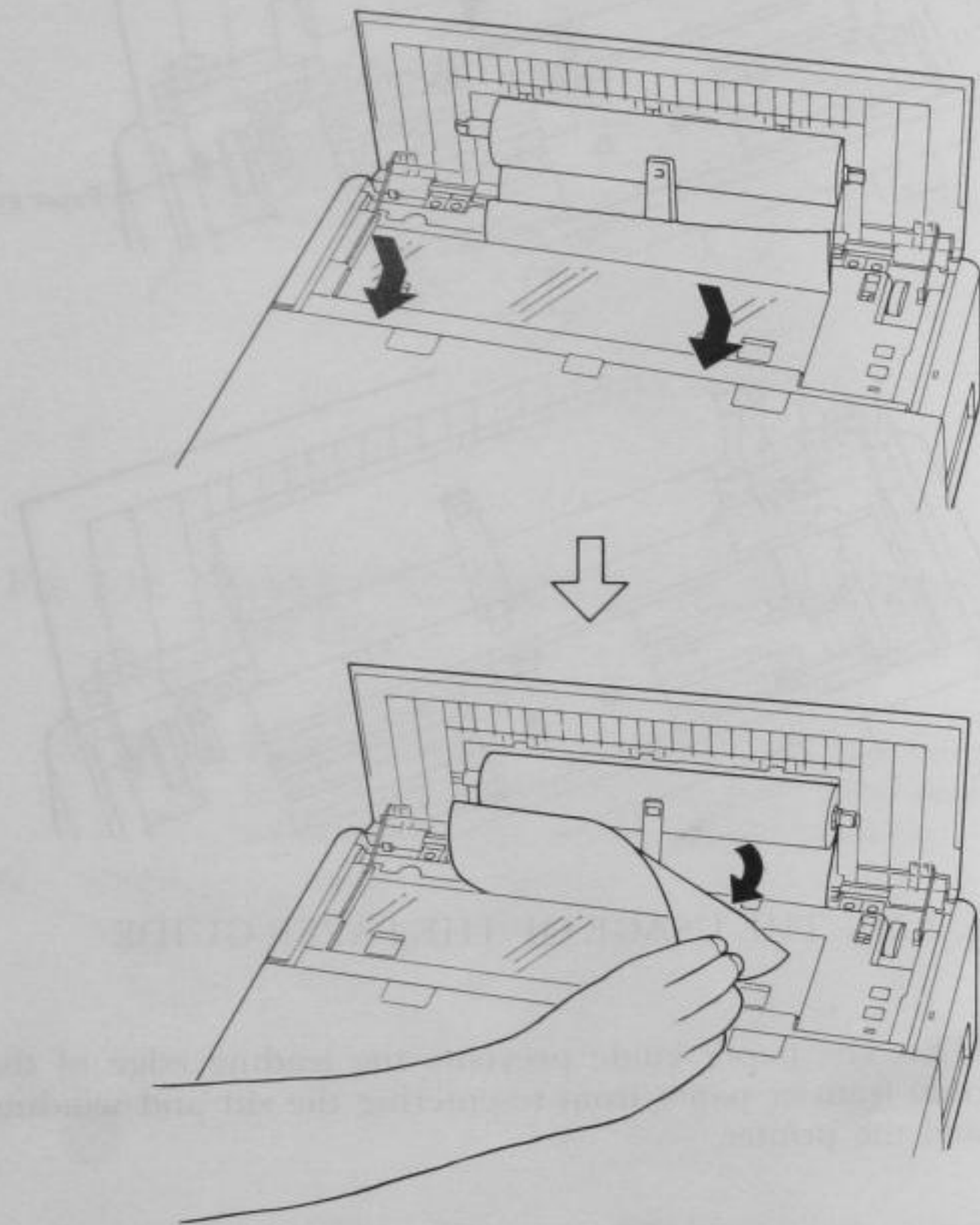


Fig. 2-21. PLASTIC COVER ATTACHMENT

NOTE: You may also advance the paper using the line feed button or form feed button. However, these buttons work only when the computer is turned on and the paper is properly inserted into the printer. For now, use the paper feed knob. We will turn on the computer in a later section.

Removing Paper from the Printer

STEP 1 When you finish the printing, tear the printed paper on the paper cutting edge of the plastic cover and detach the paper rod (with paper roll) from the printer cover. Pull the paper release lever toward you to release the paper and then pull out the paper.

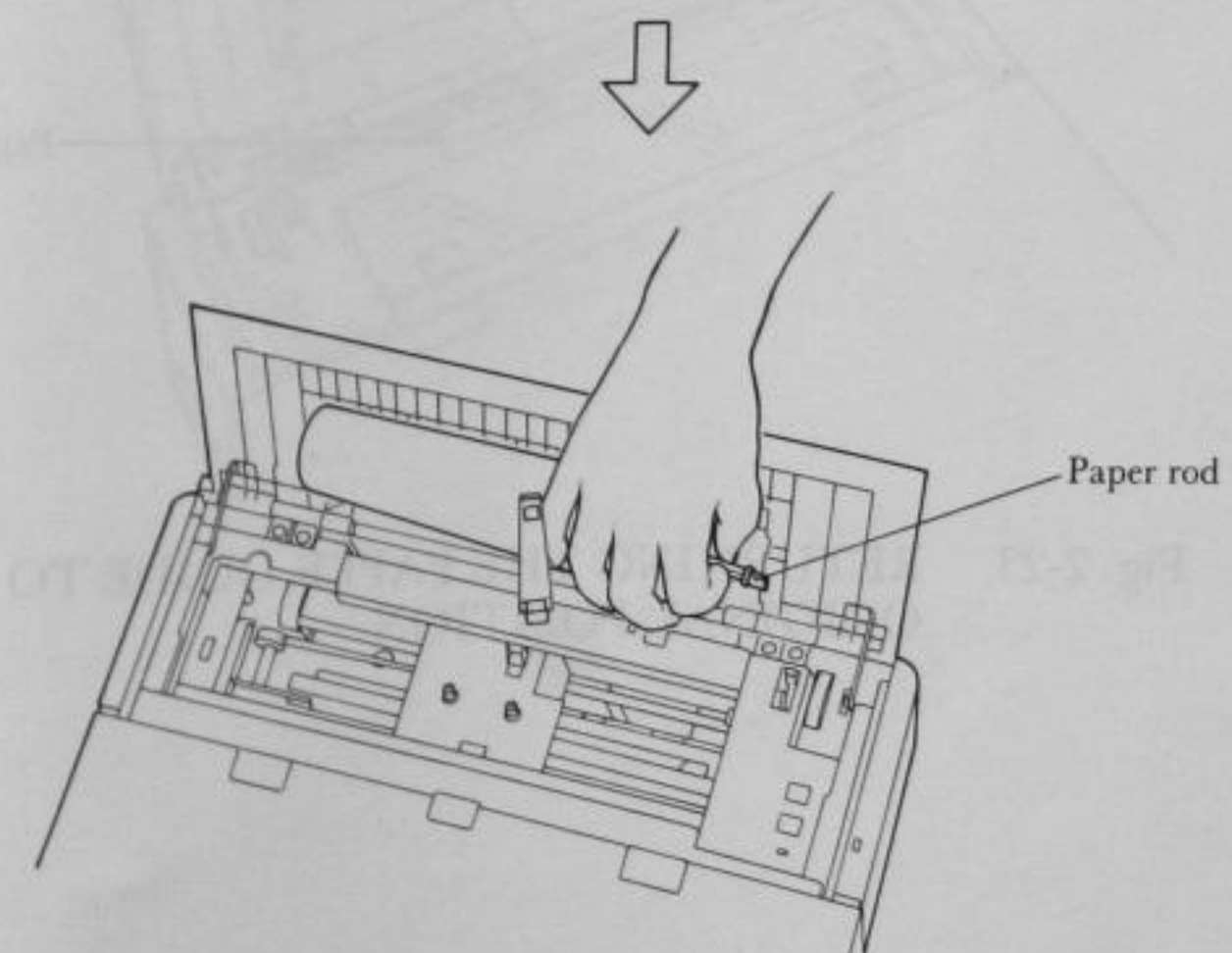
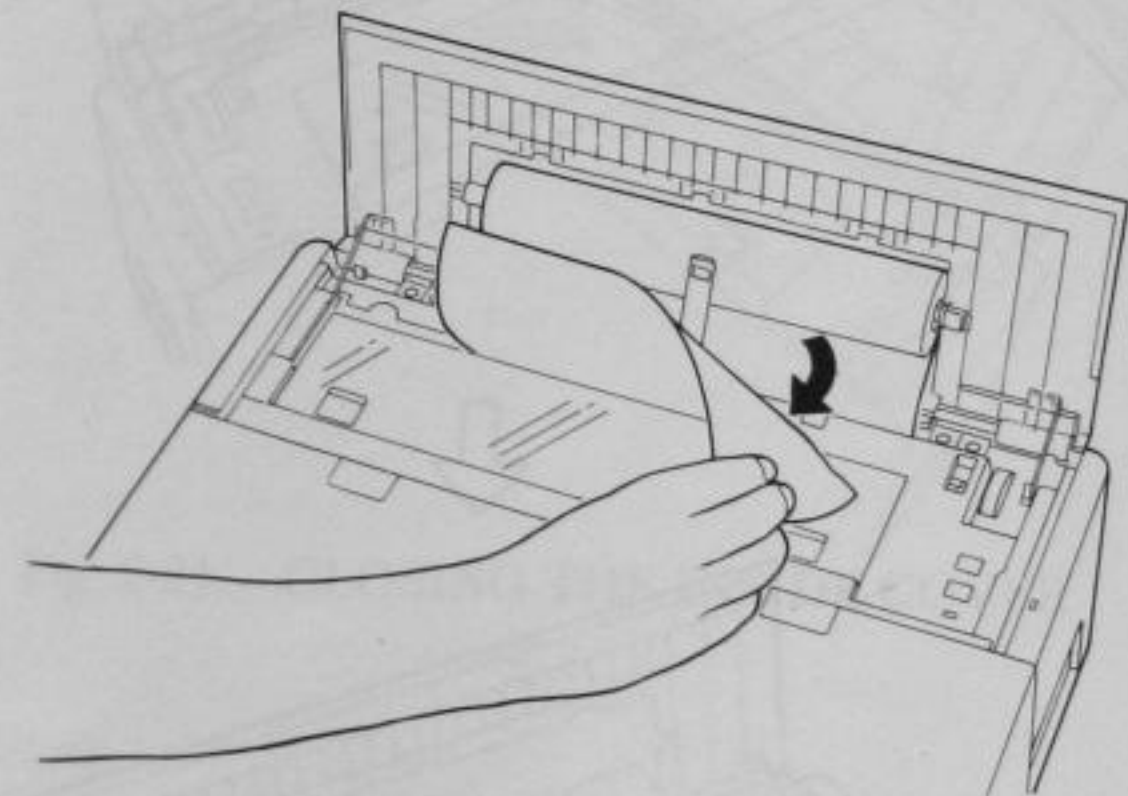


Fig. 2-22. TEARING THE PRINTED PAPER AND DETACHING THE PAPER ROD

STEP 2 Return the paper guide to its original position.

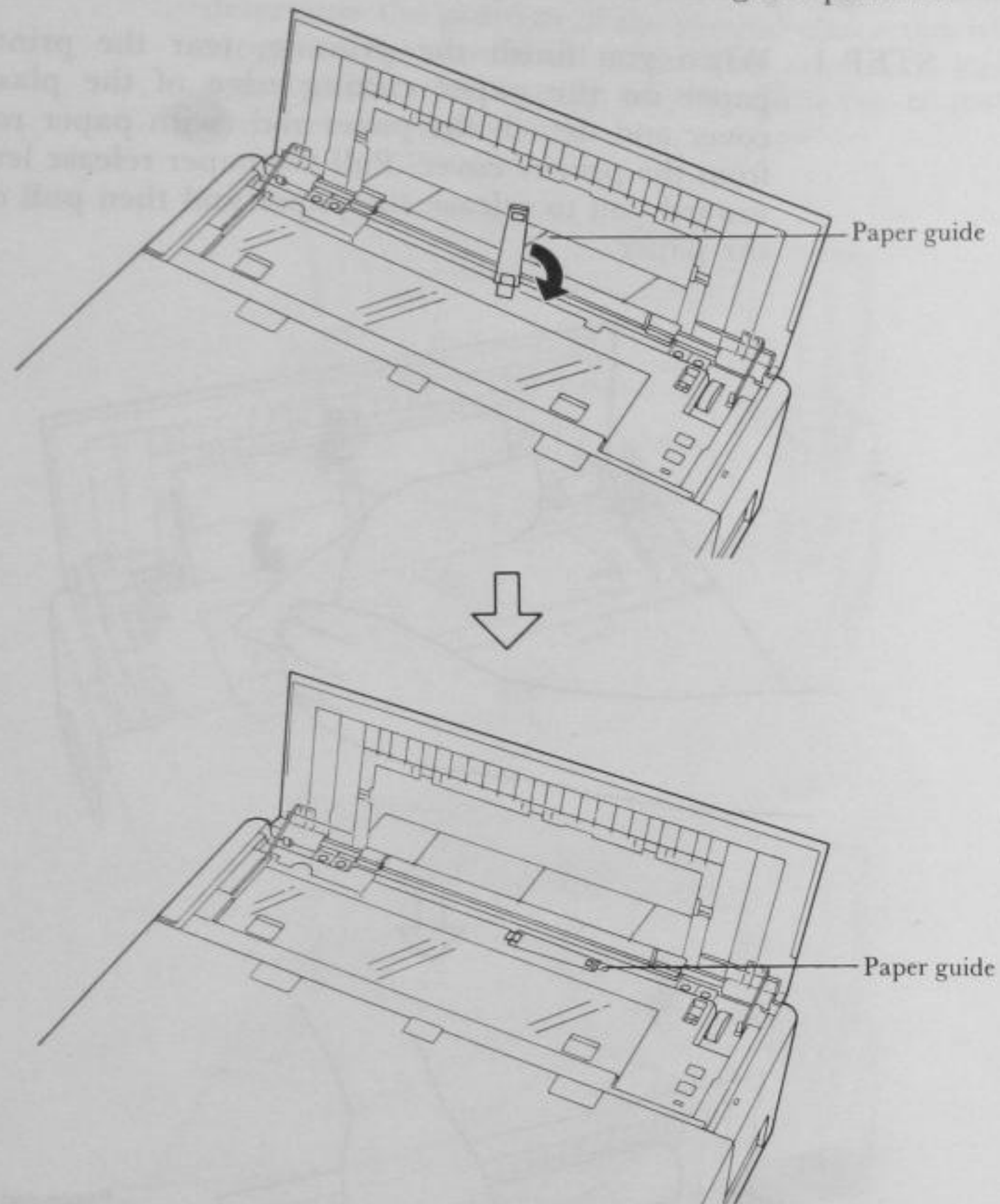


Fig. 2-23. RETURNING THE PAPER GUIDE TO ITS ORIGINAL POSITION

STEP 3 Close the inside cover.

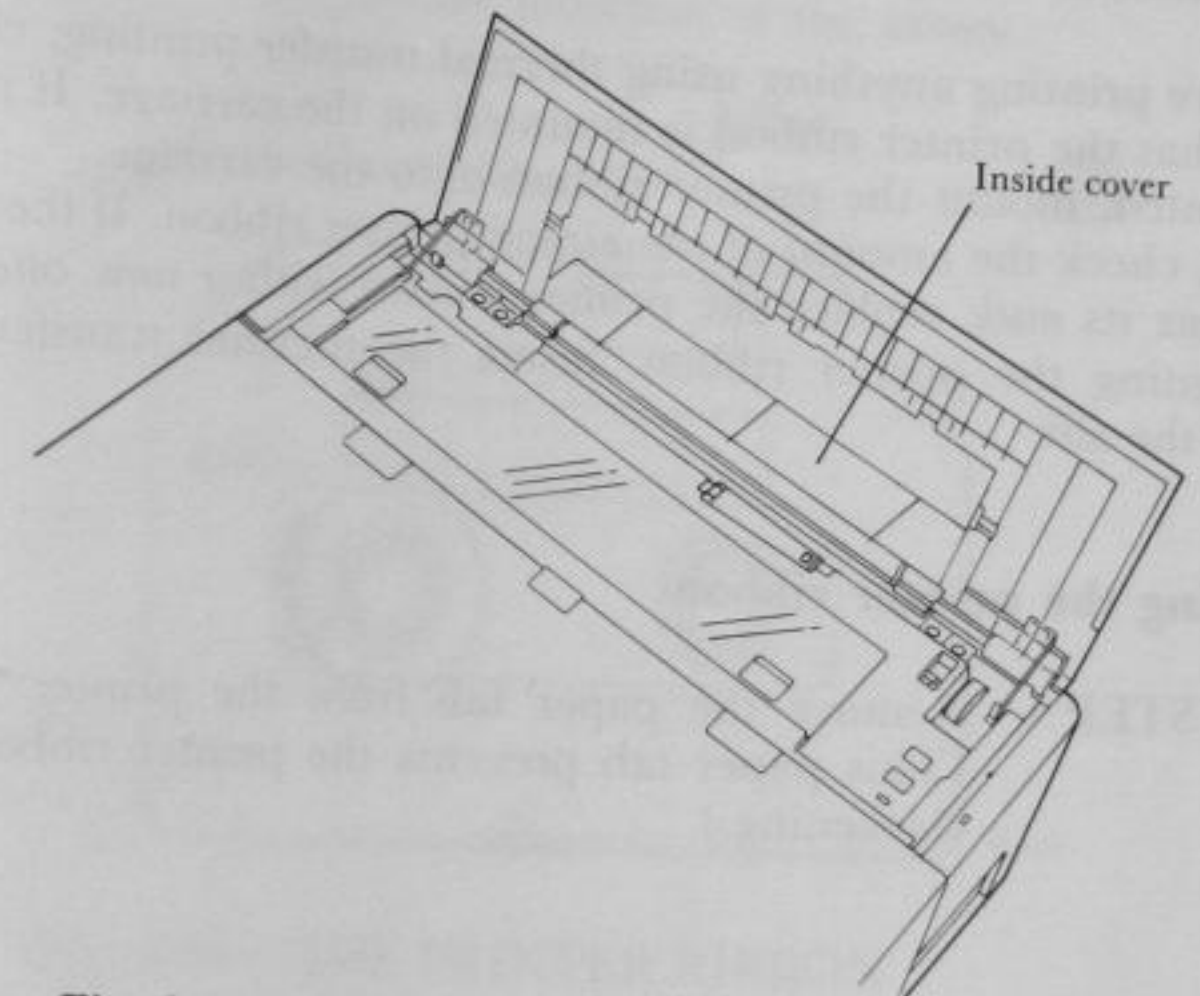


Fig. 2-24. CLOSING THE INSIDE COVER

Thermal Transfer Printing

Before printing anything using thermal transfer printing, check to see that the printer ribbon is mounted on the carriage. If it is not mounted, mount the printer ribbon onto the carriage. Also, check the amount of remaining printer ribbon. If the ribbon is near its end, replace the printer ribbon with a new one. After mounting the printer ribbon, insert the thermal transfer paper into the slit.

Setting the printer ribbon

- STEP 1 Remove the paper tab from the printer ribbon. (This paper tab prevents the printer ribbon from slackening.)

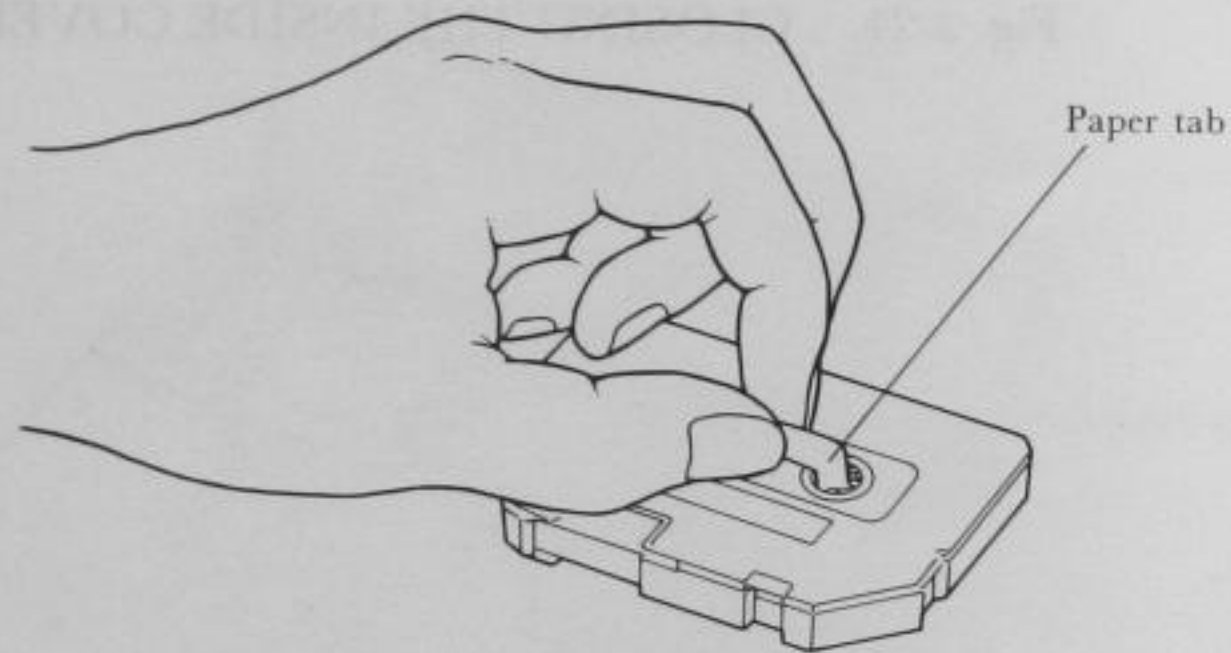


Fig. 2-25. REMOVING THE PAPER TAB

- STEP 2 Take up any slack in the printer ribbon by turning the reel in the direction of the arrow.

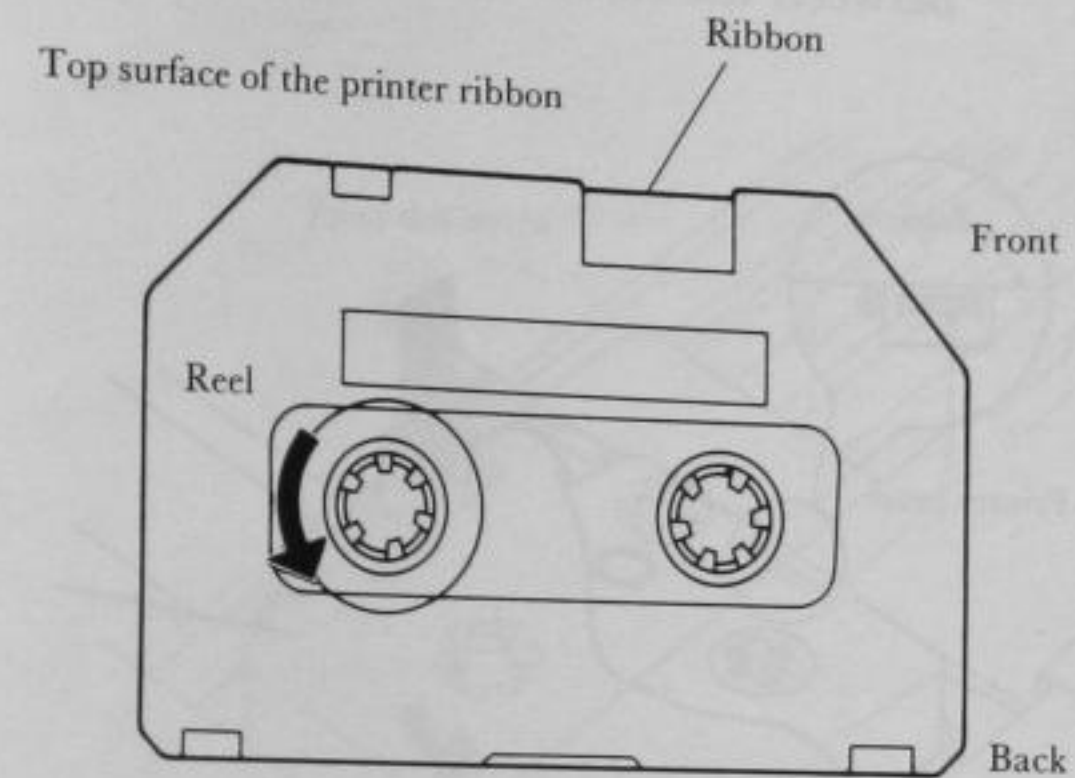


Fig. 2-26. THE PRINTER RIBBON

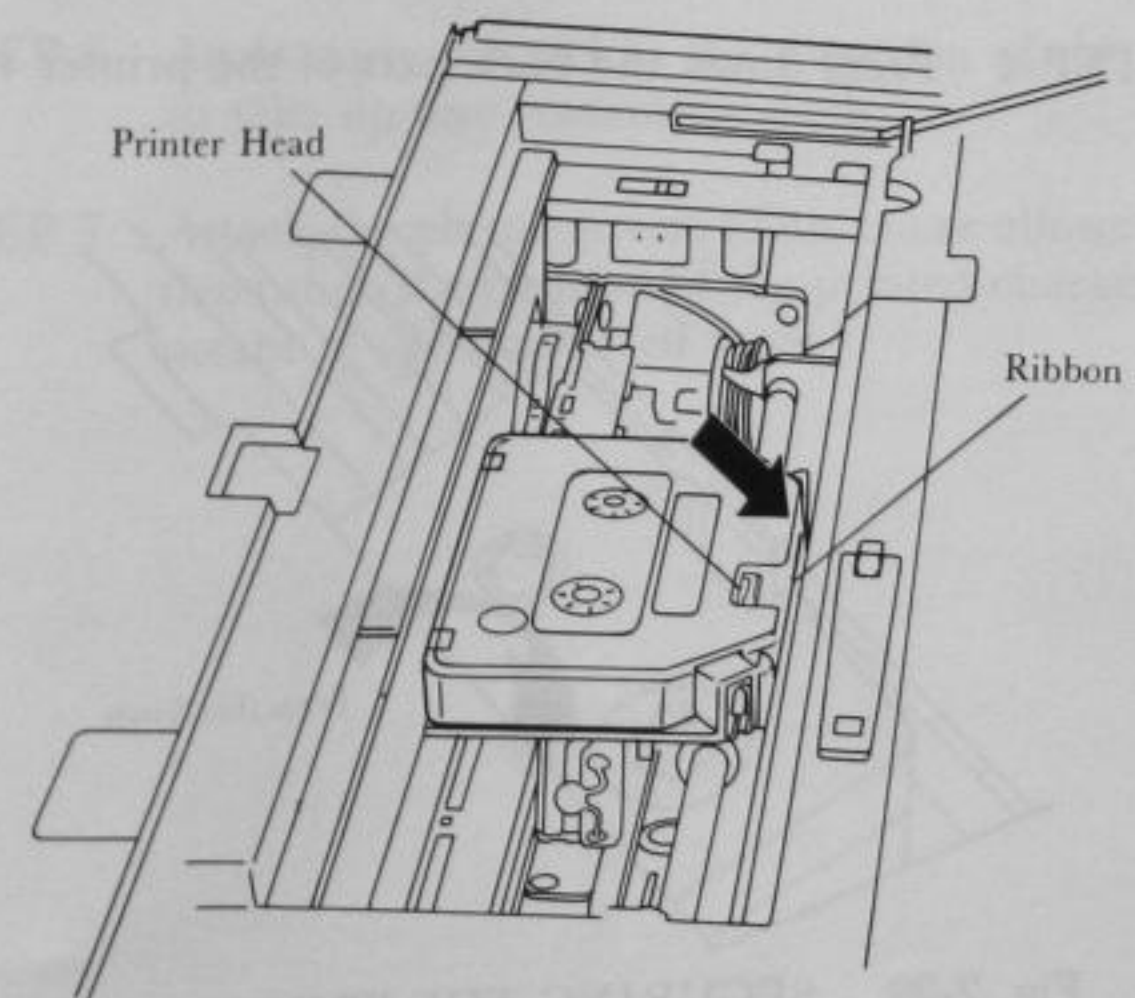


Fig. 2-27. PRINTER RIBBON AND PRINTER HEAD

- STEP 3** Position the front part of the printer ribbon on the carriage. (In other words, the Printer head is between the ribbon and the ribbon cassette case.)

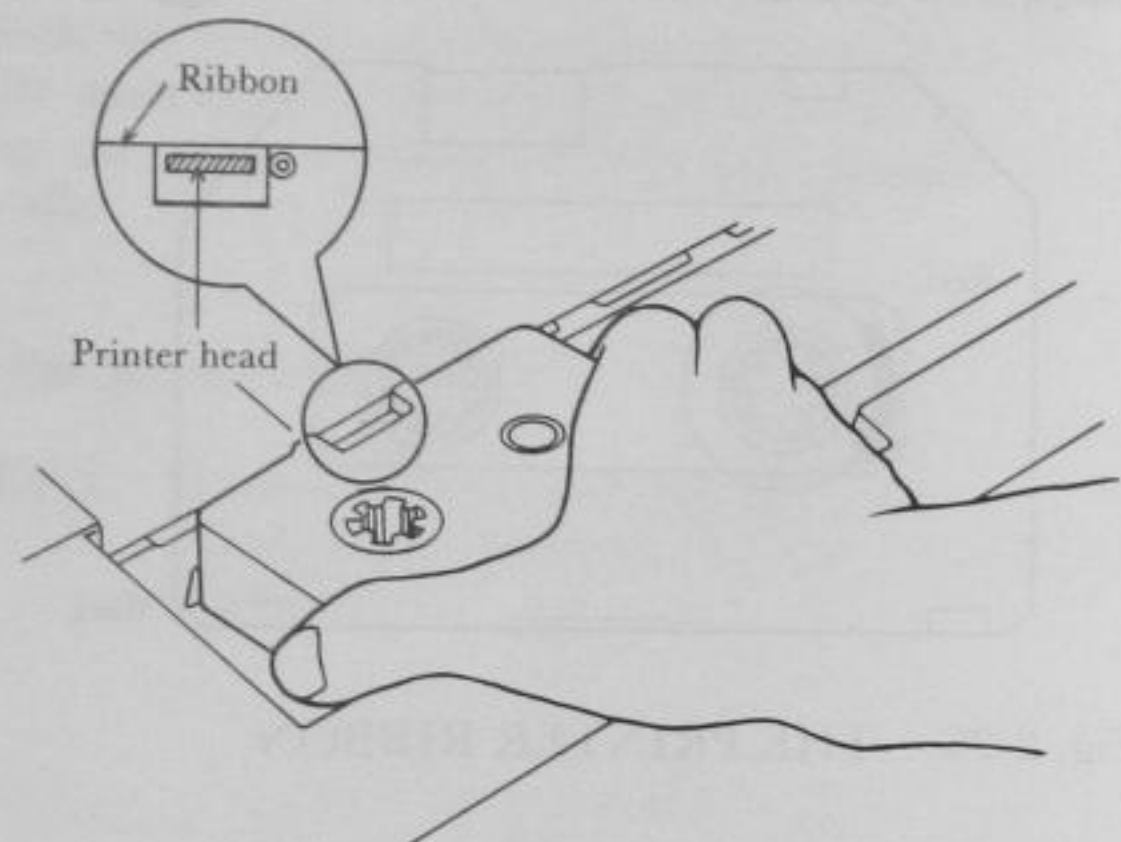


Fig. 2-28. POSITIONING THE PRINTER RIBBON

- STEP 4** Press down the back part of the printer ribbon.

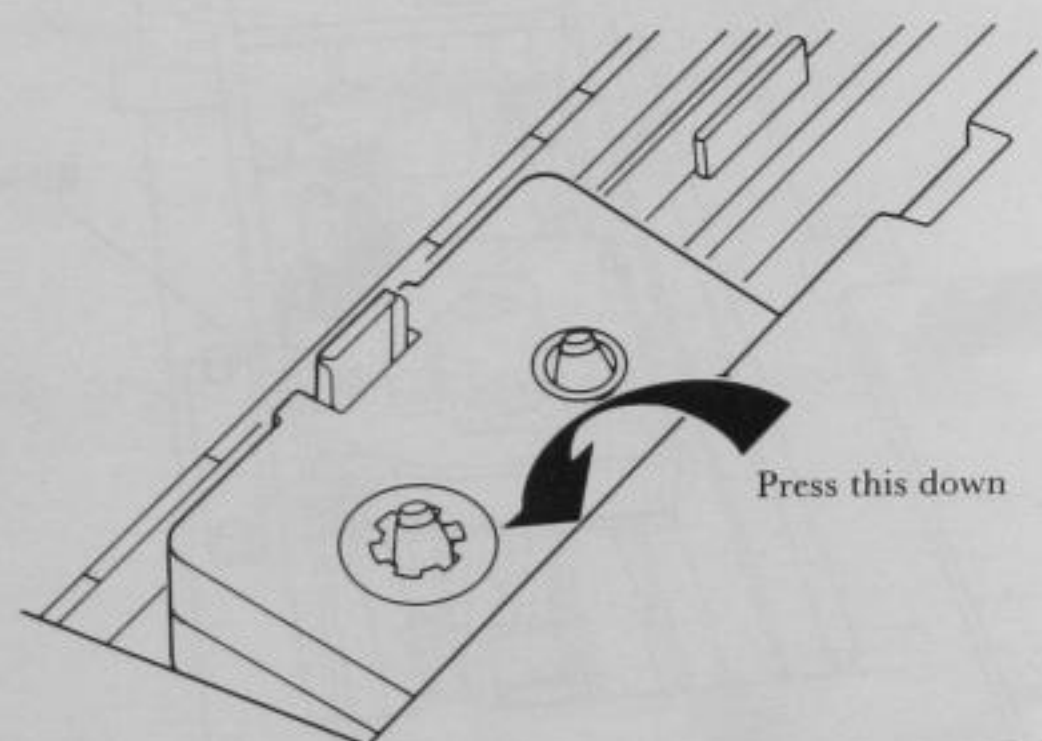


Fig. 2-29. SECURING THE PRINTER RIBBON

- STEP 5** Press down the front part of the printer ribbon. Be sure that the ribbon is in front of the printer head.

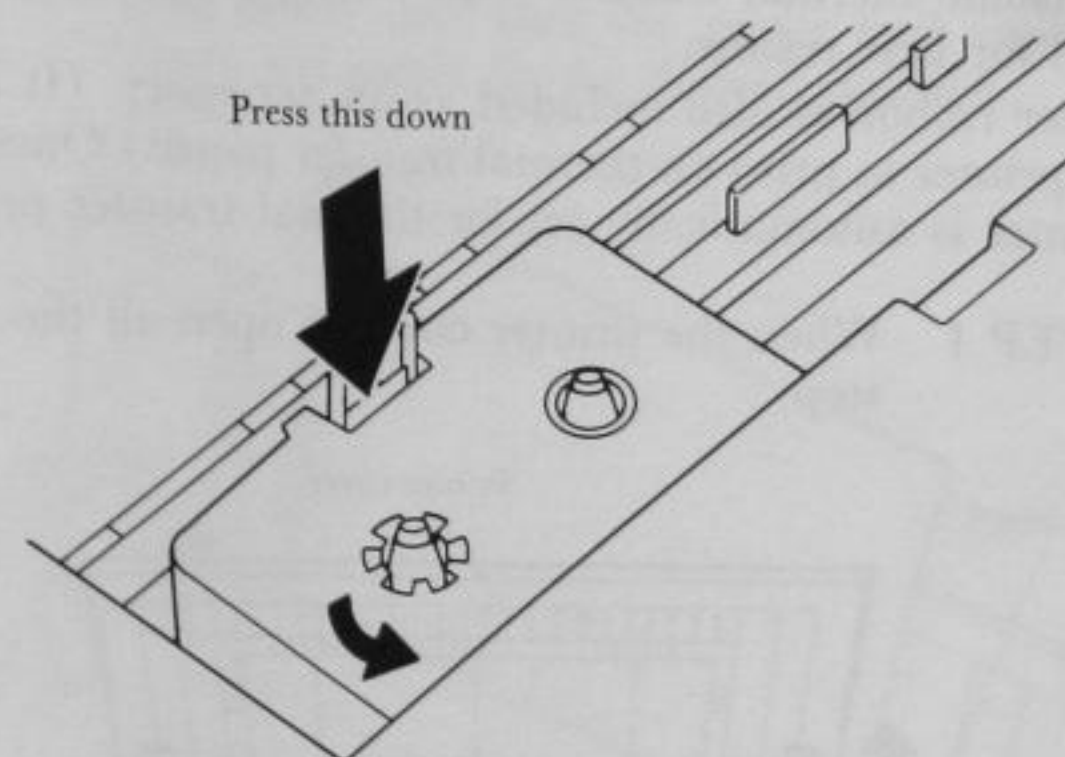


Fig. 2-30. SETTING THE PRINTER RIBBON

- STEP 6** Again turn the reel in the direction of the arrows to take up any remaining slack.
- STEP 7** Attach the plastic cover. (This cover allows you to determine the position of the printed characters by means of the numbered scale.)

Inserting the thermal transfer paper

A pack of thermal transfer paper (letter size) is included as an accessory. Although there are many kinds of paper available, use of Panasonic thermal transfer paper (RD-9442) is highly recommended for best results.

A printer ribbon is also included as an accessory. (It is required for the printer to print on thermal transfer paper.) Once mounted, the printer is automatically set for thermal transfer printing.

- STEP 1 When the printer cover is open all the way it will stop.

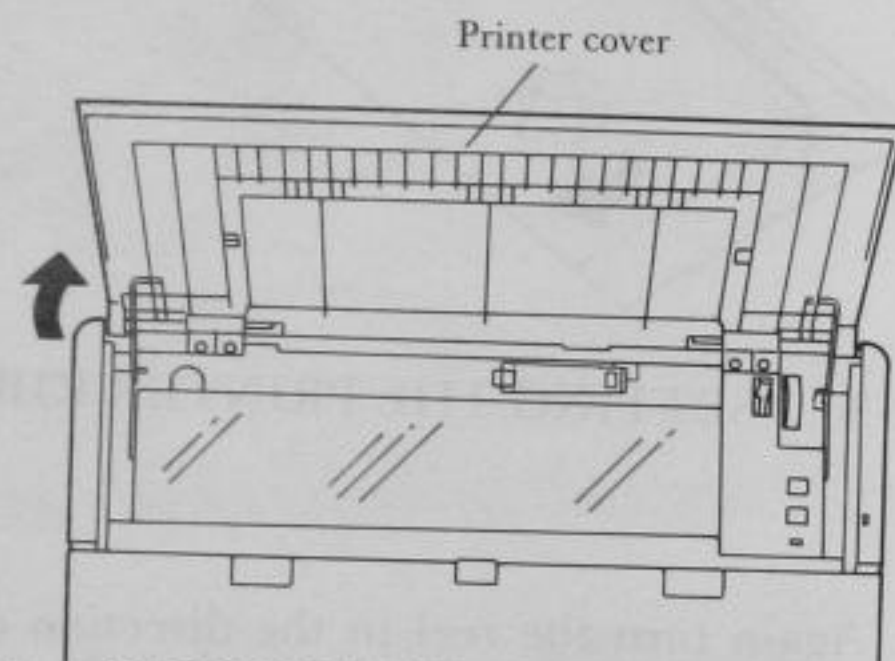


Fig. 2-31. OPENING THE PRINTER COVER

- STEP 2 Be sure that the plastic cover is attached to the printer mechanism.

- STEP 3 Pull the paper release lever towards you.

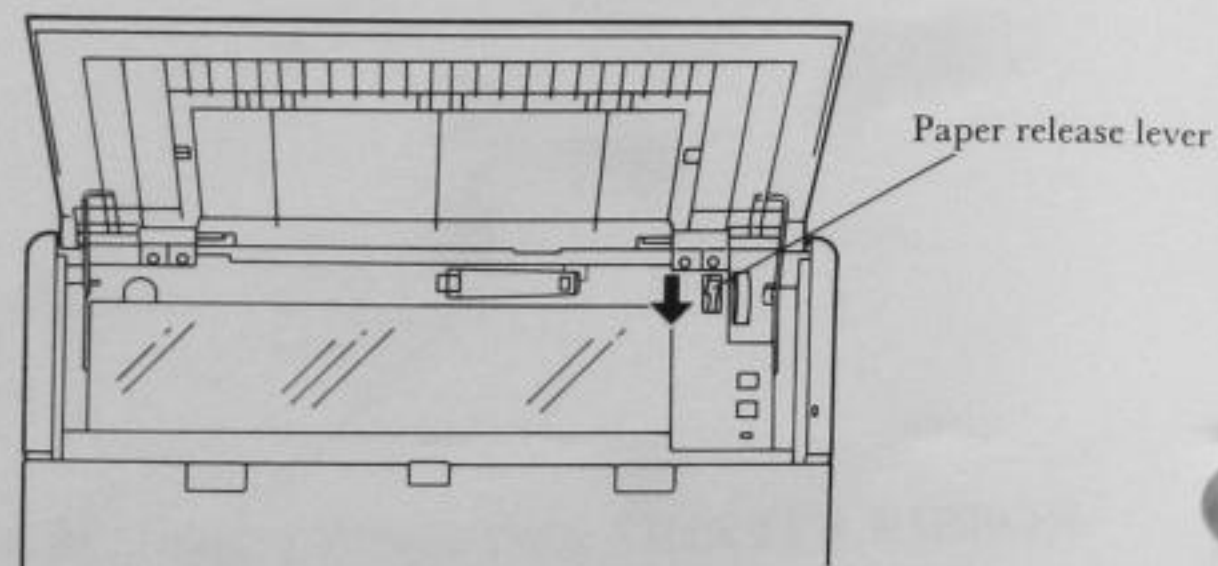


Fig. 2-32. THE PAPER RELEASE LEVER

- STEP 4 Insert the end of the paper into the slit in front of the lower edge of the cover as far as it will go.

- STEP 5 Push the paper release lever so that the paper is held firmly and turn the paper feed knob to advance the paper to the desired position.

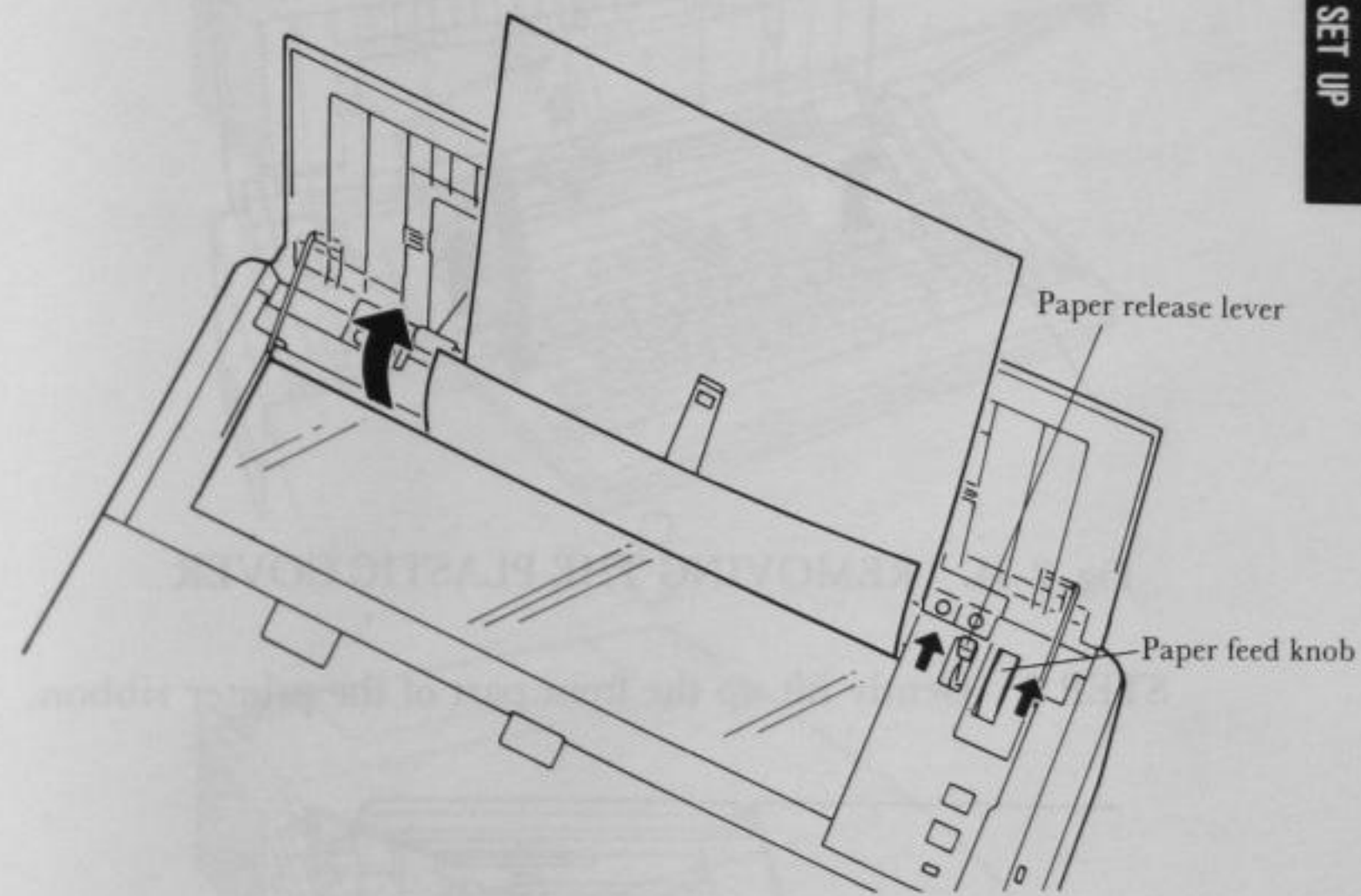


Fig. 2-33. INSERTION OF THE THERMAL TRANSFER PAPER

NOTE: When out of paper, the printing operation starts again by pressing the line feed button after the inserting paper.

Removing the printer ribbon

STEP 1 Remove the plastic cover by sliding it to the back of the computer and lifting up.

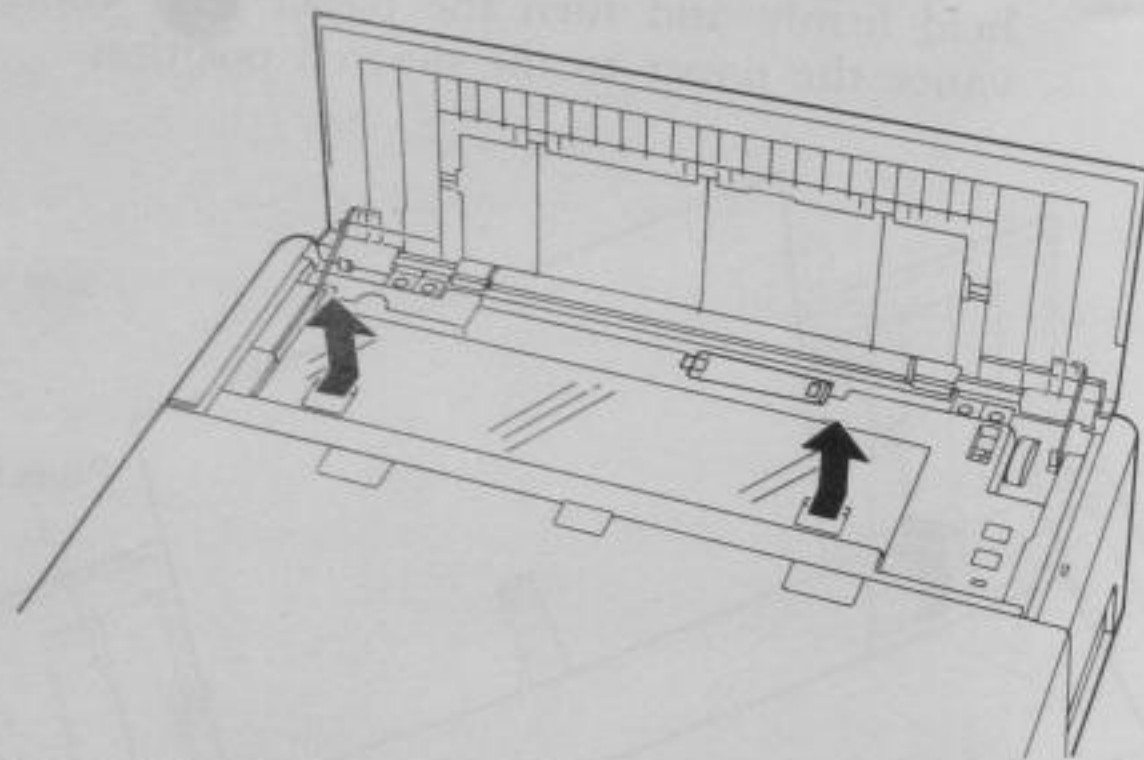


Fig. 2-34. REMOVING THE PLASTIC COVER

STEP 2 Gently lift up the front part of the printer ribbon.

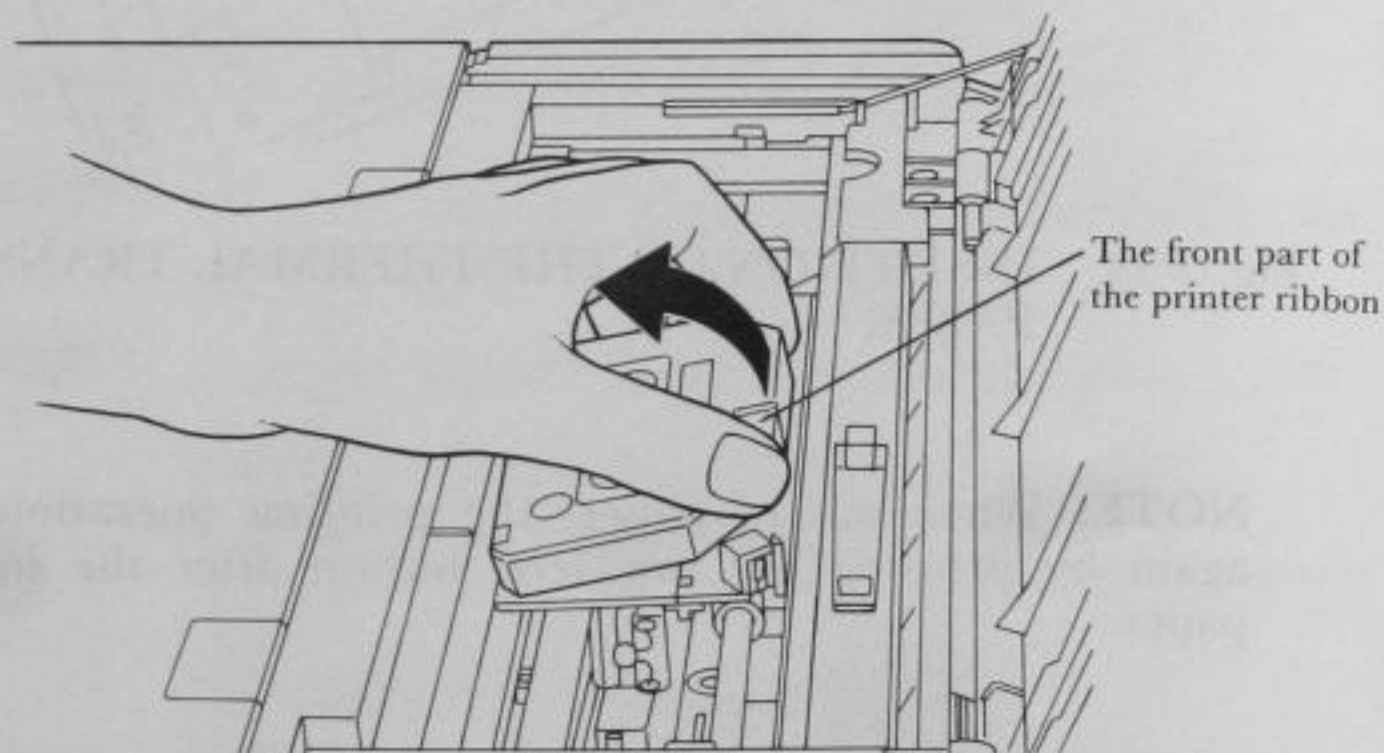


Fig. 2-35. REMOVING THE PRINTER RIBBON

STEP 3 Lift up the back part of the printer ribbon and remove it.

STEP 4 Attach the plastic cover.

STEP 3 Fold down the display panel to close. Press the display panel down by your both hands until the lock clicks.

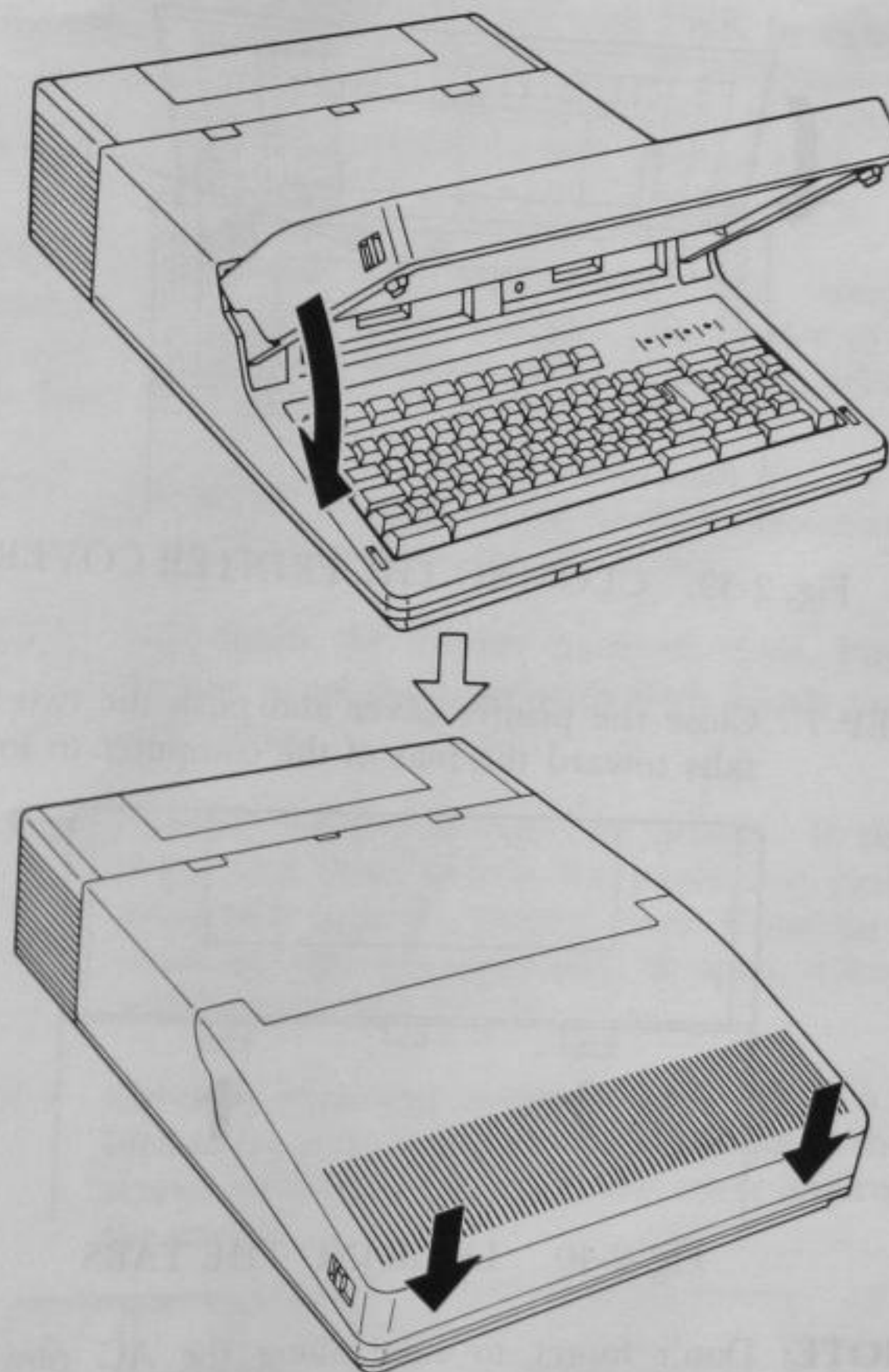


Fig. 2-38. FOLDING AND LOCKING THE DISPLAY PANEL

STEP 4 Remove the paper from the printer. If thermal paper was used, detach the paper rod (with the paper roll) from the printer cover. Close the inside cover in the printer cover, if open.

STEP 5 Return the paper guide to its original (folded down) position.

STEP 6 Attach the plastic cover if detached.

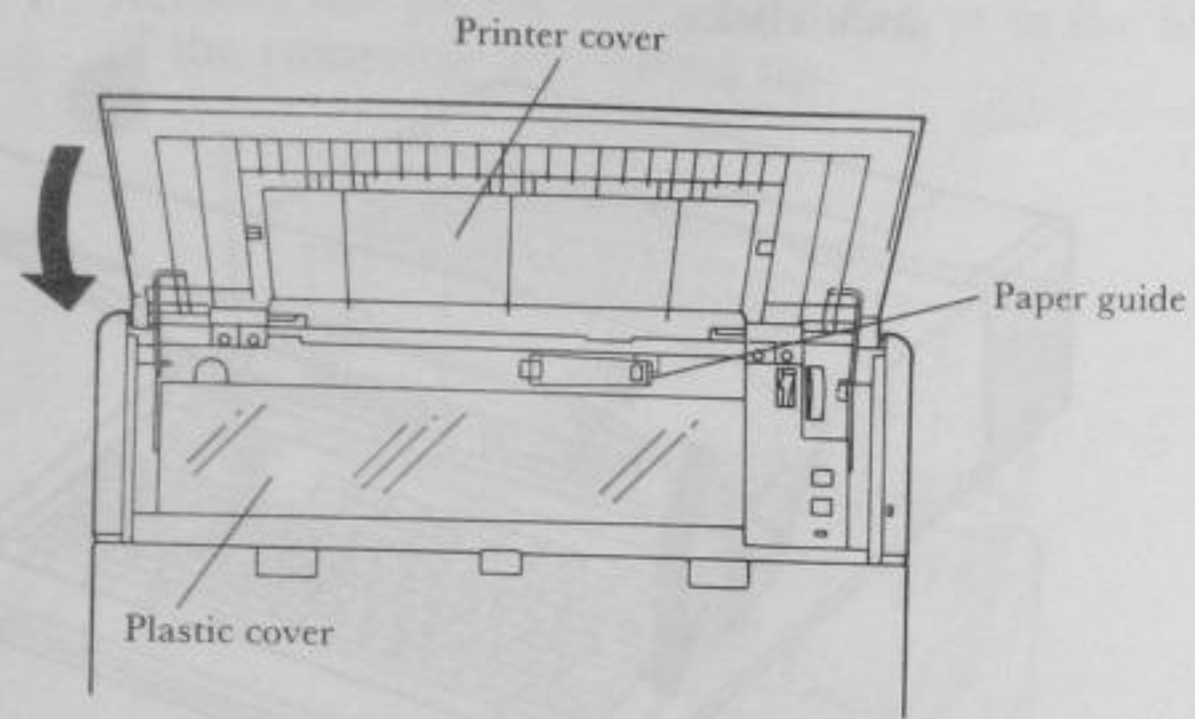


Fig. 2-39. CLOSING THE PRINTER COVER

STEP 7 Close the printer cover and push the two ribbed-tabs toward the rear of the computer to lock it.

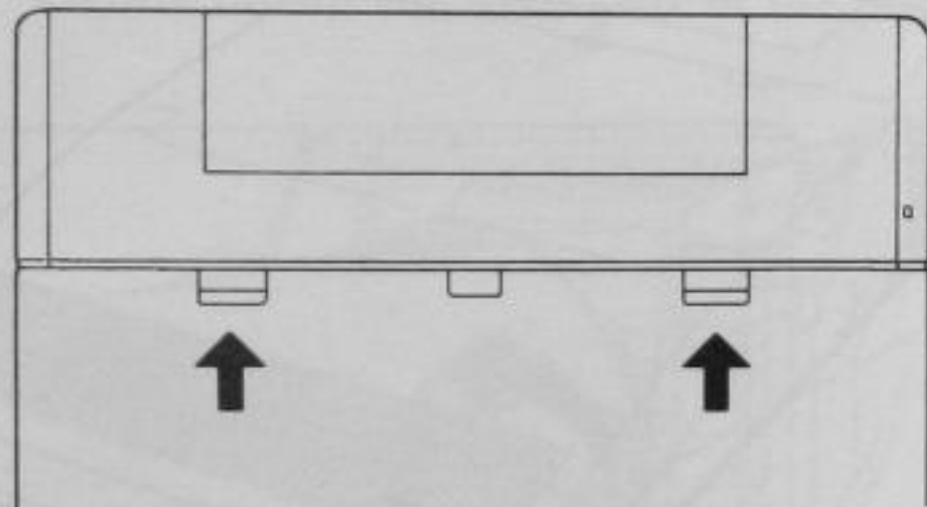


Fig. 2-40. LOCKING THE TABS

NOTE: Don't forget to take along the AC power cord, thermal paper, thermal transfer paper, paper rod, printer ribbon and other accessories that you may need on your trip.

Your Exec. Partner is now ready to travel with you! However, treat your Exec. Partner with respect. Avoid rough handling. After all, you want to be sure that your Exec. Partner is ready to work with you when you reach your destination!

MEMORY EXPANSION

Your Exec. Partner is already equipped with 256K bytes of RAM (Random Access Memory) at the factory. If you require more memory, you can increase the memory capacity to 640K bytes. You will need a Phillips screwdriver and sixteen 256K DRAM chips.

NOTE: Unless you are experienced with computers, Panasonic recommends that you ask your dealer to install and test the expansion memory before you take delivery of your Exec. Partner.

STEP 1 BE SURE THE POWER IS OFF. Disconnect the Exec. Partner from the AC outlet.

STEP 2 Fold down the display panel to close. Press the display panel down by your both hands until the lock clicks.

STEP 3 Remove the paper from the printer. If thermal paper was used, detach the paper rod (with the paper roll) from the printer cover. Close the small panel in the printer cover, if open. Close the printer cover and lock it.

STEP 4 Carefully turn the computer over. (It is a good idea to lay it on a cloth or towel.) Remove the two screws securing the RAM access cover and remove the cover.

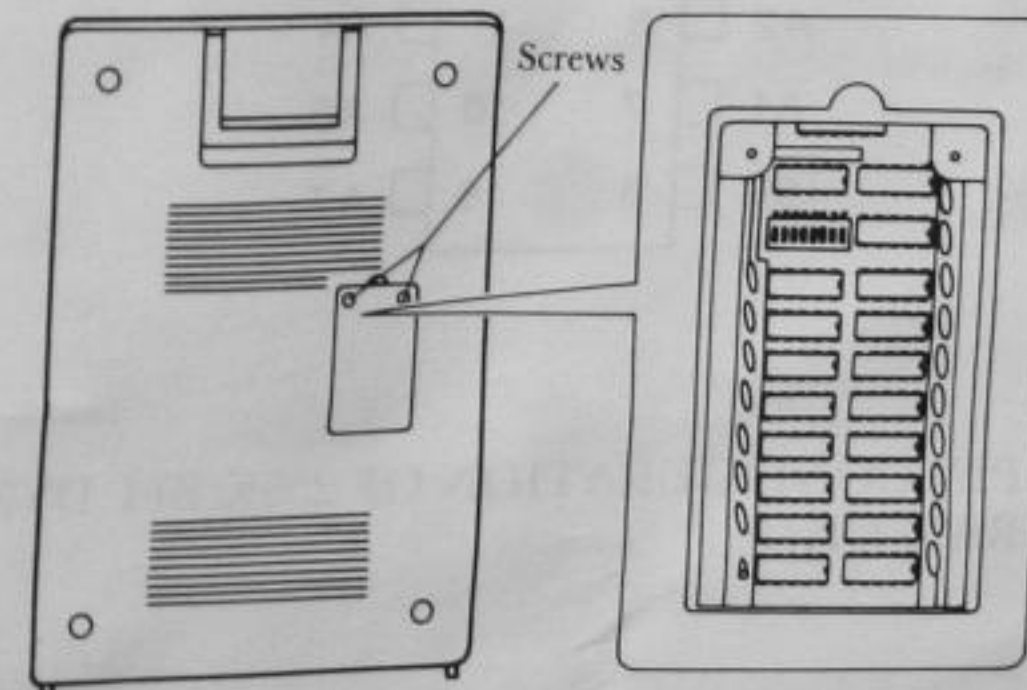


Fig. 2-41. RAM CHIPS

RAM chips

STEP 5 Remove the sixteen RAM chips (ICs in sockets).

STEP 6 Insert sixteen 256K bit RAM chips into the same sockets. Be sure the chips are oriented properly.

NOTE: All sixteen RAM chips are necessary to expand the Exec. Partner's memory. It must be expanded from 256K to the full 640K memory. These are the only available memory settings.

Specifications of 256K bit Dynamic RAM chip you can insert on the RAM board as listed below.

- Industry Standard 16-pin DIP
- 262144-word by 1 bit (256K bits)
- Single +5 V Supply
- Access time must be less than or equal to 120 ns
- Pin configuration is:

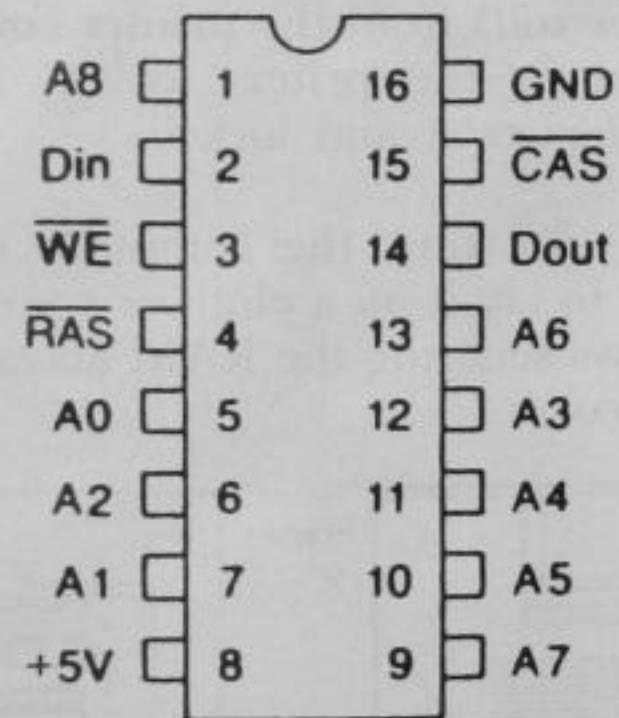


Fig. 2-42. PIN CONFIGURATION OF 256K BIT DYNAMIC RAM CHIP

STEP 7 Set DIP switch #2 (on the block of 8 switches) to the off position.

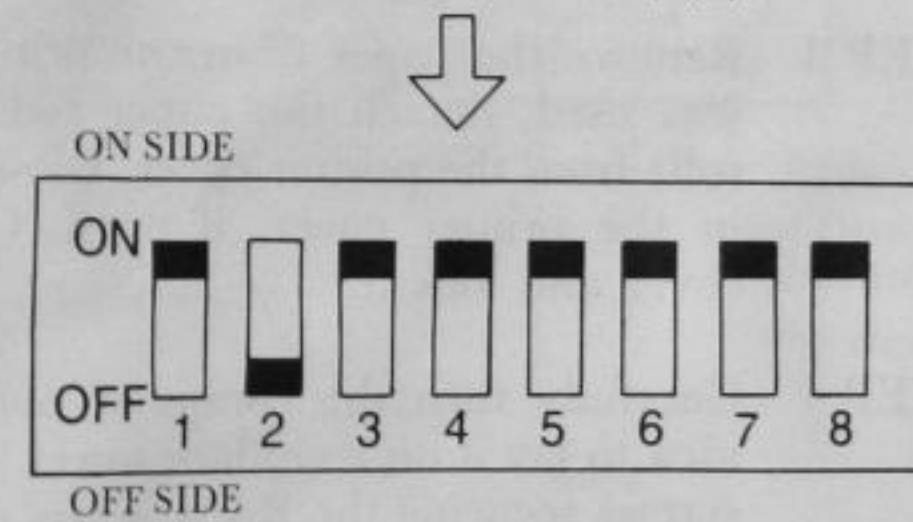
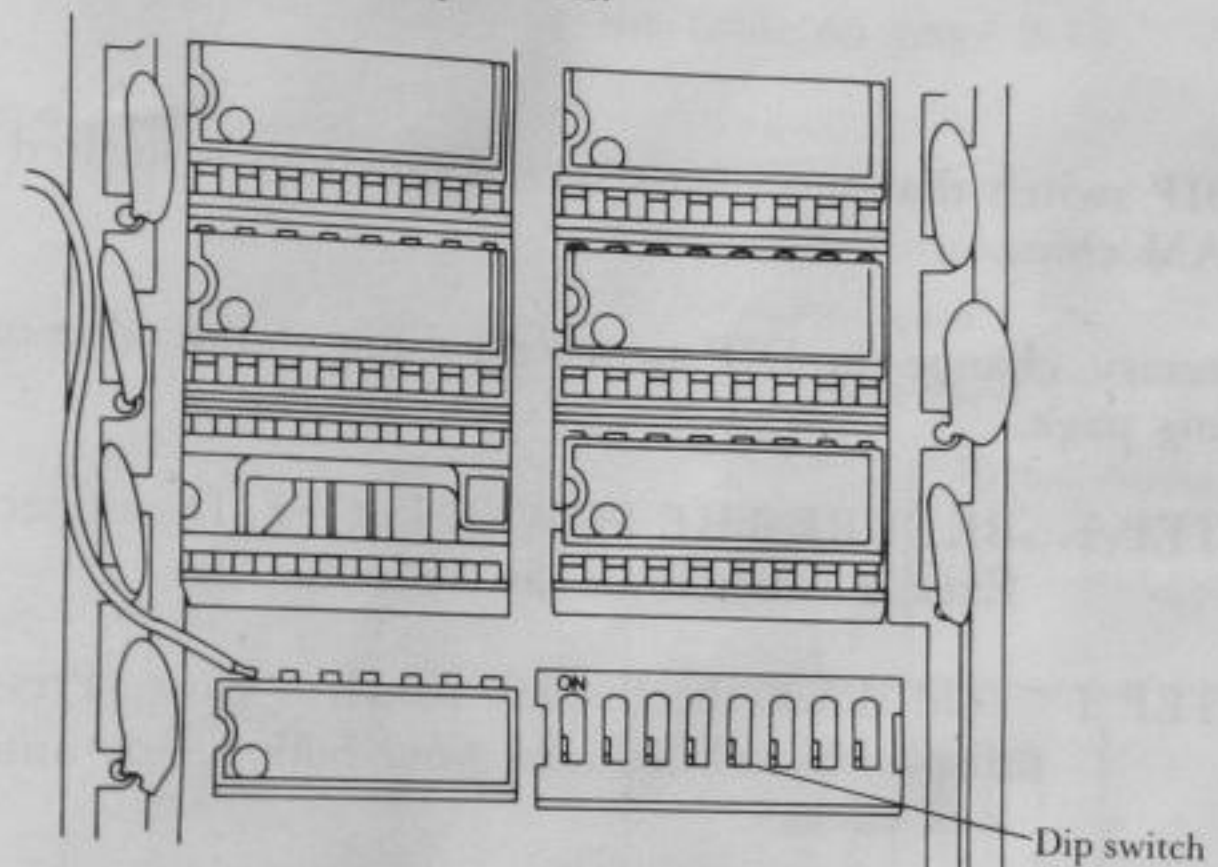


Fig. 2-43. SETTING THE DIP SWITCH

STEP 8 Attach the RAM access cover and secure with the two screws.

DIP SWITCH SETTING

The DIP switch that sets/clears several functions is located near the RAM chips.

If necessary, change the DIP switch according to the table on the following page.

- STEP 1** BE SURE THE POWER IS OFF. Disconnect the Exec. Partner from the AC outlet.
- STEP 2** Fold down the display panel to close. Press the display panel down by your both hands until the lock clicks.
- STEP 3** Remove the paper from the printer. If roll paper was used, detach the paper rod (with the paper roll) from the printer cover. Close the small panel in the printer cover, if open. Close the printer cover and lock it.
- STEP 4** Carefully turn the computer over. (It is a good idea to lay it on a cloth or towel.) Remove the two screws securing the RAM access cover and remove the cover.

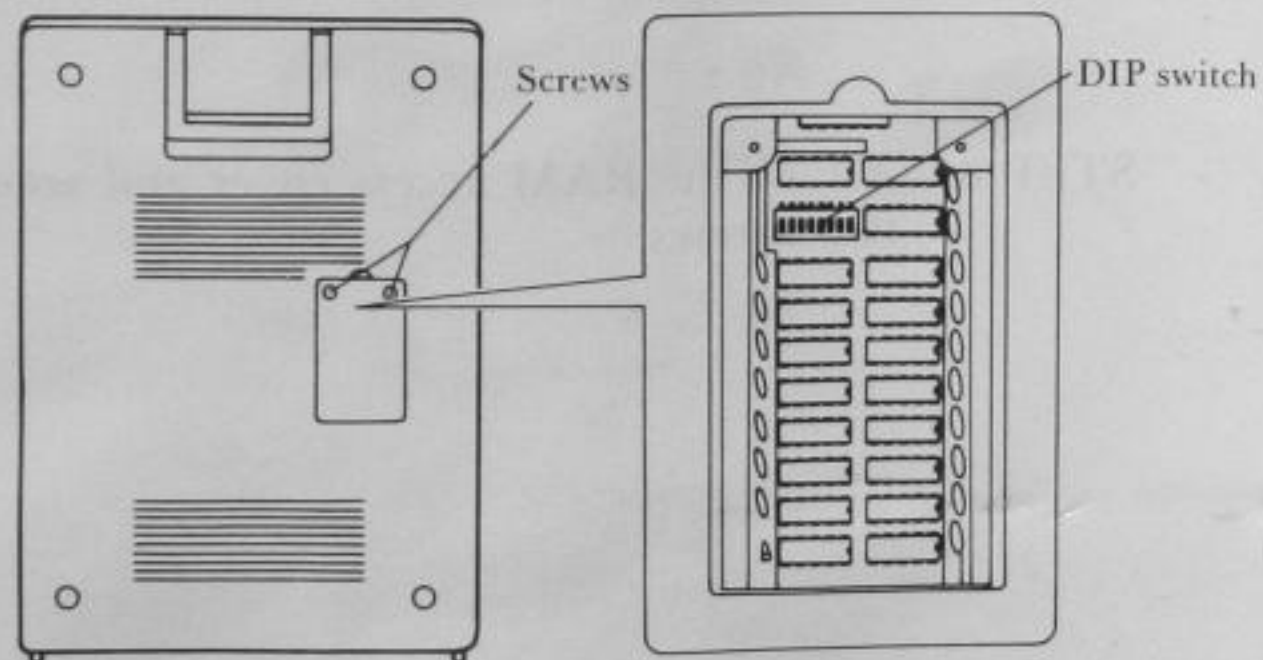
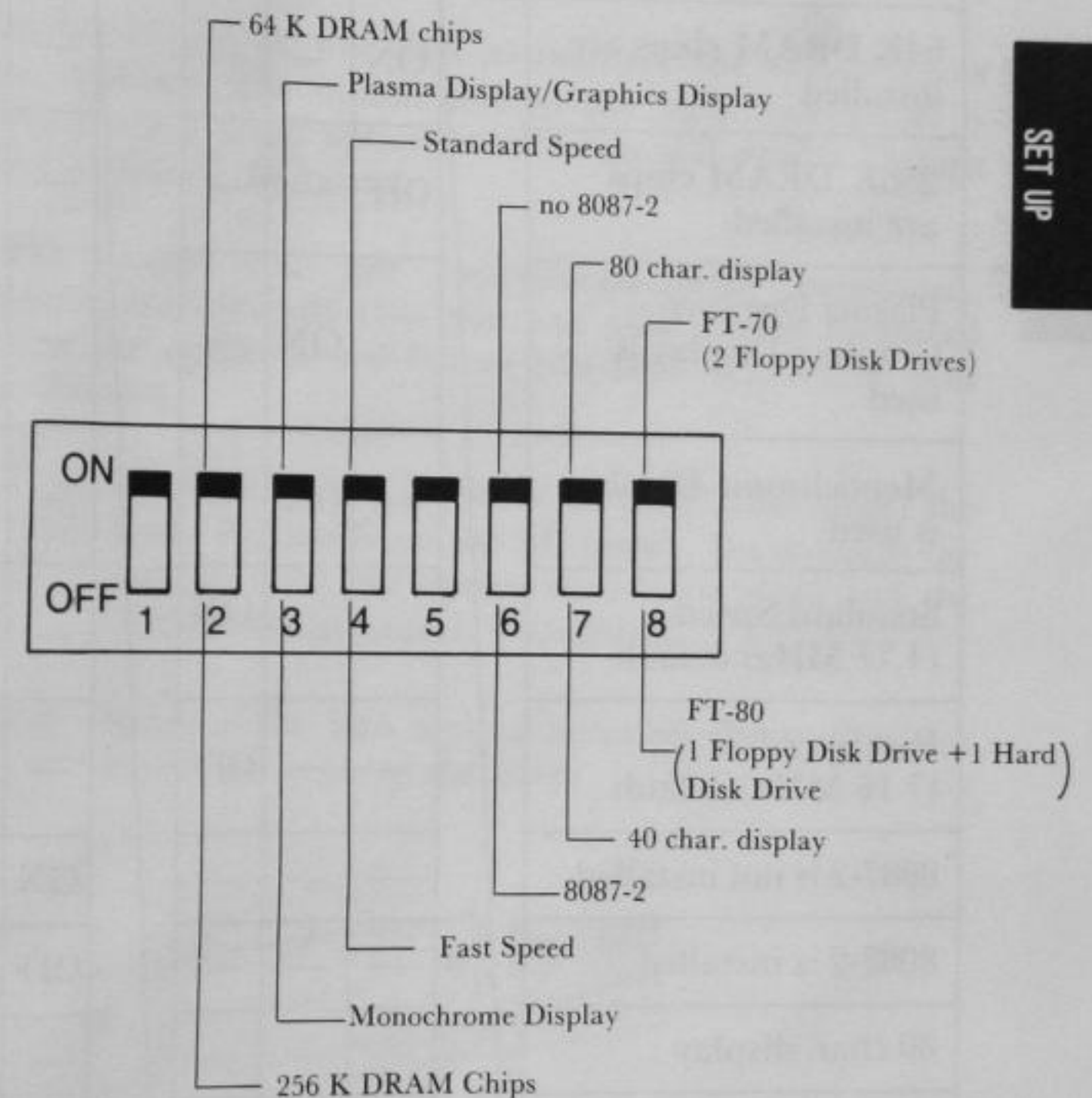


Fig. 2-44. DIP SWITCH

The Function of the DIP Switch

To set the DIP switch, see the table on page 2-44.



Function	DIP Switch Number							
	1	2	3	4	5	6	7	8
64K DRAM chips are installed		ON	—	—		—	—	—
256K DRAM chips are installed		OFF	—	—		—	—	—
Plasma Display/ Graphics Display is used		—	ON	—		—	—	—
Monochrome Display is used		—	OFF	—		—	—	—
Standard Speed (4.77 MHz) default		—	—	ON		—	—	—
Fast Speed (7.16 MHz) default		—	—	OFF		—	—	—
8087-2 is not installed		—	—	—	ON	—	—	—
8087-2 is installed		—	—	—	OFF	—	—	—
80 char. display		—	—	—	—	ON	—	—
40 char. display		—	—	—	—	OFF	—	—
FT-70 (2 Floppy Disk Drive)		—	—	—	—	—	—	ON
FT-80 (1 Floppy Disk Drive and 1 Hard Disk Drive)		—	—	—	—	—	—	OFF

STEP 6 Attach the RAM access cover and secure with two screws after setting the DIP switch.

OPTION BOARD INSTALLATION

An option board can be installed to increase the versatility of your Exec. Partner. The option slot in the right side panel is designed to accept a small option board. You will need a Phillips screwdriver to install the board.

NOTE: Unless you are experienced with computers, Panasonic recommends that you ask your dealer to install and test the option board before you take delivery of your Exec. Partner.

STEP 1 BE SURE THE POWER IS OFF. Disconnect the Exec. Partner from the AC outlet. Do not turn the unit on until the option card is in place and the installation process is completed.

STEP 2 Remove the two screws securing the option slot cover and remove the cover.

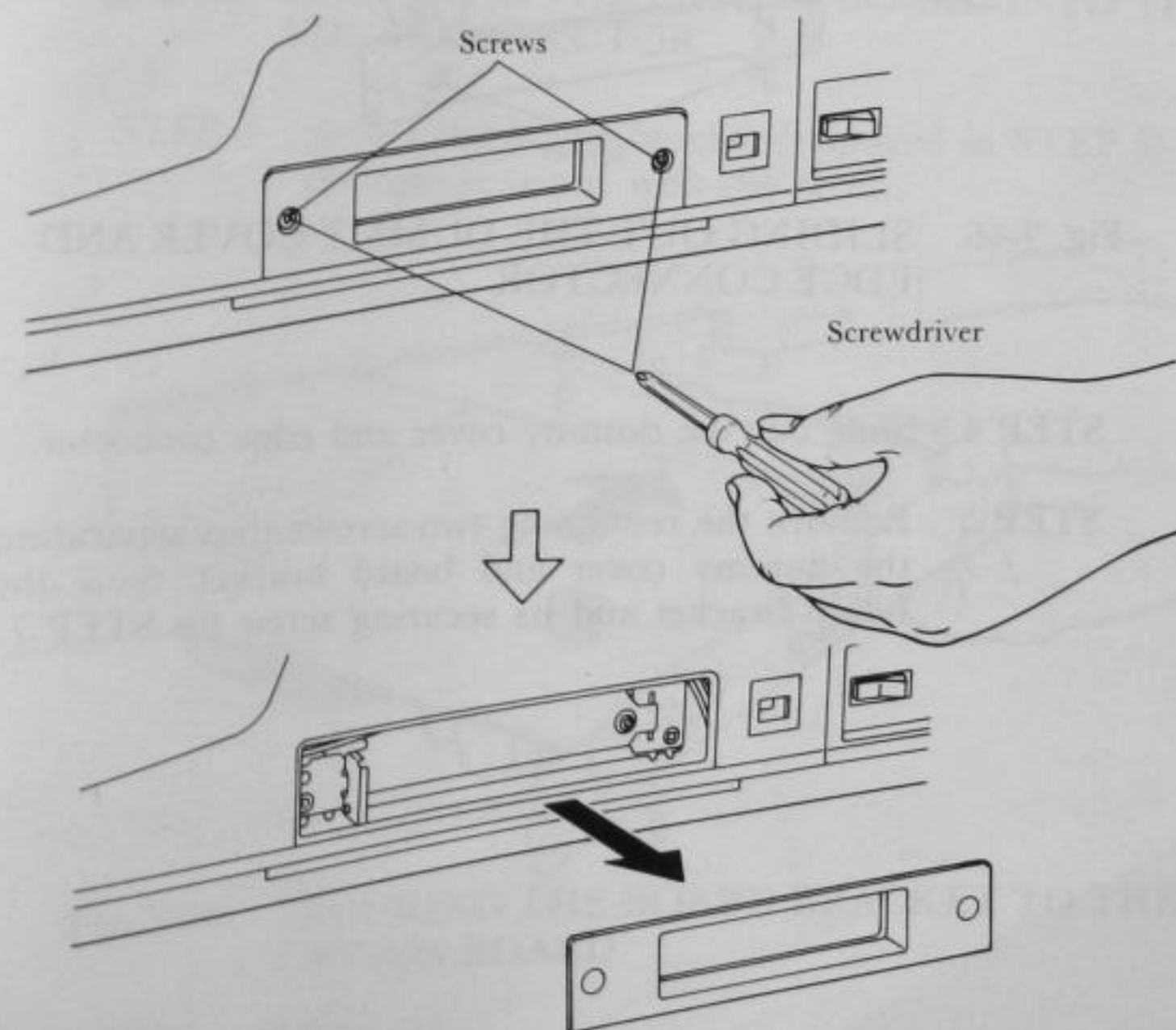


Fig. 2-45. REMOVING THE TWO SCREWS

STEP 3 Remove the two screws at opposite ends of the dummy cover.

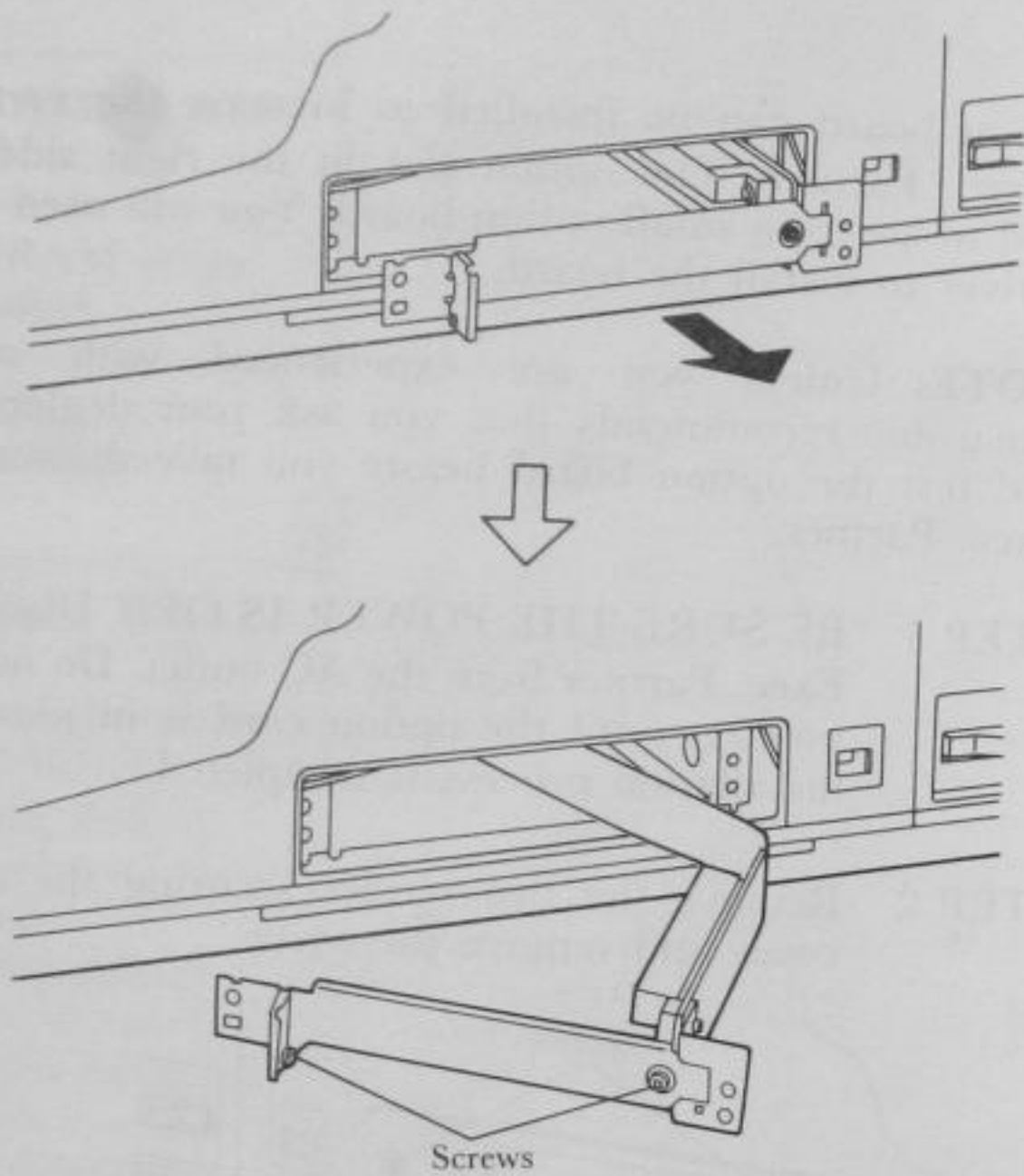


Fig. 2-46. SLIDING OUT THE DUMMY COVER AND EDGE CONNECTOR

STEP 4 Slide out the dummy cover and edge connector.

STEP 5 Remove the remaining two screws thus separating the dummy cover and board bracket. Save the board bracket and its securing screw for STEP 7.

STEP 6 Insert the option board into the edge connector.

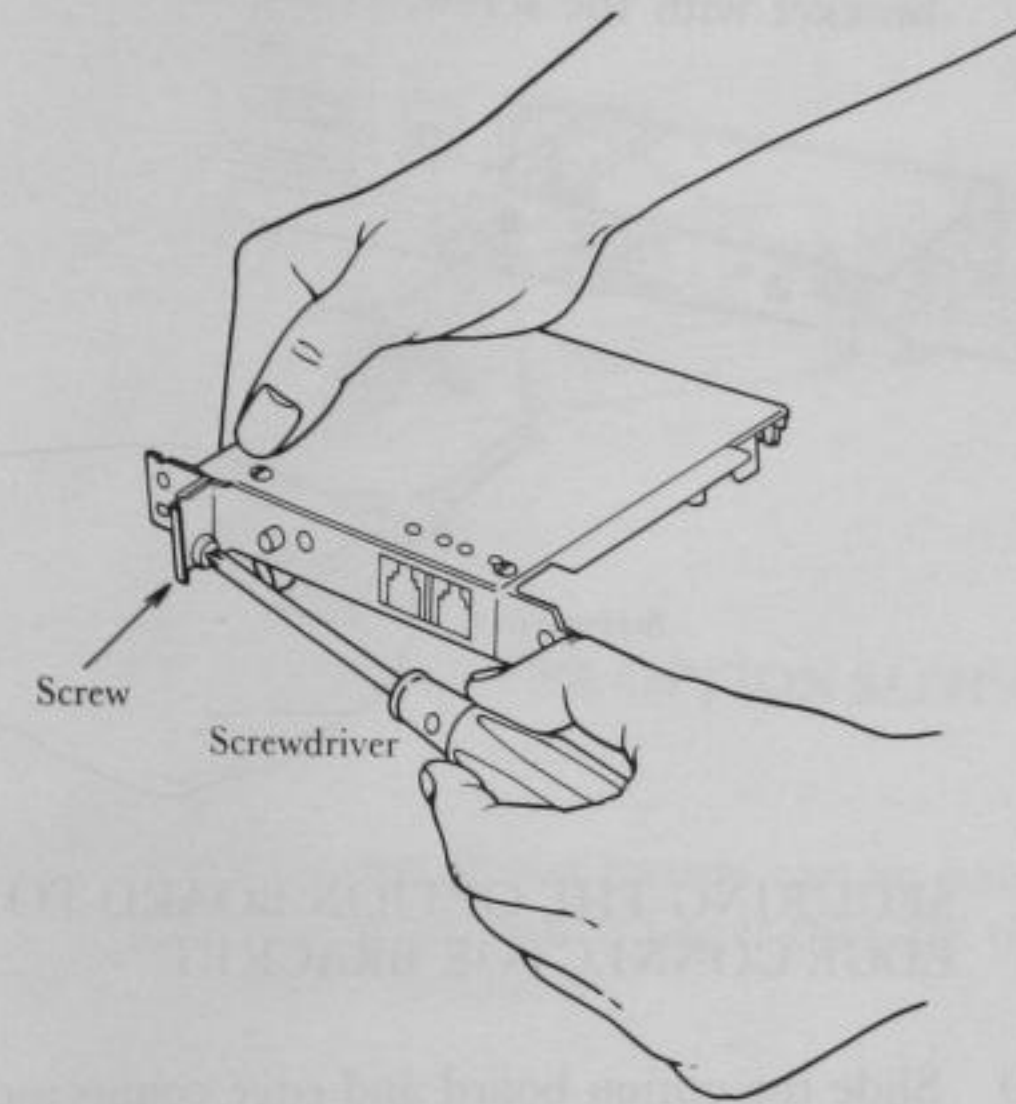


Fig. 2-47. INSERTING THE OPTION BOARD INTO THE EDGE CONNECTOR

STEP 7 Secure the board bracket (removed in STEP 5) to the option board with the screw.

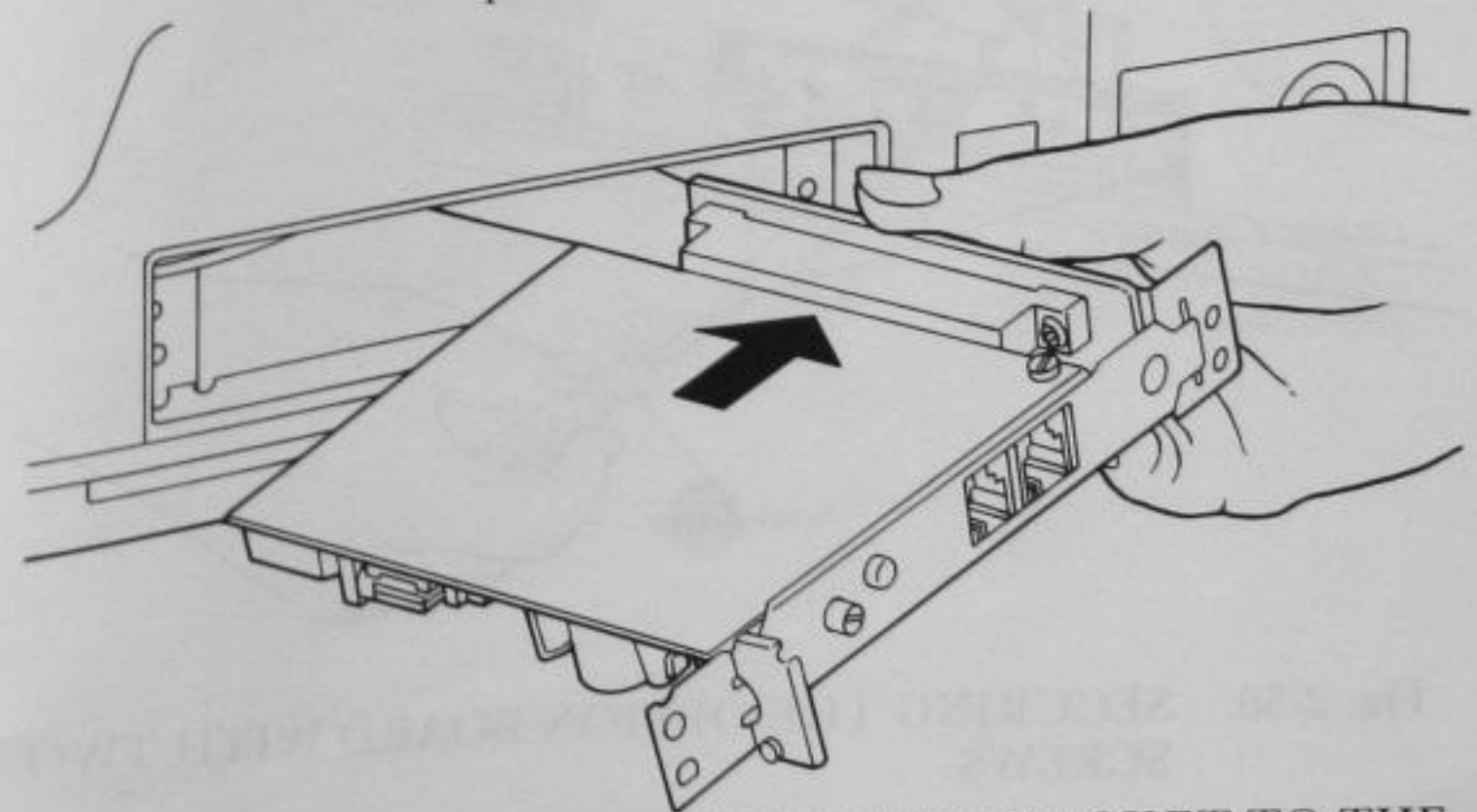


Fig. 2-48. SECURING THE BOARD BRACKET TO THE OPTION BOARD

STEP 8 Secure the option board to the edge connector bracket with the screw.

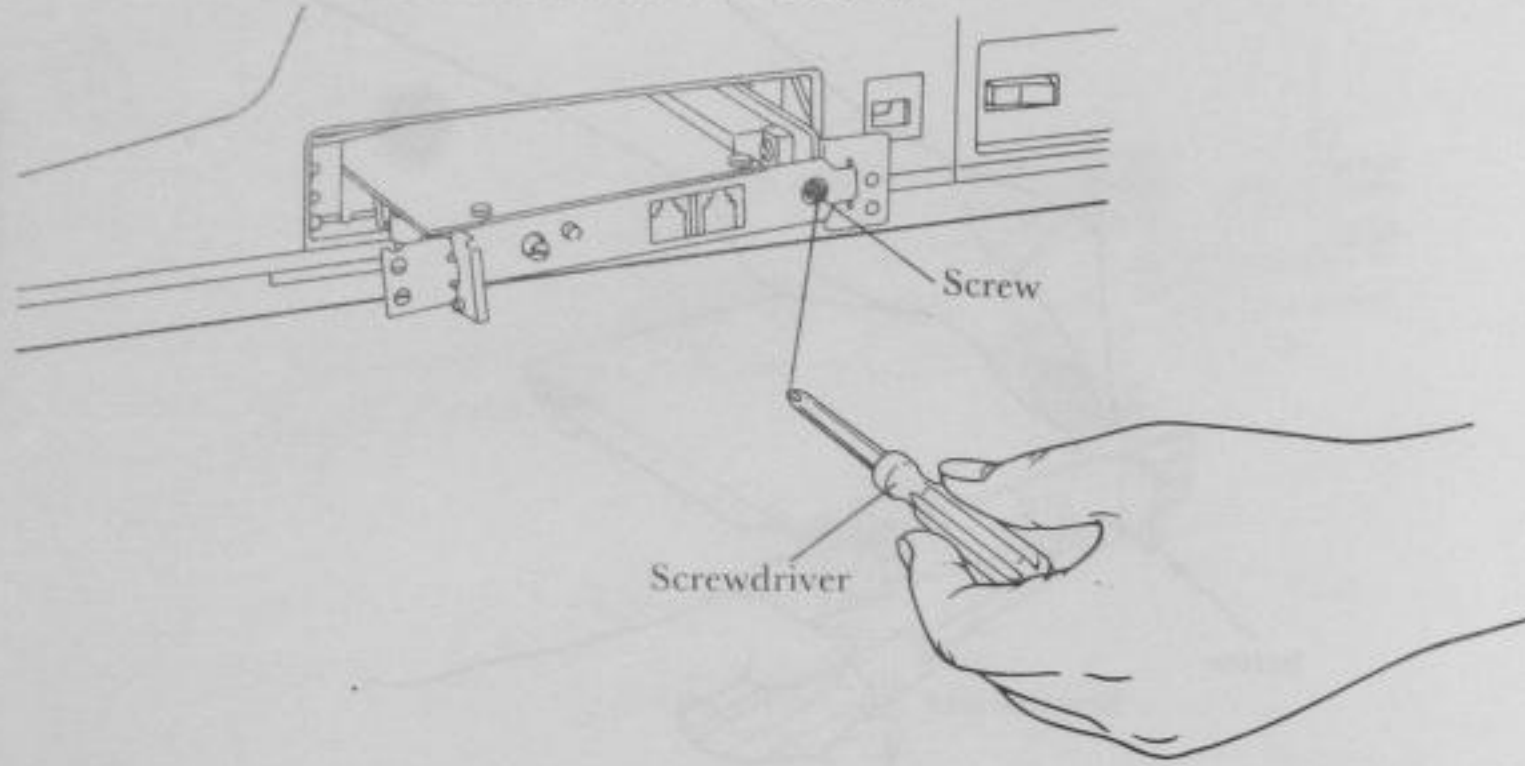


Fig. 2-49. SECURING THE OPTION BOARD TO THE EDGE CONNECTOR BRACKET

STEP 9 Slide the option board and edge connector into the computer.

STEP 10 Secure the option board with the two screws.

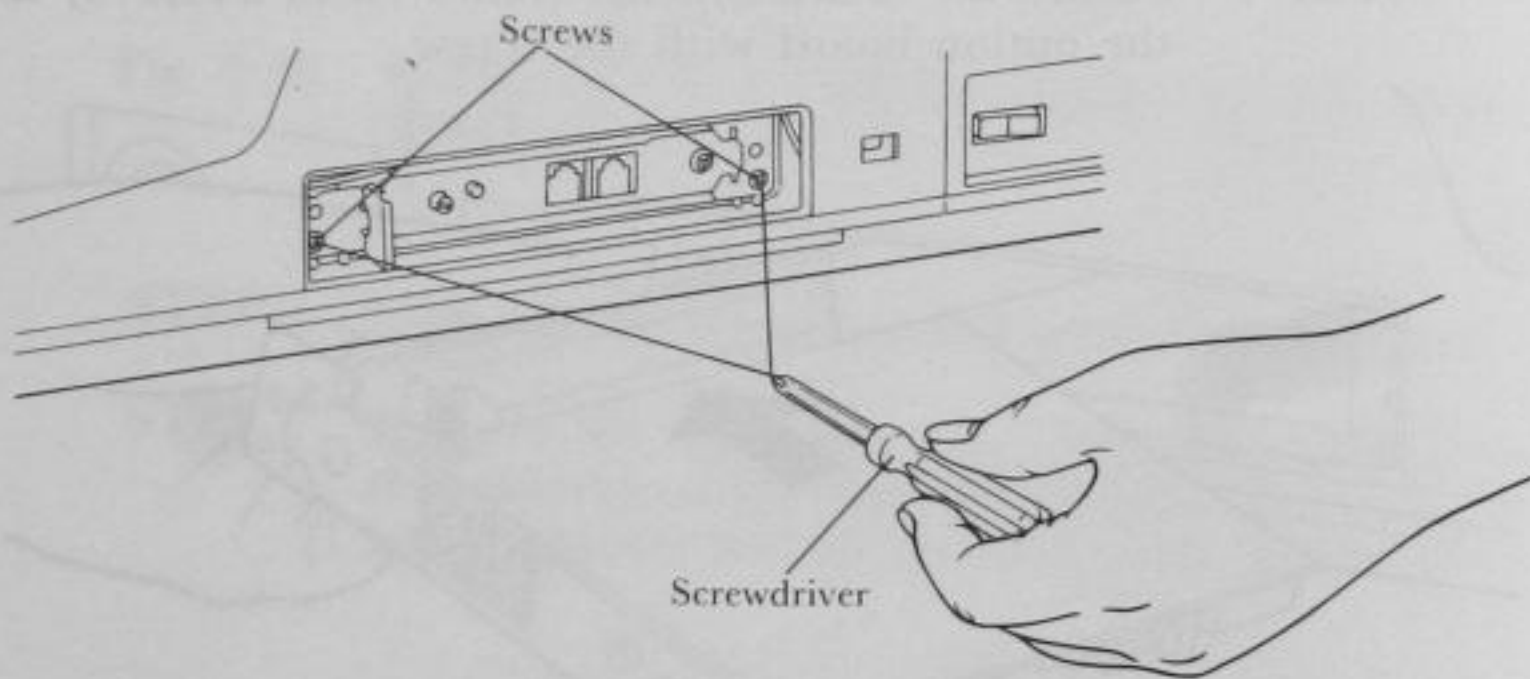


Fig. 2-50. SECURING THE OPTION BOARD WITH TWO SCREWS

STEP 11 Attach the option slot cover and secure with the two screws.

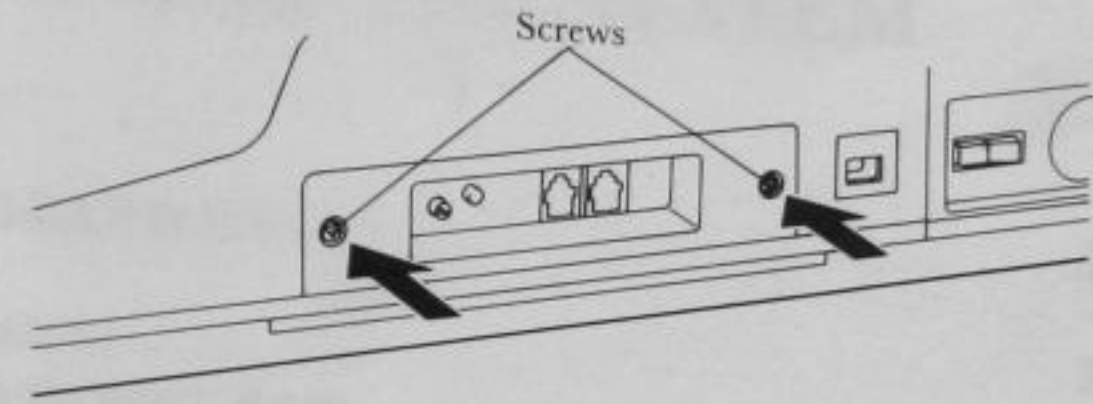
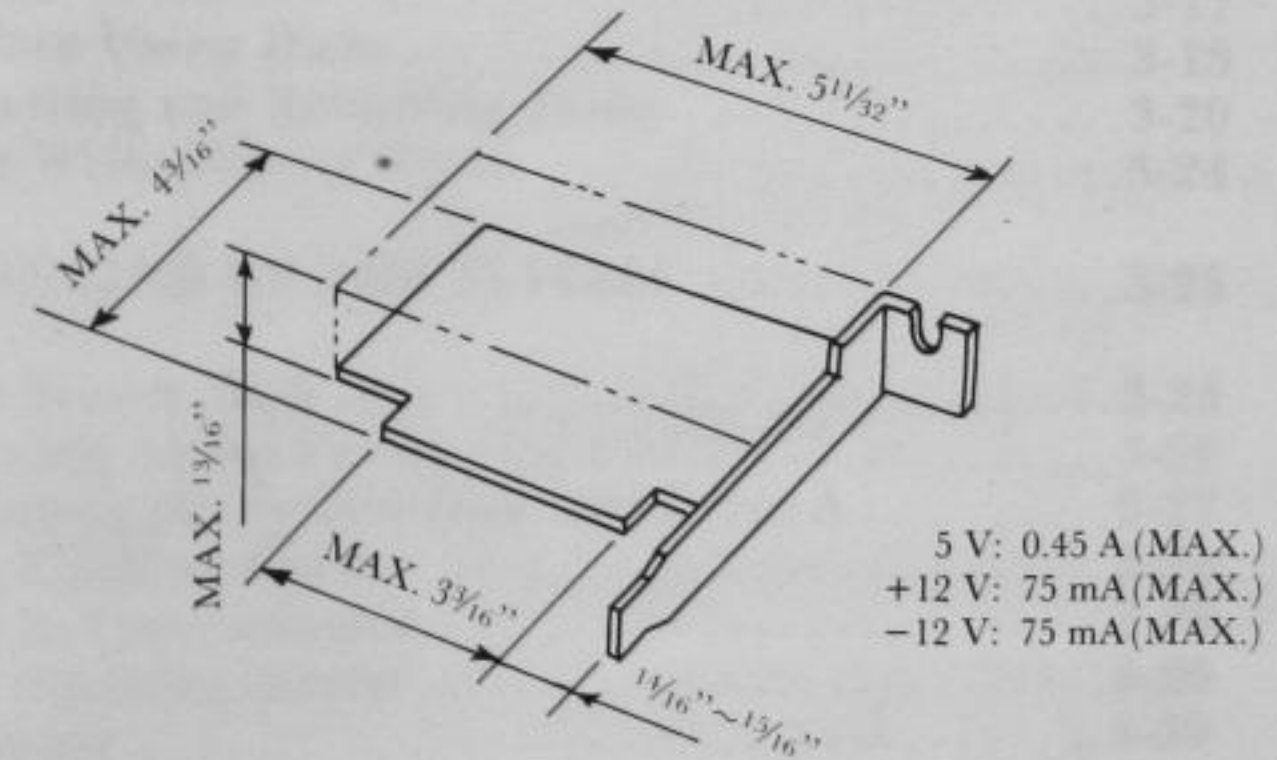


Fig. 2-51. ATTACHING THE OPTION SLOT COVER

NOTE: The following option boards can be installed on the Exec. Partner. Any short boards larger than the one shown in the figure cannot be installed.



CHAPTER 3 OPERATING THE SYSTEM

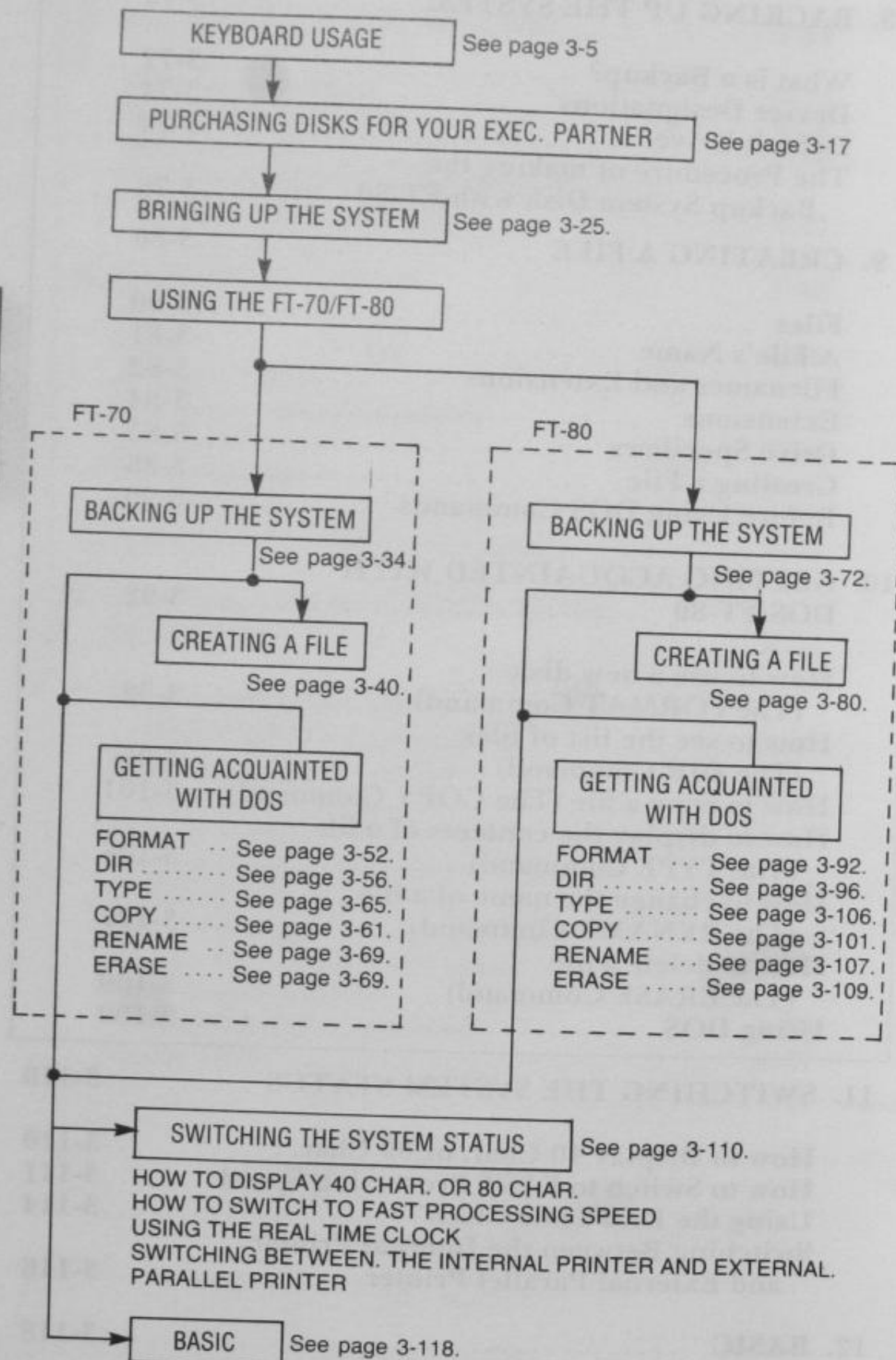
1. KEYBOARD USAGE	3-5
Indicators	3-5
Alphanumeric Keys	3-7
Keys That Can Confuse You	3-8
Control Keys	3-9
Ten Key Pad	3-12
Cursor Control	3-13
Function Keys	3-16
System Restart	3-16
2. PURCHASING DISKS FOR YOUR EXEC. PARTNER	3-17
Disks	3-17
Before Using Disks	3-18
Inserting and Removing Disks	3-20
The Write-Protect Notch	3-24
3. BRINGING UP THE SYSTEM	3-25
The System Disk	3-25
Turning on the Power	3-26
Inserting the System Disk into Drive A	3-27
The Flashing Cursor	3-28
The In Use Indicator	3-28
The Opening Screen	3-29
Prompts	3-30
If the Opening Screen Doesn't Appear	3-30
The Real Time Clock	3-31
Entering the Date	3-31
Entering the Time	3-32
4. USING THE FT-70/FT-80	3-33

FOR USING FT-70

5. BACKING UP THE SYSTEM	3-34
What is a Backup?	3-34
Device Designations	3-35
Default Drive	3-36
The Procedure of making the Backup System Disk with FT-70	3-37
6. CREATING A FILE	3-40
Files	3-40
A File's Name	3-41
Filenames and Extensions	3-42
Extensions	3-44
Drive Specifiers	3-45
Creating a File	3-46
Before Using DOS Commands	3-50
7. GETTING ACQUAINTED WITH DOS:FT-70	3-52
How to use a new disk (The FORMAT Command)	3-52
How to see the list of files (The DIR Command)	3-56
How to copy a file (The COPY Command) ..	3-61
How to display the contents of a file (The TYPE Command)	3-65
How to change the name of a file (The RENAME Command)	3-67
How to delete a file (The ERASE Command)	3-69
Using DOS	3-71

FOR USING FT-80

8. BACKING UP THE SYSTEM	3-72
What is a Backup?	3-72
Device Designations	3-74
Default Drive	3-74
The Procedure of making the Backup System Disk with FT-80	3-76
9. CREATING A FILE	3-80
Files	3-80
A File's Name	3-81
Filenames and Extensions	3-82
Extensions	3-84
Drive Specifiers	3-85
Creating a File	3-86
Before Using DOS Commands	3-90
10. GETTING ACQUAINTED WITH DOS:FT-80	3-92
How to use a new disk (The FORMAT Command)	3-92
How to see the list of files (The DIR Command)	3-96
How to copy a file (The COPY Command) ..	3-101
How to display the contents of a file (The TYPE Command)	3-106
How to change the name of a file (The RENAME Command)	3-107
How to delete a file (The ERASE Command)	3-109
Using DOS	3-109
11. SWITCHING THE SYSTEM STATUS	3-110
How to Display 40 Char. or 80 Char.	3-110
How to Switch to FAST Processing Speed ..	3-111
Using the Real Time Clock	3-114
Switching Between the Internal Printer and External Parallel Printer	3-116
12. BASIC	3-118



KEYBOARD USAGE

Many of the keys of your Exec. Partner keyboard are familiar to you, if you often use a typewriter. Beware, however! Although some keys appear to be identical to typewriter keys, in some cases their function is different. The following section details keyboard usage in general. Study the keyboard now to familiarize yourself with it's operation.

NOTE: Specific programs can affect the operation of the keyboard. Check your Operating System and Applications Program manuals for specific keyboard functions.

Indicators

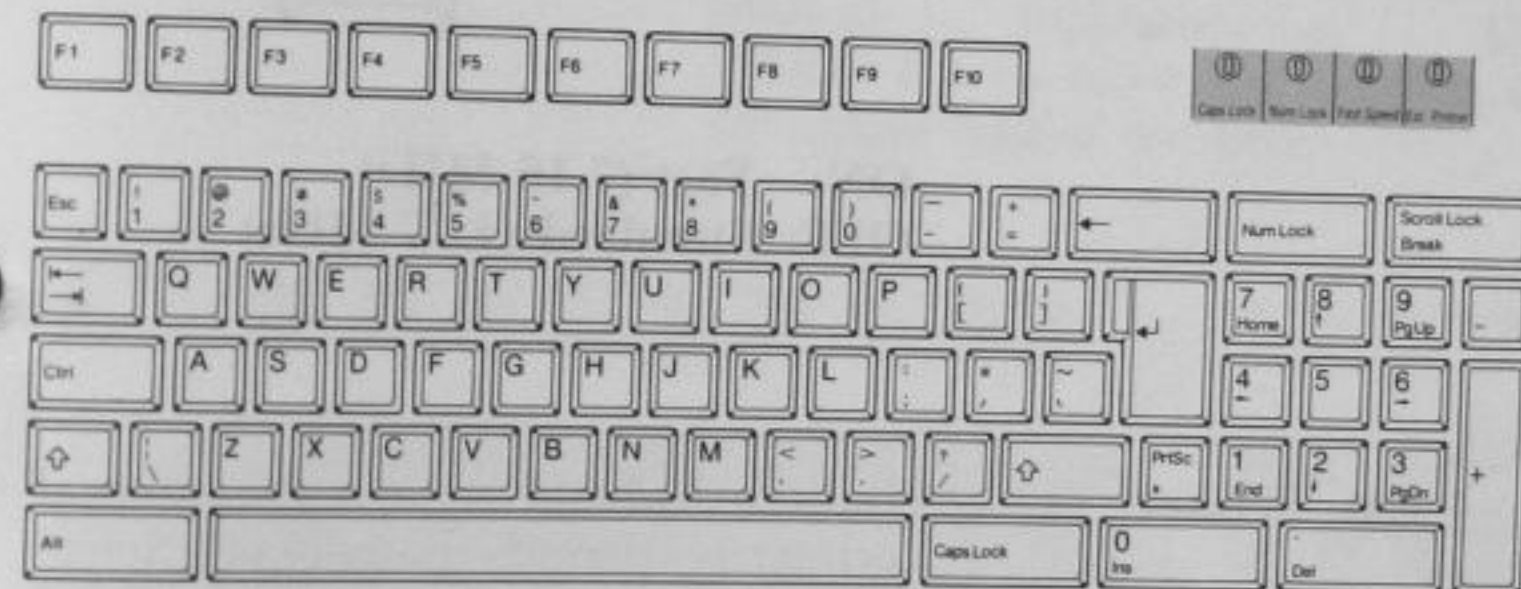


Fig. 3-1. INDICATORS

Above the <BACKSPACE>, <Num Lock> and <Scroll Lock> keys are four indicators (red). These indicators are provided to inform you of various modes.

The indicators are (from left to right):

Caps Lock

Indicates the mode of the Caps Lock key. (initially off)

ON: Upper case mode
OFF: Lower case mode

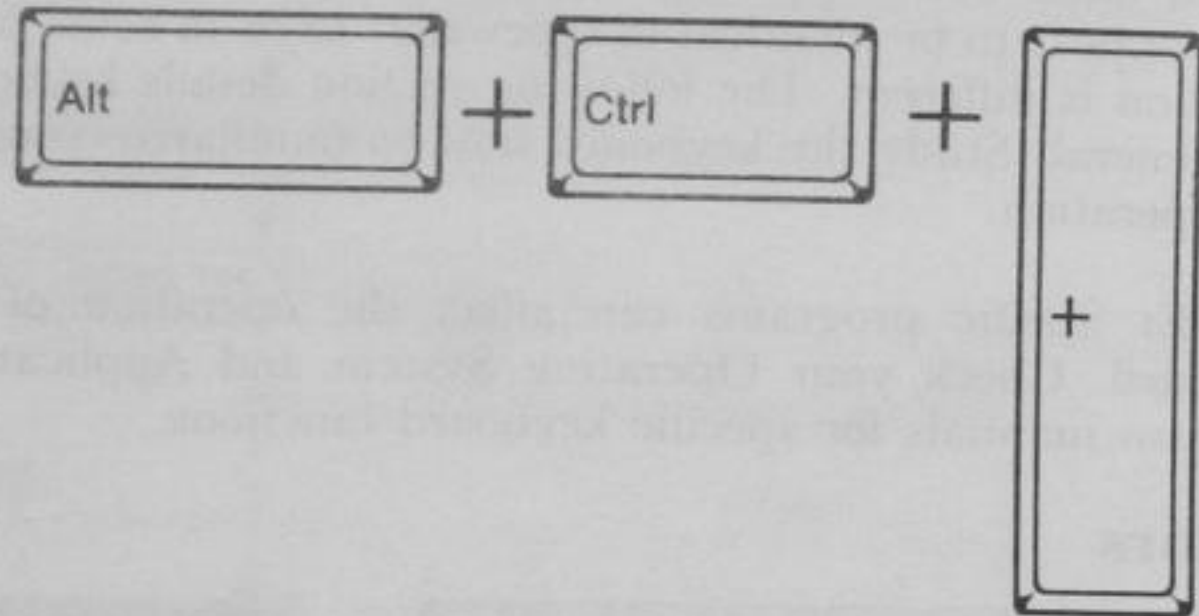
Num Lock

Indicates the mode of the cursor/numeric keys. (initially off)

ON: Numeric mode
OFF: Cursor mode

Fast Speed

Indicates the CPU clock speed. The speed is toggled between fast and standard by pressing



ON: Fast (7.16 MHz)
OFF: Standard (4.77 MHz)

Ext. Printer

Indicates whether or not the internal printer is selected. (initially off) Selection of the internal printer is toggled by pressing



ON: External printer (parallel port) selected
OFF: Internal printer selected

Alphanumeric keys

A-Z

Alphabet keys. These keys operate in the same manner as typewriter keys. Normally, these keys enter letters in the lower case. Use <SHIFT> (indicated by vertical arrows) to enter uppercase letters.

1-9, 0

Number keys. These keys operate in the same manner as typewriter keys. Use <SHIFT> keys to enter the symbols above the numbers.

Special Character keys. These keys are used for punctuation and mathematical representation. Use <SHIFT> to enter the symbols on the upper half of the keys.



Fig. 3-2. ALPHANUMERIC KEYS

Keys That Can Confuse You

The Spacebar. Press the spacebar to move the cursor to the right.

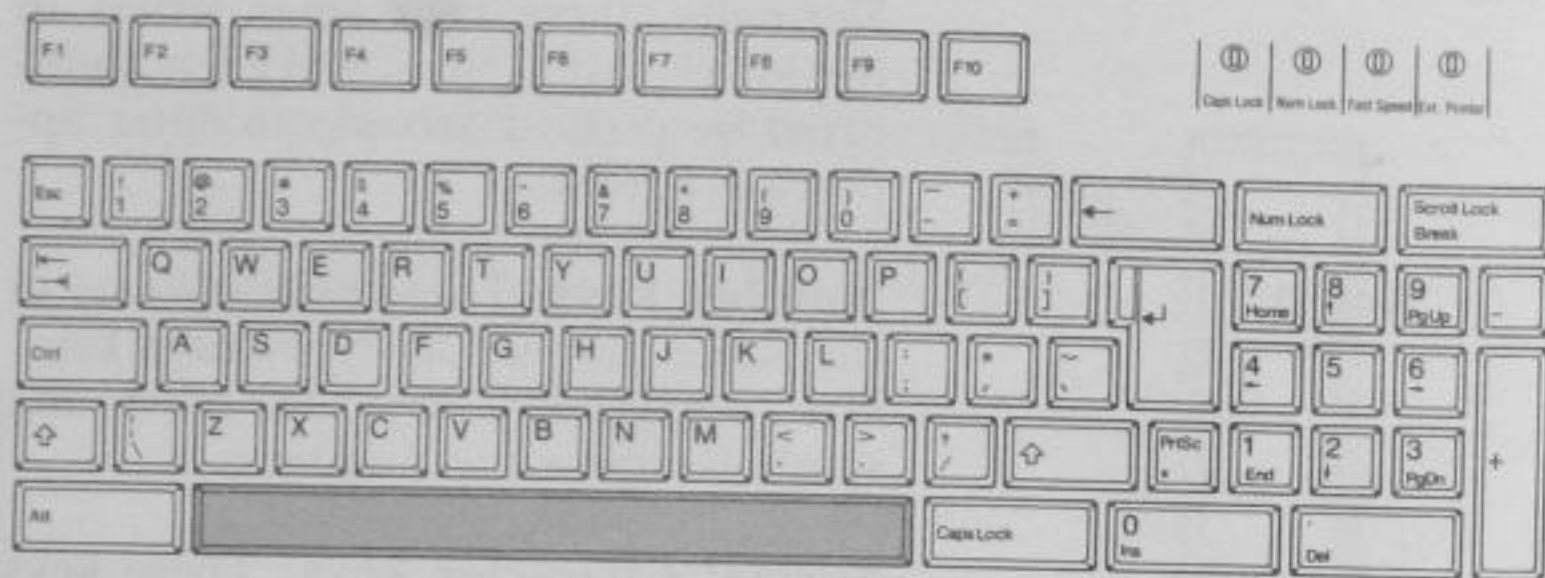
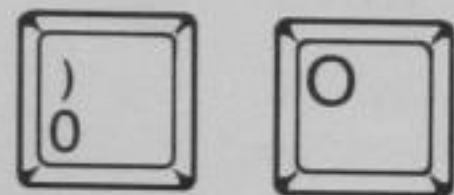
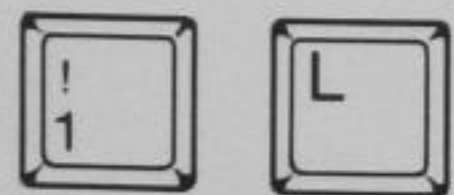


Fig. 3-3. SPACEBAR

NOTE: This key DOES NOT OPERATE like a spacebar on a typewriter. On a typewriter you use the spacebar to get to a specific location on the page. Nothing is altered on your typewritten page. The spacebar on your Exec. Partner also moves your location, but to the computer you have entered a new character, a blank character (invisible to you). Using the spacebar is like typing over a word. IT ERASES THE CHARACTERS AS IT MOVES, AND REPLACES THEM WITH BLANK CHARACTERS.



The Letter O and the Number 0. Although these keys are interchangeable on the typewriter, they are different characters to your computer. You can differentiate these two keys because the 0 has a diagonal slash running through it.



The Lowercase L and the Number 1. Like the O and 0, these keys mean quite different things to your computer. Be sure you enter the correct key.

Control Keys

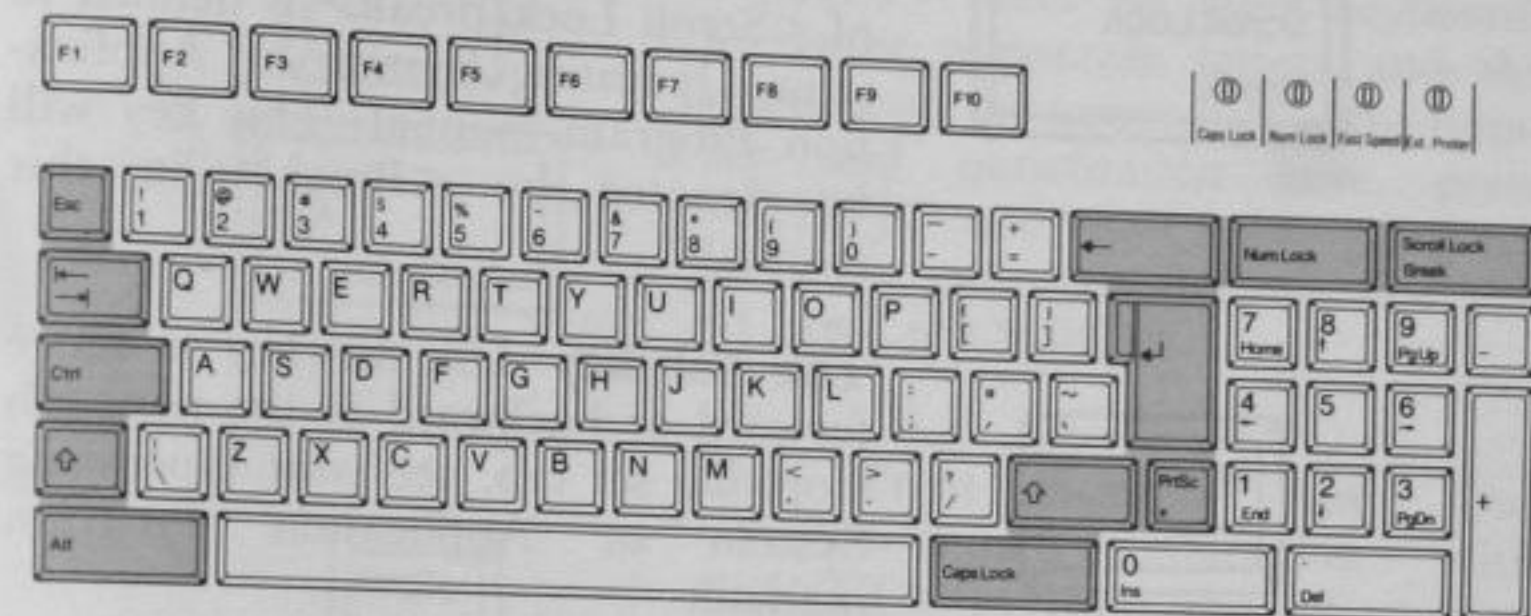
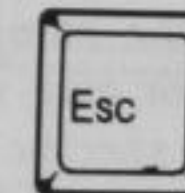
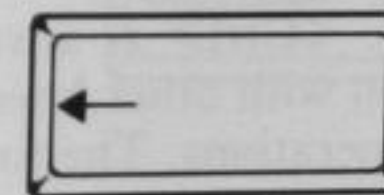


Fig. 3-4 CONTROL KEYS

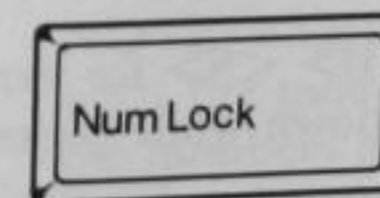
The control keys are arranged around the typewriter area. Keys are explained beginning with the top row.



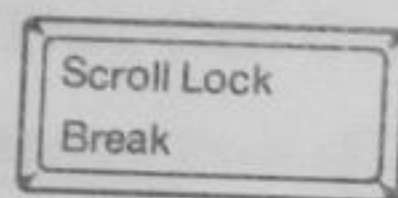
The Escape key. Like many control keys, the functions of <Esc> changes according to the program being run. See your Operating System or Application Program manual for the specific function of this key.



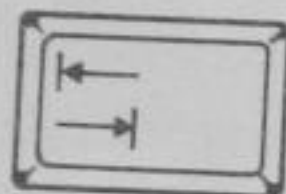
The Backspace key. This key erases the character to the left of the cursor. Use this key for making corrections.



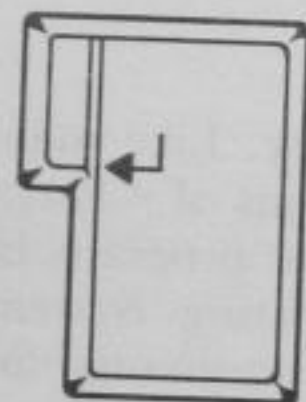
The Number Lock key. Use this key to lock the numeric keypad into the numeric mode. Pressing <Num Lock> again returns the numeric keypad to the cursor control mode. The Number Lock Indicator above the right side of the keyboard goes on by pressing the Number Lock key to get into the numeric mode.



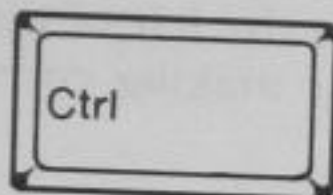
The Scroll Lock/Break key. Like the <Esc> key, the specific function of <Scroll Lock/Break> is defined in your Operating System or Application Program manual. This key will be denoted by <Break> in this manual.



The Tab key. Use this key to move forward or backward to the next tab stop. To set tab, see your Operating System or Application Program manual.

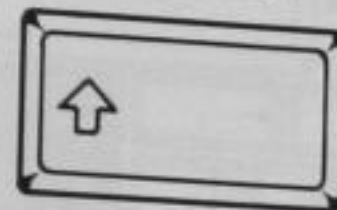
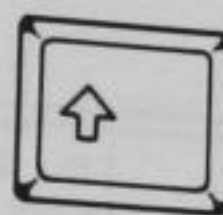


The ENTER key. THIS IS THE MOST IMPORTANT KEY ON YOUR KEYBOARD. In DOS and BASIC, this key signals the computer that you are finished entering data and are waiting for the computer to perform some action or give some response. Refer to each specific manual for an explanation of the <ENTER> function in the program you are using.

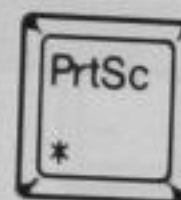


The Control key. <Ctrl> is always used in conjunction with other keys to perform specific operations. The function of this key varies according to the program definitions. See your Operating System or Application Program manual for details.

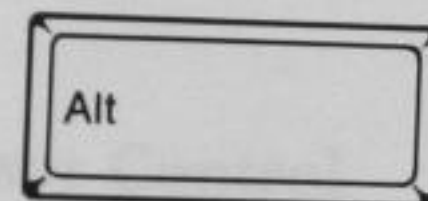
NOTE: When you see the notation <Ctrl> <S> for example, you must press <Ctrl> and hold it down while pressing <S> key.



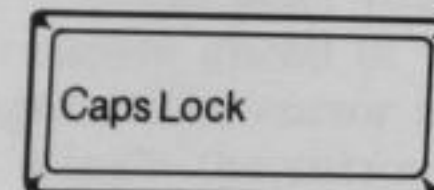
The SHIFT keys. There are two <SHIFT> keys on your keyboard. To enter uppercase letters and the symbols on the upper half of the number and punctuation keys, press <SHIFT> and while holding it down press the character desired. See also <Caps Lock> below.



The Print Screen Key. Use <PrtSc> in conjunction with <SHIFT> to print the information currently displayed on the screen. Pressing <PrtSc> without <SHIFT> enters the asterisk character.



The Alternate key. For information on this key, see your Operating System or Application Program manual.



The Caps Lock Key. <Caps Lock> lets you decide which case you would prefer as the normal entry mode. The first time you press <Caps Lock> you lock in the uppercase mode. You get lowercase letters by pressing <SHIFT>. The second time you press <Caps Lock> you lock into the lowercase mode. Pressing <SHIFT> then enters uppercase letters.

NOTE: <Caps Lock> only affects the letter keys. No matter what the mode of the <Caps Lock> key, the symbols above the numbers and the upper symbols on the punctuation keys require the use of <SHIFT>.

The Caps Lock Indicator above the right side of the keyboard goes on by pressing the Caps Lock key to get into the uppercase mode.

Numeric Pad

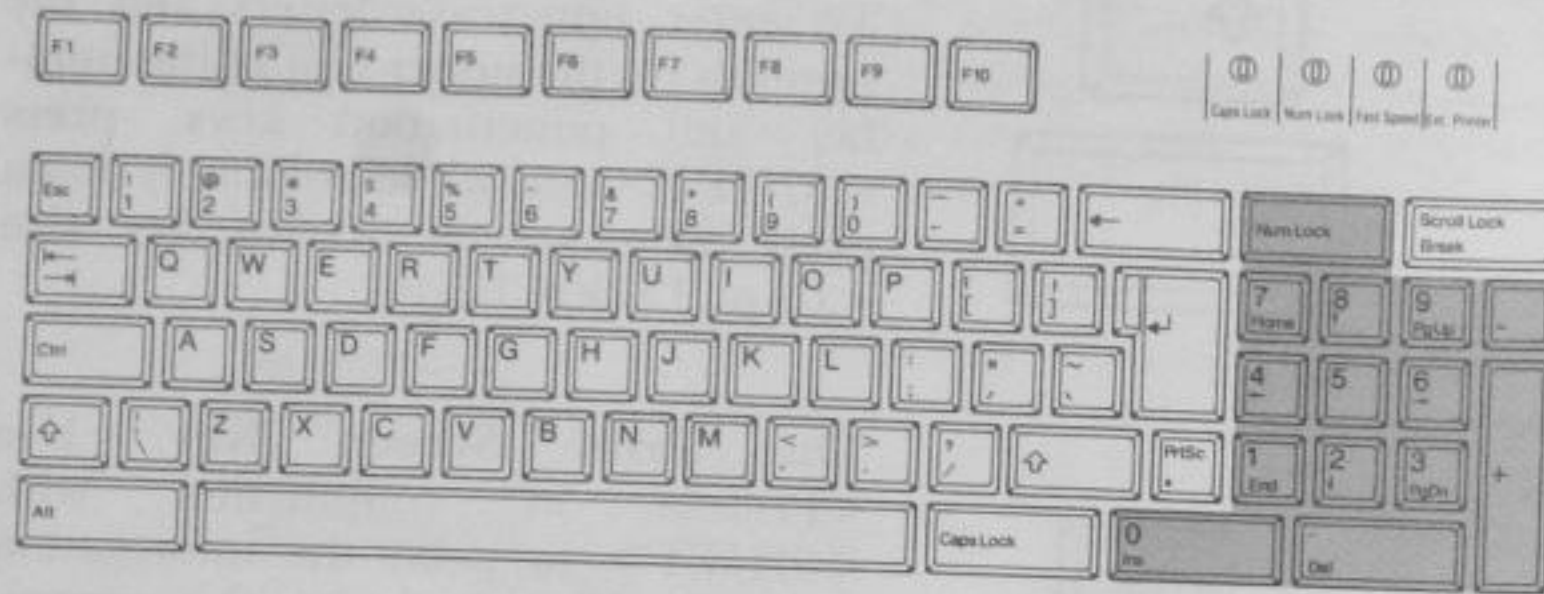
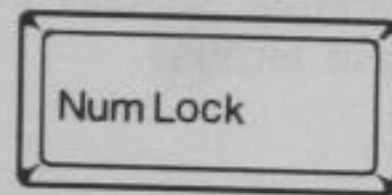


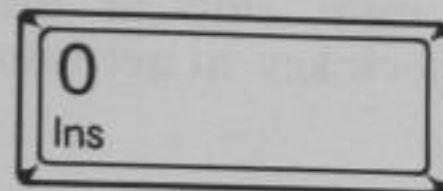
Fig. 3-5. NUMBER KEYS



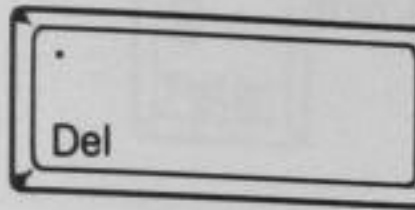
The Numeric keys. Use these keys to enter numeric data.



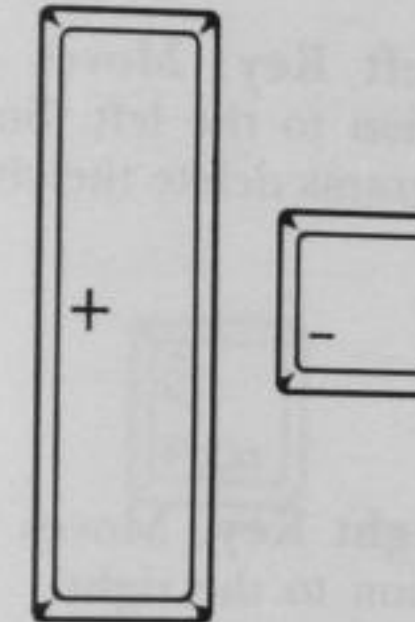
The Num Lock Key. This is the control key used in conjunction with the numeric keypad. Press <Num Lock> once and you are locked in the numeric mode. Press <Num Lock> again and you toggle back to the cursor control mode.



The Insert Key. Use <Ins> to insert characters in the middle of a line. When you use <Ins> characters are inserted **TO THE LEFT OF THE CURSOR**. Press <Ins> again to return to normal operation.



The Delete Key. Use to remove the character underlined by the cursor. When you use the current character and its space are deleted. All remaining characters to the right of the deleted position move one space to the left.



The Plus and Minus Keys. These keys have no programmed function. The display + or - on the screen.

Cursor Control

The cursor is a small blinking underline on the display which tells you your position on the screen. The cursor indicates the location of the next character you enter. When you begin to enter data, the cursor moves ahead of your typing and indicates the next available space. The cursor is also used in editing information. When using <Ins> the cursor indicates the space **BEFORE** which the character will appear. When using the cursor underlines the character to be deleted.

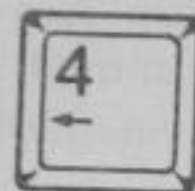
NOTE: Remember <Num Lock> determines the current mode of the numeric keypad. If you are locked in the numeric mode, you must press <Num Lock> before using these keys for cursor control.



The Cursor Up key. Moves the cursor one line up.



The Cursor Down key. Moves the cursor one line down.



The Cursor Left Key. Moves the cursor one position to the left. Some Application Programs delete the character.



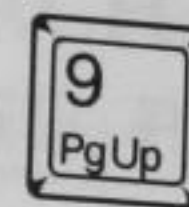
The Cursor Right Key. Moves the cursor one position to the right.



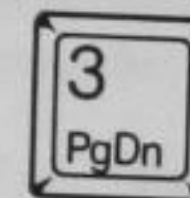
The Cursor Home key. Moves the cursor to the top left corner of the screen.



The Cursor End key. Moves the cursor to the bottom left of the screen.



The Page Up key. <PgUp> is a program controlled key, that it is may or may not be operable in the program you are using. When supported, <PgUp> allows you to "scroll" backwards in your file, looking at one "page" at a time. Check the specific manual for the program being executed for details on <PgUp> operation.



The Page Down key. Like <PgUp>, <PgDn> is a program controlled key. When supported, <PgDn> allows you to "scroll" forward in your file, one "page" at a time. Check your Application Program manual for further details on <PgDn> operation.

Some of these keys are available in the specific programs. See your Operating System or Application Program manual.

NOTE: DOS can support "The Cursor Left Key" and "The Cursor Right key" only.

Function Keys

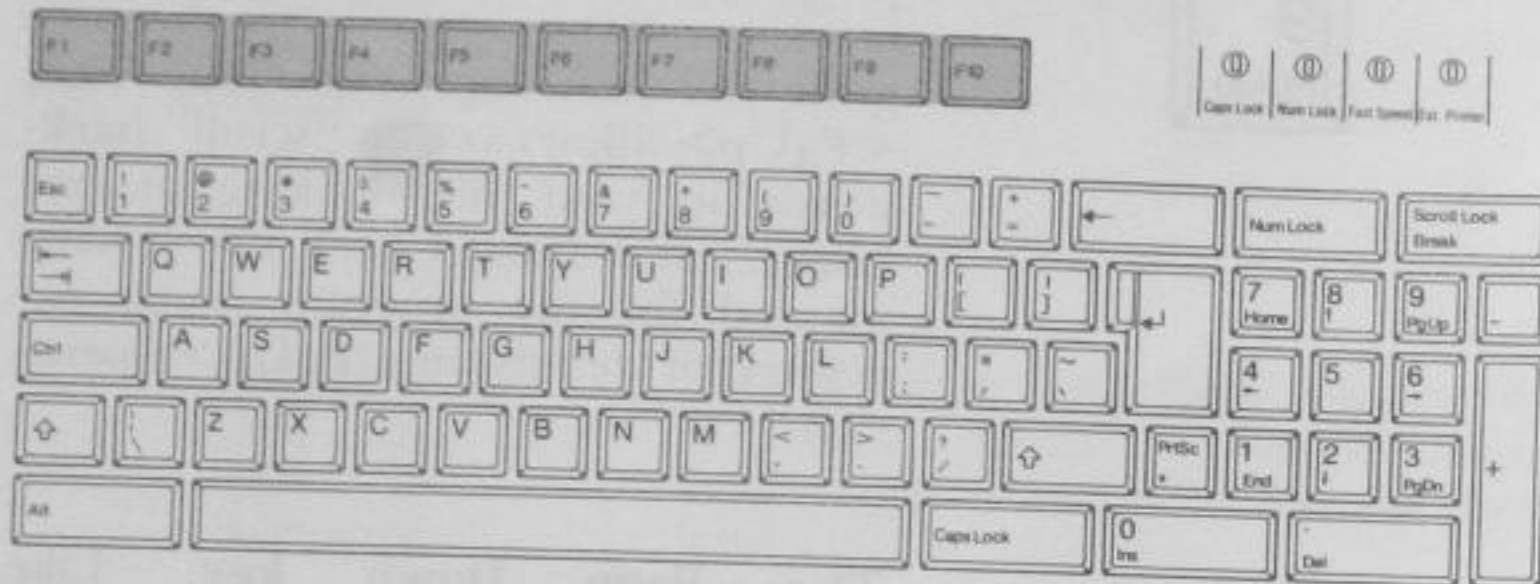


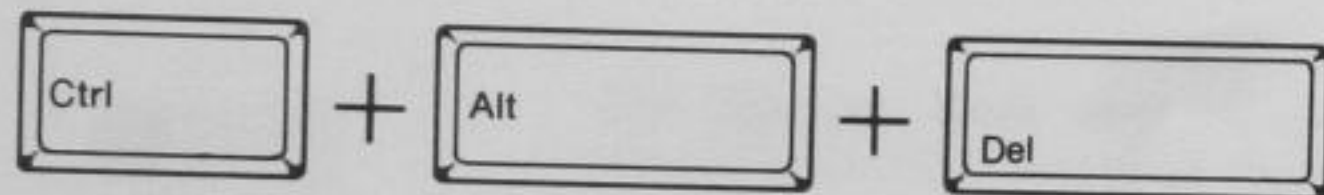
Fig. 3-6. FUNCTION KEYS

OPERATION

The function keys are always under program control. For details on function key operations see your Operating System or Application Program manuals.

System Restart

Resetting the system with the power on is called "reboot". When rebooted, the system is activated in the same condition as when the Exec. Partner is turned on. To reboot, press while pressing <Ctrl> and <Alt>.



PURCHASING DISKS FOR YOUR EXEC. PARTNER

Disks

Disks are used to read in programs and data, and write out information you wish to save from RAM. Because disks provide the instructions to the computer and supply the necessary data for programs, they are very vital link in your Exec. Partner system.

Your Exec. Partner uses 5¼ inch double-sided, double-density, soft-sectored disks. This type of disk can store approximately 360,000 characters (also termed 360K).

Double-sided disks can store information on both sides. This does not mean that you must turn over your disks, however. Information is automatically stored on both surfaces.

Double-density disks allow data to be packed together in a more economical pattern. They are called double-density because they can store twice the amount of information of single-density disks.

OPERATION

Before Using Disks

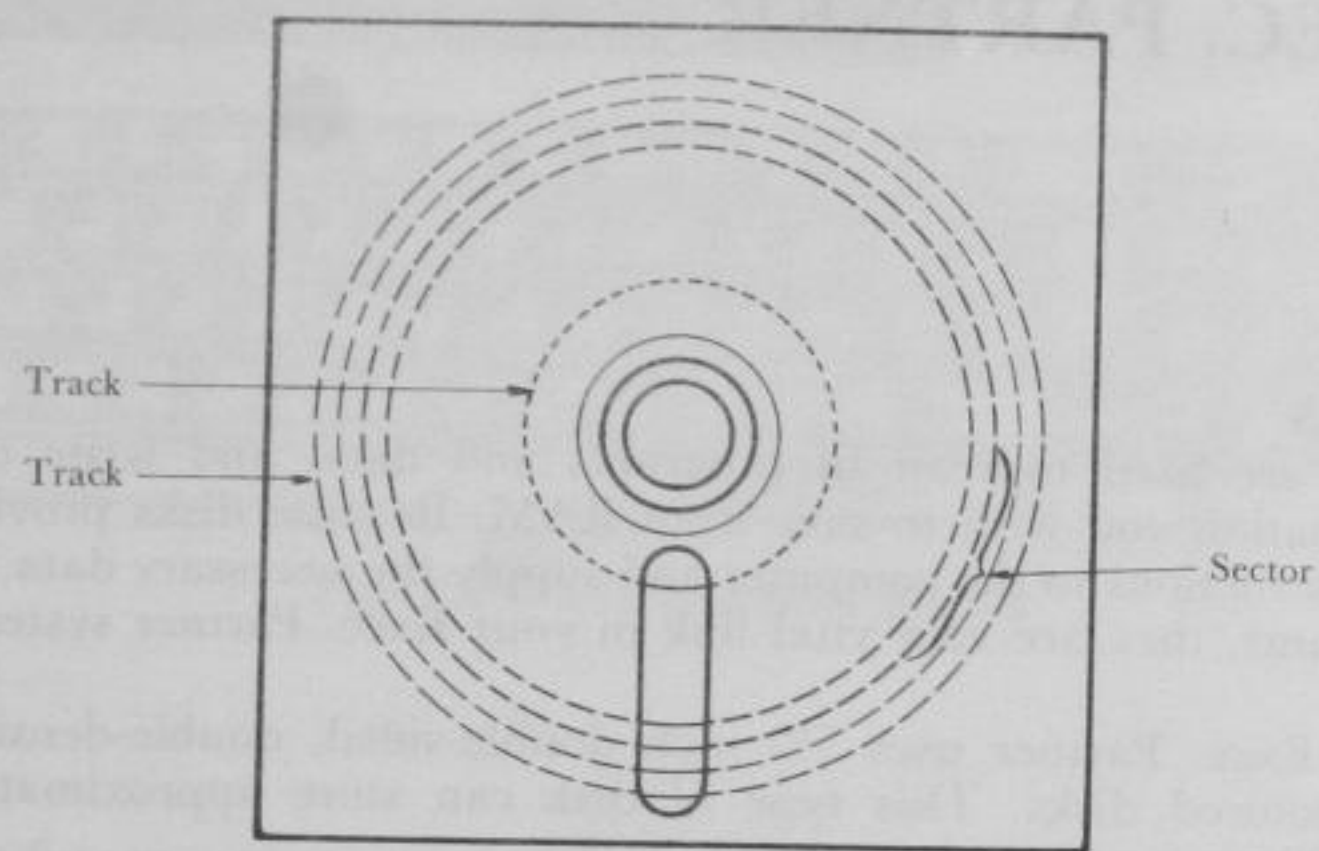


Fig. 3-7. TRACKS AND SECTORS WITHIN TRACKS

Data is written into and read from a disk divided into concentric circles called tracks as shown above.

Each track is divided into sectors. Dividing a disk into tracks and sectors is called "Disk Formatting." When the Exec. Partner is used, a disk is divided on each side into 40 tracks, each being further divided into 9 sectors.

Any new disk must always be formatted before being used. There are, however, different format procedures depending on how the disk is to be used. A disk can be used in 2 major ways:

- (1) As a system disk or as an application program disk.
- (2) As a disk to create files for data input or output.

- (1) As a system disk or as an application program disk.

If you are going to use a new disk to backup the System Disk* or an application program disk**, you do not need to format the disk using a separate operation as long as you use the DISKCOPY command***. The disk is automatically formatted while the DISKCOPY command is executed. If you do not use the DISKCOPY command but use the COPY command, you must format the disk beforehand.

- (2) As a disk to create files for data input or output.

To use a new disk only to store data, you must format it beforehand. Once formatted, you do not need to format the disk again.

*** WARNING ***

If you format a disk again, the data that you may have previously written on the disk will be completely erased.

See pages 3-52 and 3-92 for the procedures on formatting a disk.

NOTE: If a disk has only been formatted, it cannot be used to boot up the system since it doesn't contain the necessary files. The command `FORMAT/S` not only formats the disk but also transfers the system files to it.

*To backup the System Disk, you do not need to separately format the new disk. See page 3-34 or 3-72.

**An application program disk without the DOS files cannot be used to boot up the system. In this case, the System Disk is required.

***See page 7-43 for details on the DISKCOPY command.

How to Insert and Remove Disks

If you are not familiar with disk usage, this section describes procedures on how to insert and remove disk.

Inserting Disks

- STEP 1 Be sure the main power switch is in the OFF position.
- STEP 2 If the door of the disk drive is closed, open it by pushing the door lever in and then up.

*** WARNING ***

Remove the head protection sheet from the drive. This head protection sheet protects the delicate recording head while your Exec. Partner is in shipment.

- STEP 3 Grasp the disk in your hand on the manufacturer's label. The label should be facing up. The edge with the head slot enters the drive first.
- STEP 4 As you insert the disk it will come to rest against a solid stop. **DO NOT FORCE THE DISK BEYOND THE STOP.** Once the stop is reached the disk is completely inserted.
- STEP 5 Press the lever down until it locks.

Removing Disks

When you are finished using a disk, remove it carefully.

- STEP 1 Open the door by pushing the lever in and then up.
- STEP 2 Remove the disk by grasping the label.
- STEP 3 Return the disk to its storage jacket.

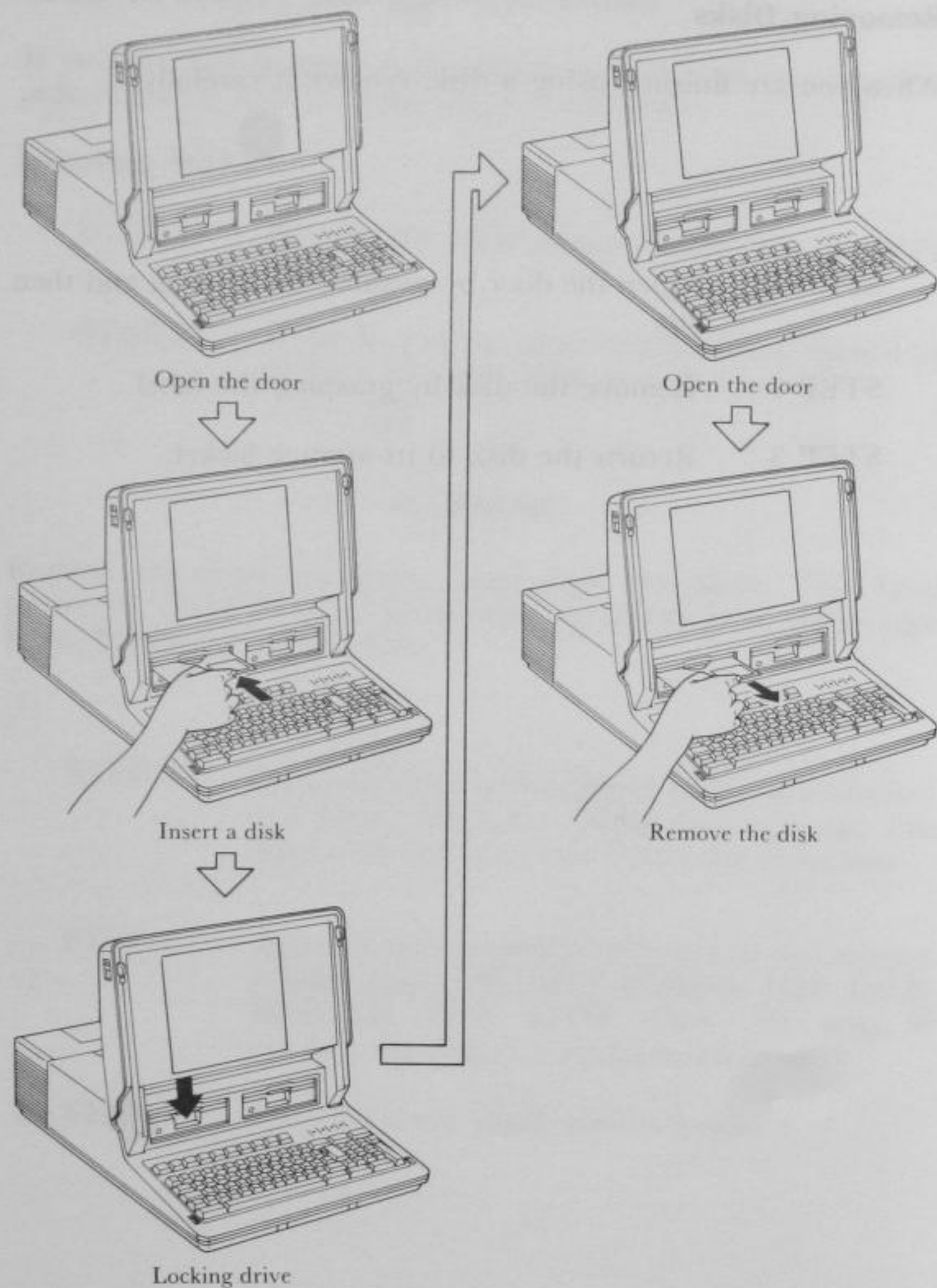


Fig. 3-8. HOW TO INSERT AND REMOVE A DISK

*** WARNING ***

BEFORE HANDLING OR USING ANY DISK SEE THE SECTION BELOW ON DISK CARE

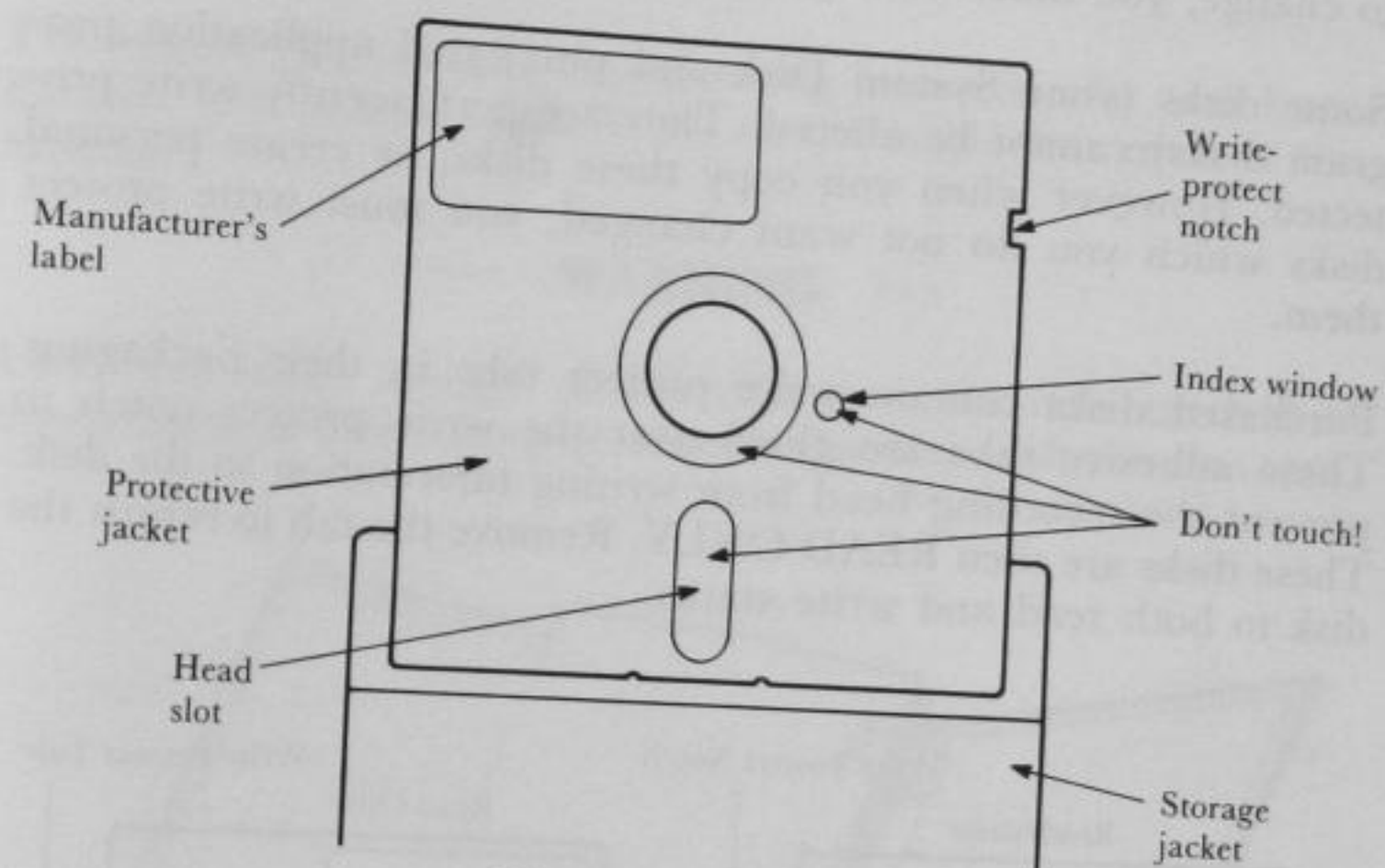


Fig. 3-9. DISK

The **manufacturer's label** identifies the type of disk and the maker.

The **storage jacket** provides protection for the disk when you are not using it.

The Write-Protect Notch

Usually your computer will both read from and write to a disk. When disks contain important information which you do not want to change, you must write protect the disks.

Some disks (your System Disk and purchased application program disks) cannot be altered. They are permanently write protected. However when you copy these disks, or create personal disks which you do not want changed, you must write protect them.

Purchased disks contain write protect tabs in their packaging. These adhesive tabs are glued over the write protect notch to prevent the recording head from writing information to the disk. These disks are then **READ ONLY**. Remove the tab to return the disk to both read and write status.

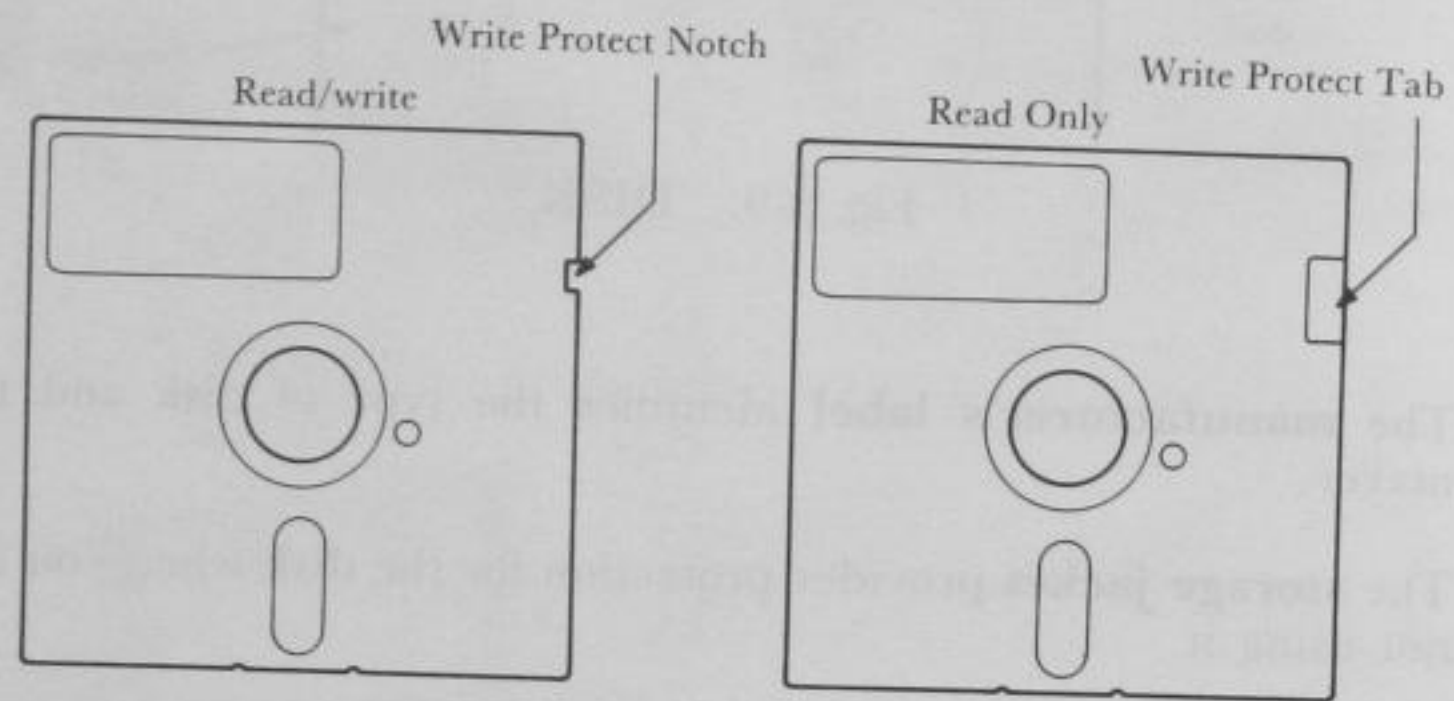


Fig. 3-10. WRITE PROTECT NOTCH AND WRITE PROTECT TAB

BRINGING UP THE SYSTEM

The System Disk

The System Disk contains the operating instruction for your Exec. Partner. It is located in the jacket on the inside rear cover of this manual.

*** WARNING ***

Until you make a copy of this disk it is the only method of operating your Exec. Partner. Handle this disk with extreme care.

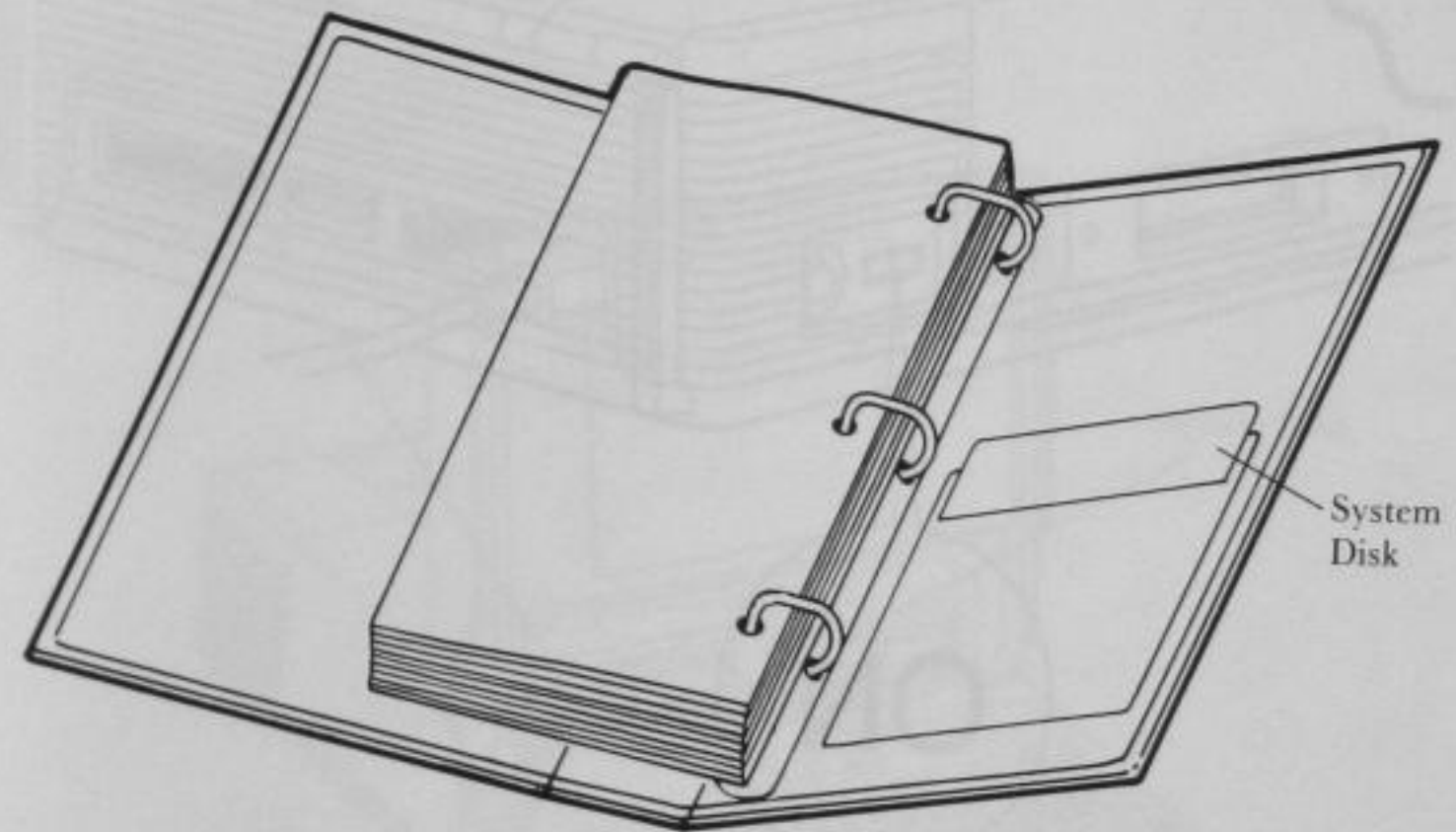


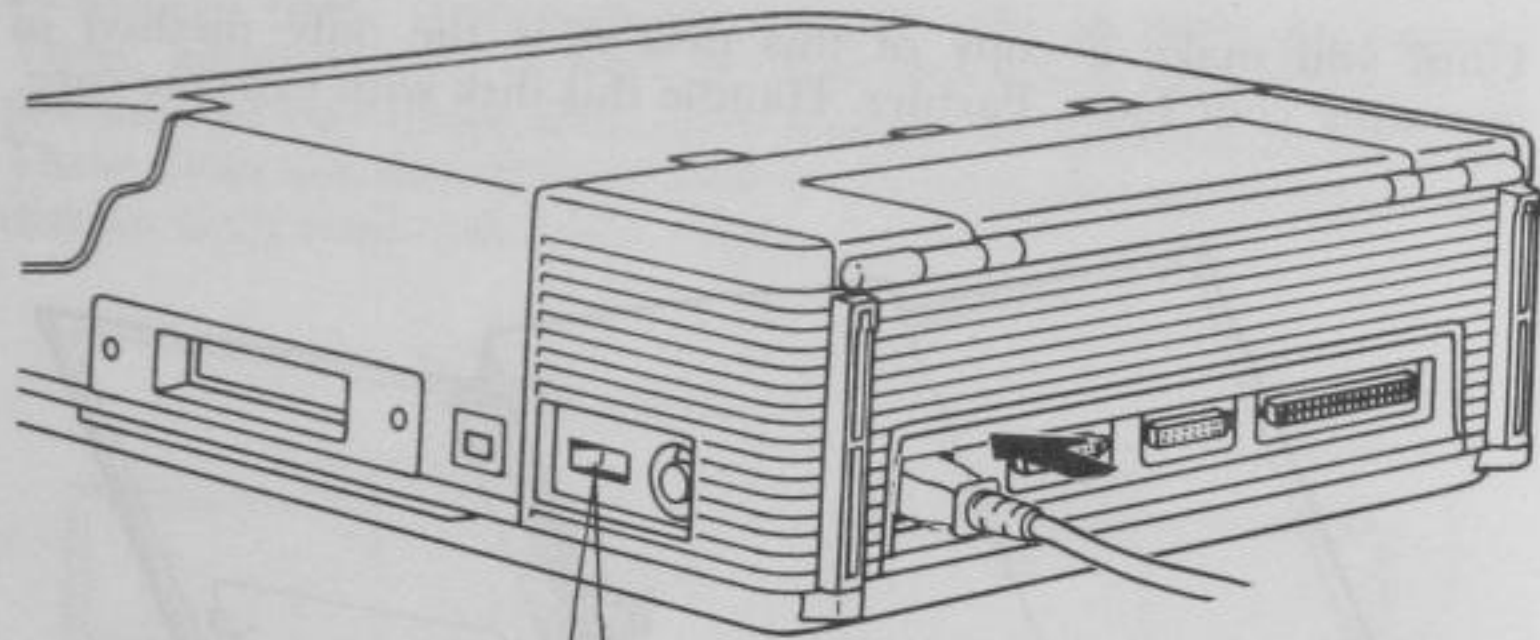
Fig. 3-11. SYSTEM DISK

Turning on the Power

... WARNING ...

You must turn on the main power switch **before** inserting the System Disk.

Make sure the computer is plugged into a wall socket and contains **no disk** or **head protection sheet**. Now turn on the main power switch.



ON

Fig. 3-12. POWER ON

Inserting the System Disk into Drive A

Insert the System Disk into Drive A. You can insert the System Disk:

1. After one short beep is heard...
2. While the cursor is blinking in the upper left corner of the screen...
3. While the RAM Check is being performed...
4. While the In Use Indicator of the disk drive is not lit up...
5. When the message Disk boot error Replace and strike any key when ready appears on the screen.

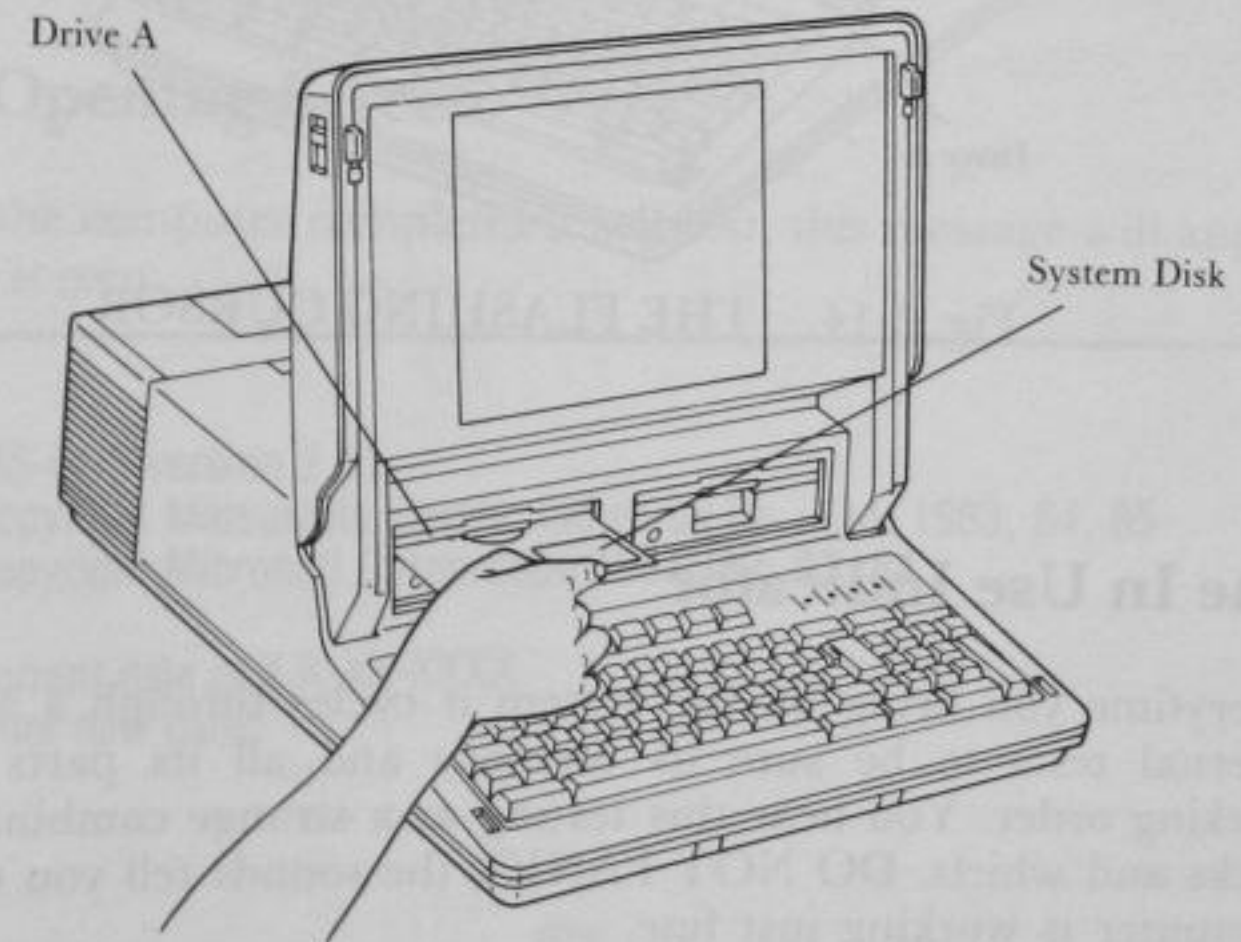


Fig. 3-13. INSERTION OF THE SYSTEM DISK

The Flashing Cursor

Immediately after the beep, the cursor will blink in the upper left hand corner of the screen. The cursor is an indicator to you of your position on the screen. Then the computer checks the capacity of RAM and the size of RAM appears on the screen.

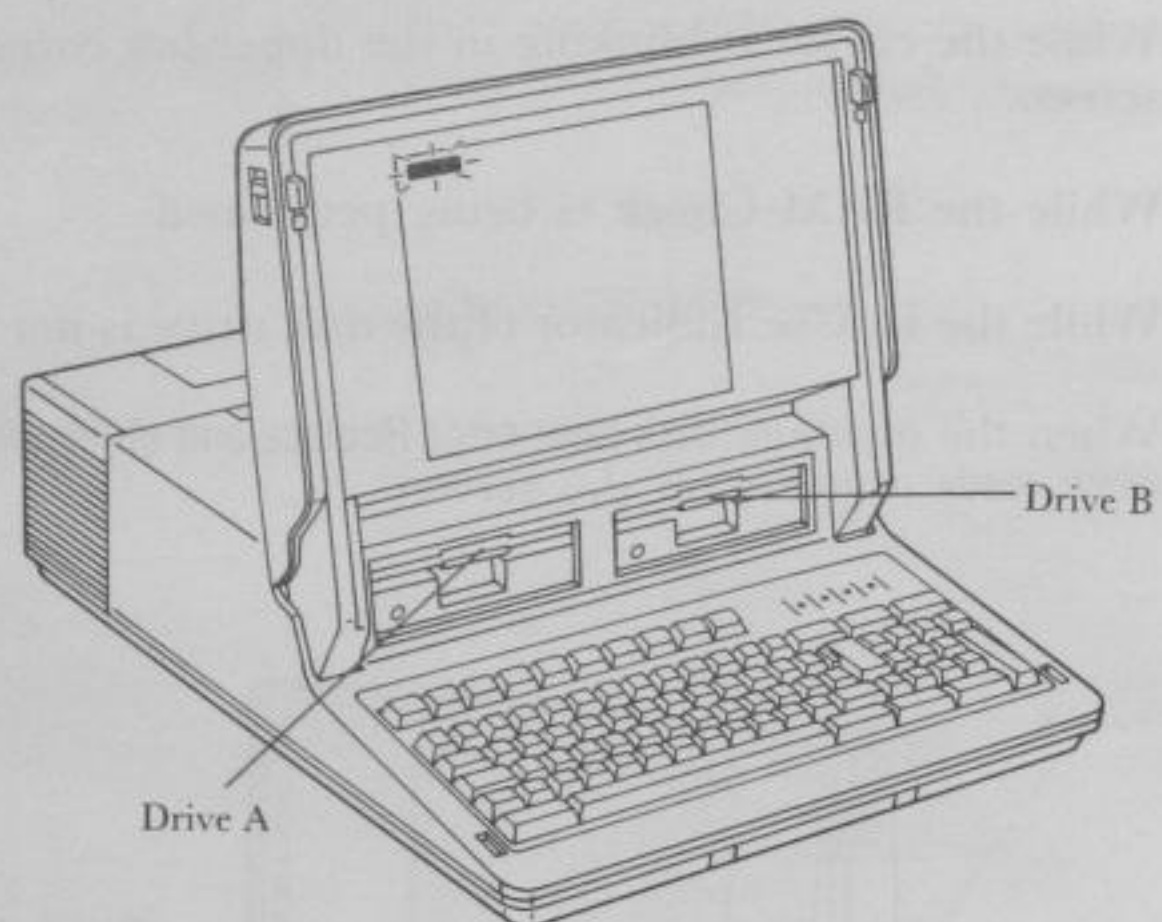


Fig. 3-14. THE FLASHING CURSOR

The In Use Indicator

Everytime you bring up the system it cycles through a series of internal tests to be sure its memory and all its parts are in working order. You hear this testing as a strange combination of clicks and whirrs. DO NOT PANIC, the sounds tell you that the computer is working just fine.

The Exec. Partner is reading its instructions from the System Disk. You will hear the disk drive transferring information. The **In Use Indicator** lights up when the computer is accessing a disk, either reading from it or writing to it.

REMEMBER: NEVER UNLOCK THE DRIVE DOOR OR ATTEMPT TO REMOVE A DISK WHEN THE IN USE INDICATOR IS ON!

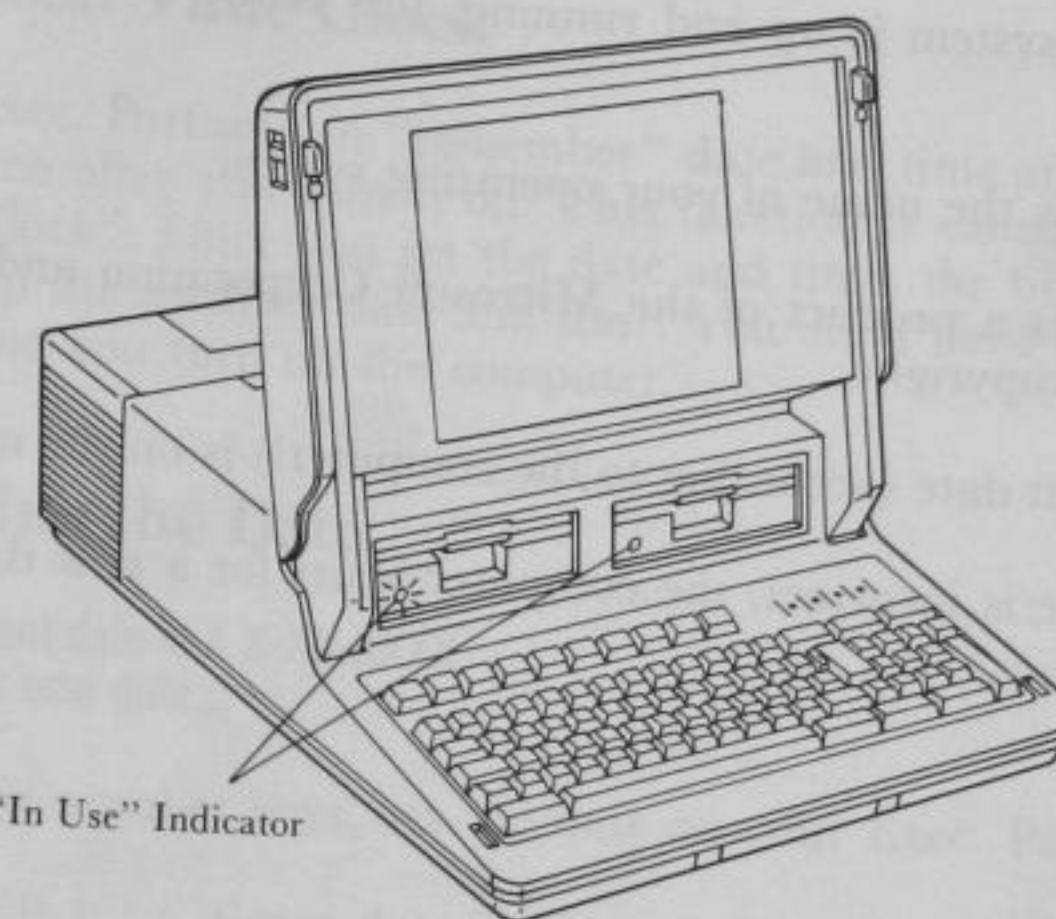


Fig. 3-15. THE "IN USE" INDICATOR

The Opening Screen

When the computer completes its self-test, this message will appear on the screen:

```
MS-DOS version 2.11
Copyright Matsushita Electric Industrial Co., Ltd. 1983, 84, 85
Copyright Microsoft Corporation 1981, 82, 83, 84
```

```
Current date is X X-XX-XXXX
Enter new date:
```

Fig. 3-16. THE OPENING SCREEN

When the system is up and running, this is the first screen you will see.

MS-DOS is the name of your operating system.

MS-DOS is a product of the Microsoft Corporation and is protected by copyright.

The current date (according to the computer!) is on the next line.

If this date is incorrect, the computer waits for a new date.

Prompts

The cursor at the end of the line tells you that the computer is waiting for a response from you.

Whenever the computer awaits input from you it displays a **prompt**. Different programs use different prompts. A>, B> or C> is the prompt for DOS, ok is the prompt for BASIC. In this case, the cursor is the prompt.

If the Opening Screen Doesn't Appear

When the start-up tests indicate a problem, your screen will not look like this. As the computer shifts into a full diagnostic mode, your screen will fill with numbers. **THIS IS VERY UNLIKELY TO OCCUR.** If it does happen you have two choices.

You can turn OFF the machine and notify your dealer.

In all likelihood, you can diagnose the problem quite quickly and easily. See Appendix A for simple step-by-step procedures.

The Real Time Clock

Your Exec. Partner can "remember" date and time and store the data even after you turn it off. This function is called the "Real Time Clock". Once you set the date and time, the built-in clock will keep the accurate date and time. You don't need to set them each time you turn on the computer.

Entering the Date

Current date is X X-XX-XXXX
Enter new date: _

Using the number keys, set the date on your Exec. Partner:

- STEP 1 Enter the month as a one or two digit number, from 1 to 12.
- STEP 2 Separate digits by a hyphen (-) or a slash (/).
- STEP 3 Enter the date as a one or two digit number, from 1 to 31.
- STEP 4 Enter the year as a two digit number, from 83 to 99.
- STEP 5 Press <ENTER>

NOTE: To enter the year, you may type four digits if you wish (e.g. 1985). The computer will accept years up to 2079, you don't need to set the day of the week, because your Exec. Partner sets it automatically.

Example: Suppose you want to set the date to June 4, 1985:

6-4-85	<ENTER>
6/4/85	<ENTER>
6-04-85	<ENTER>
6/04/85	<ENTER>
06-04-85	<ENTER>
06/04/85	<ENTER>
06/4/85	<ENTER>

You may use any of these formats above to enter the date.

Entering the Time

Current time is X:XX:XX.XX
Enter new time:

Your Exec. Partner is very precise, time is reported in hours, minutes, seconds and hundredths of a second.

To set the time on your Exec. Partner:

- STEP 1 Enter the hour as a one or two digit number, 0 to 23.
- STEP 2 Enter a COLON (:).
- STEP 3 Enter the minutes as a one or two digit number, 0 to 59.
- STEP 4 Press <ENTER>.

NOTE: This is all the information that the computer requires. You may enter seconds and hundreds of a second moreover, if you wish, as follows.

- STEP 5 Enter a COLON (:).
- STEP 6 Enter the seconds as a one or two digit number, 0 to 59.
- STEP 7 Enter a PERIOD (.).
- STEP 8 Enter the hundredths of a second as a two-digit number 00 to 99.
- STEP 9 Press <ENTER>.

Example: Suppose you want to set the time 9:30:

9:30:0.00	<ENTER>
09:30:0.00	<ENTER>
9:30:0	<ENTER>
9:30	<ENTER>

You may use any of these formats above to enter the time.

USING THE FT-70/FT-80

We will now cover the basic operation of the Exec. Partner. Please read the following sections carefully, these operations are important. Note that the operations differ depending on the model number of your Exec. Partner.

If you have model FT-70, start from page 3-34.
If you have model FT-80, start from page 3-72.

BACKING UP THE SYSTEM (FOR USING FT-70)

What is a Backup?

The very first operation you should perform on your new Exec. Partner is to make a copy of your System Disk. This provides insurance for you, in case the original disk becomes damaged. This copy is called a "backup". Making the copy is called "backing up" the system.

It is a good practice to use your "backup" System Disk in your everyday operations. Store the original System Disk in a safe location.

You will need two disks to perform this operation. The first, your System Disk, is the one you wish to backup. The disk you are "copying from" is called the original or SOURCE disk. Your source disk is already loaded in Drive A.

The second disk will become the backup. It is the disk you will be "copying to". This is the destination or TARGET disk. Have on hand a new, unformatted disk.

original system disk (copy from)=SOURCE disk

new, unformatted disk (copy to)=TARGET disk

Your System Disk is write-protected. This means you cannot write to the disk. This protects you from erasing the stored information on the disk (see WRITE-PROTECT NOTCH earlier in the chapter). If you get confused and attempt to write to your System Disk, this message will appear:

Write protect error writing drive A(B)
Abort, Retry, Ignore?

When this message appears, three responses should be selected:

A for Abort. The system ends the program that requested the device read or write.

R for Retry. The system will try again to do the device read or write operation.

I for Ignore. The system processes the program as if the error had not happened.

With pressing A, the system aborts the program, with R, it retries the operation, and with I, it ignores the problem and continues processing.

Device Designations

Before you begin working with the DOS, disk operating system, you should become familiar with drive designations.

You must tell DOS where to store file or where to put the results of a procedure. You do this by using drive designations **A:** and **B:**

A: or **a:** Tells DOS to get this information **from** or write this information **to** the disk in Drive A.

B: or **b:** Tells DOS to get this information **from** or write this information **to** the disk in Drive B (when you have two disk drives).

Default Drive

The DOS prompt A> means that DOS is currently getting and sending information from the disk in Drive A. DOS always assumes that the file you want is on the current disk. As long as you want to read or write from the current drive, you do not need to include the drive designator in commands.

When you have two drives, however, you may want to read from or write to the disk in Drive B. Then you must include B: in your instructions to the computer.

You can change the default drive. Suppose you wanted to work with a group of files on the disk in Drive B. You don't want to specify B: in every file name. You can change the default drive to B.

A>

This is the DOS prompt. It tells you that DOS is ready, but it also tells you that Drive A is the default drive.

YOU TYPE:

b:

and press <ENTER>

SCREEN DISPLAYS:

B>

B> is also a DOS prompt. It tells you that Drive B is the default drive.

To return to A as the default just type a: in response to the B> prompt.

NOTE: DOS always assumes it will find the file or write the results to the default drives unless you indicate a drive specifier in your commands.

The Procedure of Making the Backup System Disk with FT-70

The DISKCOPY command transfers the entire contents of one disk to another disk. You will use DISKCOPY to make your Backup System Disk. In the future, use DISKCOPY any time to want to copy and entire disk.

Using DISKCOPY with FT-70

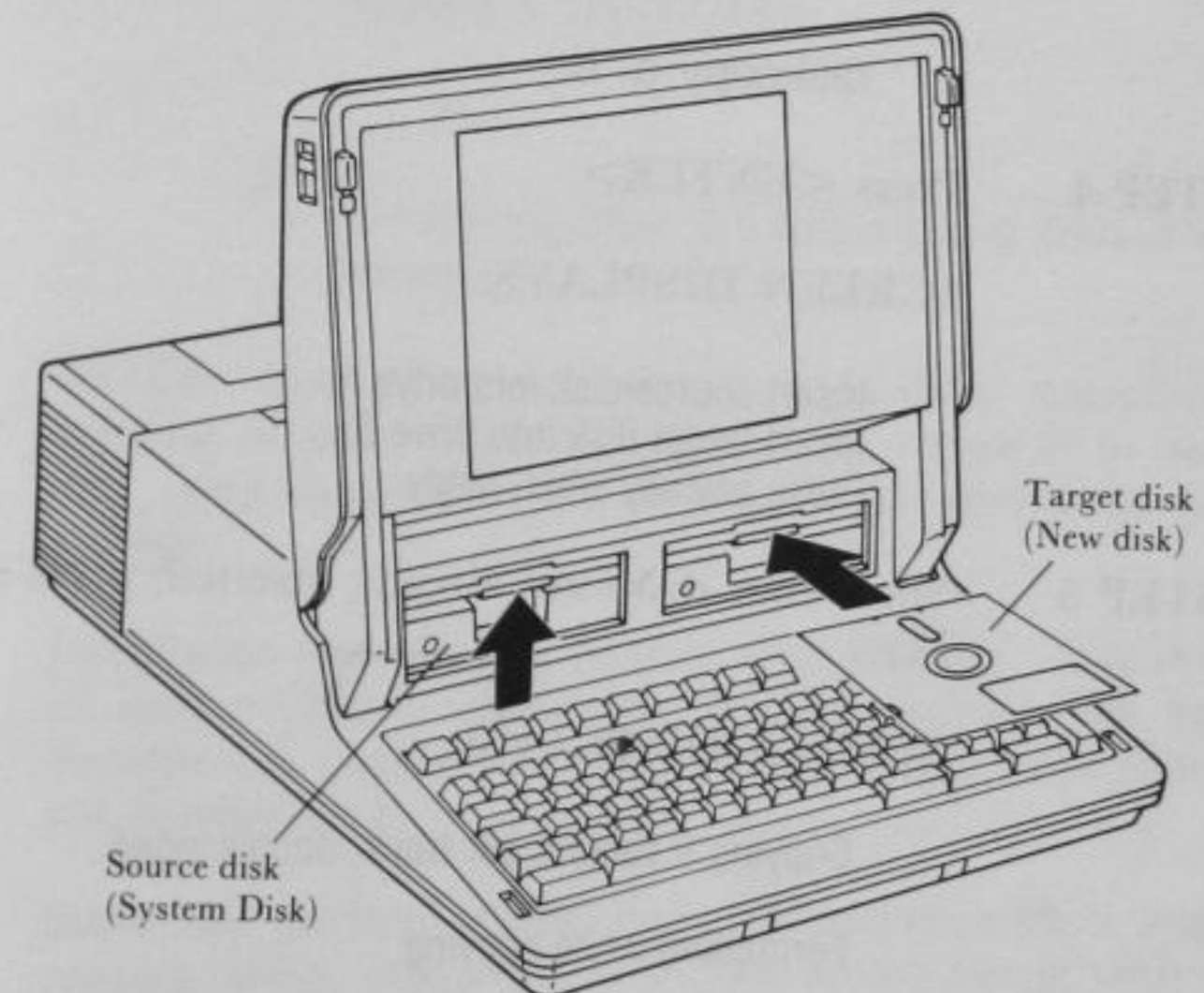


Fig. 3-17. USING DISKCOPY WITH FT-70

STEP 1 Make sure you have inserted your SOURCE disk (System Disk) into Drive A.

STEP 2 Insert a new, unformatted disk into Drive B. Hold the disk by its label (with the label facing up).

STEP 3 SCREEN DISPLAYS:

A>

YOU TYPE:

diskcopy a: b:

STEP 4 Press <ENTER>

SCREEN DISPLAYS:

Insert source disk into drive A:
Insert target disk into drive B:
Press any key when ready

STEP 5 Since your disks are already inserted, press any key.

SCREEN DISPLAYS:

Copying 9 sectors per track, double sided

Formatting while copying

The copying is done automatically. While the information from Drive A is being transferred to Drive B you will hear the drives moving, and see the In Use Indicators come off and on.

In a short time the copying will be completed.

SCREEN DISPLAYS:

Copy complete

Copy another (Y/N)?

STEP 6 TYPE:

y

if you want to make another copy

TYPE:

n

to end the diskcopying session.

SCREEN DISPLAYS:

A>

to indicate that the diskcopying session is completed.

NOTE: Now that you have successfully completed the copying of your original System Disk, return it to its jacket and store it carefully according to the directions for disk storage.

Label your copied disk "System Disk-Backup". Use this copy in your everyday operations. If your backup disk becomes damaged or inoperable make a new backup from your original System Disk.

Since the surface of the disk is covered with a magnetic coating, other magnetic fields can wreak havoc with stored information. Keep disks away from magnetic field sources such as telephones, dictation equipment, x-rays and calculators.

CREATING A FILE (FOR USING FT-70)

Files

The many types of tasks you will perform using the Exec. Partner will all be based on files. A file is a collection of data which is managed by your computer.

Like an enormous file cabinet, your computer stores all of its information in files. In order to find data, the computer must be instructed exactly where to look. (Don't forget, the computer by itself, is dumb. All it can do is follow instructions!)

Some files come built into your Exec. Partner. These files help the computer to regulate itself and operate the various parts that make up your system. These internal files are stored in ROM.

Your System Disk also contains operating files for your Exec. Partner. Once they are loaded into memory they perform automatically upon demand.

If you make a list of all the files on the System Disk (with the DIR command explained later in this chapter), you will notice that the System Disk contains many files. Among these are command files used to execute MS-DOS.

The disks which operate your application programs also store information files.

After you finish working on a program or text that you have created using BASIC or EDLIN, you can save it in a file on the disk.

To do so, you must name the file.

Once saved on a disk, you can load a file into memory any number of times whenever you need it.

A file is very convenient. You can change its contents, add to it, or even delete it whenever you want to.

A File's Name

All files on a disk are identified by their file names. Your Exec. Partner searches for a file on the disk using the file name as a clue and then processes the file.

That's why you must name the program or text that you created with a unique name.

A file is simply a collection of related data stored on a single disk. You can have many different types of files.

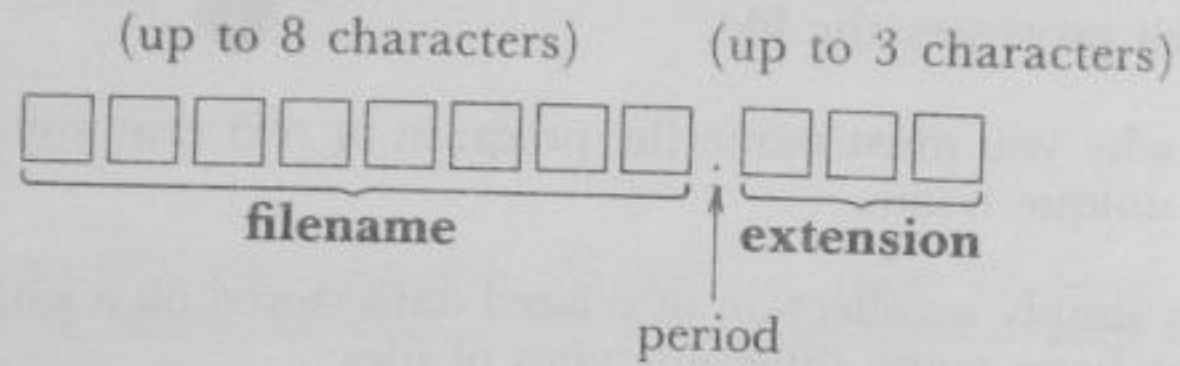
Example:	SCHED	Contains your upcoming appointments
	ACCTREC	Contains your accounts receivable
	BDAY	Contains a list of important birthdays
	FORMLET	Contains a form letter
	GAMES	Contains game programs

You may store several files on one disk. Sometimes the files may be related. Or a disk may contain all of the files which you access on a regular basis. It does not matter to the computer how you organize and store your files.

NOTE: If you store information under an existing file name, the old information will be replaced by the most current entry. This is a good way to update files, but be careful not to inadvertently lose data by assigning it to an already existing filename.

Filenames and Extensions

A filename consists of two parts.



The first section is the **filename**, the second section is the **extension**.

A **filename** may contain as many as eight **characters**. The **extension** always begins with a **period** and may contain up to **three additional characters**.

DOS allows up to eight characters in the filename, a period plus up to three additional characters in the extensions. Within these restrictions you can name a file almost anything you want. But remember, the file's name is for your reference so be sure the name is convenient to type and easy to remember.

Example: SCHED.JUL
ACCREC.CUR
BDAY.BAS
FORMLET.84
GAMES.JOD

You can use any of these characters to name your file:

The letters of the alphabet (upper or lower case)

The numbers 0 through 9

These special characters—

\$ # & @ ! % () - ' _

DOS does have some restrictions on filenames. Here are some unacceptable file names:

Q and R.COM	Filenames cannot contain spaces
"SPEECH".BAS	Filenames cannot contain quotation marks
A,B,C.EXE	Filenames cannot contain commas
.NG	Filename is missing
TOOMANYCHARACTERSINTHISNAME	Too long

Extensions

Although the filename is required, the extension is optional except for special files.

There are several extensions having special significance in MS-DOS. In other words, special extensions are added to certain types of files.

Among them, the following 3 extensions are often seen:

- .COM** Indicates a **COMmand** file. The program in a file with this extension can be executed simply by typing the filename.
- .EXE** Indicates an **EXEcutable** file. It is similar to the .COM file, and at this point can be considered to be the same.
- .BAT** Indicates a **BATch** file. A predefined sequence of commands can be executed simply by typing the filename of a file with this extension.

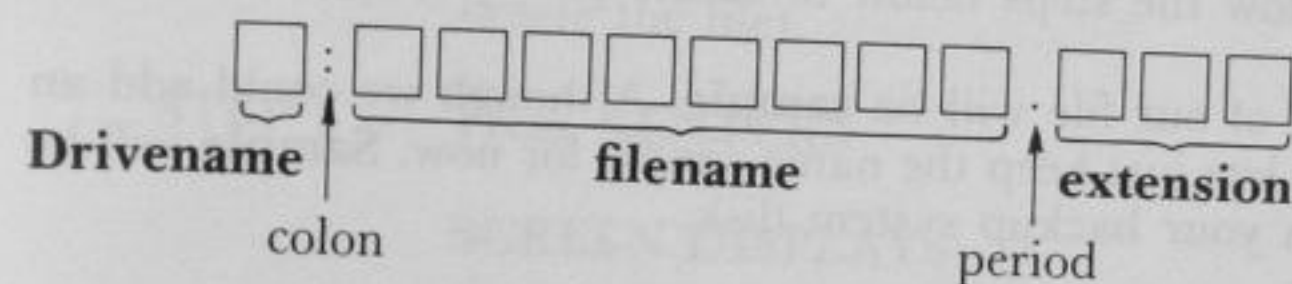
Drive Specifiers

In addition to the filename and optional extensions, the computer must know one more piece of information to retrieve or store a file. It must know WHERE to find the file.

The following type of Exec. Partners is available:

FT-70 2 floppy disk drives

To specify a file in a certain drive, you add the drive name to the beginning of the file name.



In MS-DOS, you can perform some operation on a desired file located in a certain drive by specifying the drive name in front of the file name, regardless of the current drive.

Example: Suppose the current Drive is A and you want to specify files located in Drive B. You would add drive specifiers like in the following.

A:SCHED. JUL

B:BDAY. BAS

A:GAMES. JOD

Sometimes you don't need to type the drive specifier. If the file you are creating will reside in the default or current drive, or the file you want to read is in the default drive you don't need to indicate the specifier. So if the default drive is A (indicated by the A> prompt) you could type either:

Example: A: SCHED. JUL

 SCHED. JUL

Creating a File

Managing files with DOS is simply a matter of practice. DOS is really very easy to use and in no time at all you will be wondering how you managed to do any work in pre Exec. Partner days!

While you are learning however, you may make a few small mistakes. So, to prevent any of your valuable data from being damaged, we are going to create a practice file to use while you acquaint yourself with some fundamental DOS commands.

You use EDLIN to create a file. EDLIN is a special part of DOS which allows you to create, change and display files. Chapter 9 of this manual provides full instruction for using EDLIN but for now simply follow the steps below to create your practice file.

The name of our file will be **sample**. Although we could add an extension, lets just keep the name simple for now. **Sample** will be located on your backup system disk.

NOTE: Do not proceed with these exercises unless you have made a backup of your original System Disk. See "backing up the system" earlier in this chapter.

When instructions refer to your System Disk, they are referring to this backup version.

STEP 1 Insert your System Disk in Drive A. (It is assumed Exec. Partner is already on.)

SCREEN DISPLAYS:

A>

STEP 2 YOU TYPE:

edlin sample

STEP 3 Press <ENTER>

SCREEN DISPLAYS:

New File

*

The * is the prompt for EDLIN

STEP 4 YOU TYPE:

i

The i stands for Insert, it means you want to add information to the named file (in this case create the file).

STEP 5 Press <ENTER>

SCREEN DISPLAYS:

1:*

The number 1 stands for the line number. EDLIN references all input by line numbers. Each time you press <ENTER> a new line number will appear at the beginning of the next line.

STEP 6 YOU TYPE:

This is a practice file.

STEP 7 Press <ENTER>

SCREEN DISPLAYS:

1:*This is a practice file.

2:*_

STEP 8 Continue typing in the text for the practice file.

YOU TYPE and PRESS:

We will call it sample. <ENTER>
Sample will help explain DOS. <ENTER>

NOTE: If you make a mistake while typing in the contents of the file, use <BACKSPACE> to erase the characters and then retype the entry. You must edit a line before you type <ENTER>.

Even if you do enter incorrect lines this is only a practice file. The actual contents don't really make any difference.

STEP 9 The screen should look like this:

SCREEN DISPLAYS:

```
A>edlin sample
New File
*i
1:*This is a practice file.
2:*We will call it sample.
3:*Sample will help explain DOS.
4:*_
```

STEP 10 Now indicate the end of the file.

YOU PRESS:

<Ctrl> <Z> (hold down <Ctrl> while pressing <Z>, then let go of both keys)

SCREEN DISPLAYS:

```
4:*^Z_
```

STEP 11 Press <ENTER>

SCREEN DISPLAYS:

```
*
```

The file is completed and we are back to the EDLIN prompt.

STEP 12 To check the contents of your file ask for a **listing** of the file.

YOU TYPE:

```
l (lowercase letter L)
```

and press <ENTER>

The screen will type out your sample file.

STEP 13 Having created the file, you need to exit from EDLIN and return to DOS.

YOU TYPE:

```
e
```

and press <ENTER>

This command will save the file to the default drive. The In Use Indicator will light up and you will hear the drive turning. The sample file is being written to the disk in drive A.

SCREEN DISPLAYS:

```
A>
```

to indicate DOS is now ready.

NOTE: See Chapter 9 for a description of EDLIN.

Before Using DOS Commands

You have created your file using EDLIN. Now we will use it with a number of MS-DOS commands. We use MS-DOS commands to manipulate the file loaded into memory.

The many DOS commands are described individually in detail in Chapter 7 COMMAND REFERENCE. In this section, we will consider the basic DOS commands that you will most likely use in your daily operations.

We will introduce the following commands:

FORMAT A command which must be executed before you first start using a disk, whether it is a floppy disk or hard disk. Once a disk is formatted, there is no need to format it again.

NOTE: If you format a disk which has already been formatted, you will lose all files on the disk.

DIR DIR stands for DIRectory. The DIR command displays (on the screen) the names of the files on a disk, the file creation date, and the file size.

TYPE Displays the contents of the specified file on the screen.

NOTE: This command is provided for use only on text files.

COPY Copies a file to the designated device.

NOTE: Before copying a file, the disk must be formatted. However, it need not be formatted if you use the DISKCOPY command.

RENAME Used to change the name of a file.

NOTE: The RENAME command can be used regardless of whether or not the file to be renamed is in the current drive.

ERASE Used to erase unwanted files from the disk.

NOTE: The ERASE command can be used regardless of whether or not the file to be erased is in the current drive.

We have briefly described the functions of these useful commands. A more detailed introduction describing the format of these commands, including examples, is given in the next section.



GETTING ACQUAINTED WITH DOS: FT-70

The FORMAT Command

When you purchase disks they are blank. Specific computers store information on disks in different formats. Before you can use a new disk, you must specify the format of your YOUR SPECIFIC MACHINE. To get the disk ready to receive information, use the FORMAT Command.

You only format a disk once, the very first time you use it. Since FORMAT treats the disk as a blank it will erase any information already on a disk.

You can use FORMAT to prepare a disk for new information. Just be sure you NO LONGER NEED the data on the disk.

NOTE: You do not need to format a disk before using the DISKCOPY command. DISKCOPY will perform the format operation first if necessary, then copy the data.

Using the FORMAT Command with FT-70

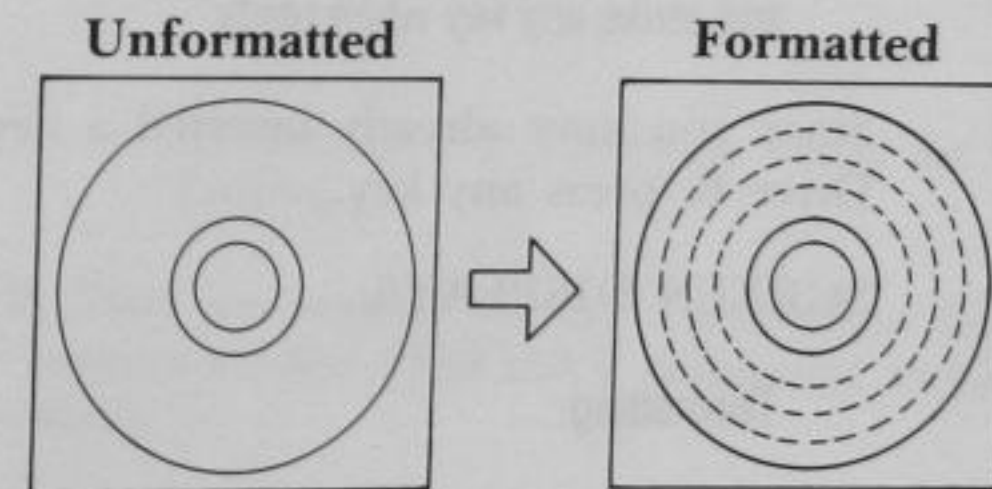
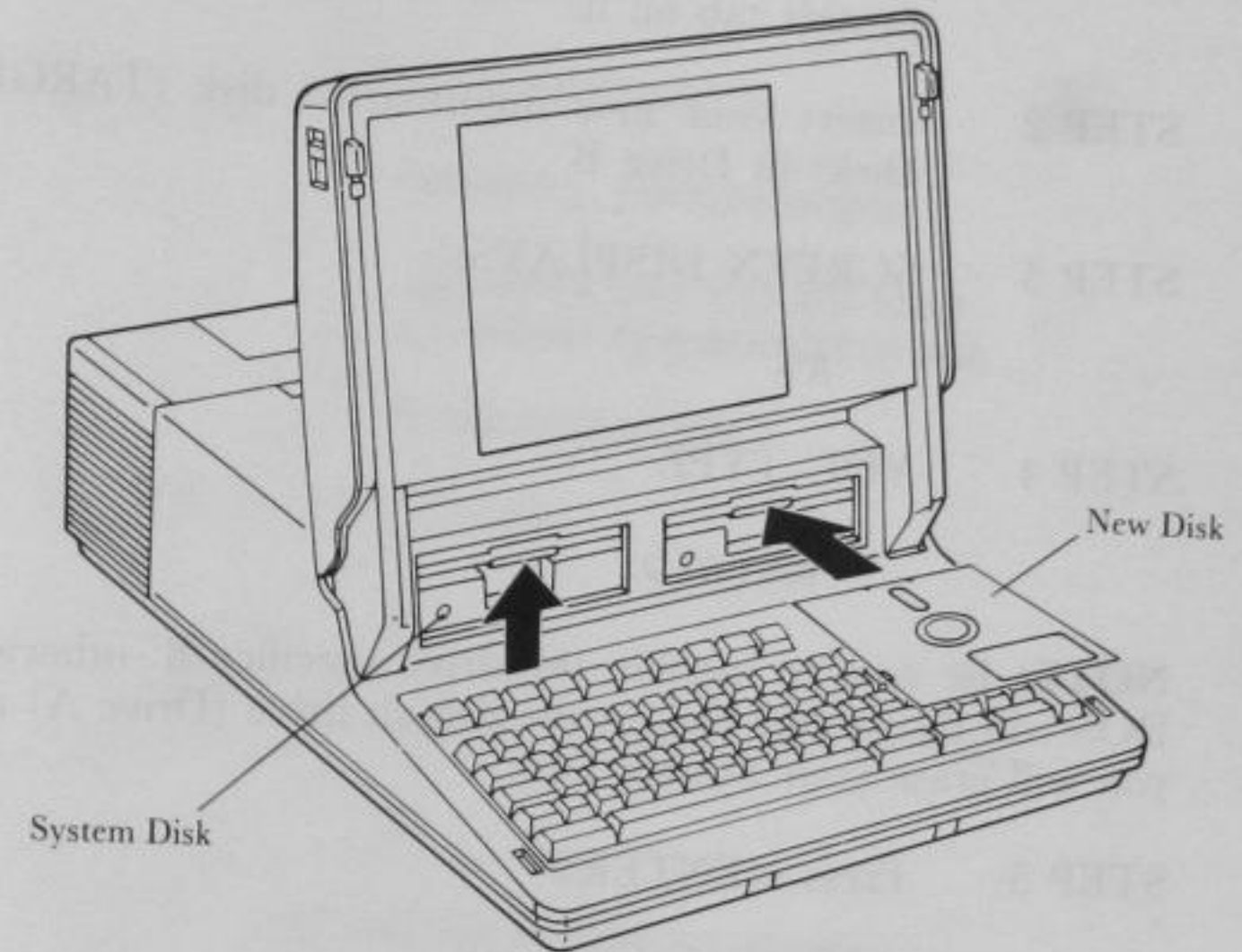


Fig. 3-18. INSERTION OF SYSTEM DISK IN DRIVE A AND BLANK DISK IN DRIVE B

STEP 1 Insert your System Disk in Drive A. It is a good idea to check and insure the disk has a write protect tab on it.

STEP 2 Insert your new unformatted disk (TARGET Disk) in Drive B.

STEP 3 SCREEN DISPLAYS:

A>

STEP 4 YOU TYPE:

format b:

NOTE: Be sure to include the drive specifier B: otherwise DOS will format the disk in the default drive (Drive A) and you will erase your System Disk.

STEP 5 Press <ENTER>

SCREEN DISPLAYS:

Insert new disk for drive B:
and strike any key when ready

STEP 6 Since you have already inserted a new disk in Drive B, press any key.

SCREEN DISPLAYS:

Formatting...

The In Use Indicator for Drive B will light up. You will hear the drive operating as it formats the disk.

The computer tells you when it has completed the operation.

SCREEN DISPLAYS:

Formatting...Format complete

xxxxxx bytes total disk space
xxxxxx bytes available on disk

Format another (Y/N)?

STEP 7 YOU TYPE:

y

if you want to format another disk.

n

to end the formatting session.

SCREEN DISPLAYS:

A>

to indicate the system is ready for another operation.

NOTE: They are several options available within the FORMAT command. See FORMAT command in Chapter 7 of this manual.

The DIR Command

How does the operating system keep track of all your files? Well, just like any other well-organized information system, the computer has a **directory**. Each disk contains the directory for its files.

You do not need to update the directory as you add, delete or rename files. The computer does it automatically, so the directory always indicates the current status of the disk.

Along with the names of the files, the directory also contains information on the sizes of the files and the total amount of space utilized on the disk.

To access this information, you use the DIR command.

View the Directory of the Current Disk

STEP 1 Insert your System Disk in Drive A.

STEP 2 YOU TYPE:

dir

or

dir a:

STEP 3 Press <ENTER>

SCREEN DISPLAYS:

Volume in drive A has no label

Directory of A: \

COMMAND	COM	16341	4-23-85	4:00p
ANSI	SYS	1721	4-23-85	4:00p
CONFIG	SYS	22	4-23-85	4:00p
RCLOCK	SYS	1389	4-25-85	1:10p
ASSIGN	COM	1077	7-02-84	10:00a
BACKUP	COM	3466	8-25-84	10:00a
CHKDSK	COM	6468	9-03-83	3:25p
DEBUG	COM	12146	9-05-83	12:40a
DISKCOMP	COM	2525	2-01-85	10:00a
DISKCOPY	COM	2405	2-01-85	10:00a
EDLIN	COM	8110	9-17-83	1:33p
EXTPRT	COM	189	4-23-85	4:00p
FAST	COM	156	4-23-85	4:00p
FDISK	COM	6793	7-02-84	10:00a
FORMAT	COM	6827	8-25-84	10:00a
GRAPHICS	COM	679	12-21-83	4:00p
MODE	COM	2306	12-21-83	4:00p
MORE	COM	4364	9-12-83	10:35a
PRINT	COM	3808	9-03-83	5:55p
.				
.				
.				
.				
SAMPLE		1	6-01-85	9:53a
	XXXXXX		XXXXX	XXXXX

A>

The directory lists several pieces of information about each file.

The file names are listed in the first column (extensions are included when present—COM is an extension or a Command file).

NOTE: It is quite possible that there are some differences between this Directory list in the Reference Guide and the one on the screen.

The second column indicates the number of bytes occupied by the file.

The third column lists the date that the file was last accessed.

The last column gives the time of last use.

The total number of files, and the amount of free space still on the disk are given at the bottom.

Notice the last file listed. It is the file we created to use in these DOS exercises. Be sure you are using the System Disk which contains the sample file.

When your directory is very large, it may not all fit on the screen at one time. Then when you type dir, the display goes flying by, rolling off the top to be replaced by new files on the bottom. To view the directory in sections:

YOU PRESS:

<Ctrl>+<Num Lock> or <Ctrl>+<S>

This will freeze the directory on the screen. To resume scrolling (adding information from the bottom as it goes off the top) press any key.

View the DIR of the Disk in Drive B

DIR displays the names of the files on the disk in the default drive (the default drive is always A unless otherwise specified.)

To view the directory of the disk in drive B:

YOU TYPE:

dir b:

Then press <ENTER>

Even after you request the directory for Drive B, the A> prompt returns because A is still the default drive.

Viewing the Wide Directory

Use of the /W parameter is convenient when the directory is too long and does not fit on the screen. With it, up to 95 files can be viewed at one time on the screen. However, the number of bytes occupied by the file and the date and time the file was created are not displayed.

STEP 1 YOU TYPE:

dir/w

or

dir a:/w

or

dir b:/w

STEP 2 Press <ENTER>

STEP 3 SCREEN DISPLAYS:

A>dir/w

Volume in drive A has no label

Directory of A:\

COMMAND	COM ANSI	SYS CONFIG	SYS RCLOCK	SYS ASSIGN	COM
BACKUP	COM CHKDSK	COM DEBUG	COM DISKCOMP	COM DISKCOPY	COM
EDLIN	COM EXTPRT	COM FAST	COM FDISK	COM FORMAT	COM
GRAPHICS	COM MODE	COM MORE	COM PRINT	COM RECOVER	COM
RESTORE	COM SYS	COM TREE	COM EXE2BIN	EXE FC	EXE
FIND	EXE LINK	EXE SORT	EXE BASIC	COM BASICA	COM
BASICA	EXE DIAG	COM DEMO	BAT EXMENU	BAS GRAPH	BAS
SPEC3	BAS SPREAD	BAS WORD3	BAS CRTBYT2	BIN BACKLOOK	BYT
DATHLOOK	BYT DATLOOK	BYT PRTLOOK	BYT ULOOK	BYT WORD3	DAT
PANASONI	WRK				

46 File(s) 55296 bytes free

A>

Verify a Specific File

You may also use the DIR command to verify that a specific file is resident on the disk. To find out the file size, date and time of a file:

YOU TYPE:

dir sample

and press <ENTER>

The screen will display the information for the sample file.

If you wanted to find out the file existed on the disk in Drive B:

YOU TYPE:

dir b: sample

and press <ENTER>

OPERATION

FT-70

The COPY Command

The **COPY** command makes a copy of an individual file (DISK-COPY makes a copy of an entire disk.)

Use COPY when you:

need another one copy of the file.

want to rearrange a file (like cutting and pasting print copy)

There are several ways to copy files:

copy a file to another disk using the same name

copy a file to another disk using a new name

copy a file to the same disk using a new name (you cannot have two files with the same name on the same disk)

copy a file to the same disk using a name already in use on the disk (this replaces the information stored under that filename)

OPERATION

FT-70

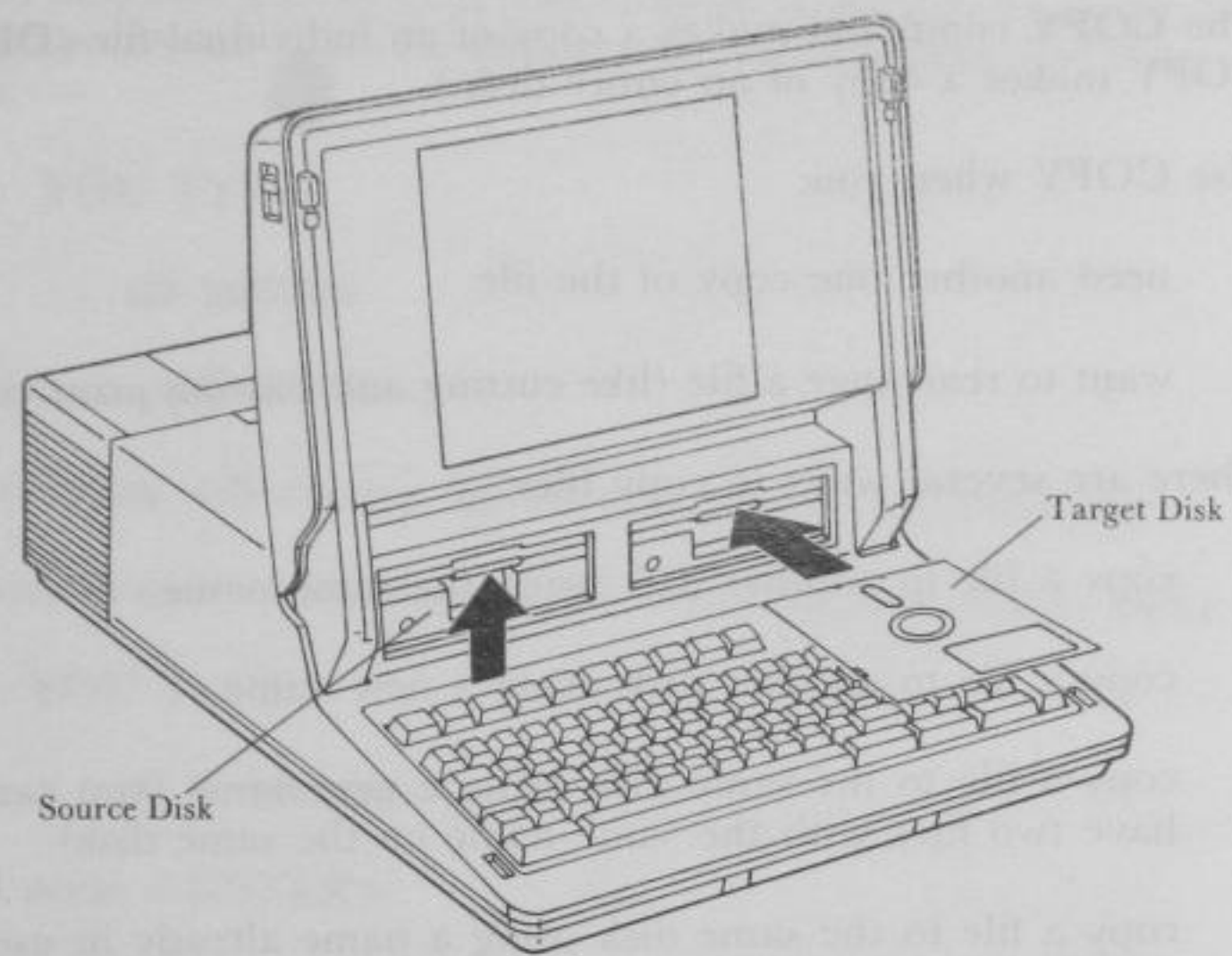


Fig. 3-19. USING THE COPY COMMAND WITH FT-70

COPY the File to Another Disk (Same Name)

STEP 1 Insert the SOURCE disk in Drive A and the TARGET disk in Drive B.

SCREEN DISPLAYS:

A>

STEP 2 YOU TYPE:

copy sample b:

STEP 3 Press <ENTER>

SCREEN DISPLAYS:

1 File(s) copied

STEP 4 (Optional)

You can verify that the file has been copied by using the DIR command.

YOU TYPE:

dir b:

and press <ENTER>

SCREEN DISPLAYS:

Volume in drive B has no label
Directory of B:

SAMPLE 106	1-01-80	12:22a
1 File(s)	361472 bytes free	

COPY the File to Another Disk (Change the Name)

You can give the file a new name when you copy it to a different disk. The file contents will still be the same.

In STEP 2 YOU TYPE:

copy sample b:example

and press <ENTER>

NOTE: If you tried this exercise, you now have two copies of the file on drive B—SAMPLE and EXAMPLE. Both files contain the contents of the original sample file on the disk in Drive A.

COPY the File to the Same Disk

You must change the name of your file if you want to make a copy of it on the same disk. Each file on a disk must have a unique name, OTHERWISE THE MOST CURRENT FILE WILL REPLACE THE EXISTING FILE. In our example we will change the name of our file to practice.

STEP 1 Insert the SOURCE disk in Drive A.

SCREEN DISPLAYS:

A>

STEP 2 YOU TYPE:

copy sample practice

STEP 3 Press <ENTER>

SCREEN DISPLAYS:

1 File(s) copied

STEP 4 (Optional)

You may check to see that both files are on the disk using the DIR command.

The TYPE Command

The **TYPE** command lets you display the contents of a file. You can also print the contents at the same time. When you are using **TYPE** you are only “looking” at the file. You cannot access a file to make changes using the **TYPE** command.

In order to use **TYPE** you must know the exact name of the file including any file specifications. To be sure the file is on the disk you are using, or to check for the exact file name, use the **DIR** command.

TYPE the Contents of a File on the Current Disk

STEP 1 Insert the System Disk in Drive A (original sample file).

SCREEN DISPLAYS:

A>

STEP 2 YOU TYPE:

type sample

STEP 3 Press <ENTER>

SCREEN DISPLAYS:

This is a practice file.
We will call it sample.
Sample helps explain DOS.

NOTE: If you want to print the contents of a file, press <Ctrl> <PrtSc> before typing “type sample”. When printing finishes, press <Ctrl> <PrtSc> again, then the printer is released.

TYPE the Contents of a File on Another Disk

If you want to see a file displayed which is on another disk you must add a drive specifier to the command.

- STEP 1 Insert the disk containing the file in Drive B (the copied disk).

SCREEN DISPLAYS:

A>

- STEP 2 YOU TYPE:

type b:sample

- STEP 3 Press <ENTER>

SCREEN DISPLAYS:

This is a practice file.
We will call it sample.
Sample helps explain DOS.

The RENAME Command

The **RENAME** command allows you to change the name of a file. You may change both the filename and the extension or only one of the components of the name.

RENAME is useful when you have two files whose names are very similar and you keep confusing them. Or maybe at the time you named the file its contents seemed perfectly clear, but now the name is confusing. Most of the time you RENAME a file just because a different name would be more useful to you.

RENAME a File on the Current Disk

- STEP 1 Insert the disk containing the file you want to rename into Drive A.
(Our System Disk contains both the **sample** file and the copied **practice** file)

SCREEN DISPLAYS:

A>

- STEP 2 We are going to rename our file called **practice** to **exercise**.

YOU TYPE:

rename practice exercise

The "old name" comes first, then a space, and the "new name".

- STEP 3 Press <ENTER>

SCREEN DISPLAYS:

A>

STEP 4 The computer does not verify the rename procedure. To check on the rename use the DIR command.

YOU TYPE:

dir

and press <ENTER>

SCREEN DISPLAYS:

(the directory for A.)

Note that **practice** no longer appears. It has been replaced by **exercise**.

RENAME a File on Another Disk

STEP 1 Insert the disk containing the file to be renamed in Drive B. (This is the disk we have been copying to.)

SCREEN DISPLAYS:

A>

STEP 2 YOU TYPE:

rename b:example exercise

STEP 3 Press <ENTER>

SCREEN DISPLAYS:

A>

STEP 4 Use the DIR command to verify the rename (remember to type dir b:).

The ERASE Command

Like everything that accumulates, you can occasionally find yourself with a lot of files—many of which are outdated, or have outlived their usefulness. **ERASE** is a convenient way to eliminate these files, leaving you usable space on your disks.

HOWEVER, ONCE YOU ERASE A FILE IT IS GONE FOREVER. USE THIS COMMAND WITH CARE.

Be especially careful when you are typing in the name of the file to be erased. This is crucial if you have files with similar names.

One note of hope. If you make backups regularly (and this is very strongly recommended) you may be able to retrieve a file which was accidentally deleted from your backup.

NOTE: ERASE and DEL (delete) are the same command.

ERASE a File on the Current Disk

STEP 1 Insert the disk containing the file to be erased in Drive A.

(First time users insert the backup System Disk).

SCREEN DISPLAYS:

A>

STEP 2 Since we are now finished with our practice file, we will erase it from the System Disk.

YOU TYPE:

erase sample

STEP 3 Press <ENTER>

SCREEN DISPLAYS:

A>

STEP 4 The computer does not verify the ERASE procedure. To verify the deletion use the dir command.

If you wish you may also ERASE the **exercise** file from the disk.

ERASE a File From Another Disk

STEP 1 Insert the disk containing the file to be deleted in Drive B.

(First time users insert the disks containing the copied files.)

SCREEN DISPLAYS:

A>

STEP 2 YOU TYPE:

erase b:sample

STEP 3 Press <ENTER>

SCREEN DISPLAYS:

A>

STEP 4 If you want to verify the deletion use the dir command (type dir b:)

If you wish you may also ERASE the **exercise** file from the disk in Drive B.

Using DOS

This has been an introduction to the many ways in which DOS can help you manage your files. Now that you are familiar with some of the DOS commands, you are ready to put DOS to work for you!

Your complete DOS reference manual begins on page 6-1.

BACKING UP THE SYSTEM (FOR USING FT-80)

What is a Backup?

The very first operation you should perform on your new Exec. Partner is to make a copy of your System Disk. This provides insurance for you, in case the original disk becomes damaged. This copy is called a "backup". Making the copy is called "backing up" the system.

It is a good practice to use your "backup" System Disk in your everyday operations. Store the original System Disk in a safe location.

You will need two disks to perform this operation. The first, your System Disk, is the one you wish to backup. The disk you are "copying from" is called the original or SOURCE disk. Your source disk is already loaded in Drive A.

The second disk will become the backup. It is the disk you will be "copying to". This is the destination or TARGET disk. Have on hand a new, unformatted disk.

original system disk (copy from)=SOURCE disk

new, unformatted disk (copy to)=TARGET disk

Your System Disk is write-protected. This means you cannot write to the disk. This protects you from erasing the stored information on the disk (see WRITE-PROTECT NOTCH earlier in the chapter.) If you get confused and attempt to write to your System Disk, this message will appear:

Write protect error writing drive A(B)
Abort, Retry, Ignore?

When this message appears, three responses should be selected:

A for Abort. The system ends the program that requested the device read or write.

R for Retry. The system will try again to do the device read or write operation.

I for Ignore. The system processes the program as if the error had not happened.

With pressing A, the system aborts the program, with R, it retries the operation, and with I, it ignores the problem and continues processing.

Device Designations

Before you begin working with the DOS, disk operating system, you should become familiar with drive designations.

You must tell DOS where to store file is or where to put the results of a procedure. You do this by using drive designations as follows.

- | | |
|----------|---|
| A: or a: | Tells DOS to get this information from or write this information to the floppy disk in Drive A. |
| B: or b: | Tells DOS to get this information from or write this information to the floppy disk in Drive B. |
| C: or c: | Tells DOS to get this information from or write this information to the hard disk in Drive C. |

*** NOTE ***

Before using drive designation C:, you must create a DOS partition on the hard disk and format the hard disk. A detailed description of this process is located on page 5-2.

Default Drive

The DOS prompt A> means that DOS is currently getting and sending information from the disk in Drive A. DOS always assumes that the file you want is on the current disk. As long as you want to read or write from the current drive, you do not need to include the drive designator in commands.

You can change the default drive. Suppose you wanted to work with a group of files on the disk in Drive C. You don't want to specify C: in every file name. You can change the default drive to C.

A>

This is the DOS prompt. It tells you that DOS is ready, but it also tells you that Drive A is the default drive.

YOU TYPE:

c:

and press <ENTER>

SCREEN DISPLAYS:

c>

C> is also a DOS prompt. It tells you that Drive C is the default drive.

To return to A as the default just type a: in response to the C> prompt.

NOTE: DOS always assumes it will find the file or write the results to the default drive unless you indicate a drive specifier in your commands.

The Procedure of making the Backup System Disk with FT-80

The DISKCOPY commands transfers the entire contents of one disk to another disk. You will use DISKCOPY to make your Backup System Disk. In the future, use DISKCOPY any time to want to copy an entire disk.

Using DISKCOPY With FT-80

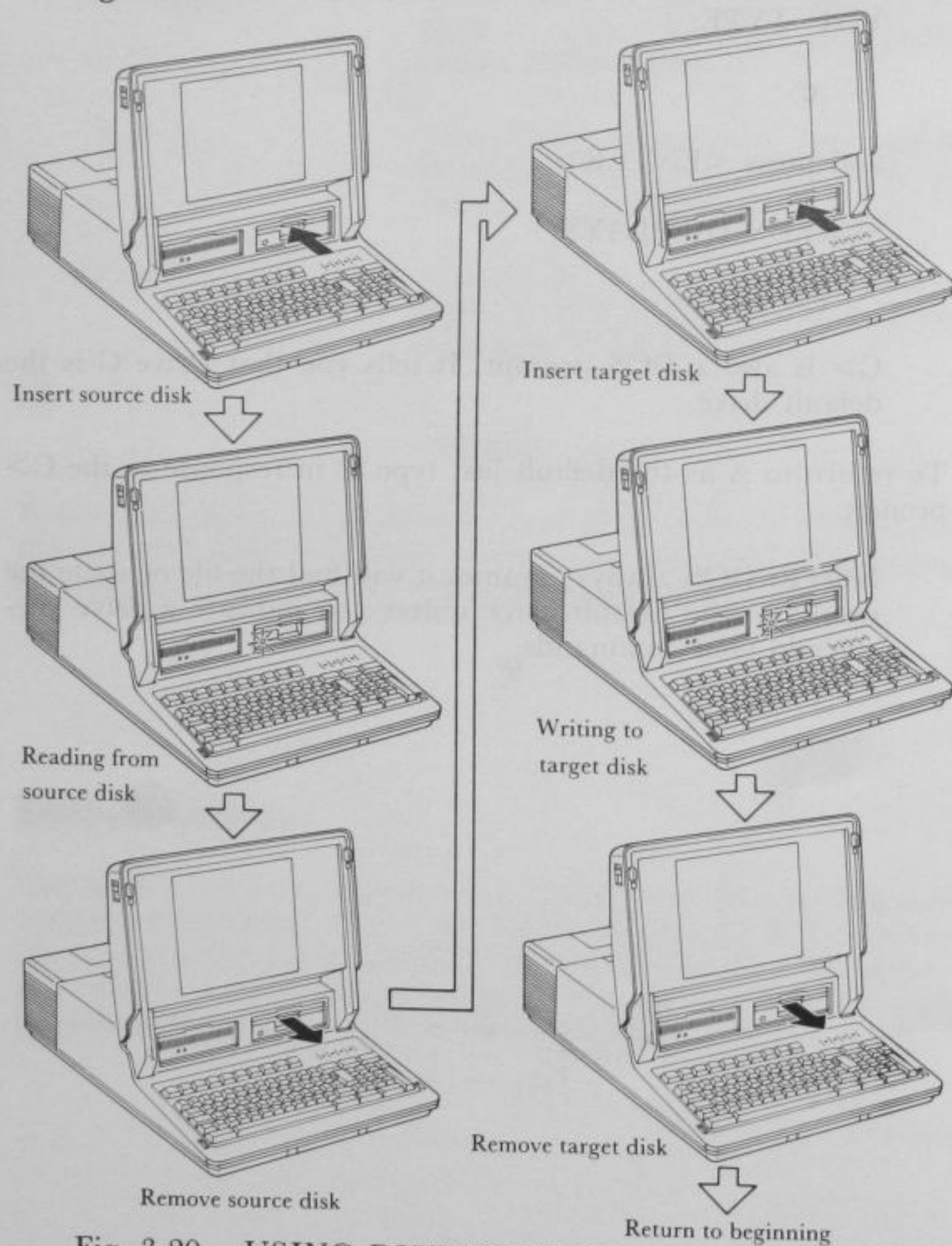


Fig. 3-20. USING DISKCOPY WITH FT-80

The same process happens when you copy with a single drive. But in order to pass on the information from the SOURCE disk, the computer must momentarily hold it in memory, while you insert the TARGET disk. This means you may have to switch disks several times during the copying procedure.

The messages are easy to follow if you remember:

original disk (copy from)=SOURCE disk
backup disk (copy to)=TARGET disk

STEP 1 Be sure you have inserted your SOURCE disk in Drive A.

STEP 2 SCREEN DISPLAYS:

A>

TYPE:

diskcopy

STEP 3 Press <ENTER>

SCREEN DISPLAYS:

Insert source disk into drive A:
Press any key when ready

STEP 4 Since your SOURCE disk is already in drive A, press any key.

You will see the In Use Indicator for Drive A come on as the original disk is being read into memory.

SCREEN DISPLAYS:

Copying 9 sectors per track, double sided
Insert target disk into drive A:
Press any key when ready

STEP 5 Remove your SOURCE disk. Remove a new, unformatted disk from its package. This will be your TARGET disk.

STEP 6 Insert your new, unformatted TARGET disk into the disk drive.

STEP 7 Press any key.

You will see the In Use Indicator come on as the information is written to the TARGET disk.

SCREEN DISPLAYS:

Formatting while copying
Insert source disk into drive A:
Press any key when ready

The computer has formatted and copied one section of the SOURCE disk onto the TARGET disk. It now needs the next section to store into memory.

STEP 8 Remove the TARGET disk from the drive and insert the SOURCE disk.

STEP 9 Press any key.

SCREEN DISPLAYS:

Insert target disk into drive A:
Press any key when ready

The computer has now written the second section to the backup disk.

STEP 10 You will need to keep "swapping" the SOURCE and TARGET disk until the entire disk has been copied.

When the procedure is completed the computer sends this message:

SCREEN DISPLAYS:

Copy complete

Copy another (Y/N)?

STEP 11 TYPE:

y

if you want to make another copy

n

to end the session.

SCREEN DISPLAYS:

A>

to indicate that the diskcopying sessions is completed.

NOTE: Now that you have successfully completed the copying of your original System Disk, return it to its jacket and store it carefully according to the directions for disk storage.

Label your copied disk "System Disk-Backup". Use this copy in your everyday operations. If your backup disk becomes damaged or inoperable make a new backup from your original System Disk.

Since the surface of the disk is covered with a magnetic coating, other magnetic fields can wreak havoc with stored information. Keep disks away from magnetic field sources such as telephones, dictation equipment, x-rays and calculators.

CREATING A FILE (FOR USING FT-80)

Files

The many types of tasks you will perform using the Exec. Partner will all be based on files. A file is a collection of data which is managed by your computer.

Like an enormous file cabinet, your computer stores all of its information in files. In order to find data, the computer must be instructed exactly where to look. (Don't forget, the computer by itself, is dumb. All it can do is follow instructions!)

Some files come built into your Exec. Partner. These files help the computer to regulate itself and operate the various parts that make up your system. These internal files are stored in ROM.

Your System Disk also contains operating files for your Exec. Partner. Once they are loaded into memory they perform automatically upon demand.

If you make a list of all the files on the System Disk (with the DIR command explained later in this chapter), you will notice that the System Disk contains many files. Among these are command files used to execute MS-DOS.

The disks which operate your application programs also store information files.

After you finish working on a program or text that you have created using BASIC or EDLIN, you can save it in a file on the disk.

To do so, you must name the file.

Once saved on a disk, you can load a file into memory any number of times whenever you need it.

A file is very convenient. You can change its contents, add to it, or even delete it whenever you want to.

A Files' Name

All files on a disk are identified by their file names. Your Exec. Partner searches for a file on the disk using the file name as a clue and then process the file.

That's why you must name the program or text that you created with a unique name.

A file is simply a collection of related data stored on a single disk. You can have many different types of files.

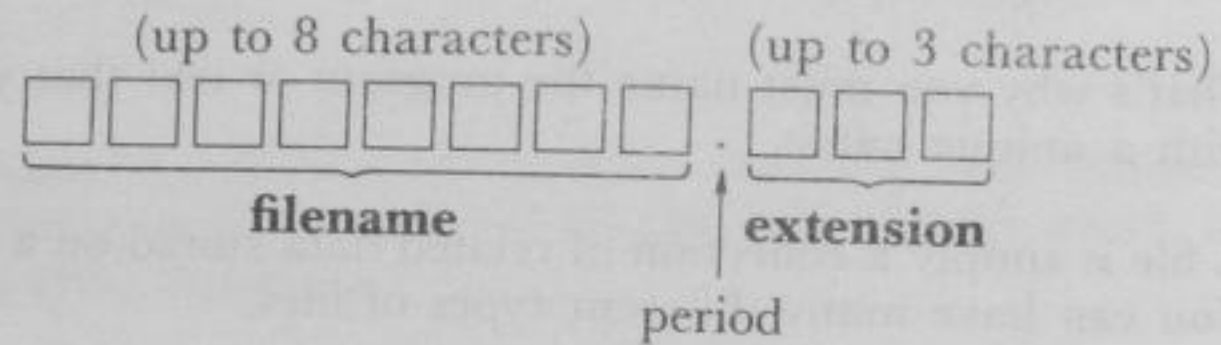
Example:	SCHED	Contains your upcoming appointments
	ACCTREC	Contains your accounts receivable
	BDAY	Contains a list of important birthdays
	FORMLET	Contains a form letter
	GAMES	Contains educational programs

You may store several files on one disk. Sometimes the files may be related. Or a disk may contain all of the files which you access on a regular basis. It does not matter to the computer how you organize and store your files.

NOTE: If you store information under an existing file name, the old information will be replaced by the most current entry. This is a good way to update files, but be careful not to inadvertently lose data by assigning it to already existing filename.

Filename and Extensions

A filename consists of two parts.



The first section is the **filename**, the second section is the **extension**.

A **filename** may contain as many as eight **characters**. The **extension** always begins with a **period** and may contain up to **three additional characters**.

DOS allows up to eight characters in the filename, a period plus up to three additional characters in the extensions. Within these restrictions you can name a file almost anything you want. But remember, the file's name is for your reference so be sure the name is convenient to type and easy to remember.

Example: SCHED.JUL
ACCREC.CUR
BDAY.BAS
FORMLET.84
GAMES.JOD

You can use any of these characters to name your file:

The letters of the alphabet (upper or lower case)

The numbers 0 through 9

These special characters—

\$ # & @ ! % () - , ' _

DOS does have some restrictions on filenames. Here are some unacceptable file names:

Q and R.COM Filenames cannot contain spaces

"SPEECH".BAS Filenames cannot contain quotation marks

A,B,C.EXE Filenames cannot contain commas

.NG Filename is missing

TOOMANYCHARACTERSINTHISNAME Too long

Extensions

Although the filename is required, the extension is optional except for special files.

There are several extensions having special significance in MS-DOS. In other words, special extensions are added to certain types of files.

Among them, the following 3 extensions are often seen:

- .COM** Indicates a **COMmand** file. The program in a file with this extension can be executed simply by typing the filename.
- .EXE** Indicates an **EXEcutible** file. It is similar to the .COM file, and at this point can be considered to be the same.
- .BAT** Indicates a **BATch** file. A predefined sequence of commands can be executed simply by typing the filename of a file with this extension.

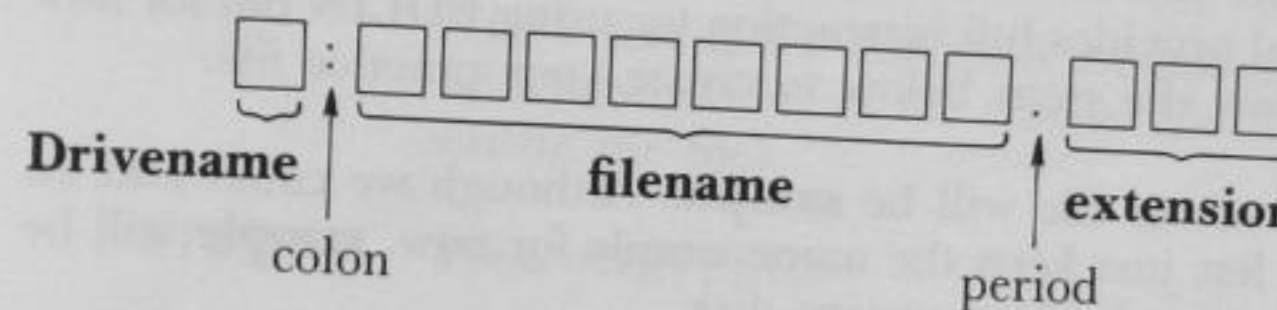
Drive Specifiers

In addition to the filename and optional extensions, the computer must know one more piece of information to retrieve or store a file. It must know **WHERE** to find the file.

The following type of Exec. Partners is available:

FT-80 1 floppy disk drive and 1 hard disk drive

To specify a file in a certain drive, you add the drive name to the beginning of the file name.



In MS-DOS, you can perform some operation on a desired file located in a certain drive by specifying the drive name in front of the file name, regardless of the current drive.

Example: Suppose the current drive is A and you want to specify files located in drive B and C. You would add drive specifiers like in the following.

B: SCHED. JUL

B: BDAY. BAS

C: GAMES. JOD

Sometimes you don't need to type the drive specifier. If the file you are creating will reside in the default or current drive, or the file you want to read is in the default drive you don't need to indicate the specifier. So if the default drive is A (indicated by the A> prompt) you could type either:

Example: A: SCHED. JUL

SCHED. JUL

Creating a File

Managing files with DOS is simply a matter of practice. DOS is really very easy to use and in no time at all you will be wondering how you managed to do any work in pre Exec. Partner days!

While you are learning however, you may make a few small mistakes. So, to prevent any of your valuable data from being damaged, we are going to create a practice file to use while you acquaint yourself with some fundamental DOS commands.

You use EDLIN to create a file. EDLIN is a special part of DOS which allows you to create, change and display files. Chapter 9 of this manual provides full instruction for using EDLIN but for now simply follow the steps below to create your practice file.

The name of our file will be **sample**. Although we could add an extension, let's just keep the name simple for now. **sample** will be located on your backup system disk.

NOTE: Do not proceed with these exercises unless you have made a backup of your original System disk. See "backing up the system" earlier in this chapter.

When instructions refer to your System Disk, they are referring to this backup version.

STEP 1 Insert your System Disk in Drive A. (It is assumed Exec. Partner is already on.)

SCREEN DISPLAYS:

A>

STEP 2 YOU TYPE:

edlin sample

STEP 3 Press <ENTER>

SCREEN DISPLAYS:

New File

*

The * is the prompt for EDLIN

STEP 4 YOU TYPE:

i

The i stands for Insert, it means you want to add information to the named file (in this case create the file).

STEP 5 Press <ENTER>

SCREEN DISPLAYS:

1:*

The number 1 stands for the line number. EDLIN references all input by line numbers. Each time you press <ENTER> a new line number will appear at the beginning of the next line.

STEP 6 YOU TYPE:

This is a practice file.

STEP 7 Press <ENTER>

SCREEN DISPLAYS:

1:*This is a practice file.

2:*_

STEP 8 Continue typing in the text for the practice file.

YOU TYPE and PRESS:

We will call it sample. <ENTER>
Sample will help explain DOS. <ENTER>

NOTE: If you make a mistake while typing in the contents of the file, use <BACKSPACE> to erase the characters and then retype the entry. You must edit a line before you type <ENTER>.

Even if you do enter incorrect lines this is only a practice file. The actual contents don't really make any difference.

STEP 9 The screen should look like this:

SCREEN DISPLAYS:

```
A>edlin sample
New File
*i
  1:*This is a practice file.
  2:*We will call it sample.
  3:*Sample will help explain DOS.
  4:*_
```

STEP 10 Now indicate the end of the file.

YOU PRESS:

<Ctrl> <Z> (hold down <Ctrl> while pressing <Z>, then let go of both keys)

SCREEN DISPLAYS:

```
4:*^Z_
```

STEP 11 Press <ENTER>

SCREEN DISPLAYS:

*

The file is completed and we are back to the EDLIN prompt.

STEP 12 To check the contents of your file ask for a **listing** of the file.

YOU TYPE:

l (lowercase letter L)

and press <ENTER>

The screen will type out your sample file.

STEP 13 Having created the file, you need to exit from EDLIN and return to DOS.

YOU TYPE:

e

and press <ENTER>

This command will save the file to the default drive. The In Use Indicator will light up and you will hear the drive turning. The sample file is being written to the disk in drive A.

SCREEN DISPLAYS:

```
A>
```

to indicate DOS is now ready.

NOTE: See Chapter 9 for a description of EDLIN.

Before Using DOS Commands

You have created your file using EDLIN. Now we will use it with a number of MS-DOS commands. We use MS-DOS commands to manipulate the file loaded into memory.

The many DOS commands are described individually in detail in Chapter 7 COMMAND REFERENCE. In this section, we will consider the basic DOS commands that you will most likely use in your daily operations.

We will introduce the following commands:

FORMAT A command which must be executed before you first start using a disk, whether it is a floppy disk or hard disk. Once a disk is formatted, there is no need to format it again.

NOTE: If you format a disk which has already been formatted, you will lose all files on the disk.

DIR DIR stands for DIRectory. The DIR command displays (on the screen) the names of the files on a disk, the file creation date, and the file size.

TYPE Displays the contents of the specified file on the screen.

NOTE: This command is provided for use only on text files.

COPY Copies a file on the disk.

NOTE: Before copying a file, the disk must be formatted. However, it need not be formatted if you use the DISKCOPY command.

RENAME Used to change the name of a file.

NOTE: The RENAME command can be used regardless of whether or not the file to be renamed is in the current drive.

ERASE Used to erase unwanted files from the disk.

NOTE: The ERASE command can be used regardless of whether or not the file to be erased is in the current drive.

We have briefly described the functions of these useful commands. A more detailed introduction describing the format of these commands, including examples, is given in the next section.

GETTING ACQUAINTED WITH DOS: FT-80

The FORMAT Command

When you purchase disk they are blank. Specific computers store information on disks in different formats. Before you can use a new disk, you must specify the format of your YOUR SPECIFIC MACHINE. To get the disk ready to receive information, use the FORMAT Command.

You only format a disk once, the very first time you use it. Since FORMAT treats the disk as a blank it will erase any information already on a disk.

You can use FORMAT to prepare a disk for new information. Just be sure you NO LONGER NEED the data on the disk.

NOTE: You do not need to format a disk before using DISKCOPY. DISKCOPY will perform the format operation first if necessary, then copy the data.

Using FORMAT with FT-80

Until you become accustomed to using a hard disk drive system, the instructions from the computer may seem confusing.

In a single floppy disk drive system, the computer calls your floppy disk drive "Drive A" or "Drive B" even though there is one floppy disk drive. Because it must read instructions and data, temporarily store them in memory, and then perform the correct procedure on the second disk, you must "swap" two disks in a single drive unit.

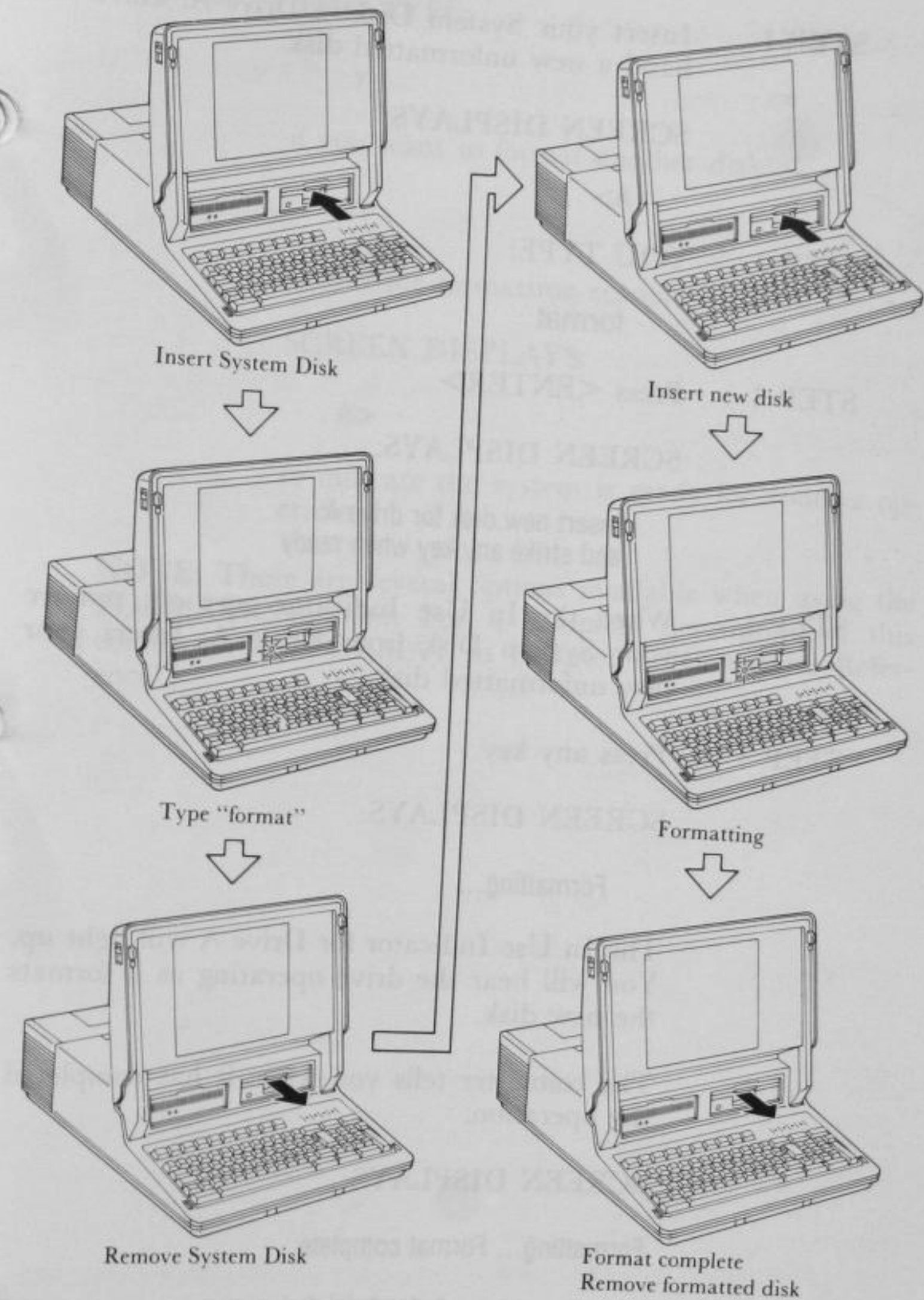


Fig. 3-21. USING FORMAT WITH FT-80

STEP 1 Insert your System Disk in Drive A. Have on hand a new unformatted disk.

SCREEN DISPLAYS:

A>

STEP 2 YOU TYPE:

format

STEP 3 Press <ENTER>

SCREEN DISPLAYS:

Insert new disk for drive A:
and strike any key when ready

STEP 4 When the In Use Indicator goes off, remove your System Disk from Drive A. Insert your new unformatted disk.

STEP 5 Press any key.

SCREEN DISPLAYS:

Formatting...

The In Use Indicator for Drive A will light up. You will hear the drive operating as it formats the new disk.

The computer tells you when it has completed the operation.

SCREEN DISPLAYS:

Formatting... Format complete

xxxxxx bytes total disk space
xxxxxx bytes available on disk

Format another (Y/N)?

STEP 6 YOU TYPE:

y

if you want to format another disk.

n

to end the formatting session.

SCREEN DISPLAYS:

A>

to indicate the system is ready for another operation.

NOTE: There are several options available when using the FORMAT command. For a complete description of this command, see FORMAT in Chapter 7 "Command Reference".

The DIR Command

How does the operating system keep track of all your files? Well, just like any other well-organized information system, the computer has a **directory**. Each disk contains the directory for its files.

You do not need to update the directory as you add, delete or rename files. The computer does it automatically, so the directory always indicates the current status of the disk.

Along with the names of the files, the directory also contains information on the sizes of the files, and the total amount of space utilized on the disk.

To access this information, you use the DIR command.

View the Directory of the Current Disk

STEP 1 Insert your System Disk in Drive A.

STEP 2 YOU TYPE:

dir

or

dir a:

STEP 3 Press <ENTER>

SCREEN DISPLAYS:

Volume in drive A has no label
Directory of A:\

COMMAND	COM	16341	4-23-85	4:00p
ANSI	SYS	1721	4-23-85	4:00p
CONFIG	SYS	22	4-23-85	4:00p
RCLOCK	SYS	1389	4-25-85	1:10p
ASSIGN	COM	1077	7-02-84	10:00a
BACKUP	COM	3466	8-25-84	10:00a
CHKDSK	COM	6468	9-03-83	3:25p
DEBUG	COM	12146	9-05-83	12:40a
DISKCOMP	COM	2525	2-01-85	10:00a
DISKCOPY	COM	2405	2-01-85	10:00a
EDLIN	COM	8110	9-17-83	1:33p
EXTPRT	COM	189	4-23-85	4:00p
FAST	COM	156	4-23-85	4:00p
FDISK	COM	6793	7-02-84	10:00a
FORMAT	COM	6827	8-25-84	10:00a
GRAPHICS	COM	679	12-21-83	4:00p
MODE	COM	2306	12-21-83	4:00p
MORE	COM	4364	9-12-83	10:35a
PRINT	COM	3808	9-03-83	5:55p
.				
.				
.				
SAMPLE		1	6-01-85	9:53a
	XXXXXX		XXXXX	XXXXX

A>

The directory lists several pieces of information about each file.

The file names are listed in the first column (extensions are included when present—COM is an extension or a Command file).

The second column indicates the number of bytes occupied by the file.

NOTE: It is quite possible that there are some changes between this Directory list in the Reference Guide and the one on the screen.

The third column lists the date that the file was last accessed. If you do not enter a new date when you turn on this computer, this date will not change even if you use the file.

The last column gives the time of last use.

The total number of files, and the amount of free space still on the disk are given at the bottom.

Notice the last file listed. It is the file we created to use in these DOS exercises. Be sure you are using the System Disk which contains the sample file.

When the directory is very large, it may not all fit on the screen at one time. Then when you type dir, the display goes flying by, rolling off the top to be replaced by new files on the bottom. To view the directory in sections:

YOU PRESS:

<Ctrl> <Num Lock> or <Ctrl> <S>

This will freeze the directory on the screen. To resume scrolling (adding information from the bottom as it goes off the top) press any key.

Viewing the Wide Directory

Use of the /W parameter is convenient when the directory is too long and does not fit on the screen. With it, up to 95 files can be viewed at one time on the screen. However, the number of bytes occupied by the file and the date and time the file was created are not displayed.

STEP 1 YOU TYPE:

dir/w

or

dir a:/w

STEP 2 Press <ENTER>

STEP 3 SCREEN DISPLAYS:

A>dir/w

Volume in drive A has no label

Directory of A: \

COMMAND	COM ANSI	SYS CONFIG	SYS RCLOCK	SYS ASSIGN	COM
BACKUP	COM CHKDSK	COM DEBUG	COM DISKCOMP	COM DISKCOPY	COM
EDLIN	COM EXTPRT	COM FAST	COM FDISK	COM FORMAT	COM
GRAPHICS	COM MODE	COM MORE	COM PRINT	COM RECOVER	COM
RESTORE	COM SYS	COM TREE	COM EXE2BIN	EXE FC	EXE
FIND	EXE LINK	EXE SORT	EXE BASIC	COM BASICA	COM
BASICA	EXE DIAG	COM DEMO	BAT EXMENU	BAS GRAPH	BAS
SPEC3	BAS SPREAD	BAS WORD3	BAS CRTBYT2	BIN BACKLOOK	BYT
DATHLOOK	BYT DATLOOK	BYT PRTLOOK	BYT UPLOOK	BYT WORD3	DAT
PANASONI	WRK				

46 File(s) 55296 bytes free

A>

Verify a Specific File

You may also use the DIR command to verify that a specific file is resident on the disk. To find out the file size, date and time of a file:

YOU TYPE:

```
dir sample
```

and press <ENTER>

The screen will display the information for the sample file.

The COPY Command

The **COPY** command makes a copy of an individual file (DISK-COPY makes a copy of an entire disk).

Use COPY when you:

need more than one copy of the file

need to backup a changed file on a disk

need to make changes in a file but want a backup of the original

want to rearrange a file (like cutting and pasting print copy)

There are several ways to copy files:

copy a file to another disk using the same name

copy a file to another disk using a new name

copy a file to the same disk using a new name (you cannot have two files with the same name on the same disk)

copy a file to the same disk using a name already in use on the disk (this replaces the information stored under that filename)

You must "swap" the source and target disks. The messages for the COPY command sound as if there are two drives (Drive A and Drive B). To make it easier for yourself, imagine your disks as drives. The source disk is "Drive A", the target disk is "Drive B".

Using the COPY Command with FT-80

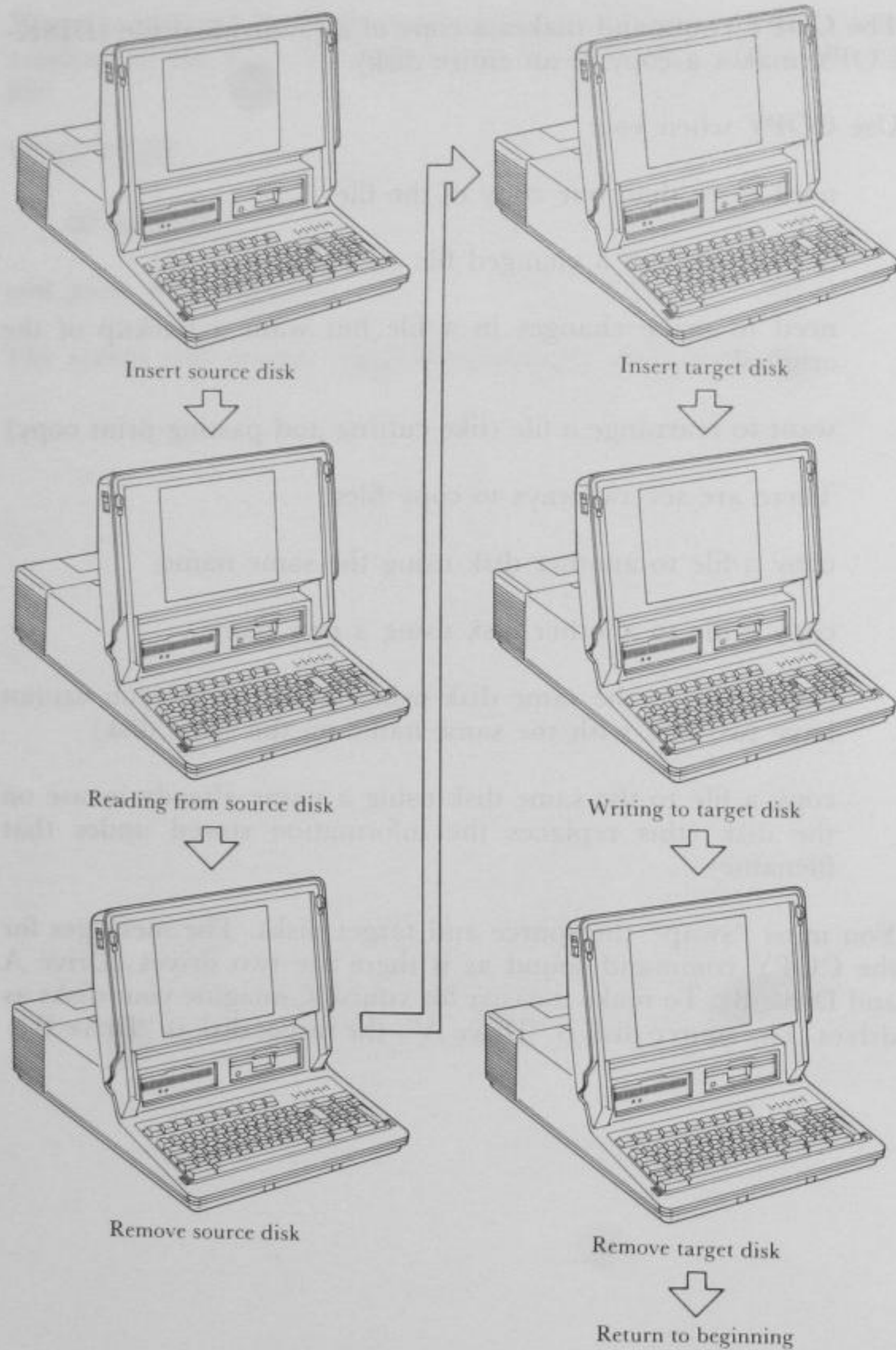


Fig. 3-22. USING THE COPY COMMAND WITH FT-80

COPY the File to Another Disk (Same Name)

STEP 1 Insert the source disk in Drive A. Have on hand a new, formatted disk to be used as your target disk.

SCREEN DISPLAYS:

A>

STEP 2 YOU TYPE:

copy sample b:

STEP 3 Press <ENTER>

SCREEN DISPLAYS:

Insert disk for drive B: and strike any key when ready.

STEP 4 Remove your SOURCE Disk and insert the new disk in Drive A.

STEP 5 Press any key to tell DOS you have swapped the disks.

STEP 6 DOS can copy only so much information at one time. If your file is large you may see this message.

SCREEN DISPLAYS:

Insert disk for drive A: and strike any key when ready.

STEP 7 DOS needs to read another section of the file into memory.

Remove the TARGET Disk from the drive and insert your SOURCE Disk.

OPERATION

FT-80

OPERATION

FT-80

- STEP 8 Press any key to tell DOS you have swapped the disks.
- STEP 9 After the additional information is read into memory, this message appears.

SCREEN DISPLAYS:

Insert disk for drive B: and strike any key when ready

- STEP 10 Remove the SOURCE disk and insert the TARGET disk to receive the next section of the file.
- STEP 11 Press any key to indicate the disks have been swapped.

Continue alternating disks until this message is displayed.

SCREEN DISPLAYS:

1 File(s) copied

A>

The A> prompt indicates that DOS is ready for another operation.

- STEP 12 (Optional)
- You can verify that the file has been copied by using the DIR command.
- Be sure the disk which contains the copied file (TARGET disk) is in Drive A.

COPY the File to Another Disk (Change the Name)

You can give the file a new name when you copy it to a different disk. The file contents will still be the same.

In STEP 2 YOU TYPE:

copy sample b:example

and press <ENTER>

NOTE: If you tried this exercise, you now have two copies of the file on the target disk—SAMPLE and EXAMPLE. Both files contain the contents of the original SAMPLE file on the source disk.

COPY the File to the Same Disk

You must change the name of your file if you want to make a copy of it on the same disk. Each file on a disk must have a unique name, OTHERWISE THE MOST CURRENT FILE WILL REPLACE THE EXISTING FILE. In our example we will change the name of our file to practice.

STEP 1 Insert the SOURCE disk in Drive A.

SCREEN DISPLAYS:

A>

STEP 2 YOU TYPE:

copy sample practice

STEP 3 Press <ENTER>

SCREEN DISPLAYS:

1 File(s) copied

STEP 4 (Optional)

You may check to see that both files are on the disk using the DIR command.

The TYPE Command

The **TYPE** command lets you display the contents of a file. You can also print the contents at the same time. When you are using **TYPE** you are only "looking" at the file. You cannot access a file to make changes using the **TYPE** command.

In order to use **TYPE** you must know the exact name of the file including any file specifications. To be sure the file is on the disk you are using, or to check for the exact file name, use the **DIR** command.

TYPE the Contents of a File

- STEP 1 Insert the disk containing the file to be displayed in Drive A.

SCREEN DISPLAYS:

A>

- STEP 2 If you want to print the file while it is displayed, press <Ctrl> <PrtSc>. (Hold down <Ctrl> while you press <PrtSc>. Then release both keys.)

- STEP 3 YOU TYPE:

type sample

- STEP 4 Press <ENTER>

SCREEN DISPLAYS:

This is a practice file.
We will call it sample.
Sample helps explain DOS.

The contents of the file are printed at the same time. After printing finishes, press <Ctrl> <PrtSc> again, then the printer is deactivated.

The RENAME Command

The **RENAME** command allows you to change the name of a file. You may change both the filename and the extension or only one of the components of the name.

RENAME is useful when you have two files whose names are very similar and you keep confusing them. Or maybe at the time you named the file its contents seemed perfectly clear, but now the name is confusing. Most of the time you **RENAME** a file just because a different name would be more useful to you.

RENAME a File on the Current Disk

- STEP 1 Insert the disk containing the file you want to rename into Drive A.
(Our System Disk contains both the **sample** file and the copied **practice** file)

SCREEN DISPLAYS:

A>

- STEP 2 We are going to rename our file called **practice** to **exercise**.

YOU TYPE:

rename practice exercise

The "old name" comes first, then a space, and the "new" name.

- STEP 3 Press <ENTER>

SCREEN DISPLAYS:

A>

STEP 4 DOS does not verify the rename procedure. To check on the rename use the DIR command.

YOU TYPE:

dir

and press <ENTER>

SCREEN DISPLAYS:

(the directory for A.)

Note that **practice** no longer appears. It has been replaced by **exercise**.

The ERASE Command

Like everything that accumulates, you can occasionally find yourself with a lot of files—many of which are outdated, or have outlived their usefulness. **ERASE** is a convenient way to eliminate these files, leaving you usable space on your disks.

HOWEVER, ONCE YOU ERASE A FILE IT IS GONE FOREVER. USE THIS COMMAND WITH CARE.

Be especially careful when you are typing in the name of the file to be erased. This is crucial if you have files with similar names.

One note of hope. If you make backups regularly (and this is very strongly recommended) you will have a copy of any file which is accidentally deleted.

NOTE: ERASE and DEL (delete) are the same command.

ERASE a File on the Current Disk

STEP 1 Insert the disk containing the file to be erased in Drive A.

SCREEN DISPLAYS:

A>

STEP 2 Since we are now finished with our practice file, we will erase it from the System Disk.

YOU TYPE:

erase sample

STEP 3 Press <ENTER>

SCREEN DISPLAYS:

A>

STEP 4 DOS does not verify the ERASE procedure. To verify the deletion use the DIR command.

If you wish you may also ERASE the **exercise** file from the disk.

Using DOS

This has been an introduction to the many ways in which DOS can help you manage your files. Now that you are familiar with some of the DOS commands, you are ready to put DOS to work for you!

Your complete DOS reference manual begins on page 6-1.

SWITCHING THE SYSTEM STATUS

How to Display 40 Char. or 80 Char.

Use the **MODE** command to set the display mode at 40 or 80 characters per line. Your System Disk must be in Drive A to use this command. 80 characters is the default setting.

Setting the Display Mode to 40 Characters Per Line

STEP 1 Insert your System Disk into Drive A.

SCREEN DISPLAYS:

A>

STEP 2 YOU TYPE:

mode 40

STEP 3 Press <ENTER>

Your display is set to 40 characters per line.

Setting the Display Mode to 80 Characters Per Line

STEP 1 Insert your System Disk into Drive A.

SCREEN DISPLAYS:

A>

STEP 2 YOU TYPE:

mode 80

STEP 3 Press <ENTER>

Your display is set to 80 characters per line.

How to Switch to FAST Processing Speed

The CPU clock frequency in the Exec. Partner can be switched to 4.77 MHz or 7.16 MHz. Setting the CPU clock frequency to 7.16 MHz greatly increases the processing speed of the Exec. Partner. When the Exec. Partner operates at Fast Speed Mode, the Fast Speed Indicator light will be on. Switch the CPU clock frequency using one of the 3 methods given below.

1. Pressing the <Alt>+<Ctrl>+<+> keys

The CPU clock frequency switches by holding down the <Alt>+<Ctrl> keys and pressing <+> key on the ten key pad. Pressing these keys again returns to the previous CPU clock frequency. This key combination acts as a toggle between 4.77 MHz and 7.16 MHz. The indicator will be lit for the fast speed.

2. Setting the DIP Switch (sets only the default CPU clock frequency)

If the DIP switch located on the main circuit board is on, the CPU clock frequency is set to 4.77 MHz, and when off, is set to 7.16 MHz.

Please refer to page 2-42.

3. Using the FAST Command

Perform the steps below to switch the CPU clock frequency to 7.16 MHz.

STEP 1 Insert your System Disk into Drive A.

SCREEN DISPLAYS:

A>

STEP 2 YOU TYPE:

FAST

STEP 3 Press <ENTER>

NOTE: Certain application programs do not allow you to switch the CPU clock frequency through the keyboard. For these programs, switch the CPU clock frequency before running the program or set the default DIP switch before booting the system.

To set the CPU clock frequency to 7.16 MHz when booting the system, set the DIP switch to off before booting or use the an AUTOEXEC.BAT file with the FAST command.

Using the AUTOEXEC.BAT File

Use the AUTOEXEC.BAT file so that the CPU clock frequency is set to the fast speed when the system is booted.

STEP 1 Insert your System Disk into Drive A. (It is assumed that the Exec. Partner is already on.)

SCREEN DISPLAYS:

A>

STEP 2 YOU TYPE:

edlin AUTOEXEC.BAT

STEP 3 Press <ENTER>

SCREEN DISPLAYS:

New File

*

The * is the prompt for EDLIN

STEP 4 YOU TYPE:

i

STEP 5 Press <ENTER>

SCREEN DISPLAYS:

1:*

STEP 6 YOU TYPE:

FAST

STEP 7 Press <Ctrl>+<Z>

SCREEN DISPLAYS:

2:*^Z_

STEP 8 Press <ENTER>

SCREEN DISPLAYS:

*

STEP 9 YOU TYPE:

e

and press <ENTER>

SCREEN DISPLAYS:

A>

STEP 10 Once you reboot the system,

SCREEN DISPLAYS:

FAST MODE ON

And the Fast Speed indicator will light up. Your Exec. Partner will now operate at the Fast Speed. If you want to return to the Standard Speed, press the <Alt>+<Ctrl>+<+> keys.

Using the Real Time Clock

A clock based on a crystal oscillator is built into your Exec. Partner. Once you set the date and time, the built-in clock will keep the accurate date and time. To enable the real time clock in your Exec. Partner to operate.

DEVICE=RCLOCK.SYS

has been set in the CONFIG.SYS file to add the real time clock function to the system.

The real time clock will stop operating if you delete the CONFIG.SYS file or delete the DEVICE=RCLOCK.SYS setting in the CONFIG.SYS file.

NOTE: The CONFIG.SYS file sets variables for MS-DOS when booted. The contents of this file is easily edited using EDLIN.

Setting the Date and Time

If the AUTOEXEC.BAT file does not exist, the date and time will be set when the system is rebooted. To set the date and time without rebooting the system, use the **DATE** command and **TIME** command.

Suppose you want to set the date to December 20, 1985 and the time to 10:10 and 50 seconds.

STEP 1 YOU TYPE:

DATE

STEP 2 Press <ENTER>

SCREEN DISPLAYS:

Current date is day mm-dd-yy
Enter new Date: _

STEP 4 YOU TYPE:

12-20-1985

STEP 5 Press <ENTER>

STEP 6 YOU TYPE:

TIME

STEP 7 Press <ENTER>

STEP 8 SCREEN DISPLAYS:

Current time is hh:mm:ss:xx
Enter new time: _

STEP 9 YOU TYPE:

10:10:50

STEP 10 Press <ENTER>

Switching Between the Internal Printer and External Parallel Printer

A printer is built into your Exec. Partner. You can also use an external printer by connecting it to the serial port or parallel port, depending on its type. Switch to an external parallel printer using either of the 2 methods given below.

1. Pressing the <Alt>+<PrtSc> Keys.

Press the <PrtSc> key holding down the <Alt> key and the Ext. Printer indicator above the keyboard on the right side will light up.

2. Using the EXTPRT command to switch to an external parallel printer.

NOTE: To set the external printer as the default printer when booting the system, follow the steps below.

STEP 1 Insert your System Disk into Drive A. (It is assumed that the Exec. Partner is already on.)

SCREEN DISPLAYS:

A>

STEP 2 YOU TYPE:

edlin AUTOEXEC.BAT

STEP 3 Press <ENTER>

SCREEN DISPLAYS:

New file

*

The * is the prompt for EDLIN.

STEP 4 YOU TYPE:

i

STEP 5 Press <ENTER>

SCREEN DISPLAYS:

1:*

STEP 6 YOU TYPE:

EXTPRT

STEP 7 Press <Ctrl> <Z>

SCREEN DISPLAYS:

2:*^Z_

STEP 8 Press <ENTER>

SCREEN DISPLAYS:

*

STEP 9 YOU TYPE:

e

and press <ENTER>

SCREEN DISPLAYS:

A>

Once you reboot the system, the Ext. Printer indicator will light up. Your Exec. Partner will now print on the connected external printer.

NOTE: If the external printer is connected to the serial port, see page 4-4.

BASIC

Your Exec. Partner can use the BASIC programming language.

Included in your Exec. Partner package is a BASIC Reference Guide. This manual provides complete programming instruction.

BASIC is contained on your System Disk. Once you have loaded BASIC into the computer's memory, it is ready to use.

NOTE: Do not attempt to use BASIC until you have made a backup of your System Disk. See "Backing Up the System" on page 3-34 (FT-70) and 3-72 (FT-80).

To get into BASIC:

STEP 1 Insert your System Disk in Drive A.

SCREEN DISPLAYS:

A>

STEP 2 YOU TYPE:

basica

or

basic

See **NOTE 1** and/or **2**.

STEP 3 and press <ENTER>. After messages are displayed, the screen displays

Ok

Ok is the BASIC prompt. You are now ready to program.

STEP 4 To return to DOS

YOU TYPE:

system

and press <ENTER>

SCREEN DISPLAYS:

A>

to indicate DOS is ready for the next operation.

NOTE 1:

We offer two kinds of BASIC, named BASIC and BASICA which means Advanced BASIC. Advanced BASIC contains the additional statements and functions.

NOTE 2:

When you make your own program, we recommend you use BASICA, because BASIC is a subset of BASICA. In case the particular programs can only run under BASIC, you have to follow the instructions of the programs.

CHAPTER 4 PRINTER

1. SWITCHING BETWEEN THE INTERNAL PRINTER AND EXTERNAL PRINTER	4-2
The Internal Printer	4-2
Using an External Printer by Connecting it to the Parallel Port	4-3
Using an External Printer by Connecting it to the Serial Port	4-4
2. SPECIAL KEYS FOR THE PRINTER	4-6
Indicator	4-6
Special Keys for the Printer	4-6
3. SETTING THE PRINTER DIP SWITCHES	4-8
How to Set the Printer DIP Switches	4-9
4. PRINTER CONTROL CODES	4-10
5. PRINTER CHARACTER SETS	4-38
Printer Character Set 1	4-38
Printer Character Set 2	4-40

SWITCHING BETWEEN THE INTERNAL PRINTER AND EXTERNAL PRINTER

You can use the printer built into the Exec. Partner or an external printer connected to the parallel port or serial port for printing. We will describe how to connect an external printer and how to switch between the internal (built-in) printer and external printer.

The Internal Printer

If the Ext. Printer indicator is not lit, the internal printer operates as the default printer.

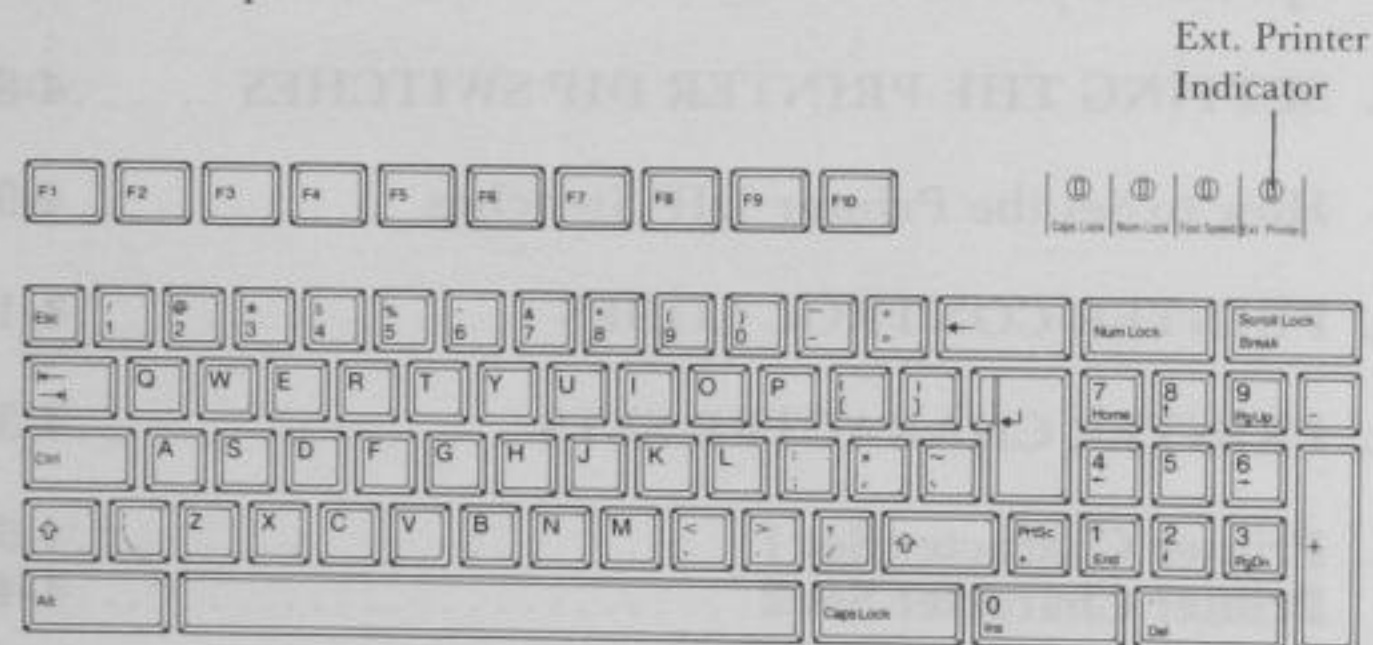


Fig. 4-1. EXT. PRINTER INDICATOR

In this state, the device names are as follows:

PRN: or **LPT1:** Internal printer

LPT2: External printer connected to the parallel port

If the external printer is the default printer, the Ext. Printer indicator will be lit. To switch the default printer to the internal printer, hold down the <Alt> key and press the <PrtSc> key. The Ext. Printer indicator goes off.

NOTE: The default printer switches from the internal printer to the external parallel printer or vice versa by pressing the <Alt>+<PrtSc> keys.

Using an External Printer by Connecting it to the Parallel Port

You can use an external printer by connecting it to the parallel port. (Be sure it is a parallel or Centronics type printer.) Connect the external printer to the parallel port and switch to the external printer using either of the 2 methods given below.

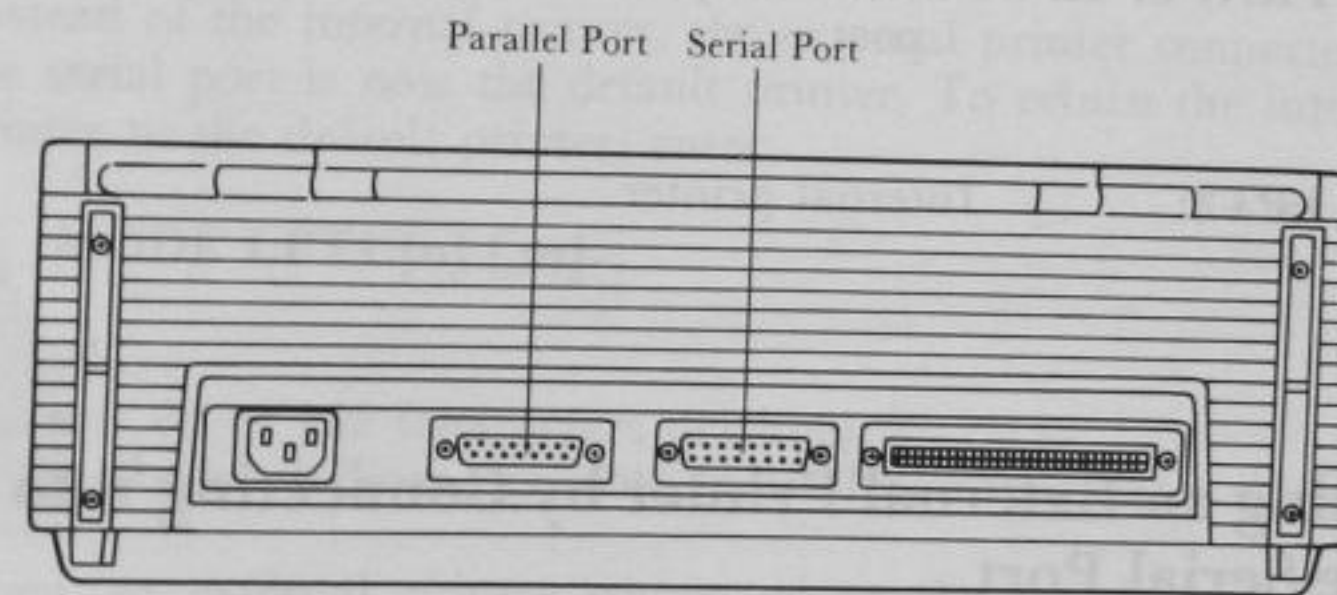


Fig. 4-2. PARALLEL PORT AND SERIAL PORT

Method 1: Pressing the <Alt>+<PrtSc> keys.

Hold down the <Alt> key and press the <PrtSc> key. The Ext. Printer indicator above the keyboard on the right side will light up.

Method 2: Using the EXTPRT command to switch to an external printer

STEP 1 Insert your System Disk into Drive A. (It is assumed that the Exec. Partner is already on.)

SCREEN DISPLAYS:

A>

STEP 2 YOU TYPE:

EXTPRT

STEP 3 Press <ENTER>

When the external printer is the default printer, the device names are as follows:

PRN: or **LPT1:** External printer connected to the parallel port

LPT2: Internal printer

Using an External Printer by Connecting it to the Serial Port

You can also use an external printer by connecting it to the serial port. (Be sure it is a serial printer.)

STEP 1 Connect the external printer to the serial port.

STEP 2 Turn on the Exec. Partner.

STEP 3 Insert your System Disk into Drive A.

SCREEN DISPLAYS:

A>

STEP 4 Set the mode of the serial interface with the MODE command. For now, we set the interface as follows.

YOU TYPE:

MODE COM1:30, N, 8, 1

This command sets the interface at a baud rate of 300, no parity, eight data bits, and one stop bit. See page 7-66 for details.

STEP 5 Press <ENTER>

STEP 6 YOU TYPE:

MODE LPT1:=COM1

STEP 7 Press <ENTER>

Instead of the internal printer, the external printer connected to the serial port is now the default printer. To return the internal printer to the default printer, enter:

MODE LPT1:[n] [,m]

where:

n is 80 or 132 (characters per line)

m is 6 or 8 (lines per inch, vertical line spacing)

When an external printer connected to the serial port is the default printer, the device names are as follows:

PRN: or **LPT1:** External printer connected to the serial port

LPT 2: External printer connected to the parallel port or the internal printer

SPECIAL KEYS FOR THE PRINTER

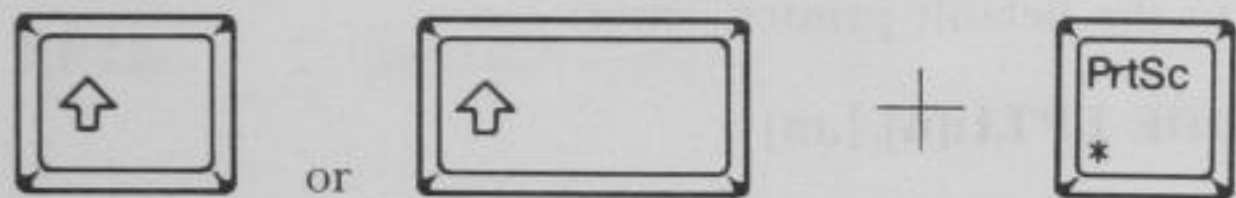
Indicator

Ext. Printer Indicates whether or not the external printer is selected (initially off)
Selection of the external printer is toggled by pressing <Alt>+<PrtSc>.
ON: External printer (parallel port) selected
OFF: Internal printer selected

Special Keys for the Printer

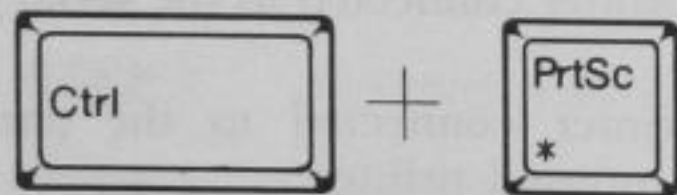
The following key combinations for printer control are available:

Print screen.



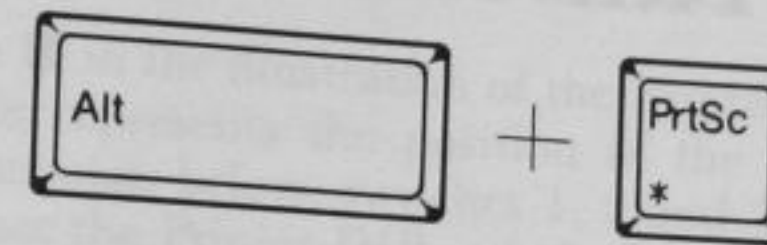
Produces a printed version (known as a screen dump) of what is on the screen. If graphics is being displayed for information on the GRAPHICS Command. See page 7-63.

Print what you are typing.



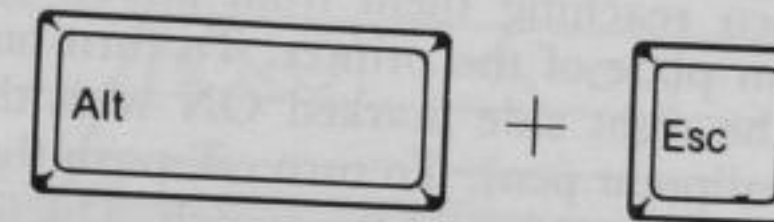
To print the characters typed on the CRT, press <PrtSc> while pressing <Ctrl>. The input characters are printed each time <ENTER> is pressed. Pressing <Ctrl> and <PrtSc> again stops the printing.

Switch printer.



The internal printer is switched to an external parallel printer or vice versa.

Reset the internal printer.



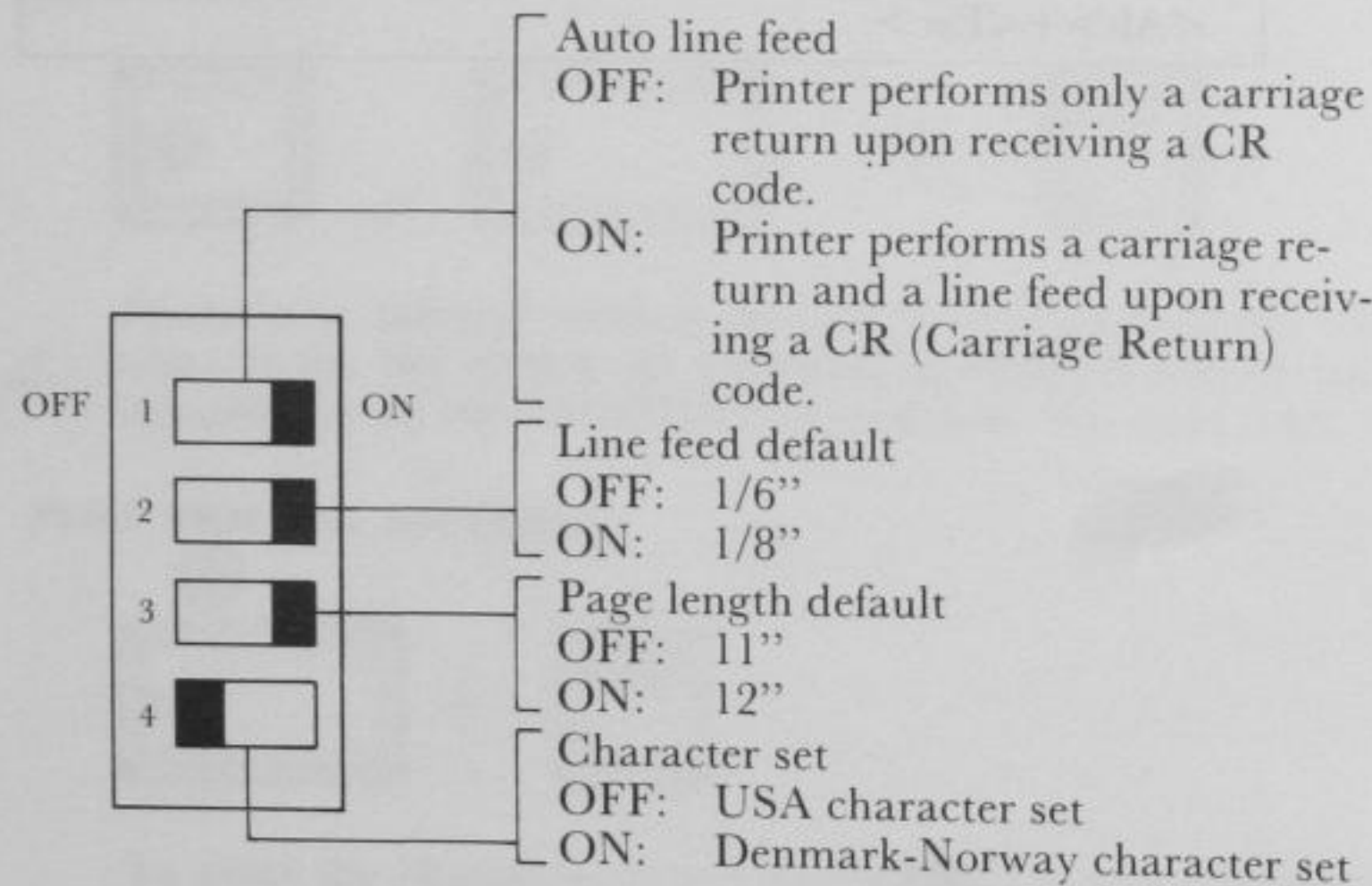
This key sequence will reset the active printer. If the printer error indicator is on or when you have changed the linefeed DIP switch of the internal printer (it is located inside the printer mechanism.), press <Alt>+<Esc>.

SETTING THE PRINTER DIP SWITCHES

The Printer DIP switches are located below the printer mechanism on the right side and are provided so that you can set the various features of the printer to suit your needs. The switches are off when you first receive the Exec. Partner.

The Printer DIP switches can be accessed by first detaching the plastic cover and then reaching them from above through the opening in the bottom plate of the printer. To turn on a switch, push the switch to the right side marked ON with the tip of a pointed object (e.g. ballpoint pen). To turn off, push the switch to the left side indicating the number of the switch. The four Printer DIP switches are numbered from 1 to 4.

The switches are:



How to Set the Printer DIP Switches

The ■ in the illustration of the Printer DIP switches shown in the table represents the position of the individual switches. In the illustration below, switches 1, 2 and 3 are on, and switch 4 is off. To set the Printer DIP switches, push the switches with the tip of a pointed object as shown in the illustration below:

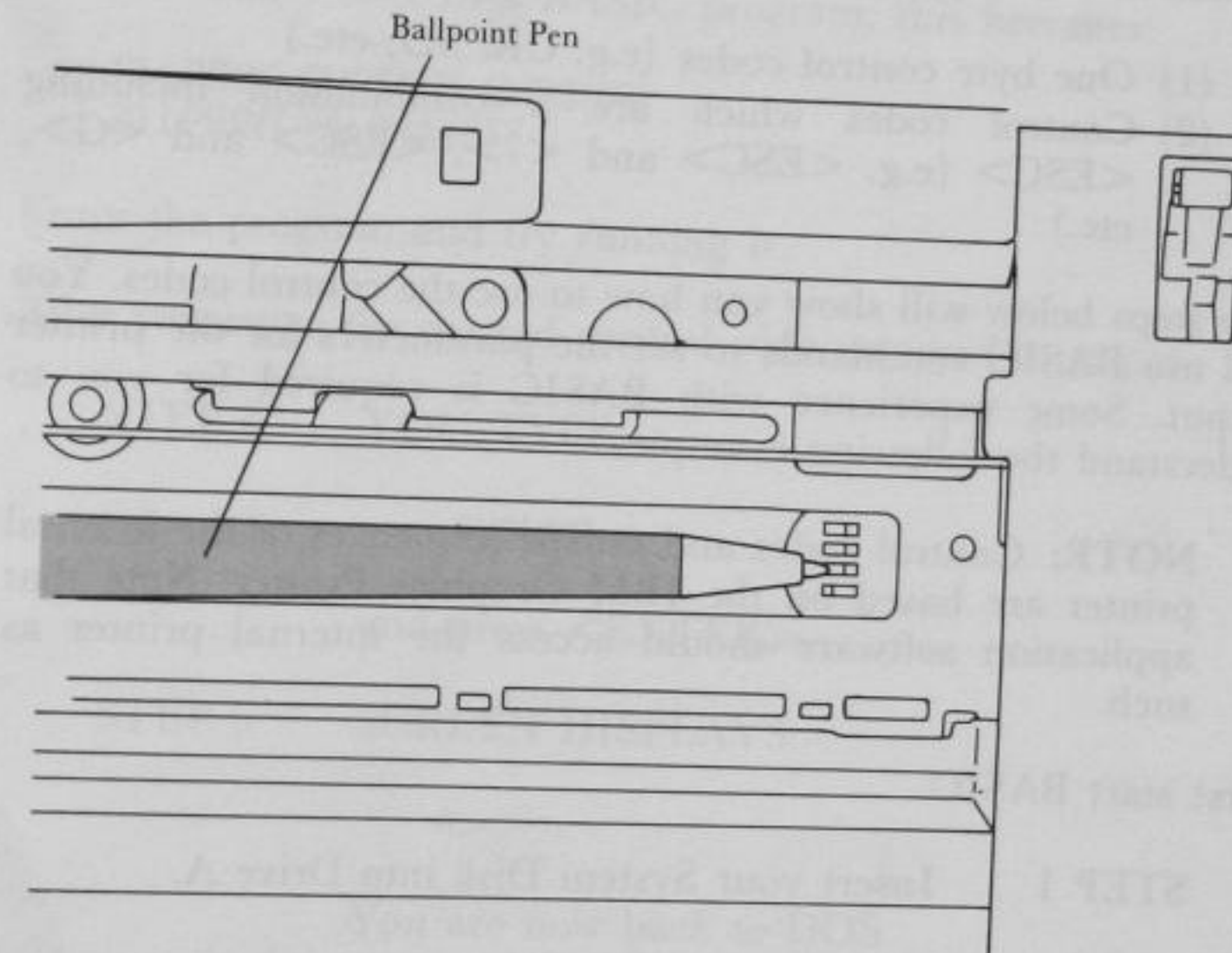


Fig. 4-3. PRINTER DIP SWITCHES

PRINTER CONTROL CODES

The print type and format of the internal printer can be set and changed using the printer control codes. Control codes can be divided into 2 major groups:

- (1) One byte control codes (e.g. CR, SO, etc.)
- (2) Control codes which are a combination including <ESC> (e.g. <ESC> and <7>, <ESC> and <G>, etc.)

The steps below will show you how to use the control codes. You will use BASIC commands to set the parameters for the printer output. Some experience with BASIC is required for you to understand the following examples.

NOTE: Control codes and escape sequences of the internal printer are based on the IBM Graphics Printer. Note that application software should access the internal printer as such.

First start BASIC.

STEP 1 Insert your System Disk into Drive A.

SCREEN DISPLAYS:

A>

STEP 2 YOU TYPE:

basica

or

basic

STEP 3 and press <ENTER>. After the opening messages are displayed,

SCREEN DISPLAYS:

Ok

Now, see this section for the desired control codes. To send control codes to the internal printer, use the BASIC statement:

```
LPRINT CHR$( )
```

Suppose we want to print the word "ALPHABET" in emphasized characters. The code for emphasized printing is the combination <ESC> and <E>. In a BASIC program, this becomes:

```
10 LPRINT CHR$(27); CHR$(69)
20 LPRINT "ALPHABET"
```

Enter the program and try running it.

After you enter the desired control code, proceed to step 4.

STEP 4 YOU TYPE:

system

and press <ENTER>.

STEP 5 SCREEN DISPLAYS:

A>

You are now back to DOS.

CR (Carriage Return)

Syntax: LPRINT CHR\$(13)

Purpose: Prints the data within the printer buffer and returns the carriage.

Example:

```
10 LPRINT "CR"; CHR$(13); "CR"
20 END

CR
CR
```

<ESC> and <0>

Syntax: LPRINT CHR\$(27); CHR\$(48)

Purpose: Sets the line spacing to 1/8 inch.

Example:

```
10 LPRINT CHR$(27); CHR$(48)
20 FOR A=1 TO 5
30 LPRINT "1/8 inch"
40 NEXT A
50 END

1/8 inch
1/8 inch
1/8 inch
1/8 inch
1/8 inch
```

<ESC> and <1>

Syntax: LPRINT CHR\$(27); CHR\$(49)

Purpose: Sets the line spacing to 7/82 inch.

Example:

```
10 LPRINT CHR$(27); CHR$(49)
20 FOR A=1 TO 5
30 LPRINT "7/82 inch"
40 NEXT A
50 END

7/82 inch
7/82 inch
7/82 inch
7/82 inch
7/82 inch
```

<ESC> and <2>

Syntax: LPRINT CHR\$(27); CHR\$(50)

Purpose: Sets the line spacing to 12/82 inch.

Example:

```
10 LPRINT CHR$(27); CHR$(50)
20 FOR A=1 TO 5
30 LPRINT "12/82 inch"
40 NEXT A
50 END

12/82 inch
12/82 inch
12/82 inch
12/82 inch
12/82 inch
```


<ESC> and <3+n>

Syntax: LPRINT CHR\$(27); CHR\$(51); CHR\$(n)
($1 \leq n \leq 246$)

Purpose: Sets the line spacing to n/246 inch.

Example:

```
10 LPRINT CHR$(27); CHR$(51); CHR$(105)
20 FOR A=1 TO 5
30 LPRINT "105/246 inch"
40 NEXT A
50 END
```

105/246

105/246

105/246

105/246

105/246

<ESC> and <A+n>

Syntax: LPRINT CHR\$(27); CHR\$(65); CHR\$(n)
($1 \leq n \leq 82$)

Purpose: Sets the line spacing to n/82 inch.

Example:

```
10 FOR A=1 TO 10
20 LPRINT CHR$(27); CHR$(65); CHR$(A);
CHR$(27); CHR$(65);
30 LPRINT "from 1 to 10, _____"
40 NEXT A
50 END
```



LF (Line Feed)

Syntax: LPRINT CHR\$(10)

Purpose: Prints the data in the print buffer and feeds a line. If there is no print data, only a line feed operation is executed.

<ESC> and <J+n>

Syntax: LPRINT CHR\$(27); CHR\$(74); CHR\$(n)
(only one line, $1 \leq n \leq 246$)

Purpose: Sets line feed n/246 inch.

Example:

```
10 FOR A=0 TO 30 STEP 5
20 LPRINT "Line Spacing"; CHR$(13);
30 LPRINT CHR$(27); CHR$(74); CHR$(A);
40 LPRINT "-----"; A
50 NEXT A
60 END
```

```
Line Spacing
----- 0
Line Spacing
----- 5
Line Spacing
----- 10
Line Spacing
----- 15
Line Spacing
----- 20
Line Spacing
----- 25
Line Spacing
----- 30
```

<ESC> and <C+n>

Syntax: LPRINT CHR\$(27); CHR\$(67); CHR\$(n)

Purpose: Sets the page length to n lines ($1 \leq n \leq 127$).

Example: To set the page length to 50 lines,
LPRINT CHR\$(27); CHR\$(67); CHR\$(50)

<ESC> and <C+0+n>

Syntax: LPRINT CHR\$(27); CHR\$(67); CHR\$(00);
CHR\$(n) ($1 \leq n \leq 22$)

Purpose: Sets the page length to n inches.

Example: To set the page length to 10 inches,
LPRINT CHR\$(27); CHR\$(67); CHR\$(00); CHR\$(10)

FF (Form Feed)

Syntax: LPRINT CHR\$(12)

Purpose: Advances the paper to the first printing line of the next page. After data in the line buffer is printed, the paper is advanced to the beginning of the next page.

VT

Syntax: LPRINT CHR\$(11)

Purpose: Prints the data in the printer buffer and executes the same operation as the line feed.

<ESC> and <D+n₁+n₂+ ... +n_x+NUL>

Syntax: LPRINT CHR\$(27); CHR\$(68); CHR\$(n₁);
CHR\$(n₂);+ ... +CHR\$(n_k)+CHR\$(0)
(1 ≤ k ≤ 28)

Purpose: Sets the horizontal tab positions. (Be sure to set the tabs in ascending order.)

PRINTER

HT

Syntax: LPRINT CHR\$(9)

Purpose: Sets the horizontal tab position to every 8th space.

Example:

```
10 FOR A=1 TO 5
20 LPRINT CHR$(9); "HT"
30 NEXT A
40 END
```

HT HT HT HT HT

SO (Shift Out)

Syntax: LPRINT CHR\$(14)

Purpose: Sets the printer to double width mode. (If a carriage return, line feed, form feed, or printing due to buffer-full is executed, this mode is automatically canceled.)

Example:

```
10 FOR A=1 TO 3
20 LPRINT CHR$(14); "DOUBLE, double"
30 LPRINT "MODE, mode"
40 NEXT A
50 END
```

```
DOUBLE, double
MODE, mode
DOUBLE, double
MODE, mode
DOUBLE, double
MODE, mode
```

PRINTER

DC4 (Device 4)

Syntax: LPRINT CHR\$(14) or LPRINT CHR\$(20)

Purpose: Cancels the double width mode. (The SO function is also automatically canceled by a CR or LF operation.)

Example:

```
10 LPRINT CHR$(14); "DOUBLE"  
20 LPRINT CHR(20); "MODE"  
30 END
```

```
DOUBLEMODE
```

<ESC> and <W+n>

Syntax: LPRINT CHR\$(27); CHR\$(87); CHR\$(n)
(n=0 or 1)

Purpose: Sets the printer to the double width mode. Double width printing begins when n=1 and ends when n=0.

Example:

```
10 LPRINT CHR$(27); CHR$(87); CHR$(1);  
20 LPRINT "WIDTH"  
30 LPRINT "WIDTH"  
40 LPRINT CHR$(27); CHR$(87); CHR$(0),  
50 LPRINT "WIDTH"  
60 END
```

```
WIDTH  
WIDTH          WIDTH
```

NOTE: When the double width mode is specified using this code, it is not automatically canceled by a CR or LF operation.

SI (Shift In)

Syntax: LPRINT CHR\$(15)

Purpose: Sets the printer to the compressed mode. (Setting the emphasized mode automatically cancels this mode.) This mode can be used together with the double strike mode.

Example:

```
10 LPRINT "WITH THIS STATEMENT"  
20 LPRINT CHR$(15); "COMPRESSED MODE"  
30 LPRINT "AUTOMATICALLY BEGINS"  
40 END
```

```
WITH THIS STATEMENT  
COMPRESSED MODE  
AUTOMATICALLY BEGINS
```

```
10 LPRINT "COMPRESSED MODE"  
20 LPRINT CHR$(15); "IS AUTOMATICALLY  
   CANCELED"  
30 LPRINT CHR$(27); CHR$(69); "BY EMPHASIZED  
   MODE"  
40 END
```

```
COMPRESSED MODE  
IS AUTOMATICALLY CANCELED  
BY EMPHASIZED MODE
```


DC2 (Device 2)

Syntax: LPRINT CHR\$(18)

Purpose: Cancels the compressed mode.

Example: 10 LPRINT CHR\$(15); "DC2 CANCELS"
20 LPRINT CHR\$(18); "COMPRESSED MODE"
30 END

```
DC2 CANCELS
COMPRESSED MODE
```

<ESC> and <E>

Syntax: LPRINT CHR\$(27); CHR\$(69)

Purpose: Sets the printer to the emphasized mode.

Example: 10 LPRINT "THIS IS NORMAL PRINTING"
20 LPRINT CHR\$(27); CHR\$(69); "NOW, THIS IS
EMPHASIZED PRINTING"
30 END

```
THIS IS NORMAL PRINTING
NOW, THIS IS EMPHASIZED PRINTING
```

<ESC> and <F>

Syntax: LPRINT CHR\$(27); CHR\$(70)

Purpose: Cancels the emphasized mode.

Example: 10 LPRINT "EMPHASIZED MODE"
20 LPRINT CHR\$(27); CHR\$(70); "IS NOW
CANCELED"
30 LPRINT "BY (ESC) AND (F)"
40 END

```
EMPHASIZED MODE
IS NOW CANCELED
BY (ESC) AND (F)
```


<ESC> and <G>

Syntax: LPRINT CHR\$(27); CHR\$(71)

Purpose: Sets the printer to the double strike mode.

Example:

```
10 LPRINT "IN THIS PAGE"
20 LPRINT CHR$(27); CHR$(71); "LETTERS PRINTED
   BY DOUBLE STRIKE MODE"
30 LPRINT "CONTRAST VERY WELL WITH THE
   OTHERS"
40 END
```

```
IN THIS PAGE
LETTERS PRINTED BY DOUBLE STRIKE MODE
CONTRAST VERY WELL WITH THE OTHERS
```

NOTE: The double strike mode can be used together with the emphasized mode. This combination produces a higher print quality.

Example:

```
10 LPRINT CHR$(27); CHR$(69); "THIS IS
   EMPHASIZED MODE"
20 LPRINT CHR$(27); CHR$(71); "DOUBLE STRIKE
   MODE HAS JUST ADDED TO"
30 LPRINT "EMPHASIZED MODE. CAN YOU SEE?"
40 END
```

```
THIS IS EMPHASIZED MODE
DOUBLE STRIKE MODE HAS JUST ADDED TO
EMPHASIZED MODE. CAN YOU SEE ?
```

<ESC> and <H>

Syntax: LPRINT CHR\$(27); CHR\$(72)

Purpose: Cancels the double strike mode.

Example:

```
10 LPRINT CHR$(27); CHR$(71); "DOUBLE STRIKE
   MODE PRINTING"
20 LPRINT CHR$(27); CHR$(72); "TURNS INTO
   NORMAL PRINTING"
30 END
```

```
DOUBLE STRIKE MODE PRINTING
TURNS INTO NORMAL PRINTING
```

<ESC> and <S+0>

Syntax: LPRINT CHR\$(27); CHR\$(83); CHR\$(0)

Purpose: Sets the superscript mode. In the superscript mode, characters are printed in the upper half of the line.

Example:

```
10 LPRINT "SUPERSCRIPIT MODE"
20 LPRINT "4=2";
30 LPRINT CHR$(27);CHR$(83);CHR$(0);
40 LPRINT "2"
50 LPRINT CHR$(27);CHR$(84)
```

```
SUPERSCRIPIT MODE
4=22
```

NOTE: The superscript mode can be used with the double width mode, compressed mode, or normal mode. It cannot be used with the double strike mode or emphasized mode.

<ESC> and <S+1>

Syntax: LPRINT CHR\$(27); CHR\$(83); CHR\$(1)

Purpose: Sets the subscript mode. In the subscript mode, characters are printed in the lower half of the line.

Example: 10 LPRINT "SUBSCRIPT MODE"
20 LPRINT "CO";
30 LPRINT CHR\$(27);CHR\$(83);CHR\$(1);
40 LPRINT "2"
50 LPRINT CHR\$(27);CHR\$(84)

```
SUBSCRIPT MODE
CO2
```

NOTE: The subscript mode can be used with the double width mode, compressed mode, or normal mode. It cannot be used with the double strike mode or emphasized mode.

<ESC> and <T>

Syntax: LPRINT CHR\$(27); CHR\$(84)

Purpose: Cancels the superscript mode or subscript mode.

<ESC> and <- + n>

Syntax: LPRINT CHR\$(27); CHR\$(45); CHR\$(0)
LPRINT CHR\$(27); CHR\$(45); CHR\$(1)
(n=0 or 1)

Purpose: Sets the underline mode. The underline begins when n=1 and ends when n=0.

Example: 10 LPRINT CHR\$(18)
20 LPRINT "ABCDEFGHIJ";
30 LPRINT CHR\$(27); CHR\$(45); CHR\$(1);
"ABCDEFGHIJ";
40 LPRINT CHR\$(27); CHR\$(71); "ABCDEFGHIJ";
50 LPRINT CHR\$(15); "ABCDEFGHIJ";
60 LPRINT CHR\$(27); CHR\$(45); CHR\$(0);
"ABCDEFGHIJ"
70 END

```
ABCDEFGHIJ ABCDEFGHIJABCDEFGHIJABCDEFGHIJ
```

<ESC> and <8>

Syntax: LPRINT CHR\$(27); CHR\$(56)

Purpose: Sets the printer to ignore the paper end.

<ESC> and <9>

Syntax: LPRINT CHR\$(27); CHR\$(57)

Purpose: Sets the printer to detect the paper end.

CAN

Syntax: LPRINT CHR\$(24)

Purpose: Clears the print buffer.

NUL

Syntax: LPRINT CHR\$(00)

Purpose: No operation. Used as a terminator for <Esc> and and <Esc> and <D>.

<ESC> and <6>

Syntax: LPRINT CHR\$(27); CHR\$(54)

Purpose: Selects Character Set 2.

<ESC> and <7>

Syntax: LPRINT CHR\$(27); CHR\$(55)

Purpose: Selects Character Set 1.

<ESC> and <<>

Syntax: LPRINT CHR\$(27); CHR\$(60)

Purpose: Returns the print head to the left margin after printing. This code is only effective each time it is specified.

Example:

```
10 LPRINT "return to"  
20 LPRINT CHR$(27); CHR$(60); "left margin"  
30 LPRINT "for next line"  
40 END
```

```
return to  
left margin  
for next line
```


<ESC> and <L+n₁+n₂>

Syntax: LPRINT CHR\$(27); CHR\$(76); CHR\$(n₁);
CHR\$(n₂) (0 ≤ n₁ ≤ 255, 0 ≤ n₂ ≤ 255)

Purpose: Changes the text mode to the bit image graphics mode (dual density). The operation is the same as <ESC> and <K+n₁+n₂> except that the maximum number of positions is 960.

Example:

```
10 WIDTH "1pt1:",255
20 A=300
30 LPRINT CHR$(27);CHR$(76);CHR$(44);CHR$(1);
40 FOR P=1 TO A
50 LPRINT CHR$(&H3F);
60 NEXT P
70 LPRINT "DUAL"
80 END
```

————— DUAL

PRINTER

<ESC> and <Y+n₁+n₂>

Syntax: LPRINT CHR\$(27); CHR\$(89); CHR\$(n₁);
CHR\$(n₂) (0 ≤ n₁ ≤ 255, 0 ≤ n₂ ≤ 255)

Purpose: Specifies double-speed, dual-density bit image printing. Up to 960 bit image positions can be printed in one line.

Example:

```
10 WIDTH "1pt1:",255
20 A=300
30 LPRINT CHR$(27);CHR$(89);CHR$(44);CHR$(1);
40 FOR P=1 TO A
50 LPRINT CHR$(&HFF);
60 NEXT P
70 LPRINT "DOUBLE SPEED"
80 END
```

————— DOUBLE SPEED

PRINTER

<ESC> and <Z+n₁+n₂>

Syntax: LPRINT CHR\$(27); CHR\$(90); CHR\$(n₁);
CHR\$(n₂)

Purpose: Specifies quad-density bit image printing. Up to 1920 bit image positions can be printed in one line.

Example:

```
10 WIDTH "1pt1:",255
20 A=300
30 LPRINT CHR$(27);CHR$(90);CHR$(44);CHR$(1);
40 FOR P=1 TO A
50 LPRINT CHR$(&HFF);
60 NEXT P
70 LPRINT "QUAD-DENSITY"
80 END
```

————— QUAD-DENSITY

<ESC> and <N+n>

Syntax: LPRINT CHR\$(27); CHR\$(78); CHR\$(n)
(1 ≤ n ≤ 127)

Purpose: Sets the skip perforation function. The number of lines to be skipped, n, must range from 1 to 127.

Example:

```
10 LPRINT CHR$(27); CHR$(67); CHR$(5);
20 LPRINT CHR$(27); CHR$(78); CHR$(2);
30 FOR A=1 TO 12
40 LPRINT "Skip Perforation Function"; A
50 NEXT A
60 END
```

```
Skip Perforation Function 1
Skip Perforation Function 2
Skip Perforation Function 3
```

```
Skip Perforation Function 4
Skip Perforation Function 5
Skip Perforation Function 6
```

```
Skip Perforation Function 7
Skip Perforation Function 8
Skip Perforation Function 9
```

```
Skip Perforation Function 10
Skip Perforation Function 11
Skip Perforation Function 12
```

<ESC> and <O>

Syntax: LPRINT CHR\$(27); CHR\$(79)

Purpose: Cancels the skip perforation function.

PRINTING AT 132 CHARACTERS PER LINE

The internal printer of the Exec. Partner normally prints at 80 characters per line. However, by using the command:

MODE LPT1: or **2:[n]** (where n is 80 or 132 characters per line)

and entering 132 for n, the printer will print at 132 characters per line.

STEP 1 Insert your System Disk into Drive A.

SCREEN DISPLAYS:

A>

STEP 2 YOU TYPE:

MODE LPT1:132

STEP 3 Press <ENTER>

Now, the printer will print at 132 characters per line. Here is a sample printout at 132 characters per line:

Example:

```
132 characters per line.132 characters per line.132 characters per line.132 characters per line.
er line.132 characters per line.132 characters per line.132 characters per line.132 characters
132 characters per line.132 characters per line.132 characters per line.132 characters per line.
er line.132 characters per line.132 characters per line.132 characters per line.132 characters
```

NOTE: If the Density selector switch is set to the Position 3 (bold character), Exec. Partner does not print at 132 characters per line.

PRINTER CHARACTER SETS

Printer Character Set 1

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL									HT	LF		FF	CR	SO	SI
1			DC2		DC4				CAN		ESC					
2	SP	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6	'	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z		:	}	~	DEL

PRINTER

Printer Character Set 2

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
8	NUL									HT	LF		FF	CR	SO	SI
9			DC2		DC4				CAN		ESC					
A	á	í	ó	ú	ñ	Ñ	ä	ö	¿	¬	½	¼	i	«	»	
B	■	■	■													
C	L	⊥	T	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥
D	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	⊥	■	■	■	■	■
E	α	β	Γ	Π	Σ	σ	μ	τ	ϕ	Θ	Ω	δ	∞	∅	∈	∩
F	≡	±	≥	≤	∩	J	÷	≈	○	■	-	√	n	2	■	SP

PRINTER

Printer Character Set 2

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL									HT	LF		FF	CR	SO	SI
1			DC2		DC4				CAN		ESC					
2	SP	!	"	#	\$	%	&	'	()	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[\]	^	_
6	'	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL

PRINTER

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
8	Ç	ü	é	â	ä	à	á	ç	ê	ë	è	ï	î	ï	Ä	Â
9	É	æ	Æ	ô	ö	ò	û	ù	ÿ	ö	ü	ç	£	¥	₪	ƒ
A	á	í	ó	ú	ñ	Ñ	à	o	ı	¬	¬	½	¼	ı	«	»
B																
C	L	⊥	T	┆	┆	┆	┆	┆	┆	┆	┆	┆	┆	┆	┆	┆
D	┆	┆	┆	┆	┆	┆	┆	┆	┆	┆	┆	┆	┆	┆	┆	┆
E	α	β	Γ	Π	Σ	σ	μ	τ	ϕ	Θ	Ω	δ	∞	∅	∈	∩
F	≡	±	≥	≤	∫	∫	÷	≈	○	■	-	√	ⁿ	²	■	SP

PRINTER

PRINTER

PRINTER

CHAPTER 5

HARD DISK (FT-80 only)

- 1. HARD DISK 5-2
 - Hard Disk Setup Operation 5-2
 - Formatting 5-7
 - FDISK Command Function 5-10
 - Making the DOS Partition (Option 1) 5-12
 - Changing the Partition Status (Option 2) 5-15
 - Removing the DOS Partition (Option 3) 5-17
 - Displaying Partition Information (Option 4) 5-19

HARD DISK (FT-80 only)

Hard Disk Setup Operation

This chapter describes how to prepare a hard disk for use with DOS on your Computer. If your system has a hard disk (FT-80), you must take several steps before the operating system can read from or write to the disk successfully. If you attempt to use the hard disk without properly setting it up, this error message will be displayed:

Invalid drive specification

One of the features of a hard disk is that it can be partitioned—divided into separate storage areas. The disk can have up to four partitions of varying sizes. Each partition can be used by a different operating system, the FDISK command provides this feature.

Follow the steps in this chapter, using the FDISK command, to set up the DOS partition. If you plan to allocate parts of the hard disk to other operating systems, assign only a specific amount of disk space to DOS—see “FDISK Command Function” section. If DOS is the only operating system to be used, assign all the available disk space to DOS by taking the steps explained in “To set up the hard disk for DOS.”

HARD DISK

Perhaps you have not yet decided if you will use other operating systems with the hard disk. In that case, start by assigning all the disk space to DOS (see “To set up the hard disk for DOS”). In the future, if you determine to use additional operating systems, use the BACKUP Command to back up files in the DOS partition; then change the DOS partition, and use RESTORE to restore the files from floppy disks.

When you have completed the steps outlined in “FDISK Command Function” or “To set up the hard disk for DOS,” you will want to format the DOS partition as described in “Formatting.”

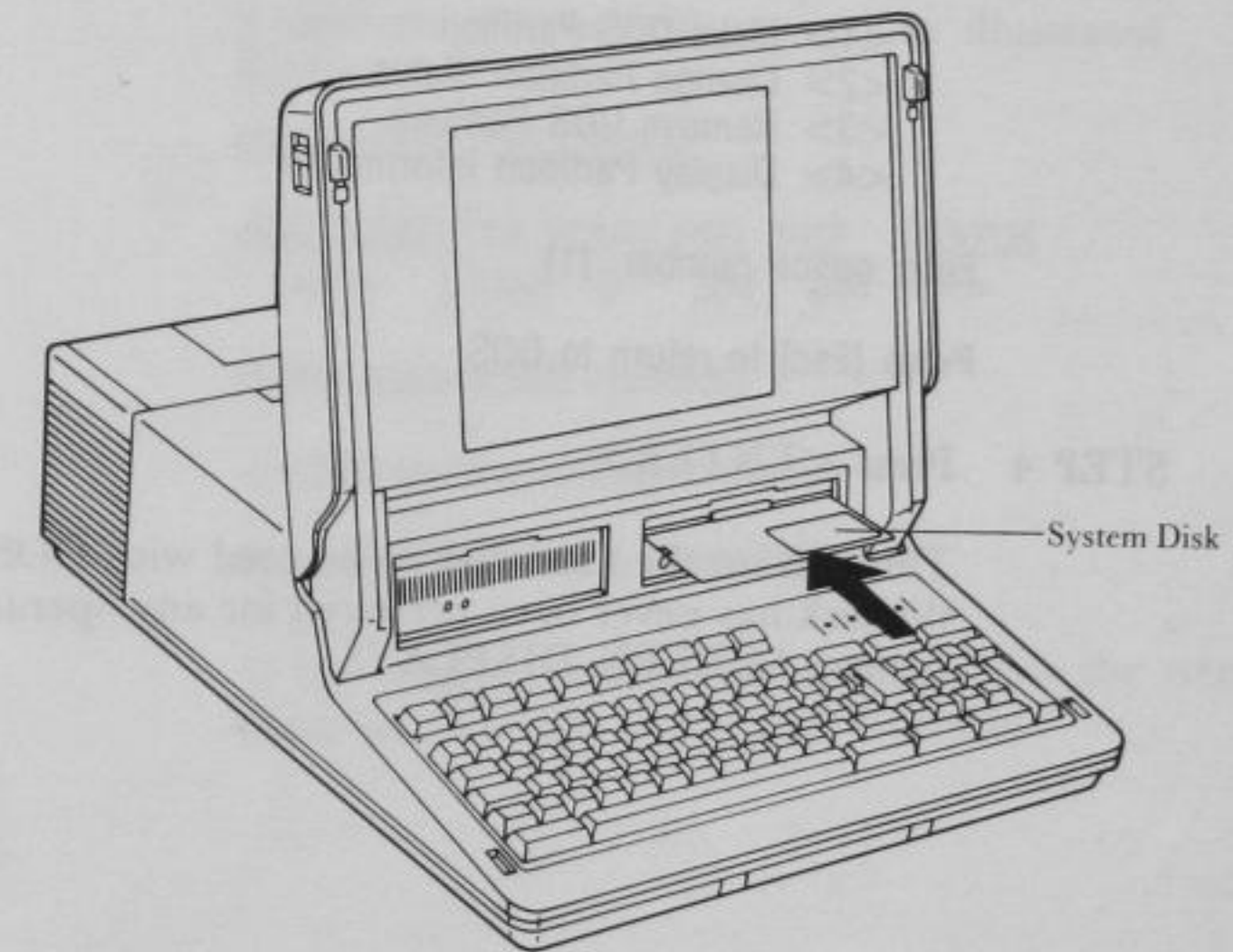
The menu-driven FDISK Command has been developed to simplify the process of setting up a hard disk. Each time the program requests you to enter information, it displays a default response. If this is the response you wish, simply press <ENTER>. To enter another choice, type it in and then press <ENTER>.

Follow the steps outlined in this section if DOS is the only operating system you will use with your hard disk. The entire hard disk will be allocated to DOS.

If you plan to use part of the disk space for some other operating system, refer to another section of this chapter, “FDISK Command Function”.

To set up the hard disk for DOS: (if DOS is the only operating system)

STEP 1 Insert your System Disk into Drive A and boot the system.



HARD DISK

Fig. 5-1. BOOTING THE SYSTEM

STEP 2 SCREEN DISPLAYS:

A>

YOU TYPE:

FDISK

STEP 3. Now press <ENTER>

SCREEN DISPLAYS:

Hard Disk Preparation Program V 1.0

(C) Copyright Matsushita Electric
Industrial Co., Ltd., 1984

((OPTIONS MENU))

Choose one of the following:

- <1> Make DOS Partition
- <2> Change Partition Status
- <3> Remove DOS Partition
- <4> Display Partition Information

Enter option number: [1]

Press [Esc] to return to DOS

STEP 4 Press <ENTER>

This sets up the hard disk to be used with DOS. If the disk has never been prepared for any operating system, SCREEN DISPLAYS:

HARD DISK

((MAKE DOS PARTITION))

PARTITION TYPE START END SIZE STATUS

No partitions

(Entire space is 305 cylinders)

Do you want to use the entire space
for DOS <Y/N>?:[Y]

Press [Esc] to return to FDISK MENU

NOTE: If partitions already exist on the disk,
a different screen will appear, as illustrated
below.

((MAKE DOS PARTITION))

PARTITION TYPE START END SIZE STATUS
<1> [DOS] 0 304 305 <A>

(Entire space is 305 cylinders)

<<DOS partition already exists>>

Press [Esc] to return to FDISK MENU

If the system displays the above, follow the steps
listed in "FDISK Command Function."

HARD DISK

STEP 5 Press <ENTER>.

This means you want the default response—Y for yes. You already decided to use only DOS with this entire hard disk. The FDISK command will now assign the entire disk to DOS.

SCREEN DISPLAYS:

System must be restarted

Insert System Disk in drive A:
Press any key when ready

Before you use the DOS partition, it must first be formatted. See "Formatting" in this chapter.

NOTE: If, at Step 5, you typed <N> and pressed <ENTER>, you assigned the starting and ending cylinder numbers for a partition.

STEP 6 Insert system disk in drive A, and press any key when ready.

Formatting

The DOS hard disk partition must be formatted before you can use it. Follow the instructions in this section only if you have created a DOS partition, but have not yet formatted it and stored data on it. If you perform a format operation on an already formatted disk partition, any data residing in the partition will be destroyed.

STEP 1 Be sure you have inserted your system Disk into drive A, and have created a DOS partition.

STEP 2 SCREEN DISPLAYS:

A>

If you want to label the hard disk,

YOU TYPE:

FORMAT C:/S/V

If you don't want to label the hard disk,

YOU TYPE:

FORMAT C:/S

and press <ENTER>.

NOTE: You can not add a volume label later.

STEP 3 SCREEN DISPLAYS:

Press any key to begin formatting C:

Press any key.

STEP 4 The red, In Use Indicator on your hard drive will light up and **SCREEN DISPLAYS:**

Formatting...

You may wait up to several minutes while DOS checks the data in all areas of the DOS partition. Then, when the operation is complete, **SCREEN DISPLAYS:**

Format complete

SCREEN DISPLAYS:

system transferred

This means that a copy of the operating system files is on the hard disk.

If you typed `FORMAT C:/S/V` in your `FORMAT` command, **SCREEN DISPLAYS:**

Volume label (11 characters, ENTER for none)?

Enter a volume label of up to 11 characters. This label identifies the hard disk when `DIR` and `VOL` are executed to display information.

SCREEN DISPLAYS:

A>

The steps above now enable you to start up the system from the hard disk.

However, to run the DOS commands, you must copy all the programs on the DOS disk to the hard disk. Once the programs are on the hard disk, all DOS commands can be run from the hard disk.

STEP 5 Keep your DOS disk in drive A and enter

`COPY *.* C:`

Wait until you see the screen display A>.

Now remove the DOS disk from drive A, leaving the disk drive door open. Press the `<Ctrl>`, `<Alt>`, and `` keys at the same time to reset the system. Now DOS will start from the hard disk. The system will prompt you for date and time.

Once you enter the date and time, the DOS prompt will change. Rather than A>, the system will display C>. DOS makes the drive it was started from the default drive.

NOTE: Drive A must be empty when you reset the system. This means the disk must be removed or the disk drive door left open. The computer will try first to load the operating system from drive A. Only if no disk is available in drive A, will the computer go to the hard disk to load the operating system.

FDISK Command Function

As was mentioned previously, a hard disk can be divided into up to four partitions. You can specify which partition the operating system will access when you start the computer. Note, however, that an operating system can access only one partition. Data cannot be transferred directly from one partition to another.

The FDISK command allows you to set up or delete only the DOS partition. A partition for another operating system must be created or erased by that operating system.

A DOS partition of any size can be set up in whatever location you choose. You may also choose to delete the DOS partition later on if, for instance, you want to change its size or location on the disk. The FDISK command provides these function:

- <1> Make DOS Partition
- <2> Change Partition Status
- <3> Remove DOS Partition
- <4> Display Partition Information

To access these functions, which are described in the following pages, start the FDISK command this way:

STEP 1 YOU TYPE:

FDISK

and press <ENTER>

SCREEN DISPLAYS:

Hard Disk Preparation Program V 1.0

(C) Copyright Matsushita Electric
Industrial Co., Ltd., 1984

((OPTIONS MENU))

Choose one of the following:

- <1> Make DOS Partition
- <2> Change Partition Status
- <3> Remove DOS Partition
- <4> Display Partition Information

Enter option number: [1]

Press [Esc] to return to DOS

Type the number corresponding to your chosen option and press <ENTER>. The default is Option 1. It will be chosen automatically if you simply press <ENTER> without typing a number. Now go on to the section that explains the option you chose.

Making the DOS Partition (Option 1)

Use this option to make a DOS partition. You will have to decide how big a partition you want and where it should be on the hard disk.

A hard disk is made up of cylinders, whose number and size varies. A 10-megabyte disk contains 305 cylinders, with each cylinder holding 34,816 bytes (characters). If you had a 10-megabyte disk and you wanted to allocate the entire disk to DOS, you would specify a size of 305 cylinders. The starting cylinder number would be 0. If a partition had already been assigned to DOS, an error message would result. The screen that appears will depend on whether the disk already has partitions. If it has one or more partitions already, go on to STEP 2. If the disk has not been initialized, the following screen is displayed.

((MAKE DOS PARTITION))

PARTITION TYPE START END SIZE STATUS

No partitions

(Entire space is 305 cylinders)

Do you want to use the entire space
for DOS <Y/N>?: [Y]

Press [Esc] to return to FDISK MENU

HARD DISK

If you will use DOS only with this hard disk, follow the steps in "To set up the hard disk for DOS" (earlier in the chapter). If you are to use other operating systems also, follow these steps:

STEP 1 YOU TYPE:

n

and press <ENTER>.

SCREEN DISPLAYS:

305 cylinders are available
from cylinder 0

Enter partition size: [305]

The default entry equals the largest available
space on the disk.

Now go on to STEP 3.

STEP 2 If your hard disk was previously initialized, a
screen like this will appear:

((MAKE DOS PARTITION))

PARTITION TYPE START END SIZE STATUS
<1> [DOS] 0 304 305 <A>

(Entire space is 305 cylinders)

<<DOS partition already exists>>

Press [Esc] to return to FDISK MENU

HARD DISK

- STEP 3 If you want a DOS partition to take up all the available space, just press <ENTER>. If you want a specific number of cylinders, type in the number and press <ENTER>.

To use part of the cylinder space for the DOS, the size of partition must be a minimum of 12 cylinders so that the entire contents of the country DOS disk can be transferred.

- STEP 4 SCREEN DISPLAYS:

Enter starting cylinder number..... :[0]

The default for the starting cylinder number will depend on that specified partition size. It will be the first cylinder of the largest space on the disk big enough to accept the size partition you want. If that is where you want the DOS partition to be, press <ENTER>. Or type in the starting cylinder number of your choice and press <ENTER>.

- STEP 5 SCREEN DISPLAYS:

Press [Esc] to return to FDISK MENU

The cursor is at the bottom of the screen, and the lines on the screen have changed to show a new partition. You have created the DOS partition.

Be sure the System Disk is in drive A.
Now press <ESC> three times to reset the system.

Even though a DOS partition has been assigned on your disk, you will not be able to use it until you take the steps explained in "Formatting," in this chapter.

To make the partition just created active (usable), follow the steps in "Changing the Partition Status (Option 2)," next.

Changing the Partition Status (Option 2)

Choose this option if you want to make a partition active when the system is started from the hard disk.

- STEP 1 SCREEN DISPLAYS:

((CHANGE PARTITION STATUS))

PARTITION	TYPE	START	END	SIZE	STATUS
<1>	[DOS]	0	99	100	N

(Entire space is 305 cylinders)

Enter partition number you want
to activate..... :[]

Press [Esc] to return to FDISK MENU

You enter the number of the partition whose operating system you want to start from and press <ENTER>.

For example, select partition number "1", and press <ENTER>.

Now go on to STEP 3.

- STEP 2 If the partition was already activated, a screen like this will appear.

((CHANGE PARTITION STATUS))

PARTITION	TYPE	START	END	SIZE	STATUS
<1>	[DOS]	0	99	100	<A>

(Entire space is 305 cylinders)

<<Partition 1 is already active>>

Press [Esc] to return to FDISK MENU

STEP 3 SCREEN DISPLAYS:

((CHANGE PARTITION STATUS))

PARTITION	TYPE	START	END	SIZE	STATUS
<1>	[DOS]	0	99	100	<A>

(Entire space is 305 cylinders)

<<Partition 1 has been activated>>

Press [Esc] to return to FDISK MENU

The screen will change to show the new active partition. Press the <ESC> key to return to FDISK menu. Press <ESC> again to reset the system.

To start the operating system in the partition just activated, do this:

Open the door to disk drive A.

Press and hold the <Ctrl> and <Alt> keys. Then press .

HARD DISK

Removing the DOS Partition (Option 3)

... WARNING ...

The following procedure destroys all data in the DOS partition. Be sure to back up all DOS files before proceeding.

STEP 1 SCREEN DISPLAY:

((REMOVE DOS PARTITION))

PARTITION	TYPE	START	END	SIZE	STATUS
<1>	[DOS]	0	304	305	<A>

(Entire space is 305 cylinders)

Data in DOS partition will be deleted. Continue <Y/N>? ...[N]

Press [Esc] to return to FDISK MENU

HARD DISK

STEP 2 YOU TYPE:

Y

Press <ENTER>. If you have not yet backed up all your files, cancel the operation by pressing either <ENTER> or <ESC>. This returns you to the FDISK options menu. If you entered Y,

SCREEN DISPLAYS:

No partitions

(Entire space is 305 cylinders)

<<DOS partition has been removed>>

Press [Esc] to return FDISK MENU

The DOS partition has been removed and the information on one screen updated. To go on, you will have to boot another system from the hard disk or restart DOS from a floppy disk.

After changing the size of the partition, re-format the partition. Refer to page 5-7.

HARD DISK

Displaying Partition Information (Option 4)

Use this option to get information about the status of your hard disk. The screen will appear similar to this:

((DISPLAY PARTITION INFORMATION))

PARTITION	TYPE	START	END	SIZE	STATUS
<1>	[DOS]	0	304	305	<A>

(Entire space is 305 cylinders)

Press [Esc] to return to FDISK MENU

The screen displays a line of data for each assigned partition. The column labeled Partition shows the relative number of that partition (whether it appears first, second, third, or fourth on the disk).

The Type column records whether each partition is DOS or non-DOS. The starting and ending cylinder numbers for each partition are shown in the Start and End columns. The Size column shows the number of cylinders taken up by each partition.

The Status column shows from which partition the system will start when the computer is started. That partition's status is shown as A for active. Others are N for non-active.

The bottom line shows the total number of cylinders on this hard disk.

HARD DISK

MEMO

CHAPTER 6

USING DOS

- 1. INTRODUCTION 6-2
- 2. THE KEYBOARD 6-2
- 3. FILES 6-6
 - Naming Files 6-6
 - Reserved Device Names 6-6
 - Wildcard Characters 6-7
- 4. DIRECTORIES 6-8
 - Directory Structure 6-8
 - Current Directory 6-10
 - Paths to Files 6-12
 - Setting a Path 6-13
 - Table of Directory Commands 6-13
 - Directory Commands 6-14

INTRODUCTION

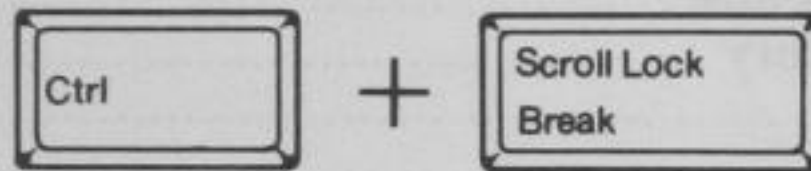
DOS is the disk operating system provided for you to use with your computer. DOS is a group of programs that allow you to communicate with your computer. DOS programs provide a way to organize and use the information on your disks.

THE KEYBOARD

Your computer keyboard has some special keys to use with DOS. These keys enable you to perform DOS functions by just pressing keys on the keyboard.

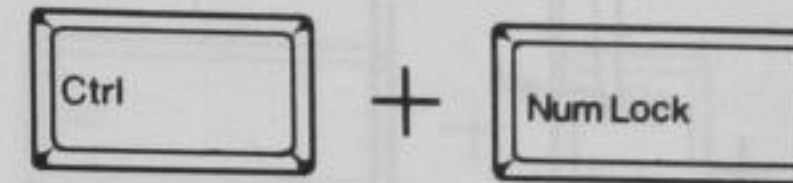
When more than one key is needed for a particular function hold down the first key and then press the second key.

Cancel the current operation.



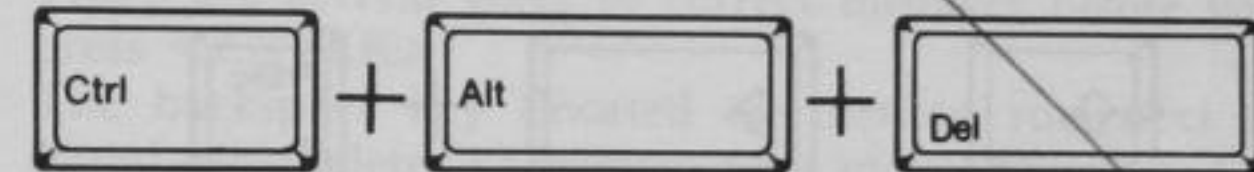
Stops the execution of the current command. DOS displays the prompt and you may enter another command.

Suspend the screen display.



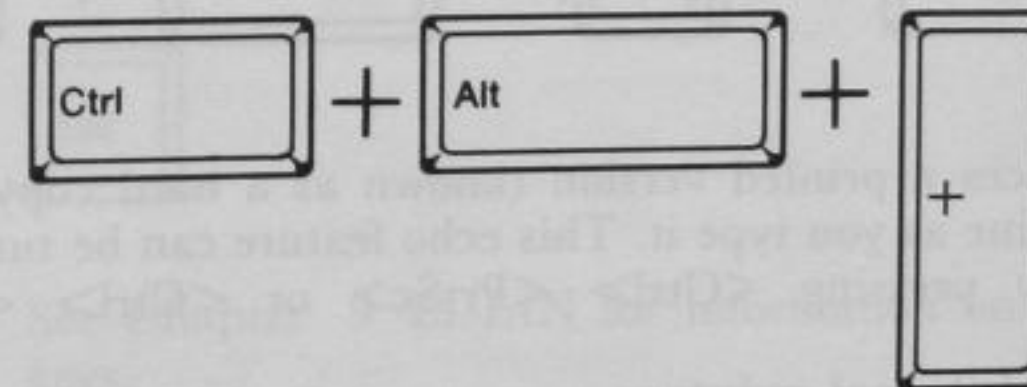
The information displayed on the screen is temporarily halted. When you are ready to continue receiving information, press any key.

System Restart (Warm Boot)



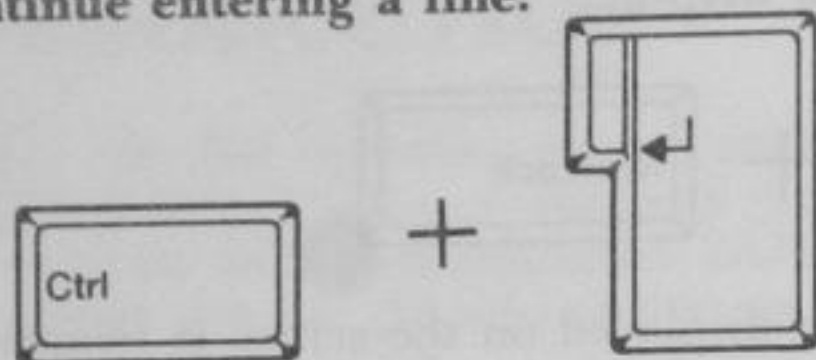
Be sure you have the DOS disk in drive A. Press and hold down the <Ctrl> and <Alt> keys and then press the key. This function will restart (known as reboot or warm boot) DOS.

Switch CPU clock frequency.



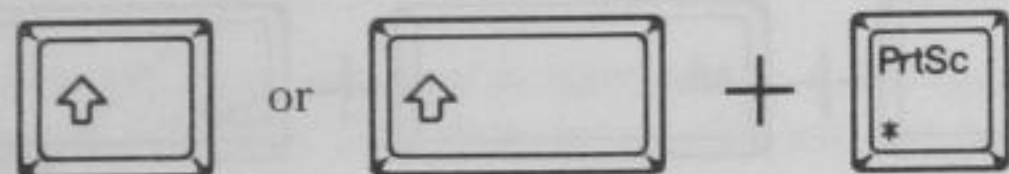
Simultaneously pressing these keys on the keyboard will toggle the speed between standard (4.77 MHz) and fast (7.16 MHz). The LED on the keyboard console will indicate the current speed.

Continue entering a line.



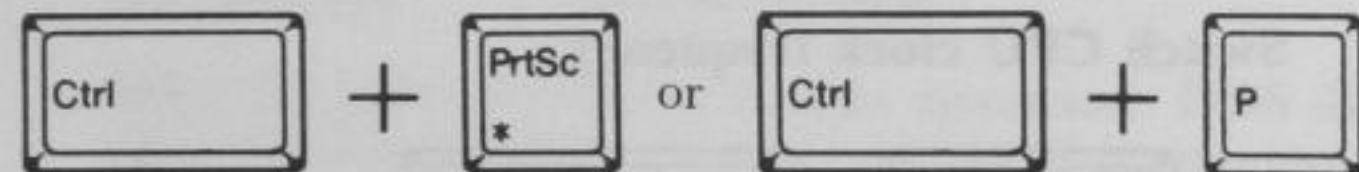
Moves the cursor to the next displayed line on the screen to continue entering the line you are typing.

Print what is on the screen.



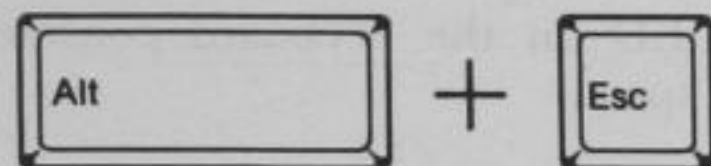
Produces a printed version (known as a hard copy) of what is on the screen (screen dump). If you wish to print graphics, GRAPHICS.COM must be loaded first.

Print what you are typing.



Produces a printed version (known as a hard copy) of each line as you type it. This echo feature can be turned off by pressing <Ctrl> <PrtSc> or <Ctrl> <P> again.

Reset the internal printer



If the printer jams or fails, or when you have changed the linefeed dip switch on the internal printer (It is located under a paper roll.), press <Alt> <Esc>.

Switch printer.

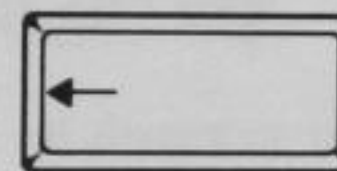


This key sequence will cause output to be routed from the internal (external) printer to the external (internal) printer. The LED on the keyboard console will indicate the active printer. The active printers accessed as device LPT1:.

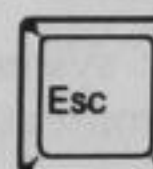
Correct mistakes.

There are several ways to correct mistakes before you press <ENTER>.

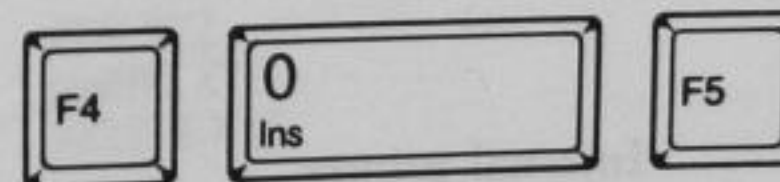
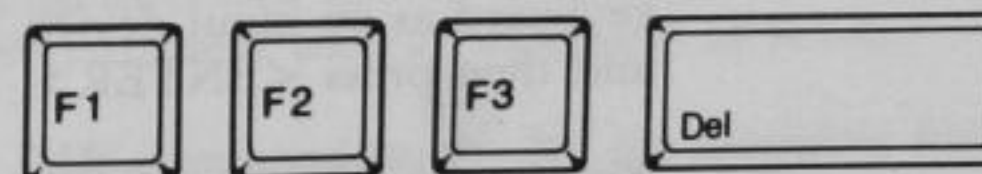
The backspace key (located on the top row next to NumLock) deletes characters as it moves the cursor to the left.



The Esc key cancels the entire line. A backslash (\) is displayed on the screen and the cursor moves one line down. You may then type the command again.



See Chapter 9 EDLIN for information on the editing keys.



FILES

Naming Files

A filename can have a maximum of eight characters plus an optional extension. The extension consists of a period (.) followed by a maximum of three characters. The extension immediately follows the filename.

For example:

MYFILE.TST

If your filename has an extension always include it.

The valid characters in file names and extensions are:

A-Z	\$	&	#	@	-
0-9	!	%	()	_
{	}	,	'		

Any other characters are invalid and the file name will be shortened (truncated).

Reserved Device Names

Reserved device names are used only for specific system input/output devices and cannot be used in any other context.

Reserved Name:	Device:
CON:	Keyboard input and screen output. To end CON: when using the keyboard as an input device, press F6 and then press <ENTER>.
AUX: or COM1:	Built-in RS232C (Serial) interface.
PRN: or LPT1:	Internal printer

LPT2: External printer connected to the parallel port.

NUL: Dummy device for testing various applications. You can use NUL: when you do not want to create a file, but the syntax of the command requires an input or output file name.

Wildcard Characters

The characters * and ? may be used with filenames and their extensions to provide enhanced flexibility when issuing DOS commands.

The ? indicates that any valid character may occupy that position. For example:

DIR AB?.XYZ

Lists all directory entries beginning with AB, with any character in the next position, with an extension of XYZ.

The DIR command might list the following files:

ABC.XYZ
AB1.XYZ
AB2.XYZ

The * indicates that any valid character may occupy that position and all remaining positions. For example:

DIR AB*.XYZ

Lists all directory entries beginning with AB and having any other characters with an extension of XYZ.

The DIR command might list the following files:

ABC.XYZ
ABCDE.XYZ
AB1234.XYZ

DIRECTORIES

Directory Structure

Directories provide a way to organize the files on each disk. The directory contains:

- Names of the files.
- Sizes of the files.
- Dates the files were created.
- Dates the files were updated.
- Locations of the files on the disk.

You can organize your files into convenient categories by setting up separate directories.

Any one directory may contain entries for files and entries for other directories (called sub-directories). This method of file organization looks like an inverted tree and is called a hierarchical directory structure. The levels are defined as follows:

ROOT

The first level in the directory structure. This directory is automatically created when you format a disk and put files in it. It can contain 112 entries (double sided disk), either file names or sub-directory names.

SUB-DIRECTORIES

A subsequent level in the directory structure. Sub-directories themselves can include the names of additional sub-directories. The sub-directories are actually files and are therefore not restricted in size, but are limited only by the available space on the disk. The sub-directory names are in the same format as file names. Files or directories with the same name can exist as long as each is in a different directory.

It is possible to "travel" around your directory structure. For example, it is possible to find any file in the system by starting at the root and traveling down through the sub-directories. You may also start where you are within the file system and travel up towards the root.

Figure 6-1 illustrates a typical hierarchical directory structure.

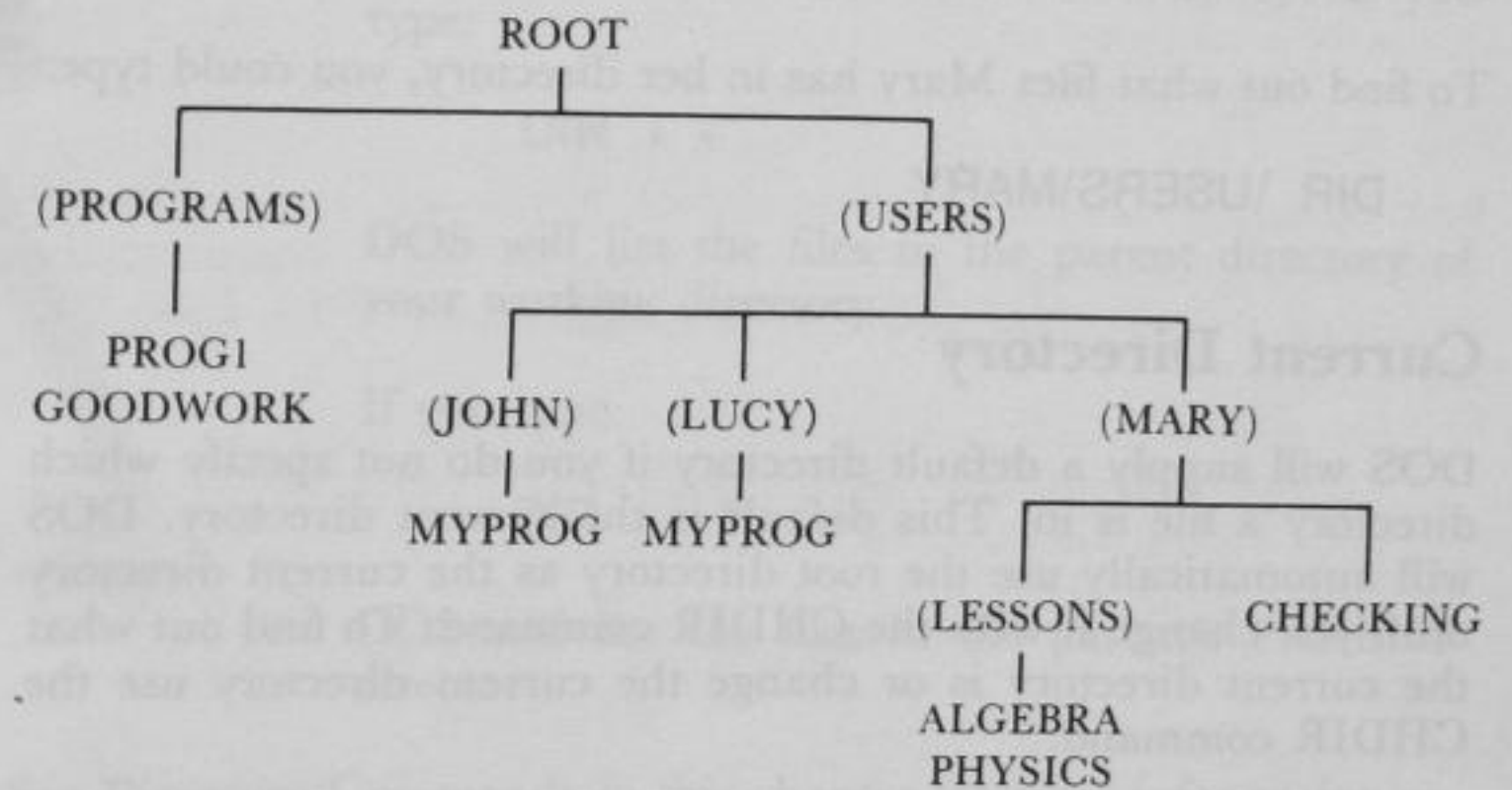


Fig. 6-1. A SAMPLE HIERARCHICAL DIRECTORY STRUCTURE
(Sub-directory names are in parenthesis)

In this example, two sub-directories of ROOT have been created. These include:

A directory of programs, named PROGRAMS.

A USERS directory containing separate sub-directories for the users of the system.

John, Lucy and Mary each have their own directories which are sub-directories of the USERS directory. John and Lucy have files in their directories, each named MYPROG. Notice that John's file is unrelated to Lucy's. Mary has a file named CHECKING as well as a sub-directory named LESSONS.

You can get a list of the files in Mary's LESSONS directory by typing:

```
DIR \USERS\MARY\LESSONS
```

Note that the backward slash mark (\) is used to separate directories from other directories and files.

To find out what files Mary has in her directory, you could type:

```
DIR \USERS\MARY
```

Current Directory

DOS will supply a default directory if you do not specify which directory a file is in. This default is the current directory. DOS will automatically use the root directory as the current directory until you change it with the CHDIR command. To find out what the current directory is or change the current directory use the CHDIR command.

For example, if your current directory is \USER\LUCY, when you type:

```
CHDIR<ENTER>
```

you will see:

```
A:\USER\LUCY
```

This is your current drive designation plus the current directory.

Each directory contains two special entries. They are special shorthand notations for the current directory and the parent directory (one level up) of the current directory.

- Notation to indicate the name of the current directory in all hierarchical directory listings. DOS automatically creates this entry when a directory is created.
- • The current directory's parent directory. If you type:

```
DIR • •
```

DOS will list the files in the parent directory of your working directory.

If you type:

```
DIR • • \ • •
```

DOS will list the files in the parent's PARENT directory.

See Directory Commands in this chapter for more information on CHDIR.

Paths to Files

To tell DOS where the files are located in the directory structure use a pathname to the file.

To find or create a file DOS requires the following information:

- The drive.
- The name of the file.
- The name of the directory.

DOS searches the current directory automatically. If the file is not in the current directory you must tell DOS the path leading to the correct directory.

A pathname (denoted by *path*) is:

A sequence of directory names separated by backslashes (\) and a filename.

The syntax of *path* is:

[\] [*directory* \]... [*directory* \]

If a pathname begins with a backslash, DOS searches for the file beginning at the root directory. Otherwise, DOS begins at the user's current directory, and searches downward from there. For example.

If your current directory is MARY, and you want to find the file PHYSICS you can use either:

\USERS\MARY\LESSONS\PHYSICS

or

LESSONS\PHYSICS

In the first case the full path from the root was specified by the leading backslash. In the second case the path from the current directory was given.

Setting a Path

The PATH command allows you to specify additional paths for DOS to search if it does not find a command in the current directory.

For example:

If you are in a working directory named \COM\PROG, and all DOS external commands are in \COM, you must tell DOS to choose the \COM path to find the DISKCOPY command. The command

PATH \COM

tells DOS to search in your working directory and the \COM directory for all commands. You only specify this path once to DOS during your terminal session. DOS will now search in \COM for the external commands.

To check what the current path is, type the word PATH and the current value of PATH will be displayed.

For more information on the DOS command PATH, refer to Chapter 7 "Command Reference".

Table of Directory Commands

COMMAND	PURPOSE	SYNTAX
CHDIR	Displays working directory; changes directories	CHDIR [[<i>d:</i>] <i>path</i>]
MKDIR	Makes a new directory	MKDIR [<i>d:</i>] <i>path</i>
RMDIR	Removes a directory	RMDIR [<i>d:</i>] <i>path</i>
TREE	Displays directory structure	TREE [<i>d:</i>] [/F]

Directory Commands

Creating a Directory

To create a sub-directory in your current directory, use the MKDIR (Make Directory) command.

For example, to create a new directory named MOREDIR under your current directory, simply type:

```
MKDIR MOREDIR
```

After this command has been executed, a new directory will exist in your tree structure under your current directory. To make directories anywhere in the tree structure specify MKDIR and then a pathname. DOS automatically creates the . and .. entries in the new directory.

Deleting a Directory

To delete a directory in the tree structure, use the RMDIR (Remove Directory) command.

For example, to remove the directory MOREDIR from the current directory, type:

```
RMDIR MOREDIR
```

The directory MOREDIR must be empty except for the . and .. entries before it can be removed. This prevents you from accidentally deleting files and directories. To remove the \USERS\JOHN directory make sure that it has only the . and .. entries, then type:

```
RMDIR \USERS\JOHN
```

To remove all the files in a directory (except for the . and .. entries), type DEL and then the pathname of the directory. For example, to delete all files in the \USERS\JOHN directory, type:

```
DEL \USERS\JOHN
```

You cannot delete the . and .. entries. They are created by DOS as part of the hierarchical directory structure.

Setting the Current Directory

To change from your current directory to another directory use the CHDIR (Change Directory) command and supply a pathname. For example:

```
CHDIR \USERS
```

changes the current directory from \USERS\JOHN to \USERS. The command CHDIR . . will always put you in the parent directory of your working directory.

Displaying the Directory Tree

To see a report describing the directory structure of a disk, use the command TREE. The report includes all the directory paths and, if the /F parameter is used, the names of all files in each sub-directory. For example:

```
TREE B:/F
```

shows all the directory paths on drive B and the names of the files in each sub-directory.

CHAPTER 7

COMMAND REFERENCE

1. INTRODUCTION	7-3
2. TYPES OF DOS COMMANDS	7-4
3. SYNTAX NOTATION	7-5
4. PARAMETERS	7-6
Reserved Device Names	7-7
Wildcard Characters	7-8
5. TABLE OF COMMANDS	7-10
6. COMMAND DESCRIPTIONS	7-14
ASSIGN	7-16
BACKUP	7-18
BREAK	7-20
CHDIR	7-21
CHKDSK	7-22
CLS	7-24
COPY	7-25
CTTY	7-33
DATE	7-34
DEL	7-35
DIR	7-36
DISKCOMP	7-40
DISKCOPY	7-43
ERASE	7-46
EXE2BIN	7-48
EXTPRT	7-51
FAST	7-52
FC	7-53
FDISK	7-57
FIND	7-58
FORMAT	7-59

GRAPHICS	7-63
MKDIR	7-64
MODE	7-66
MORE	7-72
PATH	7-73
PRINT	7-75
PROMPT	7-79
RECOVER	7-82
RENAME	7-84
RESTORE	7-85
RMDIR	7-87
SET	7-88
SORT	7-91
SYS	7-92
TIME	7-93
TREE	7-95
TYPE	7-96
VER	7-97
VERIFY	7-98
VOL	7-99

INTRODUCTION

This chapter explains the ways to use the Disk Operating System (DOS) through DOS commands. These commands provide file handling capabilities and program execution, among other things.

The DOS commands:

Manipulate files—compare, copy, display, erase, and re-name.

Manipulate disks—format, compare, and copy.

Run programs—execute DOS system programs (i.e. EDLIN or DEBUG) as well as your programs.

List directories.

Enter date, time or remarks.

Set mode of a printer, screen formats and the serial interface.

Route printer output—direct printer output to the serial interface.

Convert EXE files to COM type files.

Transfer DOS to another disk.

Request the system to wait.

TYPES OF DOS COMMANDS

Internal Commands:

These commands are built into the command processor. They execute immediately.

External Commands:

These commands are not built into the command processor but reside on disks as program files. In order to execute them they must be read from disk. The disk must be in the drive in order for DOS to read it.

Your Own Commands:

Any filename with the extension .COM or .EXE is considered to be an external command. Therefore, you can create your own commands and add them to the system by using these extensions. Programs such as JUMP.COM or HOP.EXE are treated as external commands. When entering an external command, do not include the filename extension.

SYNTAX NOTATION

The following notations show how the DOS commands should be formatted:

Any words shown in CAPITAL letters are keywords and must be entered exactly as illustrated. DOS will automatically convert all keywords to uppercase. Thus, you may enter the keywords in any combination of upper and lower cases.

Commas (,), equal signs (=), question marks (?), slashes (/), or colons (:) must be entered exactly where shown. The only punctuation not needed are the square brackets ([]).

Any items shown in lowercase *italic* letters indicate where to supply variable information. When you see *filename*, replace *filename* with the name of your file.

An elipsis (...) indicates the repetition of an item as many times as needed.

Square brackets ([]) indicate optional information. To include the optional information, type in just the information inside the brackets—NOT the brackets themselves.

PARAMETERS

DOS COMMAND

Parameters are variable or constant items used to specify limits in DOS command statements. There are both optional and required parameters. In some cases DOS will provide a default value for a parameter not included. (A discussion of the default parameters appears in the detailed descriptions of each DOS command). The following four parameters, whether entered or defaulted, are necessary for every DOS command:

d: Drive. This letter indicates when you should specify a drive. A letter followed by a colon tells DOS which disk drive to use. If you omit this parameter, the default drive is assumed. For example:

- A: represents the A drive on the system.
- B: represents the B drive.
- C: represents the C drive (hard or RAM disk).

filename A 1-8 character filename, composed of valid characters. The valid characters are:

A-Z, 0-9,
\$ & # @ - ! % () _ { } ` ' `

Any other character is considered to be a delimiter and will truncate the filename.

.ext Filename extension is optional and is made up of a period and 1-3 characters. As with filenames, only valid characters are allowed and filename extensions must immediately follow the filename.

filespec [d:] filename [.ext]

The wildcard characters (* and ?) are valid in a filename or extension.

path [\] [directory \]...[directory\
See page 6-12.

Reserved Device Names

DOS reserves the following names as system devices:

Reserved Name	Device
CON:	Console, keyboard or screen. When CON: is used as an input device, you may use the F6 function key and then press <ENTER> to generate an end-of-file marker which terminates CON: as an input device.
AUX: or COM1:	Built-in serial (RS232C) port.
NUL:	Nonexistent or dummy device. This is used when testing. When NUL: is used as an input device an immediate end-of-file mark is generated. When NUL: is used as an output device, the write operations are simulated but no actual data is written.
LPT1: or PRN:	Internal printer, used as an output device only.
LPT 2:	External printer, used as an output device only.

NOTE: The reserved device names can be used as a filename to direct output. If a filename extension or drive specifier is entered with these device names, it will be ignored. The colon at the end of the reserved device name is optional.

DOS COMMAND

Wildcard Characters

There are two special characters which allow for greater flexibility with DOS commands. The "?" and the "*" allow for special replacement characters or default characters to appear in a filename or its extension.

The "?" Character:

A "?" in a filename or in a filename extension means that any character can fill that position. For example:

```
DIR AB?DE.FGH
```

This command would list all the directory entries on the default drive with the filenames that have the first two characters AB, any other character, the last two characters DE and have an extension of FGH. The position occupied by the "?" can contain any valid character. If the above command were entered, here is an example of what might be listed:

```
ABCDE.FGH
AB3DE.FGH
AB#DE.FGH
```

The "*" Character:

An "*" in a filename or in a filename extension means that any character can fill that position and all the remaining positions of either the filename (if placed in the filename) or the filename extension (if placed in the extension). If placed in the filename, the substitution stops at the period of the extension name. For example, the command:

```
DIR AB*.EFG
```

would list all directory entries on the default drive with filenames that begin with AB, followed by any other valid character or characters and have a filename extension of EFG. The filename may be anywhere from 2-8 characters long. If the above command were entered, here is an example of what might be listed:

```
ABC.EFG
ABCDE.EFG
ABXYZ123.EFG
```

Examples:

To list all the filenames on your default drive that begin with FINANCE, enter:

```
DIR FINANCE.???
or
DIR FINANCE.*
```

To list all the filenames on Drive A that end with the filename extension of DAT, enter:

```
DIR A:?????????.DAT
or
DIR A:*.DAT
```

To list all the filenames on Drive A that begin with the letters FIN and have filename extensions that begin with the letter C, enter:

```
DIR A:FIN?????.C??
or
DIR A:FIN*.C*
```


TABLE OF COMMANDS

Command	Purpose	Syntax
ASSIGN	Specify the disk drive.	ASSIGN [x=y[...]]
BACKUP	Back up files from a hard disk to floppy disk.	BACKUP <i>d:</i> [<i>path</i>] [<i>filename</i> [.ext]] <i>d:</i> [/S]/M[/A] [/D: <i>mm-dd-yy</i>]
BREAK	Check for a control break.	BREAK [ON OFF]
CHDIR	Change or display current directory.	CHDIR [[<i>d:</i>] <i>path</i>] or CD [[<i>d:</i>] <i>path</i>]
CHKDSK	Report on disk status.	CHKDSK [<i>d:</i>] [<i>filename</i>] [/F] [/V]
CLS	Clear display screen.	CLS
COPY	Copy files.	COPY [/A] [/B] [<i>d:</i>] <i>filename</i> [.ext] [/A] [/B] [<i>d:</i>] [<i>filename</i> [.ext]] [/A] [/B] [/V] or COPY [/A] [/B] [<i>d:</i>] [<i>path</i>] <i>filename</i> [.ext] [/A] [/B] [+ <i>d:</i> [[<i>path</i>] <i>filename</i> [.ext] [/A] [/B] ...] [<i>d:</i>] [<i>path</i>] [<i>filename</i> [.ext]] [/A] [/B] [/V]
CTTY	Change input and output devices.	CTTY <i>device-name</i>
DATE	Set and display date.	DATE [<i>mm-dd-yy</i>]

Command	Purpose	Syntax
DEL	See "Erase command"	_____
DIR	List of directory entries.	DIR [<i>d:</i>] [<i>path</i>] [<i>filename</i> [.ext]] [/P]/W]
DISK-COMP	Compare disk contents.	DISKCOMP [<i>d:</i>] [<i>d:</i>] [/1] [/8]
DISK-COPY	Copy disk contents.	DISKCOPY [<i>d:</i>] [<i>d:</i>] [/1]
ERASE	Delete files.	ERASE [<i>d:</i>] [<i>path</i>] [<i>filename</i> [.ext]] or DEL [<i>d:</i>] [<i>path</i>] [<i>filename</i> [.ext]]
EXE2BIN	Convert .EXE files to .COM files.	EXE2BIN [<i>d:</i>] [<i>path</i>] <i>filename</i> [.ext] [<i>d:</i>] [<i>path</i>] [<i>filename</i> [.ext]]
EXTPRT	Switches the default printer from the internal printer to the external printer.	EXTPRT
FAST	Switches the CPU clock frequency from 4.77 MHz to 7.16 MHz.	FAST
FC	Compare file contents.	FC [<i>d:</i>] [<i>path</i>] [<i>filename</i> [.ext]] [<i>d:</i>] [<i>path</i>] [<i>filename</i> [.ext]]
FDISK	Process the partition on the hard disk.	FDISK

Command	Purpose	Syntax
FIND	Output lines containing specified string.	FIND [/V] [/C] [/N] "string" [[d:] [path] filename [.ext]...]
FORMAT	Initialize the disk.	FORMAT [d:] [/S] [/V] [/B] [/1] [/8]
GRAPHICS	Print graphic screen.	GRAPHICS
MKDIR	Create sub-directory.	MKDIR [d:]path or MD [d:]path
MODE	Set mode on Display/printer	MODE LPT#:[n] [,m] [,P]] or MODE n or MODE [n],m[,T] or MODE COMn: baud[,parity[,databits [,stopbits[,P]]]] or MODE LPT#:=COMn
MORE	Send one screen of data and then pause.	MORE
PATH	Set directory path.	PATH [d:]path[; [d:]path]...
PRINT	Set up print queue.	PRINT [[d:] filename[.ext]] [/T] [/C] [/P]...
PROMPT	Set new system prompt.	PROMPT [prompt-text]

Command	Purpose	Syntax
RECOVER	Recover files from damaged disk.	RECOVER [d:] [path]filename[.ext] or RECOVER d:
RENAME	Rename a file.	REN[AME] [d:] [path]filename[.ext] filename[.ext]
RESTORE	Restore files from floppy disk to a hard disk.	RESTORE d:[d:] [path] [filename[.ext]] [/S] [/P]
RMDIR	Remove sub-directory.	RMDIR [d:]path or RD [d:]path
SET	Insert environment strings.	SET [name]=[parameter]]
SORT	Sort data.	SORT [/R] [/+n]
SYS	Transfer system files.	SYS d:
TIME	Set and display time.	TIME [hh:mm:ss.xx]
TREE	Display directory paths and list files in sub-directory.	TREE [d:][/F]
TYPE	Display file contents.	TYPE [d:] [path] filename[.ext]
VER	Display DOS version number.	VER
VERIFY	Verify successful write to disk.	VERIFY [ON OFF]
VOL	Display volume information.	VOL [d:]

COMMAND DESCRIPTIONS

DOS COMMAND

This section provides a detailed description of the syntax, purpose, type, and comments for each DOS command. Some information is common to all DOS commands. The following notations apply to all commands:

Most commands are followed by one or more parameters.

Parameters and commands may be entered in either upper or lower case (DOS will convert all characters to upper case).

Commands and parameters must be delimited by either a space, comma, semicolon, equal sign or the tab key. You need not use the same delimiter within one command. For example, you could enter:

```
COPY oldfile.txt;newfile.txt
```

A drive and filename must be separated by colon (:), and a filename and .ext must be by period (.).

A command is executed only after you press the <ENTER> key.

Drives may be referred to as source and target drives. A source drive is the drive from which you will get information and a target drive is the drive to which you will send information.

The prompt produced by the command processor is the default drive designation letter, plus >. For example: A>.

The DOS editing keys and control keys described in Chapter 6 may be used while entering commands.

Filename extensions are optional when you create or rename a file; however, if a file has a filename extension, you must use it when referring to that particular file.

To end a command while it is executing, press <Ctrl><Break>. <Ctrl><Break> is only recognized while the system is reading from the keyboard or writing to the screen; therefore, the command may not end immediately after you press <Ctrl><Break>.

Pressing the <ENTER> key begins the command. Until you press this key you may change whatever you typed onto the screen and nothing will have happened.

Device names and wildcard characters are not permitted in a command name. You may, however, use them in command parameters.

To suspend the display of output on the screen, you can press the <Ctrl><NumLock> key. To restart the display, press any other key.

DOS COMMAND

ASSIGN (HARD DISK)

External

DOS COMMAND

Syntax: ASSIGN [x=y [...]]

Purpose: Specifies a different drive from the one designated for normal disk operations.

Comments: You can use ASSIGN to instruct DOS to route requests for a disk drive to another drive.

DOS will internally convert the first drive letter, x, to the second specified drive letter, y. You need not include a colon after the drive letter.

If you enter ASSIGN without parameters, all new drive assignments will be reset back to the normal drive assignments. Now, regular disk operations can resume.

This command is meant to give you more flexibility in using applications that do not let you specify a drive. Such applications are designed to use disk drives A and B. You may want these applications to use your hard disk. By entering, for example,

```
ASSIGN A=C
```

You can make an application use hard disk drive C.

NOTE: Drives should only be reassigned when required for the situation described above. Drives should not be reassigned for use with the PRINT command or for running normal DOS operations. A reassignment of drives could prevent commands and programs that need actual, physical disk information from getting the true device type.

When developing an application, you should avoid assigning a specific drive in the program. Allow the application user to designate the drive instead.

Examples: The following command tells DOS to route all requests for Drive B to Drive C:

```
ASSIGN B=C
```

If you now execute DIR B:. DOS will display the directory on your fixed disk, Drive C.

This example resets any reassignments so that system requests for Drive B will go to Drive B, as in normal operations:

```
ASSIGN
```

DOS COMMAND

BACKUP (HARD DISK)

External

Syntax: **BACKUP** [*d:*] [*path*] [*filename* [*.ext*]] *d:* [/S] [/M] [/A] [/D: *mm-dd-yy*]

Purpose: Allows you to back up files from a hard disk to floppy disks.

Comments: Use only DOS formatted disks. The first parameter to specify is the hard disk file you want backed up. If you do not designate a path, files are backed up from the current directory. If no filename or extension is specified, all files in the directory will be backed up. The second parameter is the disk drive to which you want the files backed up.

/S will back up not only the files in the specified directory but also all the files in sub-directories at all levels beyond the specified directory.

/M specifies that only the files modified since the last backup operation should be backed up. Using this parameter keeps you from repeatedly backing up files that will never change.

/A specifies that the backed up files will be added to files on a backup disk that is already in the designated drive. Omitting this parameter causes the system to prompt you to insert a disk when the backup program is loaded.

/D is used to back up only those files written on or after the date specified. Valid date formats are covered under the DATE command.

When you enter a BACKUP command, you will be prompted to insert a disk unless the /A parameter was designated. If you did not specify /A, BACKUP will erase existing files on the disk before backing up the hard disk file. Each time a disk is filled up, BACKUP prompts for a new disk.

It is a good practice to label each disk and record the data and disk numbers.

BACKUP will display the name of each file being backed up. You can redirect system output to the printer to get a printed copy of the list.

Exit codes are as follows:

- 0 Normal completion
- 1 No files found
- 3 Operation terminated by user
- 4 Operation terminated due to error

Use these codes with the IF command in batch processing.

Files on the backup disk cannot be used in normal processing. They should be used only by the RESTORE command.

Examples: The following command backs up all the files on hard disk drive C:

```
BACKUP C: A:/S
```

The BACKUP command allows the use of global filename characters, which cause all files matching the filename to be backed up. The following command causes all files from the current directory of hard disk drive C with an extension of .XEN to be backed up on the disk in drive A:

```
BACKUP C: *.XEN A:
```


BREAK

Internal

Syntax: **BREAK** [ON|OFF]**Purpose:** Checks for a control break whenever a program requests that DOS perform any functions.**Comments:** Usually DOS performs a check for a <Ctrl><Break> being entered at the keyboard during screen, keyboard, printer, or auxiliary device operations. This command specifies when DOS should check for a <Ctrl> <Break>.**BREAK ON**

DOS checks for a <Ctrl><Break> any time a program requests any DOS function.

BREAK OFF

DOS checks for a <Ctrl><Break> only during screen, keyboard, printer, or Asynchronous Communication Adapter operations.

To check the current state of <Ctrl><Break> enter **BREAK** with no parameter.**CHDIR (Change Directory)**

Internal

Syntax: **CHDIR** [[*d:*]*path*]

or

CD [[*d:*]*path*]**Purpose:** Changes the DOS current directory or displays the current directory path of a drive.**Comments:** If you do not indicate a specific drive, DOS assumes the default drive. If you use **CHDIR** or **CD** with no parameters, or with only a *d:* parameter, DOS shows the current directory path of the named or default drive.**Examples:** **CHDIR **

Changes the current directory of the default drive to its root directory.

CD B:\SUB1\SUB2

Changes Drive B's current directory to the path "root→SUB1→SUB2".

CD B:SUB3

Changes Drive B's directory to the current directory path plus SUB3. If the previous example had been used, the resultant path would be:

root→SUB1→SUB2→SUB3

CHKDSK

External

Syntax: CHKDSK [*d:*][*filename*][*/F*][*/V*]

Purpose: Produces a disk and memory status report.

Comments: CHKDSK first shows any error messages with the status report following. For a listing of error messages see Appendix B.

CHKDSK does not automatically correct errors. In order to correct an error, you must use the */F* (fix) parameter. Without the */F* parameter, CHKDSK will not actually make the corrections on the disk.

CHKDSK will show the number of non-contiguous areas occupied by the file or files if you include a filename, but CHKDSK will look only in the current directory for these files.

CHKDSK will provide more detail about the errors if you use the */V* parameter.

CHKDSK always assumes that the disk to check is in the named drive. On a single disk-drive system, it is important that the specified drive is not the default drive, unless you wish to check the DOS disk itself.

It is a good idea to run CHKDSK occasionally for each disk to be sure of the integrity of your file structures.

COPY

Internal

Syntax: COPY [*/A*] [*/B*] [*d:*] [*path*]*filename*[*.ext*] [*/A*] [*/B*]
[*d:*] [*path*] [*filename*[*.ext*]] [*/A*] [*/B*] [*/V*]

or

COPY [*/A*] [*/B*] [*d:*] [*path*]*filename*[*.ext*] [*/A*] [*/B*]
[+*d:*] [*path*]*filename*[*.ext*] [*/A*] [*/B*]...]
[*d:*] [*path*] [*filename*[*.ext*]] [*/A*] [*/B*] [*/V*]

Purpose: Copies files to another disk and gives the copy a different name if desired.

COPY will also copy files to the same disk. The copies must have different names unless a different directory is indicated.

You may combine files (Concatenate) while performing the copy.

COPY may also be used to transfer data between any of your system devices.

Comments: The source file is the first file named, and the target file is the second. If the second parameter has no filename, files are copied without a name change.

The wildcard characters ? and * may be used in the filename and in the extension of both the duplicate and the original files. If you use an * or ? in the source filespec, the name of the files will be shown as the files are being copied.

/A

If you use /A with a source files specification, the file is treated as a text (ASCII) file. COPY copies the content of the file until it finds the first EOF (End of File) character, Ctrl-Z (1AH). The rest of the file is not copied. If you use /A with a target file specification, COPY adds Ctrl-Z character as the last character of the file. /A is the default when concatenation (Option 3) is specified.

/B

If you use /B with a source file specification, COPY copies the entire file. If you use /B with a target file specification, COPY doesn't add Ctrl-Z character as the last character. /B is the default when concatenation is not specified.

NOTE: /A or /B takes effect on the file preceding it and on all remaining files until another /A or /B is found.

/V

Verify Option:

It indicates to DOS to verify that the sectors are written on the target disk correctly. This option provides the ability to verify that critical data has been properly recorded even though errors are rare. The COPY command will run more slowly because of overhead due to the verification process.

If you have previously specified the VERIFY ON command, /V is unnecessary.

There are three format options for the COPY command:

Option 1—Source and Destination Files Have The Same Names

The copied file will have the same filename and extension as the source file. For example:

```
COPY [d:][path]filename [.ext]
```

or

```
COPY [d:][path]filename [.ext] d:[path]
```

The first example shows that the file is copied to the current directory of the default drive. The second example specifies the target drive and/or directory.

The copied file will have the same filename as the source file because we did not indicate a name for the second file. The source drive and the target drive cannot be the same unless different directories were either implied or specified; otherwise, the copy is not allowed. For example:

```
COPY *.* B:
```

If the default drive is A, this command copies all the files from the default Drive A to Drive B. There are no changes in the filenames or in the extensions. The filenames are shown while the files are copied.

If the files are in different sub-directories they can be on the same disk. For example:

```
COPY B:\PROG1 B:\SUB1
```


This command copies the file PROG1 from the root directory on Drive B to the directory path SUB1 on the same drive. Both the copy and the original files have the same filenames. This example assumes that directory SUB1 exists on Drive B. If the SUB1 directory is not there then the file PROG1 will be copied into a file named SUB1 in the root directory of Drive B. If the second parameter names a directory that is there, the file (or files) will be copied into that directory with the same filename. If the second parameter names a directory that is not there, it will be considered a filename.

Option 2—Source and Destination Have Different Names

The copied file will have a different name from the file that is copied. For example:

```
COPY [path]filename [.ext] filename[.ext]
```

or

```
COPY [path]filename[.ext] d:[path]filename [.ext]
```

The first example shows the first filename is copied and renamed to the second filename. Because a drive was not specified, the default drive was used. The second example shows that the file is copied and also renamed, but a target drive was named. The source drive and the target drive may be the same because the name of the file was changed. The current directory may be different or the same. For example:

```
COPY PROG1.XYZ B:*.ABC
```

CTTY (Change Console)

Internal

Syntax: CTTY *device-name*

Purpose: Changes the standard input and output devices.

Comments: CTTY defines the device to be used as the primary console.

AUX, COM1, or COM2

Defines that device as the primary console.

CON

Resets the standard input and output device to the primary console.

Example: The following command causes DOS to use the AUX device for its screen and keyboard operations:

```
CTTY AUX
```

The following command moves input and output back to the original device (standard screen and keyboard):

```
CTTY CON
```

You must specify the device which has capability of both input and output operations.

DATE

Internal

Syntax: DATE [mm-dd-yy]

Purpose: Sets and displays the date.

Comments: The date is recorded in the directory for any files you create or alter. If you enter a valid date with the DATE command, the date is accepted, and the system prompt appears. Otherwise, the DATE command issues the following prompt:

```
Current date is day dd-mm-yy
Enter new date:..
```

Enter a new date in the form dd-mm-yy or dd/mm/yy,
where:

```
mm is a number from 1-12
dd is a number from 1-31
yy is a number from 80-79
   (the 19 is assumed)
   or a number from 1980-2079
```

The date is entered from the keyboard or from a batch file. The valid separators within the date are hyphens (-) and slashes (/). The system does not prompt you for the date if you use an AUTO-EXEC.BAT file. You may include the DATE command in that file.

NOTE: To leave the date unchanged, press <ENTER>.

DEL

Internal

See "ERASE Command" in this chapter.

DIR (Directory)

Internal

Syntax: DIR[d:][path] [filename[.ext]][/P][/W]

Purpose: Lists either all the directory entries, or only those for specified files.

Comments: The displayed information includes the volume identification and the amount of free space on the disk.

The line displayed for each file includes the file size in decimal bytes and the date and time at which information was last written to the file.

The notation <DIR> in the file size field indicates entries which name other directories.

Directory entries for system files are not listed, even if they are present.

Parameters:

/P The display will pause when the screen is filled. Press any key to continue with the directory listing.

/W Creates a wide display of the directory so that 5 filenames and directory names are shown on each line. (Only 80-column displays should use this parameter.)

The wildcard characters ? and * may be used in the filename and extension parameters.

The DIR command has two format options. (the /P and /W parameters may be used with either option).

Option 1—List All Files

This option lists all the files in a directory. For example:

```
DIR [path]
```

or

```
DIR d:[path]
```

The first example lists all directory entries on the default drive. The second example lists all directory entries on the specified drive. If a path is specified, the listing is of files in the specified directory. The current directory is listed if no path is specified.

The directory listing might look like this:

```
A>DIR
```

```
Volume in drive A is DISKTMP
```

```
Directory of A:\
TMP1  A      1099    10-05-83  9:13p
TMP2  A     20123    12-13-83  7:25a
TRYIT      4093     7-05-82  5:34p
PROGS  <DIR>
FILEA  A      2288    11-12-83  2:32p
5 File(s)          131215 bytes free
```

If the directory being listed is not the root directory it will include two unusual entries. The first entry contains a period in place of a filename. The second contains two periods in place of filename. The list of files shown above would follow these two entries. These two entries indicate that the directory being listed is a sub-directory, rather than the root directory.

Option 2—List Selected Files

This option lists selected files from a directory. For example:

```
DIR filename.ext
```

or

```
DIR d:filename.ext
```

If either the filename or the extension is omitted, an * is assumed.

Both examples list all the files that have the specified filename and extension. The first example uses the default drive and the second example uses the specified drive.

Using the previous example, if you enter:

```
dir tmp2.a
```

the screen might look like this:

```
A>dir tmp2.a
```

```
Volume in drive A is DISKTMP
Directory of A:\
```

```
TMP2 A      20123   12-13-83  7:25a
      1 File(s) 131215 bytes free
```

To see the entry for a file that has no extension, enter the filename followed by a period. In this case, the .ext does not default to *. For example:

```
dir file1.
```

This command displays the entry for FILE1, but not for FILE1.A.

If you wish to display all the files in directory PROGS on the above drive, you can enter:

```
dir progs
```

The screen will look like this:

```
A>dir progs
```

```
Volume in drive A is DISKTMP
Directory of A:\progs
```

```

.           <DIR>          9-17-83
..          <DIR>          9-17-83
PROG12     COM           2863  10-23-83  8:21a
          3 File(s)      140912 bytes free
```

All files in directory PROGS have been listed. The two special entries found in all sub-directories have been included. The single period entry is the directory being listed (PROGS), and the double period is this directory's parent directory. In this case, the parent is the root directory. If your current directory is PROGS and you want to view the files in the parent directory, enter:

```
dir ..
```

The following screen is displayed:

```
A>dir ..
```

```
Volume in drive A is DISKTMP
Directory of A:\
```

```
TMP1  A      1099   10-05-83  9:13p
TMP2  A     20123  12-13-83  7:25a
TRYIT A      4093   7-05-82  5:34p
PROGS <DIR>     9-17-83  8:47p
FILEA A     2288  11-12-83  2:32p
      5 File(s) 133874 bytes free
```


DISKCOMP (Compare Disk)

External

Syntax: DISKCOMP [*d:*][*d:*][/*1*][/*8*]**Purpose:** Compares the contents of one disk to the contents of another disk. After a DISKCOPY operation you may run DISKCOMP to ensure that the two disks are identical.**NOTE:** This command compares entire disks; the FC command compares files.**Parameters:**

- /1 Compares only the first side of the disks, even if the disks and drives are dual-sided.
- /8 Compares only 8 sectors per track, even if the first disk contains 9 sectors per track.

DISKCOMP compares track by track. (0-39) If the tracks are different,

Compare error(s) on Track *xx*, Side *y**XX* is the track number (0-39) and *y* is the side (0 or 1) where a mismatch has occurred. When the comparison to the entire disks terminates, the following prompt appears:

Compare more disks (Y/N)?_

If you answer Y, the next comparison is performed on the same drives that you originally specified. You do receive prompts to insert the proper disks.

If you answer N, you end the command.

NOTE:

If you omit both parameters:

a single drive comparison is performed on the default drive.

If you omit the second parameter:

the default drive is used as the secondary drive. If the default drive is the first parameter, this results in a single-drive comparison.

A single drive comparison:

all prompts are for Drive A, regardless of any drive specifiers entered and are displayed to insert the disks at the appropriate time. DISKCOMP waits for you to press any key before it continues.

If you try to compare a disk created by the COPY command with the disk you copied from, DISKCOMP usually displays a mismatch message. The COPY command produces a copy that contains the same information, but it places the information at different locations on the target disk from those locations used on the source disk. In this case, use the FC command to compare individual files on the disks.

If a disk error occurs while DISKCOMP is reading the disk a message indicates where (track and side) the error occurred. DISKCOMP continues to compare the rest of the disk. The remainder of the data cannot be read correctly from the indicated track and side so you can expect to receive a "Compare error message".

The number of sides and sectors per track to be compared is based on the disk that is to be read first (the first drive parameter entered).

If the first disk or drive can be read on only one side, or if the /1 parameter is used only the first side is read from both disks.

If the first disk contains 9 sectors per track DISKCOMP will compare 9 sectors per track unless you used the /8 parameter.

If the first disk is dual-sided, and /1 is not specified a two-sided comparison is performed. An error message is produced if either the second drive or the disk is a single-sided disk.

DISKCOPY (Copy Disk)

External

Syntax: DISKCOPY [d:][d:][/1]

Purpose: Copies the contents of one disk to another disk. The target disk is formatted if necessary.

Comments: The source drive is the first parameter. The target drive is the second parameter. You can specify the same drives or different drives. If the drives are the same, a single-drive copy is performed. You are prompted to insert the disks at the appropriate times. DISKCOPY waits for you to press any key before continuing.

/1 The parameter causes DISKCOPY to copy only the first side of the disk, regardless of the disk or drive type.

After copying, DISKCOPY prompts:

Copy another (Y/N)?_

If you answer Y, the next copy is done on the same drives that you originally specified. You are prompted to insert the proper disks.

If you answer N, the command ends.

NOTE: If the target disk has not been formatted the same as the source disk, DISKCOPY will format the target disk during the

If you omit both drive parameters, a single drive DISKCOPY operation is performed on the default drive.

If you omit the second parameter, the default drive is used as the target drive.

If you omit the second parameter and you specify the default drive as the source drive, a single-drive copy operation is performed.

On a single-drive system, all prompts will be for Drive A, regardless of any drive letter you may enter.

Fragmented disks:

are disks that have substantial file creation and deletion activity. Disk space is not allocated sequentially. The first free sector found is the next sector allocated, regardless of its location on the disk.

A fragmented disk causes degraded performance due to delays involved in finding, reading, or writing a file.

It is recommended that you use the COPY command, instead of DISKCOPY, to eliminate the fragmentation. For example:

```
COPY A:*. * B:
```

This copies all the file from the disk in Drive A to the disk in Drive B.

After a successful DISKCOPY you may run DISKCOMP to ensure that the disks are identical.

If disk errors are encountered on either disk, DISKCOPY indicates the drive, track, and side in error and proceeds with the copy. The copy (target disk) may or may not be usable, depending on whether the affected disk location was to contain valid data.

DISKCOPY automatically determines the number of sides and sectors per track to copy, based on the source disk. If only the first side of the source disk can be read, then only the first side can be copied. If the source disk is dual-sided, both sides can be copied (unless you override it with the /1 parameter).

If the source disk has ever been formatted with nine sectors per track, then all nine sectors on each track will be copied.

ERASE

Internal

Syntax: ERASE [d:][path][filename[.ext]]

or

DEL [d:][path][filename[.ext]]

Purpose: Deletes a file or files. If no path is entered, the file is deleted from the current directory.

Comments: The shortened form, DEL, is a valid abbreviation. The wildcard characters ? and * are valid in the filename and in the extension. Wildcard characters should be used with caution because multiple files can be erased with a single command.

To erase all files in the current directory, enter:

```
ERASE [d:]*.*
```

To erase all files in a specific directory, enter:

```
ERASE [d:]path
```

ERASE assumes a filename of *.* if no filename is given.

NOTE: The system files cannot be erased. If you use the filespec *.* to erase all of the files in a directory or on a disk, DOS issues the following message to verify that you actually want to erase all files:

```
Are you sure (Y/N)?
```

Enter Y if you do want to erase all of the files. Otherwise, enter N. Then press <ENTER>.

The two special entries in each subdirectory (*. and *.*) cannot be erased.

Examples: The file MYFILE.123 will be erased from the current directory of Drive A.

```
ERASE A:MYFILE.123
```


EXE2BIN

External

Syntax: EXE2BIN [d:][path] filename[.ext]
[d:][path][filename[.ext]]

Purpose: Converts .EXE files to a form that is compatible with .COM programs. This results in a saving of disk space and faster program loading.

Comments: The first file named is the input file. The default extension is .EXE. The input file is converted to COM file format (memory image of the program) and stored in the output file, [d:]filename[.ext]. The drive of the input file is used if you do not specify a drive. The input filename is used if you do not specify an output filename. The new file is given an extension of .BIN if you do not specify a filename extension in the output filename. The current directory is used if you do not specify a path.

The input must be in valid .EXE format as produced by the **linker**. The resident, or actual code and data, part of the file must be less than 64K. There must be no STACK segment.

Two kinds of conversions are possible, depending upon the specified initial CS:IP.

If CS:IP is not specified in the program (the .EXE file contains 0:0), a pure binary conversion is assumed. If segment fixups are necessary (instructions requiring segment relocation are in the program), you are prompted for the fixup value. This value is the absolute segment at which the program is to be loaded.

In this case, the DOS command processor will not be capable of properly loading the program. The program is usable only when loaded at the absolute memory address specified by a user application.

If CS:IP is specified as 0000:100H, it is assumed that the file is to be run as a COM file. The location pointer is set at 100H by the assembler statement ORG and the first 100H bytes of the file are deleted. No segment fixups are allowed because COM files must be segment relocatable. Once the conversion is complete, you may rename the resultant file to a .COM extension. Then, the command processor is capable of loading and executing the program in the same manner as the .COM programs supplied on your DOS disk.

If CS:IP does not meet one of these criteria, or if it meets the COM file criterion but has segment fixups, the following error message is displayed:

File cannot be converted

This message is also displayed if the file is not a valid .EXE file.

To produce standard COM files with the assembler:

Use the assembler statement ORG to set the location pointer of the file at 100H.

Specify the first location as the start address. This is done by using the END statement. For example:

```
ORG 100H
START:
.
.
.
END START
```

The program must not use references that are defined only in other programs.

EXTPRT

External

Syntax: EXTPRT

Purpose: Switches the default printer from the internal printer to the external printer.

Comments: The default printer switches back to the internal printer if the power is turned off or if the system is rebooted, or the <Alt>+<PrtSc> keys are pressed. This command is useful when an external parallel printer is being used. Include this in AUTOEXEC.BAT file to automatically switch the default printer at boot time. The <Alt>+<PrtSc> key sequence is used to toggle the default back to the internal printer.

FAST

External

Syntax: FAST

Purpose: Switches the CPU clock frequency from 4.77 MHz (standard) to 7.16 MHz (fast). This increases the processing speed of the Exec. Partner.

Comments: The CPU clock frequency is switchable between two settings, standard and fast. There are 3 methods of switching the CPU speed.

1. Setting the DIP switch*.
2. Simultaneously pressing the <Ctrl>+<Alt>+<+> keys on the keyboard to toggle the speed between normal and fast.
3. Using the FAST command

*Setting the DIP switch only switches the default CPU clock frequency.

NOTE: If the DIP switch is set to the fast mode, it may be necessary to boot the machine from DOS and use the <Ctrl>+<Alt>+<+> keys to toggle to the slow mode for certain programs to run properly.

FC

External

Syntax: FC [/#] [/B] [/W] [/C] [d:] [path]
[filename[.ext]] [d:] [path] [filename[.ext]]

Purpose: Compares the contents of the first file (or set of files) to the contents of the second file (or set of files).

NOTE: FC compares two sets of files, DISKCOMP compares two entire disks.

Comments: The files that you compare may be on the same drive or on different drives, and they may be in the same directory or different directories.

FC A:PROGS1 A:PROGS2

It is possible to compare all files in one directory with all corresponding files in another directory.

There are four switches that you can use with FC:

/B Forces a binary comparison of both files. The two files are compared byte-to-byte, with no attempt to re-synchronize after a mismatch. The mismatches are printed as follows:

```
--ADDRS----F1----F2-
xxxxxxx yy zz
```

(where xxxxxxxx is the relative address of the pair of bytes from the beginning of the file). Addresses start at 00000000; yy and zz are the mismatched bytes from file 1 and file 2, respectively. If one of the files contains less data than the other, then a message is printed out. For example, if file 1 ends before file 2, then FC displays:

```
... Data left in F2 ...
```

/# # stands for a number from 1 to 9. This switch specifies the number of lines required to match for the files to be considered as matching again after a difference has been found. If this switch is not specified, it defaults to 3. This switch is used only in source comparisons.

/W Causes FC to compress whites (tabs and spaces) during the comparison. Thus, multiple contiguous whites in any line will be considered as a single white space. Note that although FC compresses whites, it does not ignore them. The two exceptions are beginning and ending whites in a line, which are ignored. For example (note that an underscore represents a white)

```
__More__data_to_be_found__
```

will match with

```
More_data_to_be_found
```

and with

```
___More___data_to_be___found___
```

but will not match with

```
__Moredata_to_be_found
```

This switch is used only in source comparisons.

/C Causes the matching process to ignore the case of letters. All letters in the files are considered uppercase letters. For example,

```
Much_MORE_data_IS_NOT_FOUND
```

will match

```
much_more_data_is_not_found
```

FC reports the differences between the two files you specify by displaying the first filename, followed by the lines that differ between the files, followed by the first line to match in both files. FC then displays the name of the second file followed by the lines that are different, followed by the first line that matches. The default for the number of lines to match between the files is 3. (If you want to change this default, specify the number of lines with the /# switch.) For example,

```
-----<filename 1>
<difference>
<1st line to match file 2 in file 1>
```

```
-----<filename 2>
<difference>
<1st line to match file 1 in file 2>
```

```
-----
```


FC will continue to list each difference.

If there are too many differences (involving too many lines), the program will simply report that the files are different and stop.

If no matches are found after the first difference is found, FC will display

*** Files are different***

and will return to the prompt (for example, A>).

FDISK (HARD DISK)

External

As was mentioned previously, a hard disk can be divided into as many as four partitions. You can specify which partition the operating system will access when you start the computer. Note, however, that an operating system can access only one partition. Data cannot be transferred directly from one partition to another.

The FDISK command allows you to set up or delete only the DOS partition. A partition for another operating system must be created or erased by that operating system. For a detailed description of the FDISK command, refer to page 5-2.

FIND (Filter)

External

Syntax: **FIND**
[/V] [/C] [/N]"string"[[d:] [path]filename[.ext.]...]

Purpose: This filter sends to the standard output device all lines from the filenames specified in the command line that contain the specified string.

Comments: The following options are available:

- /V All lines not containing the string are displayed.
- /C A count of the number of matching occurrences of the string in each file are displayed. The matching lines are not displayed.
- /N The relative line number of each matching line is displayed ahead of the line from the file.

The string should be enclosed in double quotes. Two quotes in succession are taken as a single quote.

Wildcard characters are not allowed in the filenames or extensions.

Example: **FIND "My Story" story1.txt story2.txt story3.txt**

This command outputs all lines from story1.txt, story2.txt and story3.txt (in that order) that contain the string "My Story".

Since FIND creates a temporary file on the disk, the disk must not be write-protected. The temporary file is deleted when the operation finishes.

FORMAT

External

Syntax: **FORMAT** [d:] [/S][/V][/B] [/1] [/8]

Purpose: Sets up the disk in the named or default drive to a format usable by DOS. Studies the entire disk for any bad tracks. Reads the disk to accept DOS files by setting up the directory, File Allocation Table, and system loader.

Comments: All new disks must be formatted (using either the FORMAT or DISKCOPY command) before they can be used. Formatting destroys all previously existing data on the disk.

The following options are available:

- /S The operating system files are copied from the default drive to the new disk.
- /1 The disk is formatted for single-sided use, regardless of the drive type.
- /8 The disk is formatted for use at eight sectors per track. The default is nine sectors per track usage if you do not specify /8. FORMAT always creates nine physical sectors on each disk track, but it instructs DOS to use only eight sectors per track if you use the /8 parameter.

/V

A volume label is created. The label is written on the disk. The volume label is for your use in keeping track of your disks. The /V parameter cannot be used with the /8 parameter. FORMAT will prompt you to enter a volume label (volume identification). The label can have from one to eleven characters. All characters acceptable in filenames are also acceptable in the volume label. The volume label, however, does not contain a period between the eighth and ninth characters.

/B

Creates an eight or nine sector-per-track disk with space allocated for the system files. It does not place the system modules or the command processor on the disk.

The /S parameters cannot be used with the /B parameter.

NOTE: Any defective tracks are marked as reserved to prevent the tracks from being allocated to a file.

Directory entries for system files are marked as hidden files. They do not appear in the directory.

FORMAT produces a status report. The report includes:

Total disk space

Space marked as defective

Space currently allocated to files
(when the parameter /S is used)

Space available for your files

A disk formatted for single-sided use can be used in a dual-sided drive. A disk formatted for dual-sided use will not be usable in a single-sided drive.

The following command formats the disk in Drive B, the operating system files are copied and a volume label is written.

FORMAT B:/S/V

The system displays the following message:

Insert new disk for drive B:
and strike any key when ready

After you insert the appropriate disk and press any key, the system displays this message:

Formatting..

When the formatting is complete, the system displays this message:

```
Formatting...Format complete
System transferred
```

Volume label (11 characters, ENTER for none)? DATADISK

```
xxxxxx bytes total disk space
xxxxxx bytes used by system
xxxxxx bytes available on disk
```

Format another (Y/N)?

Enter Y to format another disk.

Enter N to end the FORMAT program.

GRAPHICS (Screen Print)

External

Syntax: GRAPHICS

Purpose: Prints the contents of a graphics display screen.

Comments: To print what is on the screen, press the <Shift> <PrtSc> keys after executing the GRAPHICS.COM file. If the screen is in text mode, the text will print in less than 1 minute. If the screen is in graphics mode, the following things occur each time the <Shift> <PrtSc> key is pressed:
The screen contents will be printed in up to four shades of gray in the 320×200 color graphics mode.

The screen will print sideways on the paper in the 640×200 graphics mode. The lower left corner of the screen will print on the lower right corner of the paper.

The code INT 5 will call the screen print from a program.

MKDIR (Make Directory)

Internal

Syntax: MKDIR [*d:*]*path*

or

MD [*d:*]*path*

Purpose: Creates a sub-directory on the indicated disk.

Comments: The default drive is assumed if no drive is named.

Example: This command creates an entry in the root directory for a new sub-directory named SUB1:

```
MD \SUB1
```

To add another directory level, use either of the following two examples:

If the current directory is the root directory:

```
MD \SUB1\SUB2
```

This command adds an entry for sub-directory SUB2.

If the current directory is SUB1:

```
MD SUB2
```

Both examples will do the same thing. The directory SUB1 will have an entry for sub-directory SUB2.

The first \ tells DOS to search directories in the root directory. With no leading \ in the last example, DOS begins searching in the current directory. Each directory may hold names of other directories.

The only limit to creating as many sub-directories as wanted is available disk space. Be sure that the maximum length of any single path from the root directory to the level you want is 63 characters, including all backslashes.

MODE

External

Syntax: `MODE LPT#[n][,m][,P]`

or

`MODE n`

or

`MODE [n],m[,T]`

or

`MODE COMn:baud[,parity[,databits [,stopbits[,P]]]]`

or

`MODE LPT#:=COMn`

Purpose: Determines the mode of operation for a monitor or printer. Routes printer output to an serial (RS232C) interface.

Comments: If an *n* or *m* parameter is invalid or missing, the mode operation will not be altered for that parameter. There are four format options:

Option 1—Use with the printer.

`MODE LPT#[n][,m][,P]`

where:

is 1, 2, or 3 (the printer number)
n is 80 or 132 (characters per line)
m is 6 or 8 (lines per inch vertical spacing)
P specifies continuous retry on time-out errors

For example:

`MODE LPT1:132,8`

The printer is set to 80 to 132 characters per line and 8 lines per inch of vertical spacing. 80 characters per line and 6 lines per inch are the default options for the printer.

<Ctrl><Break> will stop the retry loop. In order to stop the retry of time-out errors if you have entered parameter *P*, use MODE Option 1 without naming *P*.

Option 2—Use to switch the built-in display circuit to initiate the display mode of it.

`MODE n`

or

`MODE [n],m[,T]`

where:

n is 40, 80, BW40, BW80, CO40, CO80, or MONO

40 places the width of the display at 40 characters per line

80 places the width of the display at 80 characters per line

BW40 (CO40) The display mode is set to Black and White with 40 characters per line (color is deactivated).

BW80 (CO80) The display mode is set to Black and White with 80 characters per line (color is deactivated).

MONO alters the active display adapter to the Monochrome Display Adapter. This will always have a display width of 80 characters per line.

m is L or R used to shift the display left or right.

T asks for the use of a test pattern to align the display.

Shifting the display one character for 40 column displays or two characters for 80 column displays in either direction will enhance readability. If you enter T in the MODE command, a prompt will ask if the screen is aligned correctly. Entering Y ends the command. Entering N repeats the shift and displays the prompt again. For example:

```
MODE 80,L,T
```

This places the display mode of operation at 80 characters per line and shifts the display two characters to the left. Since the test pattern is shown again you have the chance to shift the display again without having to enter the command again.

Option 3—Use with the serial (RS232C) interface. (Asynchronous Communication Interface)

```
MODE COMn:baud[,parity
[databits[,stopbits[P]]]]
```

where:

n Set to either 1 or 2 (1: Internal serial interface (Serial Port) 2: Optional Asynchronous Communication Interface)

baud 110, 150, 300, 600, 1200, 2400, 4800, or 9600

parity Either E (even), O (odd), or N (none). The default is E.

databits Either 7 or 8. The default is 7.

stopbits Either 1 or 2. If the baud is 110, the default is 2. If the baud is not 110, the default is 1.

To indicate baud you only need to enter the first two characters. Other characters are ignored.

These are called the protocol parameters. They initialize the serial interface. You must enter at least the baud rate. To specify defaults, all other parameters may be omitted by entering only commas. For example,

```
MODE COM1:30,O,8,1,P
```

This command places the baud rate at 300, odd parity, eight databits, and one stopbit. To get the defaults for the above, enter:

```
MODE COM1:30,,,,P
```

The parity will default to even, the databits to seven, and the stopbits to one.

In order to set the serial interface for use with a serial printer, and to have the time out errors retried, enter option P. Pressing Ctrl-Break stops the retry loop. Reinitializing the serial interface without entering the P will stop the continuous retry of time out errors.

NOTE: For proper operation with a serial printer, consult the printer Users Guide for correct settings (baud rate, parity, stopbits, etc.).

Option 4—Use to redirect parallel printer output to a serial (RS232C) interface (Asynchronous Communication Interface)

```
MODE LPT#:=COMn
```

where:

- # Either 1, 2, or 3 (printer number)
- n Either 1 or 2 (1: Internal serial interface, (Serial Port), 2: optional Asynchronous Communication Interface)

The output originally sent to printer LPT# is sent to the n instead.

NOTE: To redirect parallel printer output to a serial device, first initialize the serial interface with Option 3 (see above). Include parameter P if that serial device is a printer.

To stop the redirection for the printer designated by the #, enter:

```
MODE LPT#:[n][,m]
```


MORE (Filter)

External

Syntax: MORE

Purpose: This filter reads data from the standard input device, and sends one screen-full of data to the standard output device. The transmission pauses with the message —More— on the screen.

Comments: The next screen-full of data is written to the standard output device when any character key is pressed. This process continues until all input data is read.

Example: The command line displays the contents of the TEST.ASM one screen-full at a time. When the screen is full, the message —More— appears on the bottom line. To see the next screen-full you can press any key:

```
MORE<TEST.ASM
```

Since MORE creates a temporary file on the disk, the disk must not be write-protected. The temporary file is deleted when the operation finishes.

PATH (Set Search Directory)

Internal

Syntax: PATH [d:] path[[:[d:]path]...]

Purpose: Sets a path to be searched for commands or batch files that were not found by a search of the current directory.

Comments: The list of drives and path names must be separated by semicolons. If a command cannot be found in the current directory, DOS will search the named directories in the order you entered them, but the current directory is not changed.

PATH with no parameters displays the names that were indicated in a previous PATH command. PATH with only a semicolon (PATH;) resets the search path to null. The default when DOS begins is a null search path. DOS will only search the current directory for commands and batch files.

Examples: Assume the program PROGA.COM resides in directory ACCTDIR on Drive B, and that the default drive is Drive A:

```
PATH \SUB1;\SUB1\SUB2;B:\ACCTDIR
```

This instructs DOS to look in the current directory of the drive specified, followed by A: SUB1, then A: SUB1 SUB2 then B: ACCTDIR until it finds the command you have entered. If the command entered is not found in any of the directories specified in PATH, the message "Bad command or filename" is displayed.

If you enter the command:

PROGA

DOS searches four directories. DOS finds the program PROGA in B: ACCTDIR.

NOTE: Incorrect information in the paths, such as invalid drive specifications or imbedded delimiters, will not be detected until the specified paths are actually searched.

If a path is specified that no longer exists, DOS ignores that path and goes on to the next.

PRINT

Syntax: **PRINT** [[d:][filename[.ext]][[/T][/C][[/P]...]] External

Purpose: Prints a list of data files on the printer while you are doing other jobs on the computer.

Comments: You may enter multiple filenames on the command line, each filename with appropriate parameters. Up to 10 filenames can be listed for printing at one time. Wildcard characters * and ? are allowed in the filename and extension. Only files in the current directory can be listed for printing. You can change the current directory without affecting the printing of the files already in the print queue.

The files are queued for printing in the order entered. After each file is printed, the printer paper is advanced to the next page.

Parameters:

/T Terminate mode. All listed files are canceled from the print queue (files waiting to be printed). If a file is currently being printed:

The printing stops.
A cancellation message is printed.
The paper is advanced to the next page.

/C Cancel mode. You may select which file or files to cancel. The preceding filename and all following filenames entered on the command line are canceled from the print queue until a /P is entered on the command line, or the <ENTER> key is pressed.

/P Print mode. The preceding filename and all following filenames are added to the print queue until a **/C** is entered on the command line, or the **<ENTER>** key is pressed.

If no parameters (**/**) are specified following a filename and the **<ENTER>** key is pressed, the files listed on the command line are listed for printing (**/P** is assumed).

If **PRINT** is entered with no filenames, **PRINT** displays the names of the files currently in the print queue.

The first time the **PRINT** command is executed after you start your system, the following message is displayed:

Name of list device [PRN]:

You may specify the output list device, **LPT1**, **LPT2**, **LPT3**, **PRN**, **COM1**, **COM2**, **AUX**, etc. If you press **<ENTER>** the default (**PRN**) will be selected.

NOTE: Be sure the device you name is attached to your system. Specifying a nonexistent device causes unpredictable system behavior.

If a disk error occurs while the system attempts to read the file to be printed, **PRINT** will:

Cancel the file currently printing.

Print a disk error message.

Advance the printer paper to the next page.

Print the remaining files in the print queue.

If the **/T** or **/C** parameters are used to cancel a file or files currently being printed, **PRINT** will:

Print a file cancellation message.

Advance the printer paper to the next page.

Resume printing with the first file remaining in the print queue.

NOTE: The disk containing the files being printed must remain in the specified drive until all printing is complete. Files in the print queue must not be altered or erased until after they have been printed.

The printer cannot be used for any other purpose while **PRINT** has data to print.

Examples: To use the **PRINT** command for the first time, enter:

PRINT a:myfile.tst

DOS responds with:

Name of list device [PRN]:

Press the <ENTER> key to send output to the printer.

DOS adds the file MYFILE.TST from Drive A to the print queue and outputs its content to the device "PRN" printer.

To empty the print queue, enter:

```
PRINT /T
```

To remove all TEST*. * files from the print queue that have the same drive letter as the default drive, enter:

```
PRINT test*.* /C
```

To remove all TEST*. * files from the printer queue Drive A from the print queue, enter:

```
PRINT a:test1.tst /C a:test2.tst a:test3.tst
```

To add the files TEST1.TST and TEST2.TST to the print queue, and remove TEST3.TST from the print queue.

```
PRINT test1.tst test2.tst test3.tst /C
```

To remove file TEST1.TST from the print queue, and add the files TEST2.TST and TEST3.TST to the print queue, enter:

```
PRINT test1.tst /C test2.tst /P test3.tst
```

PROMPT (Set System Prompt)

Internal

Syntax: PROMPT [*prompt-text*]

Purpose: Changes the DOS prompt.

Comments: All text on the PROMPT command line becomes the new DOS prompt. If no text is specified, the default DOS prompt is assumed. Strings with special meaning may be imbedded in text in the form \$*c*.

c is one of the following:

- \$ The "\$" character.
- t The time.
- d The date.
- p The current directory of the default drive.
- v The version number.
- n The default drive.
- g The ">" character.
- l The "<" character.

- b The "|" character.
- q The "=" character.
- h A backspace
- e The ESCape character.
- A CR LF sequence.

Example: This command would set the normal DOS prompt:

```
PROMPT $n$g
```

This command would set ABC as the system prompt:

```
PROMPT ABC
```

This command would set up a two-line prompt:

```
PROMPT Time=$t$_Date=$d
```

The prompt would be:

```
Time=(current time)
```

```
Date=(current date)
```

If you precede any of the DOS command delimiters (such as semicolon, blank, etc.) with a character plus a null string, you can create a prompt that begins with one of the delimiters. In the following example, the semi-colon will be treated as the first character of the prompt, rather than as a delimiter between the word PROMPT and its parameter. For example:

```
PROMPT $A;ABC
```

The \$A is treated as a null character, because A is not one of the defined characters in the above list. All of the characters that follow the null character will become the new DOS prompt.

RECOVER

External

Syntax: RECOVER [*d:*] [*path*]*filename*[*.ext*]

or

RECOVER *d:*

Purpose: Recovers files from a defective disk. The disk may have a defective sector or the directory may be damaged.

Comments: If a sector on a disk is bad, you can recover the file specified *filename* containing that sector. The part of the file that is in the bad sector is not recovered.

In the second format, all files on the specified disk are recovered. It is assumed that the directory is damaged.

If the filename or extension includes the wildcard characters * and ?, only the first matched file is recovered.

Examples: If the disk file to be recovered is PROG1 enter:

```
RECOVER A:PROG1
```

The disk file PROG1 on Drive A is read sector by sector, skipping the bad sectors. The filename is not changed. When the bad sectors are found, the sectors are marked and DOS will no longer allocate your data to that sector.

To recover the contents of an entire disk from Drive A, enter:

```
RECOVER A:
```

The disk file allocation table on Drive A is scanned for chains of allocation units. A new root directory is created for every chain of allocation units. The directory is named as follows:

```
FILExxxx.REC
```

Here, xxxx is a sequential number (0001~). It points to one of the recovered files.

The second format of the RECOVER should be used only if the directory of the disk has become unusable. RECOVER assumes that the entire directory is bad, and recovers all files, including files for which there may still have been valid directory entries.

RENAME (or REN)

Internal

Syntax.: `REN[AME][d:][path]filename[.ext]filename [.ext]`

Purpose: Renames a file.

Comments: **RENAME** will alter the name of the file indicated in the first parameter to the name specified in the second parameter. If the second parameter includes drive information it will be ignored.

The abbreviated **REN** may be used for the **RENAME** command. The wildcard characters ? and * are valid in the parameters. You may use a path only with the first filename. After changing its name the file will stay in the same directory.

Example: To rename the file **AUTO** on Drive **B** to **CAR**, enter:

```
RENAME B:AUTO CAR
```

To rename the file **AUTO** on Drive **B** to **AUTO.NEW**, enter:

```
REN B:AUTO *.NEW
```

To rename the file **PROG.COM** in directory **SUB2** on Drive **B** to filename **PROG1.COM**, enter:

```
REN B:\SUB2\PROG.COM PROG1.COM
```

RESTORE (HARD DISK)

External

Syntax: `RESTORE d: [d:] [path] [filename [.ext]] [/S] [/P]`

Purpose: Restores one or more files from disks to a hard disk.

Comments: **RESTORE** is used only with files put on disk with the **BACKUP** command. The backup disk drive is the first parameter. The hard disk file to be restored is the second parameter.

If no path is specified, files are restored to the current directory. If no filename or extension is specified, **RESTORE** will restore all files backed up from the directory.

/S restores backed up files in all sub-directories, as well as files in the specified directory. Sub-directories at all levels beyond the specified directory are included.

/P makes **RESTORE** stop and prompt you before restoring files that are read-only, or that have been modified since they were backed up. When **RESTORE** prompts you, you can choose whether or not to restore the file.

RESTORE will prompt you to insert the backup disk. Be sure to insert the first disk that could contain the file you want to restore. If the file is not on the disk you choose to insert, **RESTORE** will prompt for another disk.

When you execute a **RESTORE** command containing global filename characters, the program will prompt you to insert a new disk once it has restored all the files on the backup disk matching the designated filename.

ERRORLEVEL is set as follows:

- 0 Normal completion
- 1 No files found to restore
- 3 Operation terminated by user
- 4 Operation terminated due to error

Use these codes with the IF subcommand in batch processing to control error level processing.

Examples: You can use global filename characters to restore all the files matching the filename. The following command restores all files from the backup disk having an extension of .XOT and backed up from the current directory:

```
RESTORE A: C: *.XOT
```

This example shows how to restore all files on the backup disk to hard disk drive C:

```
RESTORE A: C: /S
```

The following set of commands will restore two different files from the backup disk to the default hard disk drive:

```
RESTORE A: MARY\GAMES.CAR
RESTORE A: MARY\SCHED.JUL
```

RMDIR (Remove Directory)

Syntax: RMDIR [d:] path

Internal

or

RD [d:] path

Purpose: Deletes a sub-directory from the named disk.

Comments: First, empty the directory except for the "." and ".." entries. The removed directory will be the last directory named in the path.

Examples: The following command deletes the entry for SUB3 from directory SUB2:

```
RD B:\SUB2\SUB3
```

NOTE: You cannot remove the root directory and the current directory.

SET (Set Environment)

Internal

Syntax: SET [*name*=[*parameter*]]

Purpose: Inserts strings into the command processor's environment. All the strings in the environment are available to all commands and applications.

Comments: This command is meaningful only if you want to set values that will be used by programs you have written. An application program can check all values that have been set with the SET command by issuing SET with no options. For example, SET TTY=VT52 sets your TTY value to VT52 until you change it with another SET command.

The SET command can also be used in batch processing. In this way, you can define your replaceable parameters with names instead of numbers. If your batch file contains the statement "LINK %FILE%", you can set the name that DOS will use for that variable with the SET command. The command SET FILE=DOMORE replaces the %FILE% parameter with the filename DOMORE. Therefore, you do not need to edit each batch file to change the replaceable parameter names. Note that when you use text (instead of numbers) as replaceable parameters, the name must be ended by a percent sign.

Examples: The following command adds the string STR=abc to the other strings in the environment. The lowercase str is converted to uppercase. Using SET you can enter keywords and parameters that are not meaningful to DOS, but can be found and interpreted by applications that examine the environment.

```
SET str=abc
```

The following command adds the string ABCD=\ACCTS2 to the environment. An application program can search the environment for the name ABCD, and use the supplied parameter as the directory name to use for its files.

```
SET ABCD=\ACCTS2
```

The following command removes ABCD = ACCTS2 from the environment:

```
SET ABCD=
```

NOTE: You do not need the SET command to add either PROMPT or PATH commands to the environment. DOS adds these two commands to the environment when you enter them.

When DOS starts up it places the first string in the environment. This string is a COMSPEC=parameter which describes the path that DOS uses to reload the command processor if necessary.

The message "Out of environment space" appears if you issue a SET command that would cause the combined environment strings to exceed 127 bytes. (If you have not loaded a resident program DOS expands the environment string area to hold additional strings.)

SORT (Filter)

External

Syntax: SORT [/R] [/+n]

Purpose: Reads data from the standard input device, sorts the data, and then writes the data to the standard output device.

Comments: The sort uses the ASCII collating sequence. The blanks are not included for Tab characters.

/R Reverses the sort. (Make "Z" come before "A".)

/+n An integer that starts the sort with column *n*. The default is column 1. The maximum size for a file that can be sorted is 63K.

Examples: SORT/R <PRESORT.DAT >SORT.DAT

The command line reads the file PRESORT.DAT, reverses the sort, then writes the output to the file SORT.DAT.

A>DIR| SORT/+14

The output of the directory command is piped to the SORT filter. The SORT filter sorts with column 14, which is the column the file size starts in. A directory sorted by file size is created. The output is to the console.

Since SORT creates a temporary file on the disk, the disk must not be write-protected. The temporary file is deleted when the operation finishes.

SYS (System)

External

Syntax: SYS *d:*

Purpose: Transfers the DOS system files from the disk in the default drive to the disk in the drive specified by *d*:

Comments: The directory of the disk in drive *d*: must be totally empty, or the disk must have been formatted by a `FORMAT d:/S` or `FORMAT d:/B` command to contain directory entries for the DOS files.

NOTE: SYS is used to transfer DOS files to application program disks which are sold without the DOS files. The space required for these files has already been allocated and the SYS command transfers the files to the disk.

DOS COMMAND

TIME

Internal

Syntax: TIME [*hh:mm:ss.xx*]

Purpose: Sets and displays the time. The time is recorded in the directory entry for any files you create or alter.

Comments: Any valid time is accepted, and the system prompt appears. If an invalid entry is made, the TIME command issues the following prompt:

```
Current time is hh:mm:ss.xx
Enter new time:..
```

You now can enter the current time, where:

hh is a number from 0-23 (hours)

mm is a number from 0-59 (minutes)

ss is a number from 0-59 (seconds)

xx is a number from 0-99 (hundredths of a second)

The date is entered from the keyboard or from a batch file.

NOTE: To leave the time unchanged, press <ENTER>.

If you enter incomplete information, the remaining fields are set to zero.

As long as the digits are within the defined ranges the time will be considered valid.

DOS COMMAND

The valid separators within the time are the colon (:) separating the hours, minutes, and seconds, and the period (.) separating the seconds and the hundredths of a second.

An "Invalid time" message is displayed if you enter an invalid time or delimiter.

TREE (Display Directory)

External

Syntax: TREE [d:][/F]

Purpose: Displays the directory paths found on the specified drive. Lists the files in each sub-directory if you enter the optional /F.

Comments: The default drive is assumed if no drive is entered.

Displays the full path name, along with the names of any directories defined within it.

Option:

/F The names of all files in each sub-directory also will be displayed.

Examples: TREE B:/F>SEE.DAT

All directories on Drive B are displayed. The output will be written to the file SEE.DAT in Drive B, and will contain the names of all sub-directories and files at each directory.

TYPE

Internal

Syntax: TYPE [d:][path]filename[.ext]

Purpose: Displays the contents of the file on the screen.

Comments: You may examine the contents of a file but cannot modify the file. To modify the file use EDLIN.

The only formatting done is that tab characters are expanded to an eight-character boundary; that is, columns 8, 16, 24, etc.

NOTES: You must specify a filename.

You can redirect the output to a file or the printer. Press <Ctrl> <PrtSc> or <Ctrl> <P> if you want to output the contents of the file.

Text files are legible. Other files, such as object program files, may appear unreadable due to nonalphabetic or nonnumeric characters.

Wildcard characters * and ? are NOT allowed in the filename or extension. The message "File not found" will appear if wildcard characters are entered.

Example: TYPE B:prog1.tst

The file PROG1.TST on the disk in Drive B is displayed on the screen.

VER (Version)

Internal

Syntax: VER

Purpose: Displays the version number of DOS.

Comments: The DOS version is a major version number (single digit), followed by a period, followed by a minor revision level (two digit).

VERIFY

Internal

Syntax: VERIFY [ON|OFF]

Purpose: Turns the verify switch on or off when writing to disk.

Comments: VERIFY ON stays on until it is turned off. When ON, DOS will perform a verify operation after every disk-write operation. The system will run more slowly when programs write data to the disk because the verification requires extra time.

DOS shows the current state (off or on) of the verify option when you enter VERIFY with no parameters.

Examples: VERIFY ON

Turns the verify feature on.

```
A>VERIFY
VERIFY is on
```

```
A>
```

Show the current status of the VERIFY command.

VOL (Volume)

Internal

Syntax: VOL [d:]

Purpose: Shows the disk volume identification of the named drive.

Comments: The default drive is assumed if a drive is not indicated.

Example:

```
A>VOL
```

```
Volume in drive A is DISKABC
```

```
A>
```

DOS COMMAND

CHAPTER 8

BATCH PROCESSING

1. INTRODUCTION	8-2
2. THE AUTOEXEC.BAT FILE	8-4
3. CREATING A .BAT FILE	8-5
4. EXECUTING A .BAT FILE	8-7
5. TABLE OF BATCH COMMANDS	8-8
ECHO	8-9
FOR	8-11
GOTO	8-12
IF	8-13
PAUSE	8-16
REM	8-17
SHIFT	8-18

INTRODUCTION

Often you have to type the same sequence of commands over and over to perform a commonly used task. A special file (batch file) can be created to execute the entire sequence simply by typing the name of the batch file. The commands in batch files are processed as if they were typed at a terminal. A batch file must be named with the .BAT extension, and is executed by typing the filename without its extension.

BATCH

Batch processing is useful for executing several DOS commands with one batch command. For example, a batch file (DONEW.BAT) to format and check a new disk might look like this:

```
1: REM This file checks new disks
2: REM It is named DONEW.BAT
3: FORMAT B:
4: CHKDSK B:
```

To execute this .BAT file, simply type the filename without the BAT extension:

DONEW

The result is the same as if each of the lines in the .BAT file was entered at the terminal as an individual command.

Helpful Information for Using Batch Files

1. The filename extension (.BAT) should not be entered to execute a Batch file.
2. You may set up the last command in a batch file to be the name of another batch file. This means that you can call one batch file from another after the first has completed.
3. All commands in the batch file are run.
4. Pressing <Ctrl> <C> or <Ctrl> <Break> while in the batch mode displays the following prompt:

Terminate batch job (Y/N)?

Pressing Y will ignore the remainder of the commands in the batch file. The system prompt will appear.

Pressing N will stop only the current command and batch processing will continue with the next command in the file.

5. If a disk containing a batch file that is being run is removed, DOS will prompt you to insert the disk again before the next command may be processed.

BATCH

THE AUTOEXEC.BAT FILE

The **AUTOEXEC.BAT** file is a batch file that is automatically executed each time you start the system. An **AUTOEXEC.BAT** file allows you to automatically execute programs when you start DOS. When you start or restart DOS, the command processor searches the root directory for a file named **AUTOEXEC.BAT**.

If DOS finds the **AUTOEXEC.BAT** file, the file is automatically executed and the date and time prompts are bypassed.

If DOS does not find an **AUTOEXEC.BAT** file when you first load the DOS disk, then the date and time prompts will be issued.

NOTE: If you use an **AUTOEXEC.BAT** file, DOS will not prompt you for a current date and time unless you include the **DATE** and **TIME** commands in the **AUTOEXEC.BAT** file. You should include these two commands in your **AUTOEXEC.BAT** file, since DOS uses this information to keep your directory current.

BATCH

CREATING A .BAT FILE

To create a batch file, use **EDLIN** (the Line Editor) or the **COPY** command.

If, for example, you want to automatically load **BASIC** and run a program called **TEST** each time you start DOS, you could create an **AUTOEXEC.BAT** file as follows.

First, enter:

```
COPY CON: AUTOEXEC.BAT
```

This command tells DOS to copy the information from the console (keyboard) into the **AUTOEXEC.BAT** file. The **AUTOEXEC.BAT** file must be created in the root directory of your DOS disk.

Now enter:

```
BASIC TEST
```

This is the first statement in your **AUTOEXEC.BAT** file. Whenever DOS is started it will load **BASIC** and run the **TEST** program.

Now press **<Ctrl> <Z>** and then the **<ENTER>** key to indicate the completion of the **AUTOEXEC.BAT** file.

The **TEST** program now will automatically be run whenever you start DOS.

You can enter any DOS command or series of commands in the **AUTOEXEC.BAT** file. To run your own **BASIC** program, enter the name of your program instead of **TEST** in the second line of the example.

BATCH

Creating a .BAT File with Replaceable Parameters

You may want to create an application program and run it with different sets of data. This data may be stored in various DOS files.

You can create a batch (.BAT) file with dummy (replaceable) parameters. These parameters, named %0-%9, can be replaced by values supplied when the batch file executes. For example:

```
COPY CON: SAMPLE.BAT
COPY %1.ABC %2.ABC
TYPE %2.TXT
TYPE %0.BAT
```

Press <Ctrl><Z> and then press <ENTER>.

DOS responds with this message:

```
1 File(s) copied
A>_
```

The file SAMPLE.BAT, which consists of three commands, is now on the disk in the default drive.

When you execute the file, the dummy parameters %0, %1 and %2 are replaced sequentially by the parameters you supply. The dummy parameter %0 is reserved for the drive designator, if specified, and the filename of the batch file (for example, SAMPLE).

NOTES: Up to 10 dummy parameters (%0-%9) can be specified. See the SHIFT command in this chapter if you wish to specify more than 10 parameters.

To use the percent sign as part of a filename within a batch file, you must type it twice. For example, to refer to the file FLE%.EXE, you must type it as FLE%%.EXE in the batch file.

EXECUTING A .BAT FILE

To execute the batch file SAMPLE.BAT and to specify the parameters that will replace the dummy parameters, enter the batch filename (without its extension) followed by the parameters you want DOS to substitute for %1, %2, etc.

The file SAMPLE.BAT consists of 3 lines:

```
COPY %1.ABC %2.ABC
TYPE %2.TXT
TYPE %0.BAT
```

To execute the SAMPLE batch file, type:

```
SAMPLE A:APROG B:BPROG
```

As the file executes, SAMPLE is substituted for %0, A:APROG is substituted for %1, and B:BPROG is substituted for %2.

The result is the same as if you had typed each of the commands in SAMPLE with their parameters, as follows:

```
COPY A:APROG.ABC B:BPROG.ABC
TYPE B:BPROG.TXT
TYPE SAMPLE.BAT
```

Notice that the dummy parameter %0 is reserved. It is automatically replaced by the drive designator (if specified) and the filename of the batch file.

TABLE OF BATCH COMMANDS

Command	Purpose	Syntax
ECHO	Allows or inhibits screen display of DOS commands.	ECHO [ON OFF <i>message</i>]
FOR	Iterative execution of DOS commands.	FOR %% <i>variable</i> IN (<i>set</i>) DO <i>command</i>
GOTO	Transfer control through use of labels.	GOTO <i>label</i>
IF	Conditional processing of DOS commands.	IF [NOT] <i>condition</i> <i>command</i>
PAUSE	Suspends execution of DOS commands.	PAUSE [<i>remark</i>]
REM	Displays comments during batch file execution.	REM [<i>remark</i>]
SHIFT	Provides access to more than 10 variables.	SHIFT

ECHO

Syntax: ECHO [ON|OFF|*message*]

Purpose: To allow or inhibit the screen display of the DOS commands executed from a batch file. It does not interfere with messages produced during the execution of the commands.

Comments: ECHO is ON after you have powered-on or reset the system and shows all the commands on the screen as they run. ECHO OFF stops the showing of commands on the screen, including the REM command.

The ECHO message option will show messages on the screen whether or not the current state is ON or OFF. Specific messages will display even when ECHO has been turned off.

The current ECHO state of ON or OFF is displayed if ECHO is issued with no parameters.

Example: If the batch file contains the following:

```
echo off
rem **** the echo is now off
dir b:/w
echo on
dir b:/w
```


When the batch file executes, something like this will be displayed:

```
A > echo off
Volume on drive B has no label
Directory of B:\

filename1.ext  filename2.ext
2 file(s) xxxxx bytes free

A > dir b:/w

Volume in drive B has no label
Directory of B:\

filename1.ext  filename2.ext

2 file(s) xxxxx bytes free
```

The result of using "echo off" in this example is that "echo off" is displayed, the commands "rem" and "dir b:/w" are not displayed, but the output of the dir command is still displayed.

FOR

Syntax: **FOR** %%*variable* IN (*set*) DO *command*

Purpose: To allow iterative execution of DOS commands.

Comments: Each member of the *set* is sequentially substituted in the command which follows DO. Wildcard * or ? characters are valid for members of the set and all matching filenames on the disk will be substituted in the command. Path names are not accepted with filenames in the *set*.

Only one FOR command can be specified on a command line.

Example: If you enter the command:

```
FOR %%f IN (myprog.asm acct.bas acct2.bas) DO dir %%f
```

The result is:

```
dir myprog.asm
dir acct.bas
dir acct2.bas
```


GOTO

Syntax: `GOTO label`

Purpose: Transfers control to the line following a label. A label is inserted in a batch file as a colon (:) followed by the label name.

Comments: The GOTO label causes the commands on the line immediately after *label* to process. If *label* is not named, the current batch file ends with the following message:

Label not found.

To set up a label in a batch file use a colon (:) followed by a character string. The first eight characters of the string are considered the significant characters. The first eight characters must be different than those of all the other labels.

Example: The following batch file sets up an infinite loop of messages on the screen. Your screen will display "rem this is a loop.. "and" **GOTO MYLABEL**".

```
:MYLABEL  
rem this is a loop..  
GOTO MYLABEL
```

Labels within a batch file are not shown while the batch file is executing. In the above example, the line :MYLABEL would not be shown. Therefore, unreferenced labels may be used to place comments within your batch file. They are not displayed when the file is run.

IF

Syntax: `IF [NOT] condition command`

Purpose: Allows conditional processing of DOS commands.

Comments: The DOS *command* will execute when the IF *condition* is true. When the *condition* is false the DOS *command* will be bypassed and the next command in the file will execute.

The *condition* is one of the following:

`EXIST filespec`

If *filespec* is on the indicated drive, EXIST *filespec* is true. Path names are not allowed with the *filespec* when using the EXIST condition variable.

`string1==string2`

When *string1* and *string2* are identical, this condition is true.

`ERRORLEVEL number`

If the previous program had an exit code (specified as a binary value) of *number* or higher, ERRORLEVEL *number* is true. Your own programs may set an error code that can then be checked by the IF ERRORLEVEL command.

Example: IF EXIST filespec command:

```
if exist myfile goto xyz
dir b:
.
.
:xyz
command
```

The batch file with this command assuming "myfile" is named as the %1 parameter would exit to xyz provided "myfile" is on the default drive. The command following the label ":xyz" would be executed. If "myfile" is not found, the "goto xyz" would not be executed and processing would continue with the "dir b:" command in the batch file.

IF string1==string2 command:

```
if %1==Jamie echo Jamie ate it.
```

The batch file with this command, assuming Jamie is named as the %1 parameter, would perform the ECHO batch command. Jamie ate it, would be displayed. Mary, given as the %1 parameter, would make the condition false and the ECHO batch command would not be executed. The next command would then be processed.

IF ERRORLEVEL number command:

```
tstprog
if errorlevel 1 echo tstprog failure.
dir b:
```

Assume that the above commands are in a batch file. TSTPROG is a program that sets the errorlevel when it ends. Assume that TSTPROG sets the errorlevel to 0 if it finishes successfully and sets the errorlevel to 1 if it fails. The batch file conditional if errorlevel 1 echo... tests for the situation when TSTPROG failed. If TSTPROG failed, the condition is true and the ECHO batch command is processed displaying the message immediately following the echo command. If TSTPROG finished successfully, the condition is false and the ECHO batch command would not execute. The next command in the batch file, dir b:, would be processed.

IF NOT EXIST filespec command:

```
if not exist a:%1 copy b:%1 a:
myprog1
```

This IF batch command demonstrates the NOT condition. The batch file needs a certain file to be on Drive A. The IF condition is executed before MYPROG1 to ensure that the needed file is on Drive A. If the file is not on Drive A, the condition is true and the copy command will be processed, copying the file from Drive B to Drive A. If the file is on Drive A, the copy will not be processed. MYPROG1 will then be run.

PAUSE

Syntax: PAUSE [*remark*]

Purpose: Suspends the execution of a batch file and displays the message "Strike a key when ready..."

Comments: PAUSE commands within a batch file give you the chance to display messages, change disks between commands, etc. Press any key (except <Ctrl><Break>) to continue execution of the batch file. <Ctrl> <Break> will end all execution.

The optional remark may also be shown. The optional remark is a string of characters no longer than 121 bytes.

The PAUSE command, inserted at strategic points in your file allows you to decide how much of a batch file to execute. At each PAUSE command, you have time to decide either to stop processing (by pressing <Ctrl><Break>), or to continue processing (by pressing any other key).

Example: This PAUSE command is in a batch file and this message is shown:

```
A >PAUSE Insert new disk in drive B
Strike a key when ready....
```

Because the system stops, you have time to change disks between commands.

REM (Remark)

Syntax: REM [*remark*]

Purpose: Displays comments from within a batch file or provides spacing.

Comments: The *remarks* are shown as the batch processing arrives at the REM command.

Remarks may be any character string no longer than 123 bytes.

The REM command used without remarks, provides spacing and enhances readability within your batch file.

Example: The REM command contained in a batch file displays this:

```
REM This is the checkbook update program.
```


SHIFT

Syntax: SHIFT

Purpose: Allows command lines to access more than 10 (%0 through %9) replaceable variables.

Comments: Changeable variables are numbered %0 through %9. These numbers represent relative positioning in a parameter list. To use more than 10 parameters on a command line, execute the SHIFT command to get past the tenth parameter. The parameters on the command line will be shifted one position to the left; the parameter in %1 position moves to the %0 position. Each shift command moves all the parameters to the left by one more position. For example:

```
%0=Apples
%1=Bananna
%2=Carrots
%3=Dragons
.
.
%9
```

The SHIFT results are:

```
%0=Bananna
%1=Carrots
%2=Dragons
.
.
%9
```

Example: This example illustrates the use of the SHIFT command in a batch file. A batch file, CHECK.BAT contains the following commands.

```
echo %0 %1 %2 %3
shift
echo %0 %1 %2 %3
shift
echo %0 %1 %2 %3
shift
echo %0 %1 %2 %3
shift
echo %0 %1 %2 %3
```

Call the batch file with these parameters:

```
CHECK PROGA PROGB PROGC
```

These results are produced:

```
A>echo CHECK PROGA PROGB PROGC
CHECK PROGA PROGB PROGC

A>shift

A>echo PROGA PROGB PROGC
PROGA PROGB PROGC

A>shift

A>echo PROGB PROGC
PROGB PROGC

A>shift

A>echo PROGC
PROGC

A>shift

A>echo
A>
```


MEMO

BATCH

CHAPTER 9

EDLIN

1. INTRODUCTION	9-2
2. HOW TO START USING EDLIN	9-3
3. SPECIAL EDITING KEYS	9-5
4. EDLIN COMMAND INFORMATION	9-16
5. PARAMETERS	9-18
6. TABLE OF EDLIN COMMANDS	9-20
7. COMMAND DESCRIPTIONS	9-21
Append	9-21
Copy	9-22
Delete	9-24
Edit	9-26
End	9-28
Insert	9-29
List	9-32
Move	9-35
Page	9-36
Quit	9-37
Replace	9-38
Search	9-41
Transfer	9-44
Write	9-45

EDLIN

INTRODUCTION

EDLIN is the line editor program used to create, change, and display source program files or text files.

EDLIN allows you to:

Create new source or text files and save them.

Update existing files and save both the updated and original files.

Delete, edit, insert, and display lines.

Search for, delete, or replace text within one or more lines.

You can edit files one line at a time, with up to 253 characters in each line.

Line numbers are generated and displayed by EDLIN during the editing process, but are not actually present in the saved file. When you insert or delete lines in a file, all line numbers following the inserted or deleted text are automatically adjusted by the number of lines inserted or deleted. As a result, lines always remain consecutively numbered in your file.

EDLIN

HOW TO START USING EDLIN

To start EDLIN, type:

```
EDLIN [d:] [path] filename [.ext] [/B]
```

The filename is the name of the file you want to create or edit. The screen will now display a message and prompt. The prompt for EDLIN is an asterisk (*).

The system will not allow you to edit a file with a filename extension of .BAK as EDLIN assumes that any .BAK file is a backup file. To edit a file with a filename extension of .BAK, you must rename the file to another extension, then start EDLIN and specify the new name.

To edit a new file:

Enter the name of the file you want to create. If EDLIN does not find this filename on a drive, it will create a new file with Command Description section for more information about how to use the Insert Command

```
New file
```

```
*_
```

Enter an I. This is the Insert command which allows you to insert lines of text into your new file. Refer to the EDLIN Command section for more information about how to use the Insert Command.

To edit an existing file:

Enter the name of the file you want to edit. When EDLIN finds the file you specify on the designated or default drive, the file is loaded into memory.

If the entire file can be loaded, the following message and prompt will be displayed on your screen:

```
End of input file
```

```
*_
```

EDLIN

You can then edit the file using EDLIN editing commands. Refer to the EDLIN Command Description section for more information about EDLIN editing commands.

If the file is too large to be loaded into memory, EDLIN will load lines until memory is 75% full. The * prompt will be displayed on your screen:

*_

You can then edit the portion of the file that is in memory.

To edit the remainder of the file, you must save some of the edited lines on disk to free memory. EDLIN can then load the unedited lines from disk into memory. Refer to the EDLIN Command Description section for information about how to use the Write and Append Commands.

When you complete the editing session:

Use the End command to save the original and the updated (new) files, or use the Quit command to exit the editing session without saving the updated files. Refer to the EDLIN Command Description section for more information about how to use the End and Quit commands.

If you use the End command, the original file is renamed with an extension of .BAK and the new file has the filename and extension you specified in the EDLIN command.

SPECIAL EDITING KEYS

You can recall an entire command line or modify it with the DOS special editing keys. The last command line you type is automatically placed in a special storage area called a template. The DOS special editing keys are used to edit your command lines. By using the special editing keys, you can:

Repeat a command line.

Correct a mistake in the command line without retyping the entire command line.

Edit and execute a command line which is similar to a preceding command line with a minimum of typing.

The following table summarizes the special editing keys.

Key	Description
<F2>	Repeat the current input line.
<F3>	Repeat the current input line, but with the current input line replaced by the template.
<F4>	Repeat the current input line, but with the current input line replaced by the template, and then execute the command.
<F5>	Repeat the current input line, but with the current input line replaced by the template, and then execute the command, and then repeat the command.
<F6>	Repeat the current input line, but with the current input line replaced by the template, and then execute the command, and then repeat the command, and then repeat the command.
<F7>	Repeat the current input line, but with the current input line replaced by the template, and then execute the command, and then repeat the command, and then repeat the command, and then repeat the command.
<F8>	Repeat the current input line, but with the current input line replaced by the template, and then execute the command, and then repeat the command, and then repeat the command, and then repeat the command, and then repeat the command.
<F9>	Repeat the current input line, but with the current input line replaced by the template, and then execute the command, and then repeat the command, and then repeat the command, and then repeat the command, and then repeat the command, and then repeat the command.
<F10>	Repeat the current input line, but with the current input line replaced by the template, and then execute the command, and then repeat the command, and then repeat the command, and then repeat the command, and then repeat the command, and then repeat the command, and then repeat the command.

Special Editing Keys

Function	Key	Description
Copy one character	<F1>	Copies one character from the template to the command line.
Copy up to	<F2>	Copies all characters from the template to the command line, up to the character specified.
Copy template	<F3>	Copies all remaining characters in the template to the command line.
Skip one character		Skips (does not copy) a character.
Skip up to	<F4>	Skips (does not copy) the characters in the template, up to the character specified.
Void input	<Esc>	voids the current input; leaves the template unchanged.
Insert mode	<Ins>	Enters or exits insert mode. Allows characters to be inserted into a line.
New template	<F5>	Makes the new command line the new template, but the command line is not executed.

Key: <F1>

Purpose: Copies one character from the template to the command line.

Comments: Pressing the <F1> key copies one character from the template (the top line) to the command line (the bottom line). When the <F1> key is pressed, one character is inserted in the command line. Insert mode is automatically turned off.

Example: Assume that the screen shows:

```
1:*This is a sample file.
1:*_
```

At the beginning of the editing session, the cursor (indicated by the underline) is positioned at the beginning of the command line. Pressing the <F1> key copies the first character (T) from the template to the command line.

```
1:*This is a sample file.
<F1> 1:*T_
```

Each time the <F1> key is pressed, one more character appears:

```
<F1> 1:*Th_
<F1> 1:*Thi_
<F1> 1:*This_
```


Key: <F2>

Purpose: Copies multiple characters up to a specified character, from the template to the command line.

Comments: Pressing the <F2> key copies all characters up to a specified character from the template to the command line. The specified character is the next character typed after <F2>. This character is not copied or displayed on the screen.

Pressing the <F2> key causes the cursor to move to the location of the specified character. If the template does not contain that character, nothing is copied. Insert mode is automatically turned off.

Example: Assume that the screen shows:

```
1:*This is a sample file.  
1:*_
```

At the beginning of the editing session, the cursor (indicated by the underline) is positioned at the beginning of the line. Pressing the <F2> key copies all characters up to the specified character.

```
1:*This is a sample file.  
<F2>p 1:*This is a sam_
```

Key: <F3>

Purpose: Copies all remaining characters from the template to the command line.

Comments: Pressing the <F3> key copies all remaining characters from the template to the command line. Regardless of the cursor position at the time the <F3> key is pressed, the rest of the command line appears. The cursor is positioned after the last character on the line. Insert mode is then automatically turned off.

Example: Assume that the screen shows:

```
1:*This is a sample file.  
1:*_
```

At the beginning of the editing session, the cursor (indicated by the underline) is positioned at the beginning of the line. Pressing the <F3> key copies all characters from the template to the command line.

```
1:*This is a sample file. (template)  
<F3> 1:*This is a sample file_ (command line)
```


Key:

Purpose: Skips (does not copy) one character in the template.

Comments: Pressing the key skips one character in the template. Each time you press the key, one character is not copied from the template. When the edited command line is entered, the characters not copied from the template are deleted.

This key is the opposite of the <F1> key.

Example: Assume that the screen shows:

```
1:*This is a sample file.  
1:*_
```

At the beginning of the editing session, the cursor (indicated by the underline) is positioned at the beginning of the line. Pressing the key skips the first character (T).

```
<Del> 1:*This is a sample file.  
1:*_
```

The cursor position does not change. Only the template is affected. To see how much of the line has been skipped, press the <F3> key, which copies all remaining characters from the template to the command line.

```
<Del> 1:*This is a sample file.  
1:*_   
<F3> 1:*his is a sample file._
```

Key: <F4>

Purpose: Skips (does not copy) multiple characters in the template up to a specified character.

Comments: Pressing the <F4> key skips all characters up to a specified character in the template. If the template does not contain the specified character, nothing is skipped over. When the edited command line is entered, the characters not copied from the template are deleted.

This key is the opposite of the <F2> key.

Example: Assume that the screen shows:

```
1:*This is a sample file.  
1:*_
```

At the beginning of the editing session, the cursor (indicated by the underline) is positioned at the beginning of the line. Pressing the <F4> key skips all characters in the template up to the specified character.

```
1:*This is a sample file.  
<F4>p 1:*_
```

The cursor position does not change. To see how much of the line has been skipped, press the <F3> key, which copies all remaining characters from the template to the command line.

```
1:*This is a sample file.  
<F4>p 1:*_   
<F3> 1:*ple file._
```


Key: <Esc>

Purpose: Voids all input and empties the command line.

Comments: Pressing the <Esc> key empties the command line. The template remains unchanged.

<Esc> also prints a back slash (\), moves the cursor to the first position of the next line, and turns insert mode off. Pressing the <F3> key copies the original template to the command line again. The command line now displays the original template.

Example: Assume that the screen shows:

```
1:*This is a sample file.
1:*_
```

At the beginning of the editing session, the cursor (indicated by the underline) is positioned at the beginning of the line. Assume that you want to replace the line with "New Information".

```
1:*This is a sample file.
1:*New Information_
```

To cancel the line you just entered (New Line), and to keep "This is a sample file.", press <Esc>. Notice that a backslash appears on the New Information line to tell you it has been cancelled.

```
1:*This is a sample file
<Esc> 1:*New Information\
-
```

Press <ENTER> to keep the original line, or to perform any other editing functions. If <F3> is pressed, the original template is copied to the command line:

```
<F3> This is a sample file._
```

Key: <Ins>

Purpose: Enters insert mode or enters replace mode.

Comments: When you are not in the insert mode, any characters you type replace characters in the template. Replace mode is in effect when you begin to edit a line. Pressing the <Ins> key causes EDLIN to enter insert mode. To return to replace mode, simply press <Ins> again.

Insert mode:

The current cursor position in the template does not change. Characters are inserted *ahead* of the character the cursor points to. The cursor moves as each character is inserted. When you have finished inserting characters, the cursor will be in its original position in the template.

Replace mode:

All the characters you type will overstrike and replace characters in the template. If the <ENTER> key is pressed, the remainder of the template will be deleted.

Example: For an example of inserting text, assume that the screen shows:

```
1:*This is a sample file.
1:*_
```

At the beginning of the editing session, the cursor (indicated by the underline) is positioned at the beginning of the line. Assume that you press the <F2> and f keys:

```
1:*This is a sample file.
<F2>f 1:*This is a sample _
```


To insert the word "edit" and a space, press <Ins> and enter "edit":

```
1:*This is a sample file.
1:*This is a sample _
<F2>f 1:*This is a sample edit _
<Ins>edit <SPACE>
```

If you now press the <F3> key, the rest of the template is copied to the line:

```
1:*This is a sample edit
<F3> 1:*This is a sample edit file._
```

To exit insert mode, simply press the <Ins> key again.

For an example of replacing text, assume that the screen shows:

```
1:*This is a sample file.
1:*_
```

At the beginning of the editing session, the cursor (indicated by the underline) is positioned at the beginning of the line. Assume that you then press <F2>m, lary, and then <F3>:

```
1:*This is a sample file.
<F2>m 1:*This is a sa_
lary 1:*This is a salary_
<F3> 1:*This is a salary file._
```

Notice that you *replaced* "mple" with "lary."

Key: <F5>

Purpose: Creates a new template.

Comments: Pressing the <F5> key makes the current command line the new template, replacing the contents of the old template. An @ ("at sign" character) is displayed at the end of the line. The command line is now empty and insert mode is turned off.

NOTE: <F5> performs the same function as the <ESC> key, except that the template is changed and an @ ("at sign" character) is printed instead of a \ (backslash).

Example: Assume that the screen shows:

```
1:*This is a sample file.
1:*_
```

At the beginning of the editing session, the cursor (indicated by the underline) is positioned at the beginning of the line. Assume that you then press <F2>m, lary, and then <F3>:

```
1:*This is a sample file.
<F2>m 1:*This is a sa_
lary 1:*This is a salary_
<F3> 1:*This is a salary file._
```

At this point, assume that you want this line to be the new template, so you press the <F5> key:

```
<F5> 1:*This is a salary file.@
```

The @ indicates that this new line is now the new template. Additional editing can be done using the new template.

EDLIN COMMAND INFORMATION

EDLIN commands perform editing functions on lines of text.

Helpful information for using EDLIN commands:

1. The EDLIN prompt is an asterisk (*).
2. All commands are a single letter except the Edit Line command. Commands and string parameters may be uppercase or lowercase, or a combination of both.
3. Delimiters (spaces and commas) are only required between two adjacent line numbers. For example, to delete line 6, the command 6D is the same as 6 D.
4. Commands execute after you press <ENTER>.
5. Stop commands by pressing <Ctrl> <Break>.
6. Suspend the display by pressing <Ctrl> <NumLock>. To restart the display press any other character.
7. Use the editing keys for editing *within a line* and EDLIN commands for editing on *entire lines*.
8. You can reference line numbers relative to the current line (the line with the asterisk).
Use a minus sign with a number to indicate lines before the current line. Use a plus sign with a number to indicate lines after the current line. For example:

-10, +10L

Lists 10 lines before the current line, the current line, and 10 lines after the current line.

9. Multiple commands may be issued on one command line, without any special separators, unless you issue a command to edit a single line using a line number. In that case, a semicolon (;) must separate commands. The string of a Search or Replace command may be ended by a <Ctrl> <Z> instead of a <ENTER>. For example:

15;-5,+5L

Edits line 15 and then lists lines 10 through 20 on the screen.

SThis string<Ctrl><Z>-5,+5L

Searches for "This string" and then displays 5 lines before and 5 lines after the line containing the matched string.

10. You can insert a control character into text by using <Ctrl> <V>. Pressing <Ctrl> <V> tells DOS to recognize the next *capital* letter typed as a control character.

S<Ctrl><V>Z

Will find the first occurrence of CONTROL-Z in a file.

You can use a control character in any of the string arguments of Search or Replace by using <Ctrl><V>. For example:

R<Ctrl><V>C<Ctrl><Z>xyz

Will replace all occurrences of Ctrl-C in a file by xyz.

It is possible to insert <Ctrl><V> into the text by typing Ctrl-V-V.

11. The Ctrl-Z character normally means "end-of-file". If you have Ctrl-Z characters elsewhere in your file, you must tell EDLIN that these other control characters do not mean end-of-file.
Use the /B switch to tell EDLIN to ignore any Ctrl-Z characters in the file and to show you the entire file.

PARAMETERS

Parameter	Description
<i>line</i>	<p>Indicates a line number must be typed. Line numbers must be separated by a comma or a space. A comma or space must also be used to separate line numbers from other options and from the command.</p> <p>The line number may be specified one of three ways:</p> <ol style="list-style-type: none">1. Enter a number. You may enter an integer from 1-65529. If you enter a number larger than the largest existing line number the line will be given the next consecutive line number.2. Enter a period. A period (.) indicates the current line number. The current line is the last line edited, and is not necessarily the last line displayed. The current line is marked on your screen by an asterisk (*) between the line number and the first character.3. Enter a pound sign. A pound sign (#) indicates the line after the last line number. This entry has the same effect as entering a number larger than the last line number. *
<i>n</i>	<p>Indicates when you must specify the number of lines. Use this parameter with the Write and Append commands only.</p>

string *String* indicates an entry of one or more characters representing text to be found, replaced, deleted or to replace other text. Used only with the Replace and Search commands.

Each *string* must be ended by a <Ctrl> <Z> or an <ENTER> (see the Replace command for more information).

No spaces should be left between strings or between a string and its command letter, unless you want those spaces to be part of the string.

TABLE OF EDLIN COMMANDS

EDLIN commands are summarized in the following table. They are described in further detail in following the table.

EDLIN Commands

Command	Purpose
<i>line</i>	Edits line number
A	Appends lines
C	Copies lines
D	Deletes lines
E	Ends editing
I	Inserts lines
L	Lists text
M	Moves lines
P	Pages text
Q	Quits editing
R	Replaces lines
S	Searches text
T	Transfers text
W	Writes lines

EDLIN

COMMAND DESCRIPTIONS

Append

Syntax: [n]A

Purpose: The **Append** command adjusts the file being edited in memory by adding, at the end of the existing lines in memory, the specified number of lines from the disk.

Comments: Edlin puts the maximum number of lines possible into memory for editing, until memory is 75% full. If memory is at 75% capacity, no action occurs.

The **Append** command is important when the file being edited is too large to fit in memory. It allows you to edit the lines remaining on disk.

First, use the **Write** command to write all edited lines in memory to disk. Then, use the **Append** command to load the unedited lines from disk to memory. When the last file is in memory, the screen displays the message "End of input file".

EDLIN

Copy

Syntax: `[line], [line], line [,count]C`

Purpose: The **Copy** command copies a range of lines, placing them just ahead of a specified line number. The lines can be copied more than one time by using the *count* option.

Comments: If you do not enter line numbers in the first or the second *line* parameter, the default is the current line.

If you do not specify a number in *count*, the default is one line. The system then copies the text once.

After each copy, the file is renumbered automatically. The current line is the first of the copied lines.

Do not overlap line numbers or your screen will display an "Entry error" message. For example:

`3,20,15C`

would result in an error message.

Example: The current file is:

- 1:*This sample file is
- 2: used to show copying lines.
- 3: See what happens when you use
- 4: the COPY command
- 5: to copy text in your file.

To copy this file once, issue the following command;

`1,5,6C`

The file now reads:

- 1: This is a sample file
- 2: used to show copying lines.
- 3: See what happens when you use
- 4: the Copy command
- 5: to copy text in your file.
- 6:*This is a sample file
- 7: used to show copying lines.
- 8: See what happens when you use
- 9: the Copy command
- 10: to copy text in your file.

When you copy text within other text, the copied lines will appear *before* the line specified in the third *line* parameter.

The current file is:

- 1:*This file shows
- 2: how to use COPY
- 3: to the middle
- 4: of your file.
- 5: Now you know
- 6: very well.

The command `2,3,6C` results in the following file:

- 1: This file shows
- 2: how to use COPY
- 3: to the middle
- 4: of your file.
- 5: Now you know
- 6:*how to use COPY
- 7: to the middle
- 8: very well.

Delete

Syntax: [line] [,line]D

Purpose: The **Delete** command deletes a specified range of lines in a file.

Comments: When you use this command, lines are permanently deleted. If you want to reinstate the lines, you must use the **Quit** (Q) command to end the edit and begin again.

When lines are deleted, the line immediately after the deleted text becomes the current line. The current line and any following lines will then be automatically renumbered.

If you do not enter a line number in one or both of the *line* parameters, the default is the current line. For example:

,4D results in all lines being deleted from the current line through line 4. You must enter the comma to indicate the first line number is omitted.

4D or 4,D results in only line 4 being deleted. The line that follows then becomes the current line.

D results in only the current line being deleted. The line that follows then becomes the current line.

Examples: The current file is:

- 1:*This is a sample file
- 2: used to show dynamic
- 3: line numbers.
- 4: See what happens when you
- 5: use Delete and Insert.
- 6: Use the D and I commands to
- 7: edit the text
- 8: in your files.

To delete multiple lines, such as lines 5 and 6, type:

5,6D

The result is:

- 1: This is a sample file
- 2: used to show dynamic
- 3: line numbers.
- 4: See what happens when you
- 5:*edit the text
- 6: in your files.

To delete a single line, type:

3D

The result is:

1. This is a sample file
- 2: used to show dynamic
- 3:*See what happens when you
- 4: edit the text
- 5: in your files.

To delete a range of lines beginning with the current line, type:

,4D

The result is:

1. This is a sample file
- 2: used to show dynamic
- 3:*in your files.

Notice that the lines are automatically renumbered whenever the delete command is executed.

Edit

Syntax: [line]

Purpose The **Edit** command allows you to edit a line of text by displaying the line to be edited.

Comments: Enter the line number you want to edit or a period (.) to indicate the current line. If you do not enter a line number or a period, and you press <ENTER>, the line after the current line will be ready to edit.

EDLIN displays the line numbers and the text on the screen. EDLIN also displays the line number again, below the line of text, ready for editing input.

If you press <ENTER> while the cursor is located at any position other than the beginning or end of the line, you will erase the remainder of the line.

You may use any of the EDLIN editing keys to edit the line. The existing text of the line serves as the template until the <ENTER> key is pressed.

If no changes to the current line are needed and the cursor is at the beginning or end of the line, press the <ENTER> key to accept the line as is.

Example: The following file is ready to edit:

```
1:*This is a sample file.  
2: used to show the editing  
3: of line  
4: four.
```

To edit line 4, type:

```
4 <ENTER>
```

The contents of line 4 are now displayed on the screen, preceeded by an asterisk. The line number is repeated below, ready for input.

```
4:*four.  
4:*
```

Using the special editing keys, you could then perform the following edit:

```
<Ins>number <SPACE> 4: number _  
<F3><ENTER>          4: number four.
```

After completing your edit, <ENTER> will save the edited line and make it the current line. If you want, you can type more text after the changed line, as the system is in insert mode when the cursor is at the end of the line.

If you want to do additional editing to the changed line without changing the original line, press <F5>.

If you want to cancel your changes to the line, press <Esc> or <Ctrl> <Break>. The original line will then be unchanged.

End

Syntax: E

Purpose: The **End** command ends the editing session and saves the edited file.

Comments: If the disk does not contain enough free space for the entire file, the write will be aborted. Some or all of the edited file will be lost.

The system will save the edited file by writing it to the drive and filename you specified at the start of EDLIN. If the drive was not selected, the file will be saved on the disk in the default drive.

The original file will be given a .BAK filename extension. If you created a new file instead of updating an existing file, no .BAK file is created.

If you want to end the editing session without saving the edited file, use the **Quit** command.

EDLIN

Insert

Syntax: [*line*]I

Purpose: The **Insert** command inserts text immediately before the specified line. The **Insert** command must be used when you create a new file before you can insert text.

Comments: If you do not enter a line number, or if you enter the line number as a period (.), the insert is made immediately before the current line number.

If the line number you enter is greater than the highest existing line number, or if a pound sign (#) is specified as the line number, the insert is made after the last line in memory. The last line inserted then becomes the current line.

When you are in insert mode, successive line numbers are displayed automatically each time you press <ENTER>.

To exit insert mode enter <Ctrl> <Break>. After you have exited insert, the line immediately following the inserted lines becomes the current line. All line numbers following the inserted section are then automatically renumbered.

Examples: The following file is ready to edit:

- 1: *This is a sample file
- 2: used to show dynamic line numbers.
- 3: See what happens when you use
- 4: Delete and Insert
- 5: (the D and I commands)
- 6: to edit text
- 7: in your file.

EDLIN

To insert text before a specific line that is *not* the current line, such as line 7, enter:

7i

The result is:

7:*

Now, you can enter the new text for line 7:

7:*and renumber lines

To end the insertion, press <Ctrl> <Break> on the *next* line:

8:*<Ctrl><Break>

Now you can use the list command, L, to list the file. the screen displays:

- 1: This is a sample file
- 2: used to show dynamic line numbers.
- 3: See what happens when you use
- 4: Delete and Insert
- 5: (the D and I commands)
- 6: to edit text
- 7: and renumber lines
- 8:*in your file.

To insert lines immediately before the current line, enter:

i

The screen displays:

8:*

To insert the following text and then terminate the insert with a <Ctrl> <Break> on the next line:

8:*so they are consecutive
9:*<Ctrl><Break>

Now you can use the List command, L, to list the file again and see the resulting lines. The screen displays:

- 1: This is a sample file
- 2: used to show dynamic line numbers.
- 3: See what happens when you use
- 4: Delete and Insert
- 5: (the D and I commands)
- 6: to edit text
- 7: and renumber lines
- 8: so they are consecutive
- 9:*in your file.

To add new lines to the end of the file, enter:

10i

The screen displays:

10: *_

Now you can enter the following new lines:

- 10: * The insert command can place new lines
- 11: * in the file.

End the insertion by pressing <Ctrl> <Break> on line 12. The new lines will appear at the end of all previous lines in the file. Now use the List command, L, to see the resulting lines:

- 1: This is a sample file
- 2: used to show dynamic line numbers.
- 3: See what happens when you use
- 4: Delete and Insert
- 5: (the D and I commands)
- 6: to edit text
- 7: and renumber lines
- 8: so they are consecutive
- 9: in your file.
- 10: The insert command can place new lines
- 11: in the file.

List

Syntax: [line] [,line]L

Purpose: The **List** command displays on your screen a specified range of lines. The current line is not changed.

Comments: This command displays file lines from the first line number specified through the second line number specified.

If you do not specify a line number for the first *line* the default is 11 lines before the current line. The beginning comma must be inserted to indicate the omitted first *line*.

If you specify a line number for the first *line* which is more than 11 lines before the current line, the screen display will be the same as if you omitted both *line* parameters.

If you do not specify a line number for the second *line*, the screen displays 23 lines, beginning with the specified *line*.

If you do not specify a line number in either *line* parameter, the screen displays 23 lines: 11 lines before the current line, the current line, and 11 lines after the current line. If there are less than 11 lines before the current line, the screen displays extra lines after the current line, to total 23 lines.

Example: The following file is ready to edit:

```
1: This is a sample file
2: used to show dynamic line numbers.
3: See what happens when you use
4: Delete and Insert
5: (the D and I commands)
.
.
.
15:*This current line contains an asterisk.
.
.
.
26: to edit text
27: in your file.
```

To list a range of lines from line 2 through line 5, enter:

```
2,5L
```

The screen displays:

```
2: used to show dynamic line numbers.
3: See what happens when you use
4: Delete and Insert
5: (the D and I commands)
```

To list a range of lines beginning with the current line through line 26, enter:

```
.26L
```

```
15:*The current line contains an asterisk.
.
.
.
26: to edit text
```


To list a range of 23 lines from a specified line number, enter:

3L

The screen displays:

3: See what happens when you use
4: Delete and Insert
.
.
.
25: This is a way

To list a range of 23 lines centered around the current line, enter:

L

The screen displays:

4: Delete and Insert
5: (the D and I commands)
.
.
.
13: The current line is listed in the middle.
14: The current line remains unchanged.
15:*The current line contains an asterisk.
.
.
.
26: to edit text.

Since you did not specify any line numbers, the screen displayed 11 lines before the current line, the current line, and 11 lines after the current line.

Move

Syntax: [line], [line], lineM
Purpose: The **Move** command moves a range of lines ahead of a specified line in the file being edited.

Comments: This command is used to move a block of text from one location in the file to another. If you do not specify a number in the first or second *line*, the default is the current line. You *must* enter a line number in the third *line*.

The file lines are automatically renumbered after the text is moved. The first of the moved lines becomes the current line.

Example: ,+40,90M

This command moves the text from the current line plus 40 lines to line 90.

30,50,90M

This command moves lines 30-50 to line 90.

If the line numbers overlap, EDLIN will display an "Enter error" message.

EDLIN

EDLIN

Page

Syntax: [line] [,line]P

Purpose: The **Page** command pages through a file displaying a specified range of lines. It also changes the current line.

The **List** command also displays a range of lines but does not change the current line.

Comments: If you do not specify a number in the first *line*, the default is the current line plus one.

If you do not specify a number in the second *line*, the system will display 23 lines.

The last line displayed by the **Page** command, and marked by an asterisk, becomes the new current line.

EDLIN

Quit

Syntax: Q

Purpose: The **Quit** command quits the editing session. This command does *not* save any editing changes. It exits to the DOS operating system.

Comments: After you enter the command, EDLIN displays a prompt on the screen, to ensure you really do not want to save the changes.

If you want to leave the editing session without saving any changes, enter 'Y'. No .BAK file will be created. (Refer to the END command for information about the .BAK file.) Your previous backup copy will no longer exist.

If you want to continue the editing session, enter 'N' or any other character except 'Y'.

Example: *Q
Abort edit (Y/N)?Y
A>_

EDLIN

Replace

Syntax: [*line*] [,*line*] [?] R [*string1*] [<Ctrl><Z>*string2*]

Purpose: The **Replace** command replaces all occurrences of a string of text in the specified range with a different string of text or blanks.

Comments: Each occurrence of *string1* is replaced by *string2*. The lines in which replacements occur will be displayed. When all occurrences of *string1* in the specified range are replaced by *string2*, the **Replace** command terminates and the asterisk prompt reappears. The last line changed becomes the current line.

If you include both strings, the first string must be separated from the second string with a <Ctrl><Z>. The second string *must* be ended with a <Ctrl><Z><ENTER> combination or with just <ENTER>.

If *string1* is omitted, **Replace** will use the old *string1* as its value. If this is the first replace being done in this EDLIN session, then the replacement process will be terminated immediately.

If *string2* is omitted, then *string1* may be ended with <ENTER>. In this case all occurrences of *string1* are deleted.

Default values for the *line* parameters:

If the first *line* is omitted then the first *line* defaults to the line *after* the current line.

If the second *line* is omitted, the second *line* defaults to the last line in memory.

If both *line* parameters are omitted, the replace will occur from the line following the current line to the last line in memory.

The question mark parameter:

If you include the question mark (?), the Replace command will stop at each line that contains *string1* that matches. The screen will display the line and then display the prompt O.K.?

If you press Y or <ENTER>, then *string2* will replace *string1*, and the search for a match will continue.

If you press any key besides Y or <ENTER> after the O.K.? prompt, *string1* will be left as it was in the line.

If *string1* occurs more than once in a line, each occurrence of *string1* will be replaced individually, and the O.K.? prompt will be displayed after each replacement. With this method you can choose to replace particular occurrences of *string1*.

This process will continue until the end of the range of lines or until the end of the file. EDLIN displays the asterisk prompt after the last occurrence of *string1*.

Example: The following file is ready to edit:

- 1:*This is a sample file
- 2: used to show dynamic line numbers.
- 3: See what happens when you use
- 4: Delete and Insert
- 5: (the D and I commands)
- 6: to edit text
- 7: in your file.
- 8: The insert command can place new lines
- 9: in the file.

To replace all occurrences of and with AND in a specified range, type:

```
3,9 Rand<Ctrl><Z>AND<ENTER>
```

The result is:

```
4: Delete AND Insert
5: (the D AND I commands)
5: (The D AND I commANDs)
8: The insert commAND can place new lines
```

Note that in the above replacement, some unwanted substitutions have occurred. To avoid these the same original file can be used with a slightly different command.

In the next example, to replace only certain occurrences of the and with AND, enter:

```
2? Rand<Ctrl><Z>AND<ENTER>
```

The result is:

```
4: Delete AND Insert
O.K.? Y
5: (The D AND I commands)
O.K.? Y
5: (The D AND I commANDs)
O.K.? N
8: The insert commAND can place new lines
O.K.? N
*
_
```

To see the result of these changes enter the list command L:

```
4: Delete AND Insert
5: (The D AND I commands)
8: The insert command can place new lines
```

Search

Syntax: [*line*] [,*line*] [?] *Sstring*

Purpose: The **Search** command searches the specified range of lines for a specified string of text.

Comments: The *string* must be ended with the <ENTER> key. The first line that matches *string* is displayed and becomes the current line. The **Search** command will terminate when a match is found unless the question mark parameter is included. If there is no match the message "Not found" will be displayed.

If *string* is omitted, Search will take the old string if there is one. (Note that "old" here refers to a string specified in a previous Search or Replace command.) If there is not an old string (i.e., no previous search or replace has been done), the command will terminate immediately.

Default values for the *line* parameters:

If the first *line* is omitted (as in *,line S string*), the first *line* will default to the line *after* the current line.

If the second *line* is omitted (as in *line S string* or *line, S string*), the second *line* will default to # (line after last line of file), which is the same as *line, # S string*.

The question mark parameter:

If the question mark parameter (?) is included in the command, after EDLIN displays the first line with a matching string it will prompt you with the message O.K.?. If you press either the Y or <ENTER> key, the line becomes the current line and the search terminates. If you press any other key, the search continues until another match is found, or until all lines have been searched.

Example:

The following file is ready to edit:

- 1: *This is a sample file
- 2: used to show dynamic line numbers.
- 3: See what happens when you use
- 4: Delete and Insert
- 5: (the D and I commands)
- 6: to edit text
- 7: in your file.
- 8: The insert command can palce new lines
- 9: in the file.

To search for the first occurrence of the string "and", enter:

3,9 Sand<ENTER>

The following line is displayed:

4: Delete and Insert

To search through several occurrences of the string "and" untill the correct string is found, enter:

1, ? Sand

The result is:

4: Delete and Insert
O.K.?_

If you press any key (except Y or <ENTER>), the search continues, so enter N here:

O.K.? N

The search continues:

5: (the D and I commands)
O.K.?_

To terminate the search you enter Y.

To search for string ABC without the verification (O.K.?), type:

SABC

EDLIN will report a match and will continue to search for the same string if you issue the S command again:

S

EDLIN reports another match.

S

EDLIN reports the string is not found.

Note that *string* defaults to any string specified by a previous Replace or Search command.

Transfer

Syntax: `[line]T[d:] filename`

Purpose: The **Transfer** command merges the contents of a specified file with the file currently being edited.

Comments: This command is used to put the contents of one file into another file, or into the text you are creating. The transferred text is inserted ahead of the specified line in the file being edited.

The file lines are then automatically renumbered.

If you do not specify a line number, the default is the current line.

CHAPTER 10 Write

Syntax: `[n]W`

Purpose: The **Write** command writes a specified number of lines to disk from the lines that are being edited in memory. **Write** begins with line number one (1).

Comments: Use this command if the file you are editing is too large to fit in memory. EDLIN automatically writes file lines into memory until memory is 75% full. This command allows you to edit the remainder of the file.

The **Write** command writes a specified number of edited lines in memory to disk. You can then load additional unedited lines from disk into memory by using the **Append** command.

If you do not specify the number of line to write, the system will write lines to disk until 25% of available memory is used. If available memory is already less than 25% used, no action will be taken.

File lines are automatically renumbered so that the first remaining line becomes line number 1.

MEMO

EDLIN

CHAPTER 10

LINK

1. INTRODUCTION	10-2
2. FILES	10-4
Input File Extensions	10-4
Output File Extensions	10-5
VM.TMP (Temporary) File	10-5
3. DEFINITIONS	10-6
Segment	10-6
Class	10-6
Group	10-6
4. COMMANDS	10-7
Prompts	10-7
Switches	10-9
Characters	10-12
5. HOW TO START THE LINKER	10-14
6. SAMPLE LINKER SESSION	10-19

LINK

INTRODUCTION

NOTE: If you are not going to compile and link programs, you do not need to read this chapter.

Read this entire chapter before you use LINK.

You write your programs in source code which is passed through a compiler (or assembler) to produce object modules. The object modules must go through the LINK process to produce a run file.

The Linker is a program that:

Combines separately produced object modules into a program you can run (relocatable executable object code).

Searches library files for definitions of unresolved external references.

Resolves external cross-references.

Produces a listing that shows both the resolution of external references and error messages.

LINK uses as much available memory as is possible. When available memory is exhausted, LINK creates a temporary disk file named VM.TMP.

Figure 10-1 illustrates the various parts of the LINK operation.

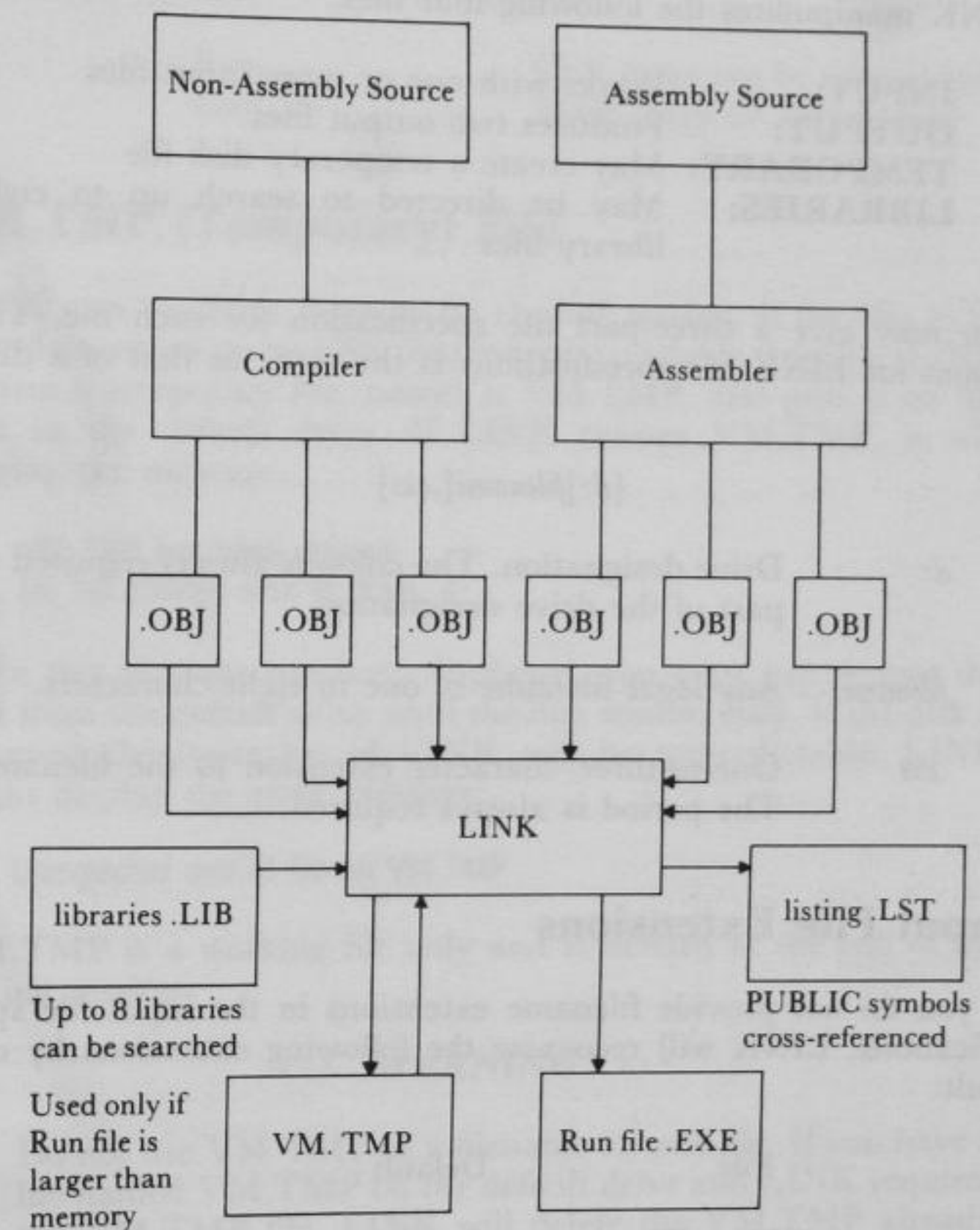


Fig. 10-1. THE LINK OPERATION

FILES

LINK manipulates the following four files.

- INPUT:** Works with one or more input files
- OUTPUT:** Produces two output files
- TEMPORARY:** May create a temporary disk file
- LIBRARIES:** May be directed to search up to eight library files

You may give a three-part file specification for each file. The format for LINK file specifications is the same as that of a disk file:

[d:]filename[.ext]

- d:* Drive designation. The colon is always required as part of the drive designation.
- filename* Any legal filename of one to eight characters.
- .ext* One-to-three character extension to the filename. The period is always required.

LINK

Input File Extensions

If you do not provide filename extensions in the input file specifications, LINK will recognize the following extensions by default:

File	Default
Object File	.OBJ
Library	.LIB

Output File Extensions

LINK appends the following default extensions to the output files:

File	Default
Run	.EXE (may not be overridden)
List	.MAP (may be overridden)

VM.TMP (Temporary) File

LINK uses available memory for the link session. If the files to be linked create an output file that exceeds available memory, LINK creates a temporary file, names it VM.TMP, and puts it on the disk in the default drive. If LINK creates VM.TMP, it will display the message:

VM.TMP has been created.
Do not change disk in drive, *d:*

Once this message has been displayed, you must not remove the disk from the default drive until the link session ends. If the disk is removed the operation of LINK will be unpredictable. LINK might display the error message:

Unexpected end of file on VM.TMP

VM.TMP is a working file only and is deleted at the end of the linking session.

... WARNING ...

Do not use VM.TMP as a filename for any file. If you have a file named VM.TMP on the default drive and LINK requires the VM.TMP file, LINK will delete the VM.TMP already on disk and create a new VM.TMP. The contents of the previous VM.TMP file will be lost.

LINK

DEFINITIONS

In DOS, memory can be divided into segments, classes, and groups.

Segment

A contiguous area of memory that is up to 64K bytes in length. A segment may be located anywhere in memory on a 16-byte (paragraph) boundary. The segment contents are addressed by a segment register/offset pair. A program's location in memory is decided at load time by the relocation loader (located in COMMAND.COM).

Class

A collection of segments. Segments with different segment names may or may not have the same class names. The class name is determined by the assembler or compiler. The class affects the order and relative placement of segments in memory. All segments assigned the same class name are loaded into memory contiguously.

Group

A collection of segments that fit within a 64K byte area of memory. The segments do not need to be contiguous to form a group. The lowest address of the segments in that group is the address of that group. A program may consist of one or more groups.

If you are writing in assembly language, you may assign the group and class names in your program. In high-level languages (BASIC, COBOL, FORTRAN, Pascal), the naming is done automatically by the compiler.

COMMANDS

Prompts

LINK displays four prompts that appear one at a time. When the last prompt is answered, LINK begins linking automatically. When the link session is finished, LINK exits to the operating system. If LINK has finished successfully, the operating system prompt appears. If the LINK is unsuccessful, the appropriate error message is displayed.

LINK prompts you for the names of Object, Run, and List files and for Libraries.

The prompts are listed in order of appearance. The default response is shown in square brackets ([]) following the prompt. The Object Modules prompt has no default and requires you to type a filename.

Object Modules [.OBJ]

[d:] [path] filename [.ext] [+ [d:] [path] filename [.ext]]...

Enter a list of the object modules to be linked. If the filename extension is omitted, LINK assumes that the filename extension is .OBJ. If an object module has any other filename extension, the extension must be given.

Modules must be separated by plus signs (+) or blanks.

LINK loads segments into classes in the order encountered. You can use this information to set the order in which the object modules will be read by LINK.

Run File [filename. EXE]

[d:] [path] [filename [.ext]]

The filename you enter creates a file for storing the Run (executable) file that results from the link session. All Run files have the filename extension .EXE, even if you specify an extension other than .EXE.

If no response is entered to the Run File prompt, LINK uses the first filename you typed in response to the Object Modules prompt as the RUN filename.
For Example:

B:ACCTS

in response to the Run File prompt creates the Run File ACCTS.EXE on drive B.

List File [NUL.MAP]

[d:] [path] [filename [.ext]]

An entry for each segment in the input (object) modules is listed. Each entry also shows the addressing in the Run file.

Libraries [.LIB]

[d:] [path] filename [.ext] [+] [d:] [[path] filename[.ext]]...

The valid responses are up to eight library filenames separated by plus signs (+) or blanks, or a carriage return. A carriage return produces a search of the default library for files which have been created previously. The filename extension defaults to .LIB for library files.

Library file searches (to resolve external references) are in the order listed. When LINK finds the module that defines the external symbol, it processes that module as another object module.

If LINK cannot find a library file on the disks in the disk drives, it displays the message:

Cannot find library *library-name*
Type new drive letter:

Press the letter for the drive designation (for example, B).

LINK

CHAPTER 11

DEBUG

1. INTRODUCTION 11-2

2. STARTING THE DEBUG PROGRAM 11-3

3. COMMAND PARAMETERS 11-4

4. TABLE OF DEBUG COMMANDS 11-7

5. DEBUG COMMANDS 11-9

 Assemble 11-10

 Compare 11-13

 Dump 11-14

 Enter 11-16

 Fill 11-18

 Go 11-19

 Hex 11-21

 Input 11-22

 Load 11-23

 Move 11-25

 Name 11-26

 Output 11-28

 Quit 11-29

 Register 11-30

 Search 11-32

 Trace 11-33

 Unassemble 11-34

 Write 11-36

DEBUG

INTRODUCTION

DEBUG is an interactive tool developed for debugging user software. DEBUG allows the user to check, modify, and test binary programs without having to recompile each and every time.

DEBUG requires a minimum of 13K bytes of memory. DEBUG can operate with one disk drive, however most practical applications require two drives.

Any DEBUG command can be terminated (aborted) by issuing a <Ctrl> <C>. The screen display scrolling action can be temporarily halted to facilitate reading by issuing a <Ctrl> <S>. The screen will resume scrolling by pressing any other key.

STARTING THE DEBUG PROGRAM

A DEBUG session can be started by typing:

DEBUG

The DEBUG utility will then prompt the user for commands by displaying a hyphen (-). Since no filename has yet been specified the DEBUG commands NAME and LOAD can be used to select the file to be debugged. (See the NAME and LOAD commands).

Another way to begin is to include the filename when starting DEBUG:

DEBUG [*d:*] [*path*] [*filename[.ext]*] [*parm1*] [*parm2*]

parm1 and *parm2* are optional parameters.

DEBUG sets registers and flags to the following initial values:

Segment registers CS, DS, ES, and SS are initially set to the first contiguous segment of memory after the end of the DEBUG program.

Instruction Pointer (IP) is set to the value 0100H.

Stack Pointer (SP) is set to the end of the segment, or the bottom of the transient portion of the program loader, whichever is lower. The segment size at offset 6 is reduced by hex 100 to allow for a stack of that size.

Remaining registers (AX, BX, CX, DX, BP, SI, and DI) are set to zero. If you specify a filespec when starting the DEBUG, the CX register contains the size of the file in bytes. If the file is greater than 64K, the size is contained in registers BX and CX (the high portion in BX).

Flags are set to their cleared values. (See the Register command.)

Default disk transfer address is set to 80H in the code segment.

COMMAND PARAMETERS

Parameter	Definition
address	<p>All numeric values are hexadecimal. Enter a one- or two-part designation in one of the following formats:</p> <p>An alphabetic segment register designation, a colon, then an offset value:</p> <p style="padding-left: 40px;">CS:0100</p> <p>A segment address, a colon, then an offset value:</p> <p style="padding-left: 40px;">4BA:0100</p> <p>An offset value only:</p> <p style="padding-left: 40px;">100</p> <p>In this case, the default segment is used. CS is the default segment for the commands G, L, T, U, and W. DS is the default segment for all other commands.</p> <p>NOTE: Memory locations specified by the address must be valid otherwise unpredictable results will occur.</p>
byte	A one or two character hexadecimal value.
drive	A one-digit hexadecimal value to indicate which drive a file will be loaded from or written to. The valid values are 0-3. These values designate the drives as follows: 0=A:, 1=B:, 2=C:, 3=D:.

filespec	A filename which can include a drive designation and filename extension. (See the Name command.)
list	One or more byte and/or string values. For example: ECS:100 42 45 52 54 41
range	<p><i>address address</i></p> <p>A range can be specified by a lower bound address and an upper bound address. For example:</p> <p style="padding-left: 40px;">DS:510 590</p> <p><i>address L value</i></p> <p>A range can also be specified by a lower bound address and a hexadecimal number specifying the number of bytes to be included. For example:</p> <p style="padding-left: 40px;">CS:400 L 11</p> <p>NOTE: The limit for range is hex 10000. To specify a value of 10000 hex in four hexadecimal characters, enter 0000 (or 0).</p>
registername	See the Register command.

record

A 1- to 3-digit hexadecimal value used to indicate the logical record number on the disk and the number of disk sectors to be written or loaded. Logical records correspond to sectors. However, their numbering differs since they represent the entire disk space.

string

Any number of characters enclosed in quote marks. Quote marks may be either single (') or double ("). If the delimiter quote marks must appear within a <string>, the quote marks must be doubled. For example, the following strings are legal:

'This is a "string" is okay.'
'This is a 'string' is okay.'

However, this string is illegal:

'This is a 'string' is not.'

Similarly, these strings are legal:

"This is a 'string' is okay."
"This is a ""string"" is okay."

value

Either a hexadecimal value up to 4 digits used to specify the port number or the number of times to repeat a command.

TABLE OF DEBUG COMMANDS

Command	Purpose	Format
Assemble	Assembles statements	A [address]
Compare	Compares memory	C range address
Dump	Displays memory	D [address] or D [range]
Enter	Changes memory	E address [list]
Fill	Changes memory blocks	F range list
Go	Executes with optional breakpoints	G [=address] [address [address...]]
Hex	Hexadecimal add-subtract	H value value
Input	Reads/displays input byte	I value
Load	Loads file or absolute diskette sectors	L [address [drive record record]]
Move	Moves memory block	M range address
Name	Defines files and parameters	N filespec [filespec...]
Output	Sends output byte	O value byte
Quit	Ends DEBUG program	Q
Register	Displays registers/flags	R [registername]

(cont.)

Search	Searches for characters	S <i>range list</i>
Trace	Executes and displays	T [=address] [value]
Unassemble	Unassembles instructions	U [address] or U [range]
Write	Writes file or absolute disk sectors	W [address [drive record record]]

DEBUG COMMANDS

The following notes are common to all DEBUG commands:

A command is specified by a single letter in upper or lower case. One or more parameters may follow.

Delimiters are only required between two consecutive hexadecimal values. Delimiters may be used to separate commands and parameters. The following commands are equivalent:

```

dcs: 200 210
d cs:200 210
d,cs: 200,210
    
```

Press <Ctrl> <Break> to abort any command.

Use the <ENTER> key to activate a command.

To stop the scrolling action of the display use <Ctrl> <NumLock>. Press any other key to resume scrolling.

The DEBUG program prompt is the hyphen (-).

The DEBUG program resides on your System Disk.

Assemble

Syntax: A[address]

Purpose: To assemble Assembler language statements directory into memory. DEBUG supports standard 8086/8087/8088 assembly language.

Comments: DEBUG accepts hexadecimal numeric input. Beginning at the specified address all assembler statements are placed in contiguous locations (addresses) in memory. The default address is the location following the last instruction assembled by a previous Assemble command or the area at CS:0100 if no previous Assemble command was used. When all statements have been entered, press <ENTER> to return to the DEBUG prompt.

DEBUG responds to invalid statements by displaying:

Error

All numeric values entered are hexadecimal and can be entered as 1-4 characters.

The segment override mnemonics are CS:, DS:, ES:, and SS:.

Prefix mnemonics must be entered in front of the opcode to which they refer. They may be entered on a separate line.

String manipulation mnemonics must state the string size. For example:

MOVSW

to move word strings.

MOVSB

to move byte strings.

The mnemonic for the far return is RETF.

The assembler will automatically assemble short, near, or far jumps and calls depending on byte displacement to the destination address. These may be overridden with the NEAR or FAR prefix. For example:

```
0100:0500 JMP 502           ; a 2 byte short jump
0100:0502 JMP NEAR 505     ; a 3 byte near jump
0100:0505 JMP FAR 50A      ; a 5 byte far jump
```

The NEAR prefix can be abbreviated to NE, but the FAR prefix cannot be abbreviated.

Operands must specify either word or byte memory locations. The type must be stated with the prefix "WORD PTR" or "BYTE PTR". DEBUG accepts the abbreviations "WO" and "BY". For example:

```
NEG BYTE PTR [128]
DEC WO [SI]
```

To display whether an operand refers to a memory location or to an immediate operand, DEBUG uses the convention that operands enclosed in square brackets refer to memory. For example:

```
MOV AX,21; Load AX with 21H
MOV AX,[21]; Load AX with the contents of memory
location 21H
```


The pseudo-instructions DB and DW are available. The DB opcode assembles byte values. The DW opcode assembles word values. For example:

```
DB 1,2,3,4, "THIS IS AN EXAMPLE"  
DB 'THIS IS A QUOTE:'  
DB "THIS IS A QUOTE:"  
  
DW 1000,2000,3000,"BACH"
```

For 8087 opcodes the WAIT or FWAIT prefix must be explicitly specified. For example:

```
FWAIT FADD ST,ST(3) ; This line will  
                    ; assemble  
                    ; an FWAIT  
                    ; prefix.
```

LD TBYTE PTR [BX] ; This line will not.

Assemble supports all forms of the register indirect commands. For example:

```
ADD  BX,34[BP+2].[SI-1]  
POP  [BP+DI]  
PUSH [SI]
```

All opcode synonyms are supported. For example:

```
LOOPZ    100  
LOOPE   100  
  
JA       200  
JNBE    200
```

Compare

Syntax: C range address

Purpose: Used to compare two blocks of memory.

Comments: Ranges are used to select two blocks for comparison. If the areas of memory are identical the prompt reappears. If there are differences, their addresses and contents are displayed, in the form:

addr1 byte1 byte2 addr2

addr1 byte1 refers to the location and contents of the mismatching locations in range.

byte2 addr2 refers to the byte found in *address*.

Example: C100, 1FF 300
or
C100L100 300

The 100H bytes of memory beginning at DS:100 are compared with the 100H bytes beginning at DS:300.

Dump

Syntax: D [*address*]

or

D [*range*]

Purpose: Used to display the contents of a block of memory.

Comments: The hexadecimal and ASCII representations of the specified block are displayed. In the ASCII portion, unprintable characters are indicated by a period (.).

Each line begins on a 16-byte boundary and shows 16 bytes. There is a hyphen between the 8th and 9th bytes.

NOTE: If the starting address of the dump is not on a boundary, the first line may have fewer than 8 or 16 bytes. In this case, the second line of the dump begins on a boundary.

The Dump command has two format options:

Option 1 D *address*

or

D

The contents are displayed starting with the specified address.

If no address is specified, the starting location is the address following the last address displayed by a previous D command. Each subsequent D displays the bytes immediately following those last displayed. If no previous D command has been issued, 0100H is used as an offset.

NOTE: If you specify only an offset for the starting address, the segment in DS register is used.

Option 2 D *range*

The contents of the specified address range are displayed.

Example: If you type the command:

```
DCS: 100 110
```

DEBUG displays the dump in the following format:

```
04BA:0100 54 4F 4D 20 53 ... 4E 44 TOM SAWYER ...
```

If you type the command:

```
DCS:100 L 20
```

the display is formatted as described above, but 20H bytes are displayed.

Enter

Syntax: E *address* [*list*]

Purpose: The Enter command can be used in two ways:

To replace the contents of one or more bytes, starting at the specified address, with the values contained in the list.

Displays and allows modification of bytes in a sequential manner.

Comments: If the optional *list* of values is typed, the replacement of byte values occurs automatically.

For examples,

```
E ds:100 E3 "abc" 7E
```

fills ds:100 through ds:104 with E3 61 62 63 7E.

If the *address* is typed without the optional *list*, DEBUG displays the address and its contents, and waits for your input. At this point, the Enter command waits for you to perform one of the following actions:

1. Replace a byte value with a value you type. Simply type the value after the current value. If the value typed in is not a legal hexadecimal value or if more than two digits are typed, the illegal or extra character is not echoed.
2. Press the <SPACEBAR> bar to advance to the next byte. To change the value, simply type the new value as described in (1.) above. If you space beyond an 8-byte boundary, DEBUG starts a new display line with the address displayed at the beginning.

3. Type a hyphen (-) to return to the preceding byte. If you decide to change a byte behind the current position, typing the hyphen returns the current position to the previous byte. When the hyphen is typed, a new line is started with the address and its byte value displayed.

4. Press the <ENTER> key to terminate the Enter command. The <ENTER> key may be pressed at any byte position.

Example: E cs:100

DEBUG displays:

```
04BA:0100 EB._
```

To change EBH to 41H, enter 41. 1.

```
04BA:0100 EB.41_
```

To step through the subsequent bytes, press the <SPACEBAR> to see:

```
04BA:0100 EB.41 10. 00. BC._
```

To change hex BC to hex 42, enter 42.

```
04BA:0100 EB.41 10. 00. BC.42_
```

To back up and change the hex 10 to hex 6F enter two hyphens and the hex 6F.

```
04BA:0100 EB.41 10. 00. BC.42-  
04BA:0102 00.-  
04BA:0101 10.6F_
```

Press the <ENTER> key to end the Enter command. The hyphen (-) prompt will be displayed.

Fill

Syntax: F *range list*

Purpose: This command is used to specify a range of bytes in memory for replacement by a list.

Comments: If the number of bytes specified in the memory range is greater than the number of bytes in the list then DEBUG will attempt to repeat the list until the memory bytes are filled. If the list is bigger than the number of bytes specified in memory then the list is truncated.

NOTE: If you specify only an offset for the starting address of the range, the segment in DS register is used.

Example: F04BA:100 L 5 F8 "ABC" 2D

Memory locations 04BA:100 through 04BA:104 are filled with the 5 bytes in the list. The ASCII values of the list characters are stored. Locations 100-104 will contain F8 41 42 43 2D.

Go

Syntax: G [=address] [address [address...]]

Purpose: This command is used to execute the current program. The program will stop and display registers, flags etc. when a BREAKPOINT is encountered.

Comments: The Go command has two format options:

Option 1 G [=address]

This option executes the program you are debugging when you don't set breakpoints.

When you test the program with different parameters each time, use this option. (Refer to the Name command.) If not using =address be sure the CS:IP values are set properly before executing the G command.

Option 2 G [=address] address [address...]

This option will halt program execution at a breakpoint. The current registers and flag status are displayed. Up to ten breakpoints are allowed.

NOTES: When a program has terminated normally ("Program terminated normally" message has been displayed), you have to reload the program to execute it again.

Breakpoints must be set only at addresses containing the first byte of an 8088 opcode.

The stack pointer must be valid and have 6 bytes available to the Go command.

If you specify only an offset for a breakpoint, the segment in the CS register is used.

Example: GCS:7550

The program in memory executes up to the address 7550 in the CS segment. DEBUG displays the registers and flags. To resume program execution at the instruction after the breakpoint, type the GO command.

Hex

Syntax: H *value value*

Purpose: This command displays the sum and difference of the two hexadecimal values.

Example: H 0A 8
0012 0002

The hexadecimal sum of 000A and 0008 is 0012, and their difference is 0002.

Example: Assume that the following commands are typed:

```
A>DEBUG  
NFILE.COM
```

Now, to load FILE.COM, type

Input

Syntax: I *value*

Purpose: Inputs and displays (in hexadecimal) one byte from the port specified by *value*.

Example: I2F8

If the byte at the port is 42H then DEBUG displays:

42

Load

Syntax: L [*address* [*drive record record*]]

Purpose: This command is used to load a file into memory. It can also be used to load specified sectors from a disk.

Comments: Set BX:CX to the number of bytes read. The file must have been named either when DEBUG was started or with the N command. Both the DEBUG invocation and the N command format a filename properly in the normal format of a file control block at CS:5C.

If the L command is typed without any parameters, DEBUG loads the file into memory beginning at address CS:100 and sets BX:CX to the number of bytes loaded. If the L command is typed with an address parameter, loading begins at the memory *address* specified. If L is typed with all parameters, absolute disk sectors are loaded, not a file. The *records* are taken from the *drive* specified (the drive designation is numeric here 0=A:, 1=B:, 2=C:, etc.); DEBUG begins loading with the first *record* specified, and continues until the number of sectors specified in the second *record* have been loaded.

Example: Assume that the following commands are typed:

```
A>DEBUG  
-NFILE.COM
```

Now, to load FILE.COM, type:

```
L
```


DEBUG loads the file and then displays the DEBUG prompt. Assume that you want to load only portions of a file or certain records from a disk. To do this, type:

```
L04BA:100 2 0F 6D
```

DEBUG then loads 109 (6D hex) records beginning with logical record number 15 into memory beginning at address 04BA:0100. When the records have been loaded, DEBUG simply returns the - prompt.

If the file has a .EXE extension, it is relocated to the load address specified in the header of the .EXE file: the *address* parameter is always ignored for .EXE files. The header itself is stripped off the .EXE file before it is loaded into memory. Thus the size of an .EXE file on disk will differ from its size in memory.

If the file named by the Name command or specified when DEBUG is started is a .HEX file, then typing the L command with no parameters causes DEBUG to load the file beginning at the address specified in the .HEX file. If the L command includes the option *address*, DEBUG adds the *address* specified in the L command to the address found in the .HEX file to determine the start address for loading the file.

Move

Syntax: M *range address*

Purpose: This command is used to move the block of memory specified by *range* to the location beginning at *address*.

Comments: Overlapping moves (moves where part of the block overlaps some of the current addresses) are always performed without loss of data. Addresses that could be overwritten are moved first.

Moves from higher addresses to lower addresses, move data beginning at the block's lowest address and work towards the highest. Moves from lower addresses to higher addresses, move data beginning at the block's highest address and work towards the lowest. The sequence of the move is important because the MOVE command copies the data from one area into another, in the sequence described, and writes over the new addresses.

Example: MCS:100 110 CS:500

DEBUG first moves address CS:110 to address CS:510; then CS:10F to CS:50F, and so on until CS:100 is moved to CS:500. The DUMP command can be used to view the results of a MOVE command.

Name

Syntax: N *filespec* [*filespec*...]

Purpose: This command specifies the filename of the file to be used with later LOAD and GO commands. This command must be used when DEBUG is started without a specified filename.

Comments: All specified filespecs and parameters are placed exactly as entered, including delimiters, in a parameter save area at CS:81. CS:80 containing the number of characters entered.

Example:

```
-NPROGA.EXE  
-L  
-NFILE1.DAT FILE2.DAT  
-G
```

The effect of these commands is:

NAME sets PROGA.EXE as the filename.
LOAD loads PROGA.EXE into memory.
NAME is used again to specify the parameters to be used by PROGA. EXE.
GO executes PROGA.EXE as if FILE1.DAT and FILE2.DAT has been typed at the DOS command level.

NOTE: If a WRITE command were executed at this point, the file would be saved with the filename FILE1.DAT. To avoid this problem, use the NAME command before LOAD or WRITE.

The regions of memory that can be affected by the NAME command are:

- CS:5C File Control Block for file 1.
- CS:6C File Control Block if file 2 is specified.
- CS:80 Count of characters in the NAME command.
- CS:81 Actual characters typed in the NAME command.

To execute PROG as if the following line had been typed:

```
PROG PARM1 PARM2/C
```

enter:

```
DEBUG PROG.COM  
-NPARM1 PARM2/C  
-G
```


Output

Syntax: O *value byte*

Purpose: This command sends the *byte* to the output port specified by *value*.

Example: O2F8 4F

Sends the byte value 4FH to output port 2F8H.

Quit

Syntax: Q

Purpose: This command ends the DEBUG program.

Comments: Note that the QUIT command stops DEBUG operation without saving the current program. The Write command must be used to save the program before stopping DEBUG.

DEBUG returns to the DOS command level.

Example: -Q
A>

Register

Syntax: R [*registername*]

Purpose: This command can be used to:

Display and modify the hex contents of any single register.

Display the hex contents of all registers, flags, and the next instruction to be executed.

Display and modify the flag settings.

Comments: If no *registername* is specified, the R command dumps the register save area and displays the contents of all registers and flags.

The valid register names are (IP and PC both refer to the instruction pointer):

AX	BP	SS
BX	SI	CS
CX	DI	IP
DX	DS	PC
SP	ES	F

The flags are:

FLAG NAME	TO SET	TO CLEAR
Overflow	OV	NV
Direction	DN (decrement)	IN (increment)
Interrupt	EI (enabled)	DI (disabled)
Sign	NG (negative)	PL (plus)
Zero	ZR	NZ
Auxillary Carry	AC	NA
Parity	PE (even)	PO (odd)
Carry	CY	NC

Example: R

DEBUG displays all registers, flags and the instruction for the current location.

RF

DEBUG displays all the flags. To change the value of a flag, just enter a valid flag value. To exit the R command press <ENTER>. The flags you did not list values for will remain unchanged.

R BX

Displays the contents of register BX. To change the contents of the register, just enter a 1-4 character hexadecimal value. To exit the R command press <ENTER>. The register contents will be unchanged if you do not enter a new value.

Search

Syntax: S *range list*

Purpose: This command is used to search for a specified character string within a range in memory.

Comments: The address of each occurrence of a match is displayed. When no match is found a hyphen (-) is displayed.

Example: S CS:100 110 41

Searches addresses from CS:100 through CS:110 for hex 41.

The addresses of matches are displayed:

```
04BA:0100
04BA:010D
```

Trace

Syntax: T [=address] [value]

Purpose: This command executes instructions and displays the contents of all registers and flags after each instruction.

Comments: The number in *value* determines the number of instructions to execute. Pressing <Ctrl> <NumLock> will suspend the scrolling so that you can study the registers and flags for any particular instruction.

Example: T

Assume that the current position is 04BA:011A. DEBUG will display:

```
AX=0E00 BX=00FF CX=0007 DX=01FF
SP=039D BP=0000 SI=005C DI=0000
DS=04BA ES=04BA SS=04BA CS=04BA
IP=011A NV UP DI NG NZ AC PE NC
04BA:011A CD21 INT 21
```

T=110A 10

DEBUG executes sixteen instructions (10 in hex). The registers and flags for each instruction are displayed.

Unassemble

Syntax: U[*address*]

or

U[*range*]

Purpose: This command is used to take the specified contents of memory and translate them into their equivalent assembler command. The addresses and hexadecimal values are displayed with assembler-like statements.

Comments: If the contents of the addresses specified do not contain valid instruction codes then errors will occur.

If you do not include a *range* of bytes, DEBUG translates 20 hex bytes. If no address is specified then the default address is the address after the last one specified by a U command. With this technique it is possible to produce continuous translations.

If you include the *range*, all the instructions in the *range* will be unassembled.

Example: U 04BA:0100 L10

DEBUG displays:

04BA:0100	206472	AND [SI+72], AH
04BA:0103	69	DB 69
04BA:0104	7665	JBE 016B
04BA:0106	207370	AND [BP+DI+70], DH
04BA:0109	65	DB 65
04BA:010A	63	DB 63
04BA:010B	69	DB 69
04BA:010C	66	DB 66
04BA:010D	69	DB 69
04BA:010E	63	DB 63
04BA:010F	61	DB 61

If you type U 04BA:0100 0108 DEBUG will display:

04BA:0100	206472	AND[SI+72],AH
04BA:0103	69	DB 69
04BA:0104	7665	JBE 016B
04BA:0106	207370	AND [BP+DI+70],DH

Write

Syntax: W [*address* [*drive record record*]]

Purpose: This command is used to write the contents of the memory area specified to the indicated drive.

Comments: Writing to absolute sectors is very dangerous because you are bypassing the file handler. Be sure that the sectors specified on the disk DO NOT contain data that you want to save.

The current file must have been named either when DEBUG was started or with the NAME command.

If W is used with no parameters:
BX:CX must be set to the number of bytes to be written.
The file is written from CS:100.

If W is used with only an address:
The file is written beginning at the address.

If W is used with parameters:
The write begins from the address specified.
The file is written to the drive specified.
(Drive designation is numeric
0=A:
1=B:
2=C: etc.)

A single W command can write a maximum of hex 80 sectors.

Example: W

DEBUG writes the file to disk and then displays the DEBUG prompt.

WCS:100 1 37 2B

DEBUG writes the contents of memory from CS:100. The data starts at sector hex 37 and continues for hex 2B sectors. The information is written on Drive B.

MEMO

DEBUG

CHAPTER 12

INPUT AND OUTPUT OPTIONS

- 1. REDIRECTION OF INPUT AND OUTPUT DEVICES12-2
- 2. PIPING INPUT AND OUTPUT12-4
- 3. FILTERS 12-5

IN / OUT

REDIRECTION OF INPUT AND OUTPUT DEVICES

The following command lines are used to redirect standard input and output. Standard input to a program is from the keyboard and standard output is to the screen. Standard input can be redirected to a file from which a program will draw data.

... WARNING ...

The program will terminate if it is allowed to read more data than is contained in the file. If the program terminates, press <Ctrl> <Break> to return to the DOS prompt.

To redirect standard input:

```
<[d:] [path] filename
```

Standard output can be redirected to any file.

To create/open a file and assign it to standard output:
(the previously created file will be erased)

```
>[d:] [path] filename
```

To create/open a file and assign it to standard output:
(output will be added to the end of a previously created file)

```
>>[d:] [path] filename
```

Example:

The following command will direct the display of DIR to the printer:

```
DIR >PRN
```

The following command will direct the display of DIR to the file LISTIT:

```
DIR >LISTIT
```

The following command will cause the program PROG1 to obtain its input from the datafile INFO.TXT:

```
PROG1 <INFO.TXT
```

NOTE: If standard input/output is not performed via DOS function calls in an application program (e.g. writing directly to the video buffer) then redirection will not work.

PIPING INPUT AND OUTPUT

Programs may be chained so that the output specified program can be stored in a temporary datafile for input to another specified program. (Temporary files can be recognized in the root directory of the default drive by the form %PIPEx.\$\$\$.) The vertical bar (|) is used to chain two programs for automatic redirection of standard input and output.

Example:

Assume the program named SORT reads all its standard input, sorts it, and then writes it to the standard output device. The command:

```
DIR | SORT
```

generates a sorted directory listing. All standard output generated by the DIR command will be sent to the standard input of the SORT program.

To send the sorted directory to a file, you would enter:

```
DIR | SORT >FILE
```

To create a file to contain only the directory entries for sub-directories, you would enter:

```
DIR | FIND "DIR" | SORT >FILE
```

APPENDICES

CONTENTS

APPENDIX A—User Diagnostics.

APPENDIX B—Error Messages.

APPENDIX C—Pin Configurations.

APPENDIX D—Interrupts and Function Calls.

APPENDIX E—Control Blocks.

APPENDIX F—Esc. Sequence.

APPENDIX G—For Operations of Popular Application Software.

APPENDIX H—Character Set/Plasma Display.

APPENDIX I—Demonstration Software.

APPENDIX J—Specifications.

APPENDIX K—Index.

APP. A

APP. B

APP. C

APP. D

APP. E

APP. F

APP. G

APP. H

APP. I

APP. J

APP. K

APPENDIX A

USER DIAGNOSTICS

Use the User Diagnostics to help you troubleshoot problems with the system. For the most accurate results, use the User Diagnostics when the system is in the OFF state.

During the course of the test you may encounter certain error messages. Always write down or print out the message and the procedure you must perform when the message appears. Sometimes a simple phone conversation can allow your technician to correct the message and help you get your computer working again.

Even if the message needs to be revised, a list of error messages and their location within the testing sequence can be helpful in troubleshooting again.



USER DIAGNOSTICS

Built-into the ROM of your Exec. Partner, is a series of tests that help pinpoint problems in the operation of the computer. You run the User Diagnostic tests whenever your computer fails to operate or respond correctly.

During the course of the tests you may encounter certain error messages. Always write down or print out the message, and the procedure you were performing when the message appeared. Sometimes a simple phone conversation can allow your Panasonic dealer to interpret the message and help you get your computer operating again.

Even if the computer needs to be serviced, a list of error messages and their location within the testing sequence can be helpful in accelerating repairs.

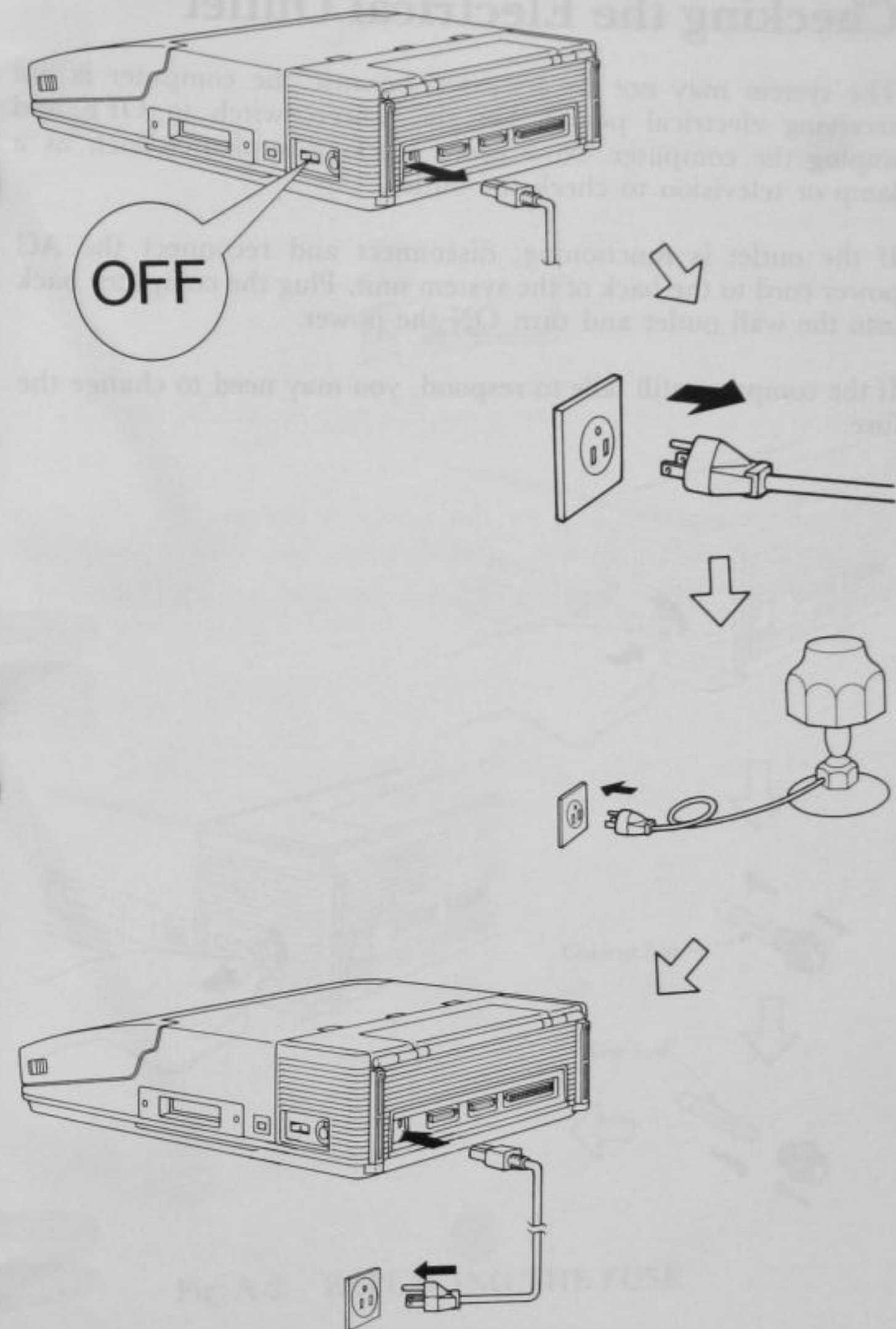


Fig. A-1. CHECKING THE ELECTRICAL OUTLET

Checking the Electrical Outlet

The system may not be activated because the computer is not receiving electrical power. Set the power switch to OFF and unplug the computer. Now use a working appliance such as a lamp or television to check the outlet.

If the outlet is functioning, disconnect and reconnect the AC power cord to the back of the system unit. Plug the computer back into the wall outlet and turn ON the power.

If the computer still fails to respond, you may need to change the fuse.

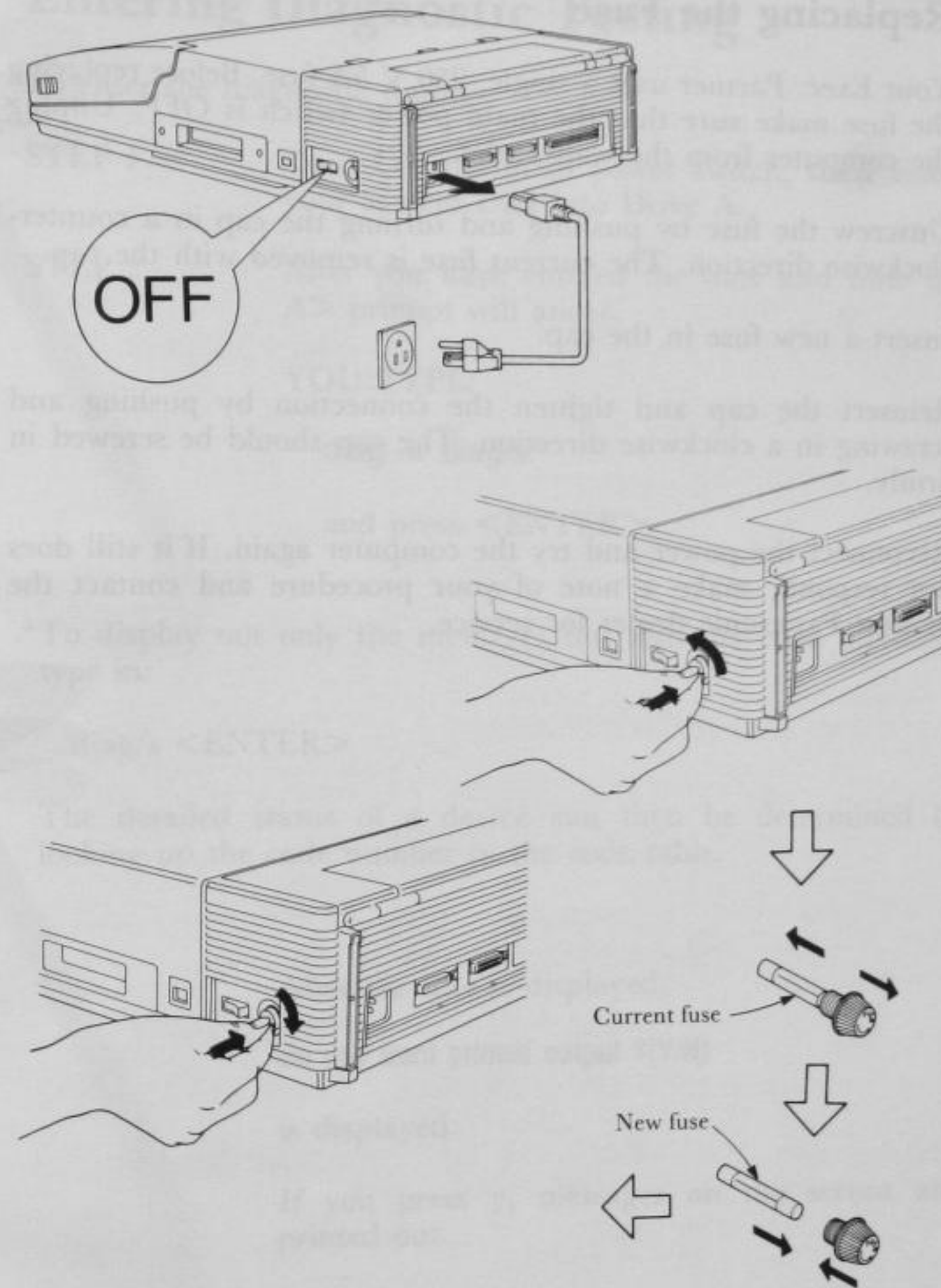


Fig. A-2. REPLACING THE FUSE

Replacing the Fuse

Your Exec. Partner uses a single, 250 V 5A fuse. Before replacing the fuse make sure that the main power switch is OFF. Unplug the computer from the wall outlet.

Unscrew the fuse by pushing and turning the cap in a counter-clockwise direction. The current fuse is removed with the cap.

Insert a new fuse in the cap.

Reinsert the cap and tighten the connection by pushing and screwing in a clockwise direction. The cap should be screwed in firmly.

Reconnect the power and try the computer again. If it still does not respond, make a note of your procedure and contact the nearest Panasonic dealer for service.

Entering Diagnostic Testing

To enter the diagnostic testing:

STEP 1 Turn ON the main power switch, **then** insert your System Disk into Drive A.

STEP 2 After you have entered the date and time the A> prompt will appear.

YOU TYPE:

diag or diag/s*

and press <ENTER>

*To display not only the messages but also the code numbers, type in:

diag/s <ENTER>

The detailed status of a device can then be determined by looking up the code number in the code table.

After the title is displayed,

Do you want printed output?(Y/N)

is displayed.

If you press y, messages on the screen are printed out.

STEP 3

SCREEN DISPLAYS:

DIAGNOSTIC MENU

- 1-TEST ALL DEVICES
- 2-TEST AUTOMATICALLY
- 3-CHANGE MENU
- 4-EXIT
- 5-MAIN BOARD
- 6-xxxKB MEMORY
- 7-KEYBOARD
- 8-PLASMA DISPLAY (or GRAPHICS DISPLAY)
- 9-2 FLOPPY DISK DRIVE(S)
- 10-2 PARALLEL PORT(S)
- 11-1 SERIAL PORT(S)
- 12-REALTIME CLOCK

SELECT MENU: 1

If your Exec. Partner has a Hard Disk Drive Unit,
SCREEN DISPLAYS:

DIAGNOSTIC MENU

- 1-TEST ALL DEVICES
- 2-TEST AUTOMATICALLY
- 3-CHANGE MENU
- 4-EXIT
- 5-MAIN BOARD
- 6-xxxKB MEMORY
- 7-KEYBOARD
- 8-PLASMA DISPLAY (or GRAPHICS DISPLAY)
- 9-1 HARD DISK DRIVE
- 10-1 FLOPPY DISK DRIVE
- 11-2 PARALLEL PORT(S)
- 12-1 SERIAL PORT(S)
- 13-REALTIME CLOCK

SELECT MENU: 1

The System Diagnostics Menu

This is the menu for the testing procedures. (A **menu** is a list of choices). This menu lists the various components of your Exec. Partner system.

If you chose:

- 1- TEST ALL DEVICES Each device is checked in sequence. The computer will wait to proceed after each test. Notice that this selection is automatically programmed as the default choice. Press <ENTER> to begin the testing.
- 2- TEST AUTOMATICALLY All the devices will be tested without further input from you. This testing will continue to operate until you press <Ctrl><Break> to stop the cycle.
- 3- CHANGE MENU Allows you to add or delete items from the menu. See A-29.
- 4- EXIT If you decide you do not wish to proceed with the diagnostic testing, press this selection to return to DOS.
- 5- MAIN BOARD Tests the main board.
- 6- xxxKB MEMORY Tests the memory. (the number of KB will change depending on the configuration of your system.)

7-KEYBOARD

Tests the keyboard.

8-PLASMA DISPLAY/
GRAPHICS VIDEO*

Tests the built-in circuit for the built-in plasma display, or an optional graphics card and graphics display.

9-MONOCROME
DISPLAY (OPTION)**

Tests an optional monochrome card and monochrome display.

10-1 HARD DISK DRIVE
(FT-80 ONLY)

Tests the hard disk drive connected to the system.

11-2 FLOPPY DISK
DRIVE(S)

Tests the floppy disk drive connected to the system.

12-2 PARALLEL PORT(S)

Tests the internal printer (parallel port 1) and the built-in circuit for an optional printer (parallel port 2). The second test checks the internal capacity to run an additional printer if it is not connected. Thus the menu will read "2 parallel port(s)" even if no additional printers are part of the system configuration.

13-1 SERIAL PORT(S)

Tests the built in RS232C port.

14-REAL TIME CLOCK

Tests the built-in real time clock.

*If a graphics display is connected to the Exec. Partner, the graphics display will operate instead of the plasma display. The test for the graphics display will be executed instead of that for the plasma display.

**If a monochrome display is connected to the Exec. Partner, "Monochrome Display" will be displayed below "KEYBOARD TEST". See page A-42 for details on the diagnostic test for the monochrome display.

To select the option on the menu:

Move the inverse display to your desired option by the <↑> key (upward) or <↓> key (downward). At the same time the value of "SELECT MENU" at the bottom of the menu will be automatically changed. (You can type the value of your desired option instead of the <↑> or <↓> key.) Then press <ENTER>.

In most cases, you will test the entire system, using the first option on the menu. The testing example in this manual will cover the entire testing procedure as contained in the first selection. To run tests on specific devices, see the sections dealing with that test.

If the correct menu has appeared on your screen, proceed to STEP 4.

If the screen does not display the menu, or the menu does not match your system configuration, write down your procedure, any error messages displayed, and call your Panasonic dealer for service.

Terminating Diagnostic Testing

If during the course or the testing you wish to stop the procedure, hold down <Ctrl> and press the <Break> key. You may need to finish the specific tests being run (complete the keyboard testing sequence, for example) but at the conclusion of the current test, you will return to the menu.

You must hold <Ctrl> and press <Break> to stop the automatic testing cycle started with Option 2.

If your screen matches this display, proceed to STEP 17.

If your screen does not look like this, write down your procedure, any error messages displayed, and call your Panasonic dealer for service.

STEP 17

SCREEN DISPLAYS:

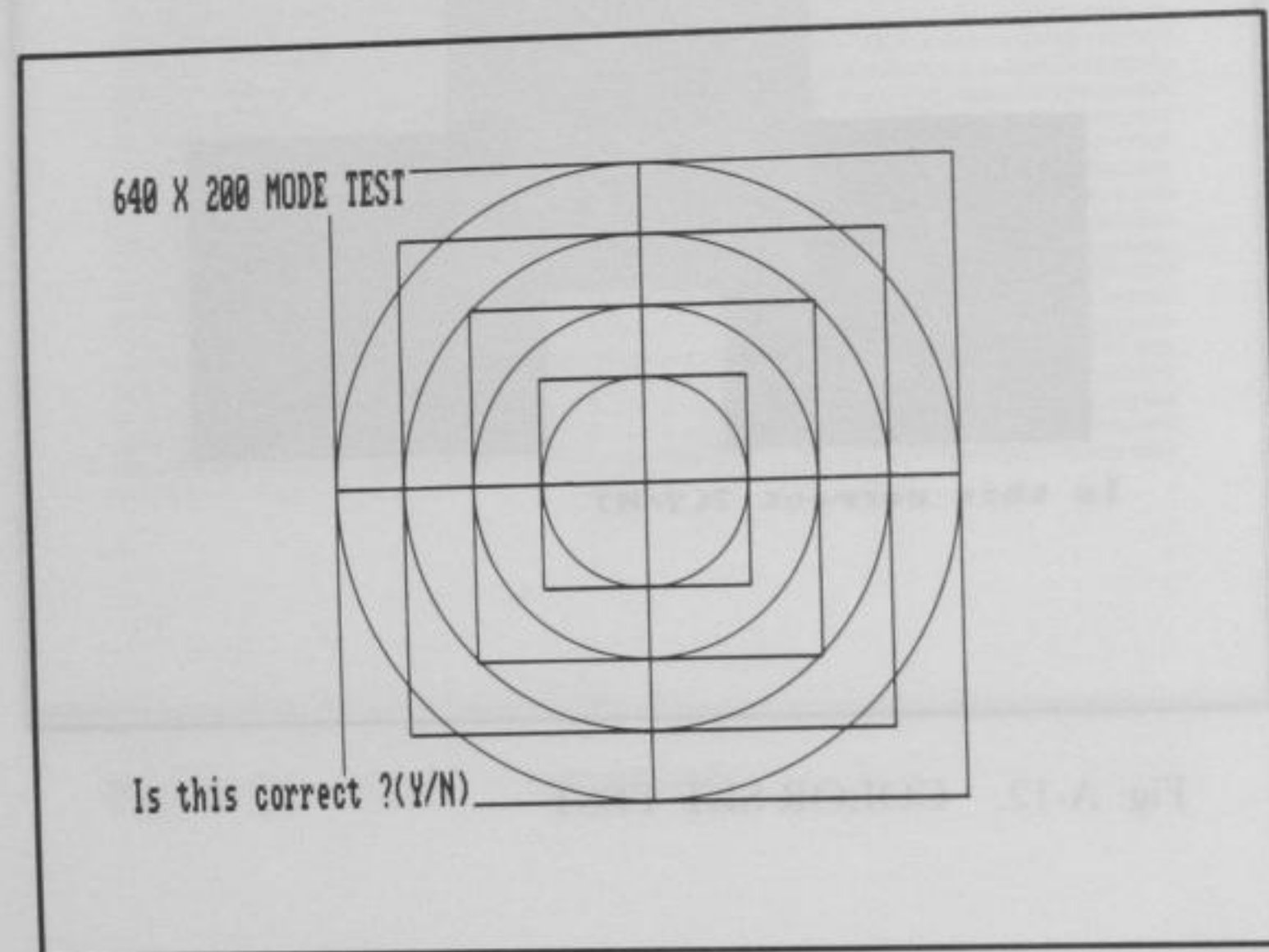


Fig. A-13. 640×200 MODE TEST

If your screen matches this display, proceed to STEP 18.

If your screen does not look like this, write down your procedure, any error messages displayed, and call your Panasonic dealer for service.

STEP 18

Press any key.

SCREEN DISPLAYS:

```
*****
* CONTENTS OF DISK(S) WILL BE DESTROYED *
*****
```

Insert scratch disk into each drive
Press any key when ready

OR

Insert scratch disk into drive(s)
Press any key when ready

This is a test of your disk drives. During this test the drives write to the disk. That means that any information on the disks will be written over and destroyed.

IF YOU HAVE NOT ALREADY DONE SO,
REMOVE YOUR SYSTEM DISK FROM
DRIVE A.

Insert a new disk in Drive A (and Drive B if you have an FT-70.) These disks will be formatted as the part of the testing process.

When the disks are ready, proceed to STEP 19.

STEP 19

Press any key

You will hear the drives operating and the In Use Indicators will come on.


```

TESTING EXTERNAL PARALLEL PORT
! " # $ % & ' ( ) * + , - . / 0 1 2 3 4 5 6 7 8 9 : ; < = > ? @ A B C D E F G H I J K L M N O P Q R S T U V W X Y Z [ \ ] ^ _ ` a b c d e f
! " # $ % & ' ( ) * + , - . / 0 1 2 3 4 5 6 7 8 9 : ; < = > ? @ A B C D E F G H I J K L M N O P Q R S T U V W X Y Z [ \ ] ^ _ ` a b c d e f g h
! " # $ % & ' ( ) * + , - . / 0 1 2 3 4 5 6 7 8 9 : ; < = > ? @ A B C D E F G H I J K L M N O P Q R S T U V W X Y Z [ \ ] ^ _ ` a b c d e f g h i
! " # $ % & ' ( ) * + , - . / 0 1 2 3 4 5 6 7 8 9 : ; < = > ? @ A B C D E F G H I J K L M N O P Q R S T U V W X Y Z [ \ ] ^ _ ` a b c d e f g h i j
! " # $ % & ' ( ) * + , - . / 0 1 2 3 4 5 6 7 8 9 : ; < = > ? @ A B C D E F G H I J K L M N O P Q R S T U V W X Y Z [ \ ] ^ _ ` a b c d e f g h i j k
! " # $ % & ' ( ) * + , - . / 0 1 2 3 4 5 6 7 8 9 : ; < = > ? @ A B C D E F G H I J K L M N O P Q R S T U V W X Y Z [ \ ] ^ _ ` a b c d e f g h i j k l
! " # $ % & ' ( ) * + , - . / 0 1 2 3 4 5 6 7 8 9 : ; < = > ? @ A B C D E F G H I J K L M N O P Q R S T U V W X Y Z [ \ ] ^ _ ` a b c d e f g h i j k l m
! " # $ % & ' ( ) * + , - . / 0 1 2 3 4 5 6 7 8 9 : ; < = > ? @ A B C D E F G H I J K L M N O P Q R S T U V W X Y Z [ \ ] ^ _ ` a b c d e f g h i j k l m n
! " # $ % & ' ( ) * + , - . / 0 1 2 3 4 5 6 7 8 9 : ; < = > ? @ A B C D E F G H I J K L M N O P Q R S T U V W X Y Z [ \ ] ^ _ ` a b c d e f g h i j k l m n o

```

Fig. A-14. PRINTER OUTPUT (2)

Your output should match this illustration.

STEP 22

This test checks the built-in serial interface and if you attach the plug for a loop back test, the serial port can be tested.

NOTE: The plug for a loop back test must be wired as follows:

- Connect pin 2 and pin 3
- Connect pin 4, pin 5 and pin 8
- Connect pin 6, pin 20 and pin 22

SCREEN DISPLAYS

SERIAL PORT(S) TEST

TESTING INTERNAL SERIAL PORT

If no error, screen displays

Need loopback test?(Y/N)

If you press y, screen displays,

Connect loopback plug.
Press any key when ready

Press any key then the loopback test will begin.

If you press n, the loopback test will not be performed.

STEP 23

Press any key.

SCREEN DISPLAYS:

REALTIME CLOCK TEST

Current Setting: Date=XX-XX-XXXX Time=XX:XX:XX

Press any key when ready

STEP 24

Press any key.

This returns you to the System Diagnostics Menu.

Change Menu

You can add or delete options to or from the System Diagnostics Menu by Option 3, CHANGE MENU.

When you select Option 3 on the System Diagnostics Menu, the following will be appeared. If you delete the option press d, if you append, press a.

Press 'D (lete)' or 'A (ppend)'

If you press d or a, the change menu will be displayed. Select your desired option.

HARD DISK

If your model is FT-80, your Exec. Partner executes hard disk test. The test proceeds as follows.

STEP 1 Press any key.

SCREEN DISPLAYS:

HARD DISK TEST

.....
 • WRITE TEST DESTROYS DATA ON CYLINDER 305 •

Do you want to include write test?(Y/N)

If you press Y, screen displays

DRIVE-C WRITE TEST

If you press N, write test will not be performed.

STEP 2 SCREEN DISPLAYS:

DRIVE-C DISK MEDIA TEST

DRIVE-C RANDOM SEEK TEST

DRIVE-C Test done

Press any key when ready

If your screen does not look this, and contains any error message, write down your procedure, any error messages displayed, and call your Panasonic dealer for service.

MEMO

STEP 3

SCREEN DISPLAYS:

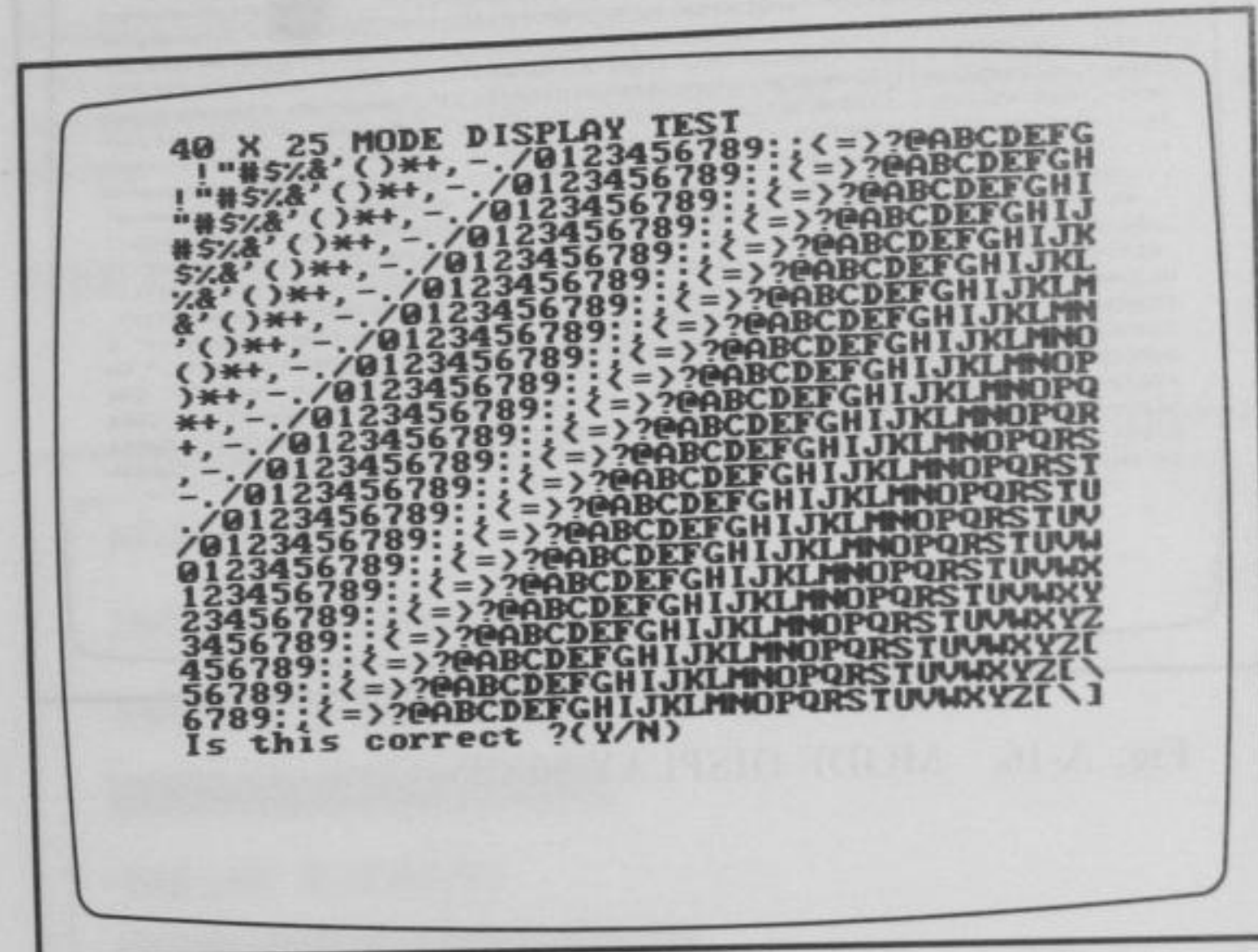


Fig. A-17. MODE DISPLAY 40×25

If your screen matches this display, press y and proceed to STEP 4.

If your screen does not look like this, press n and write down your procedure, any error messages displayed, and call your Panasonic dealer for service.

STEP 4

SCREEN DISPLAYS:

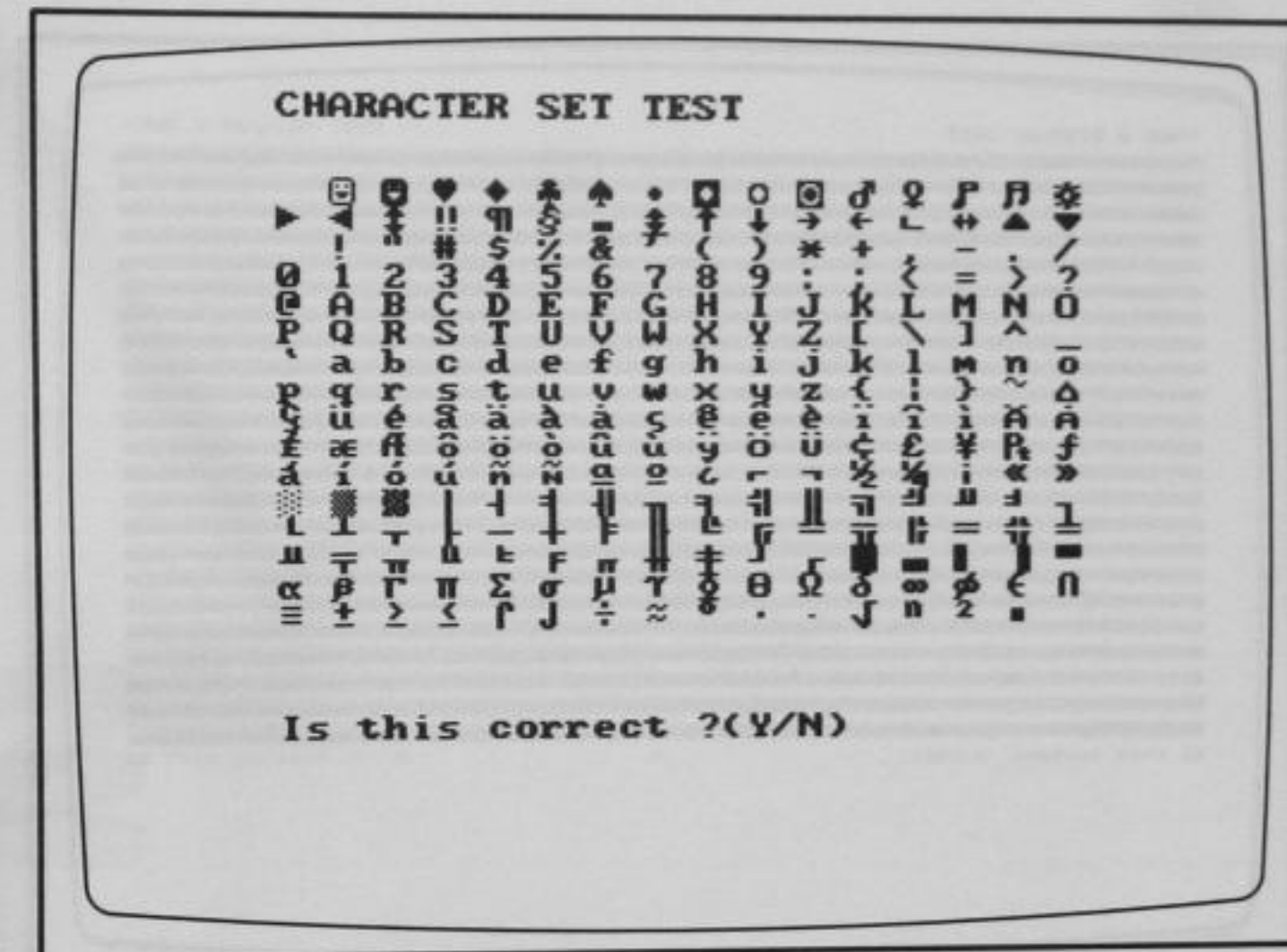


Fig. A-18. CHARACTER SET SCREEN

If your screen matches this display, press y and proceed to STEP 5.

If your screen does not look like this, press n and write down your procedure, any error messages displayed, and call your Panasonic dealer for service.

STEP 7 Brightness/color pattern is displayed.

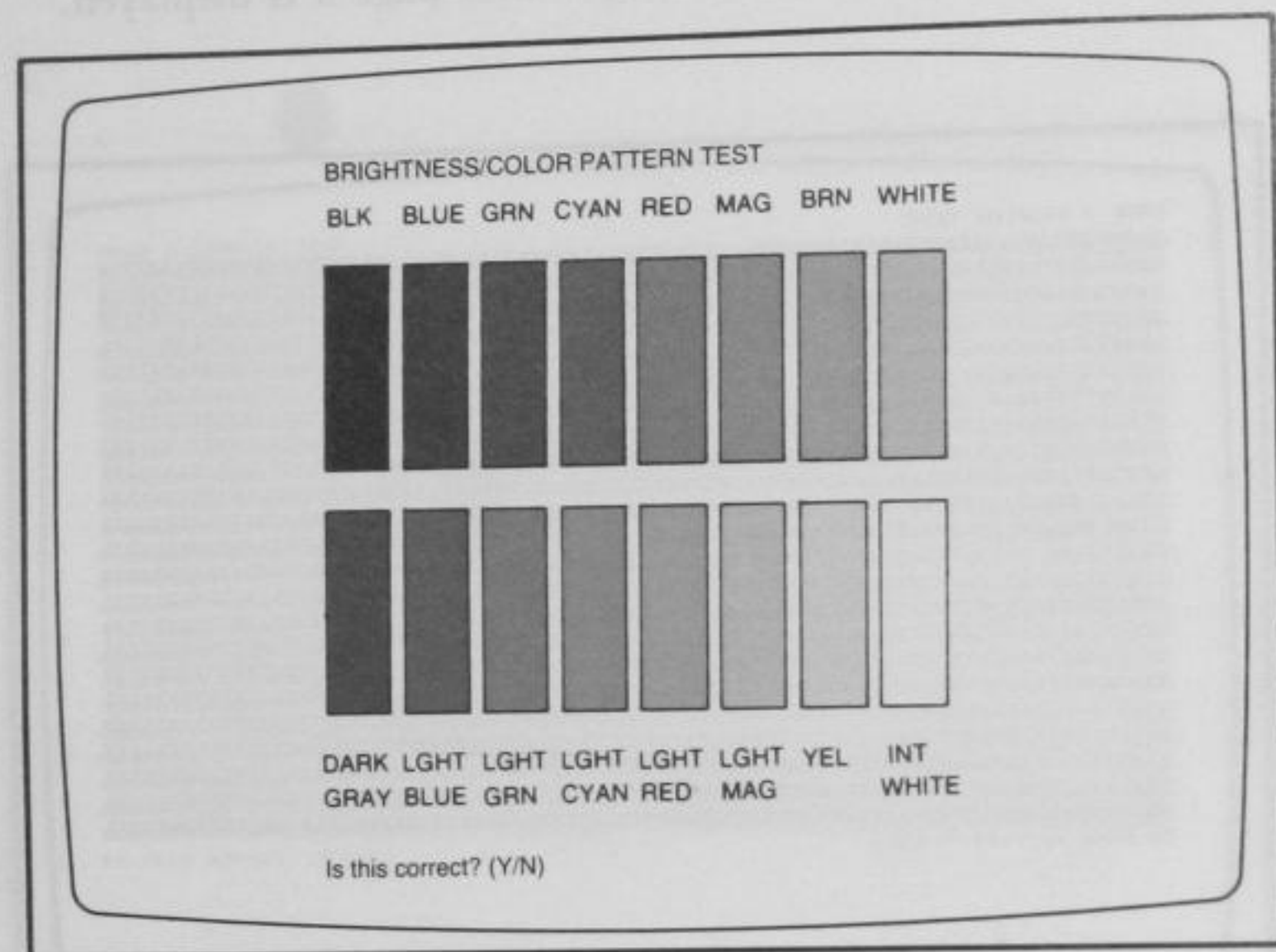


Fig. A-21. COLOR TEST PATTERN SCREEN

If your screen matches this display, press y and proceed to STEP 8.

If your screen does not look like this, press n and write down your procedure, any error messages displayed, and call your Panasonic dealer for service.

STEP 8

SCREEN DISPLAYS:

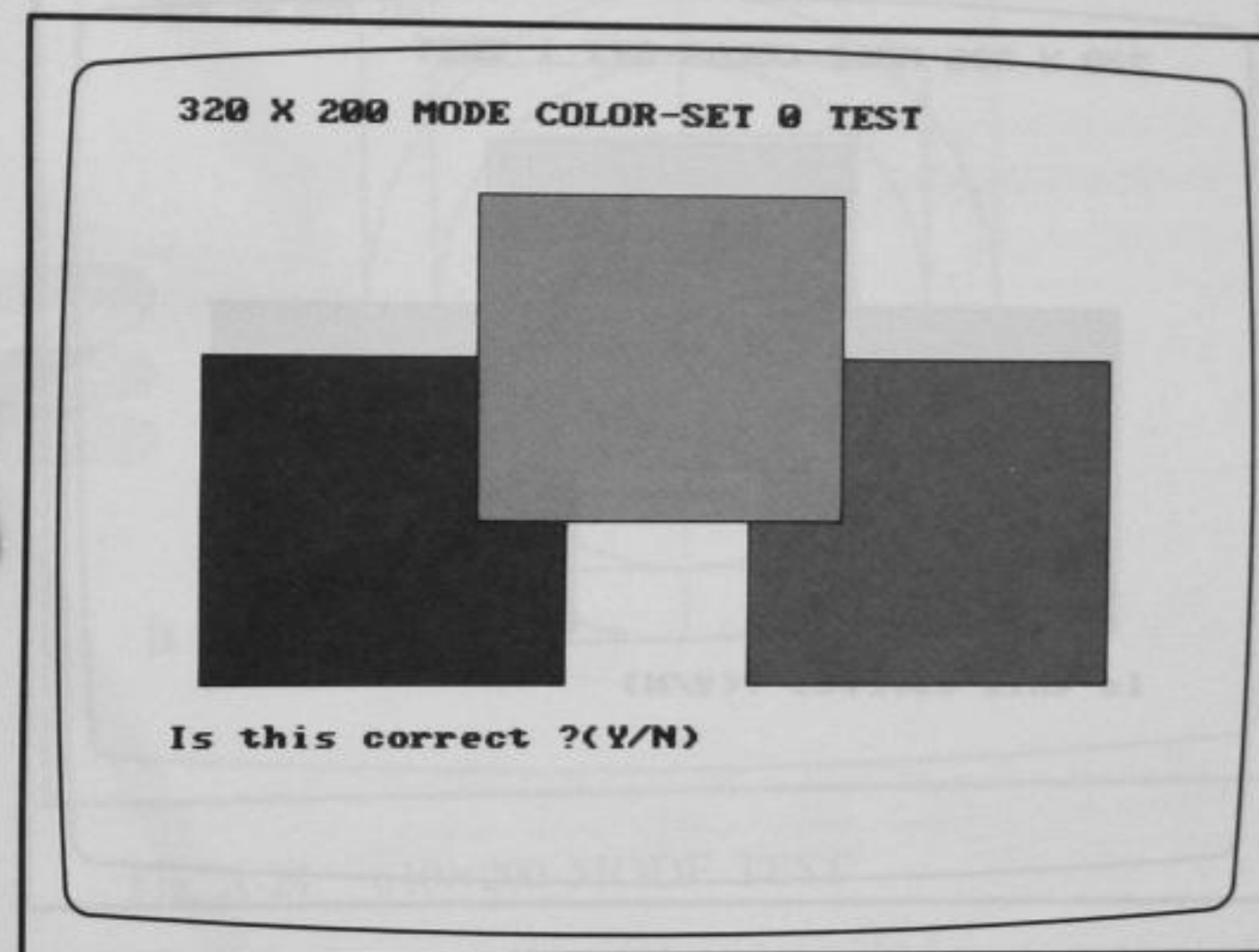


Fig. A-22. COLOR SET 0 TEST

If your screen matches the display, proceed to STEP 9.

If your screen does not look like this, write down your procedure, any error messages displayed, and call your Panasonic dealer for service.

STEP 9

SCREEN DISPLAYS:

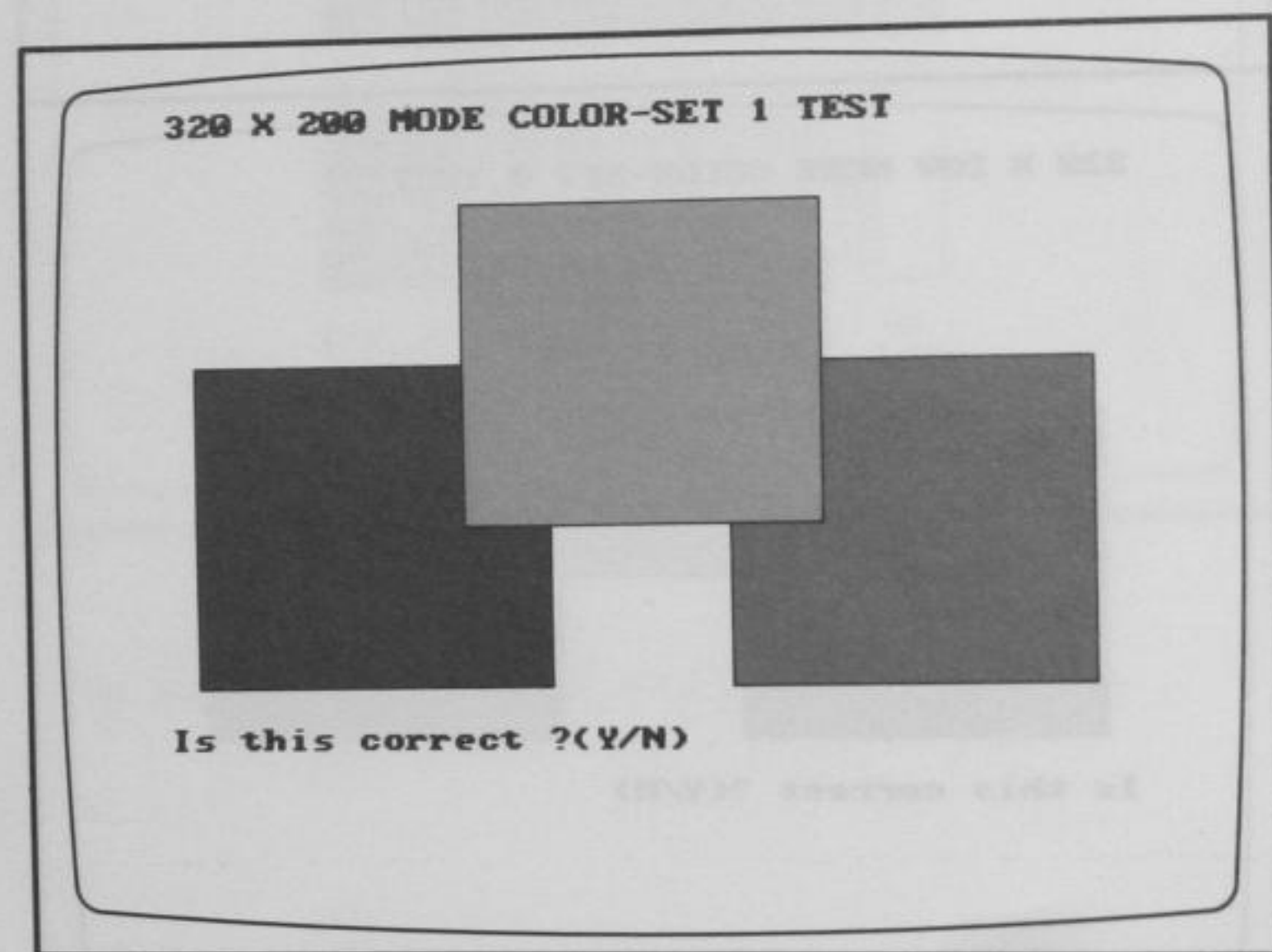


Fig. A-23. COLOR SET 1 TEST

If your screen matches this display, proceed to STEP 10.

If your screen does not look like this, write down your procedure, any error messages displayed, and call your Panasonic dealer for service.

STEP 10

SCREEN DISPLAYS:

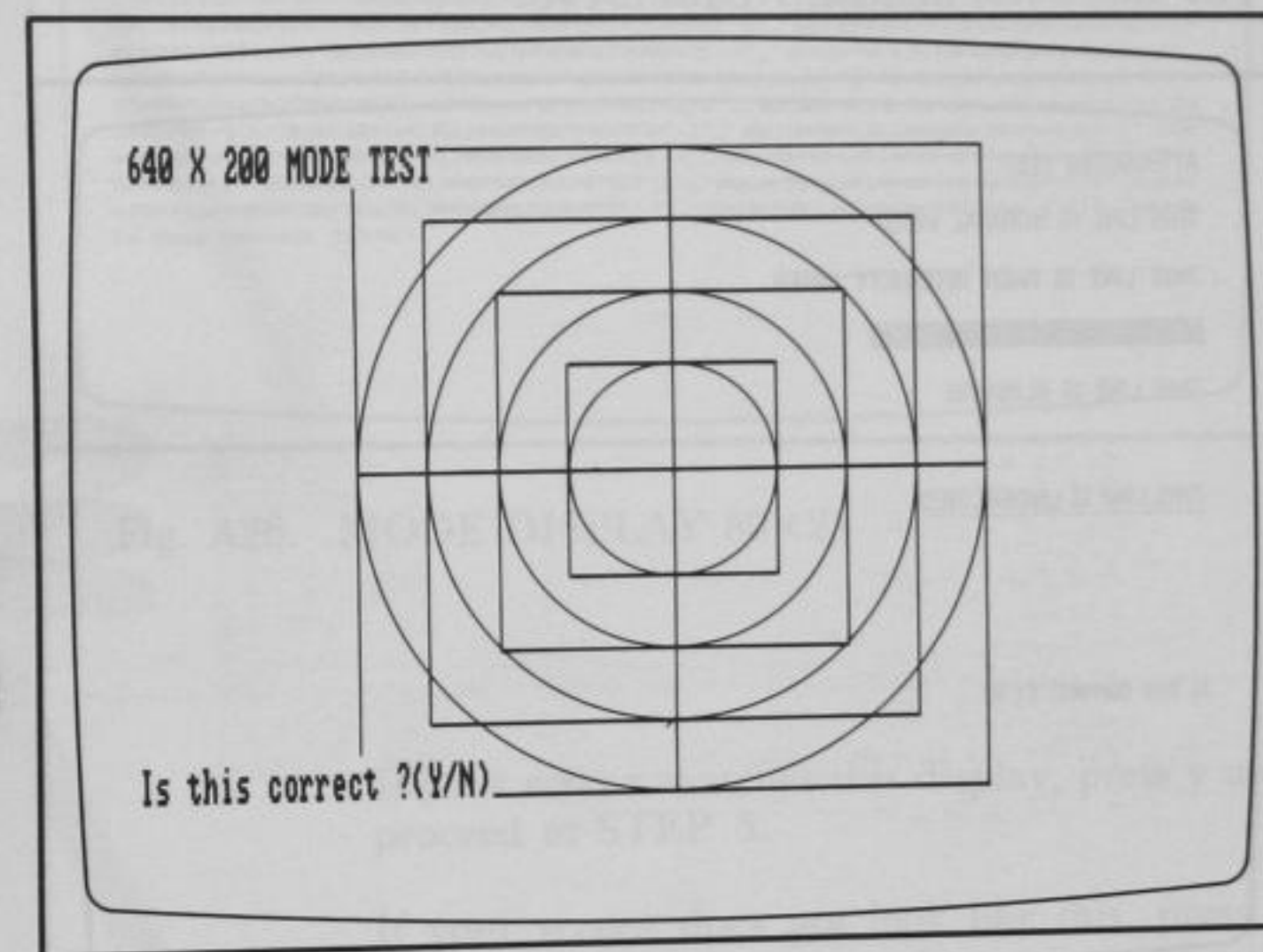


Fig. A-24. 640×200 MODE TEST

If your screen matches this display, proceed to next step or System Diagnostics Menu.

If your screen does not look like this, write down your procedure, any error messages displayed, and call your Panasonic dealer for service.

STEP 3

SCREEN DISPLAYS:

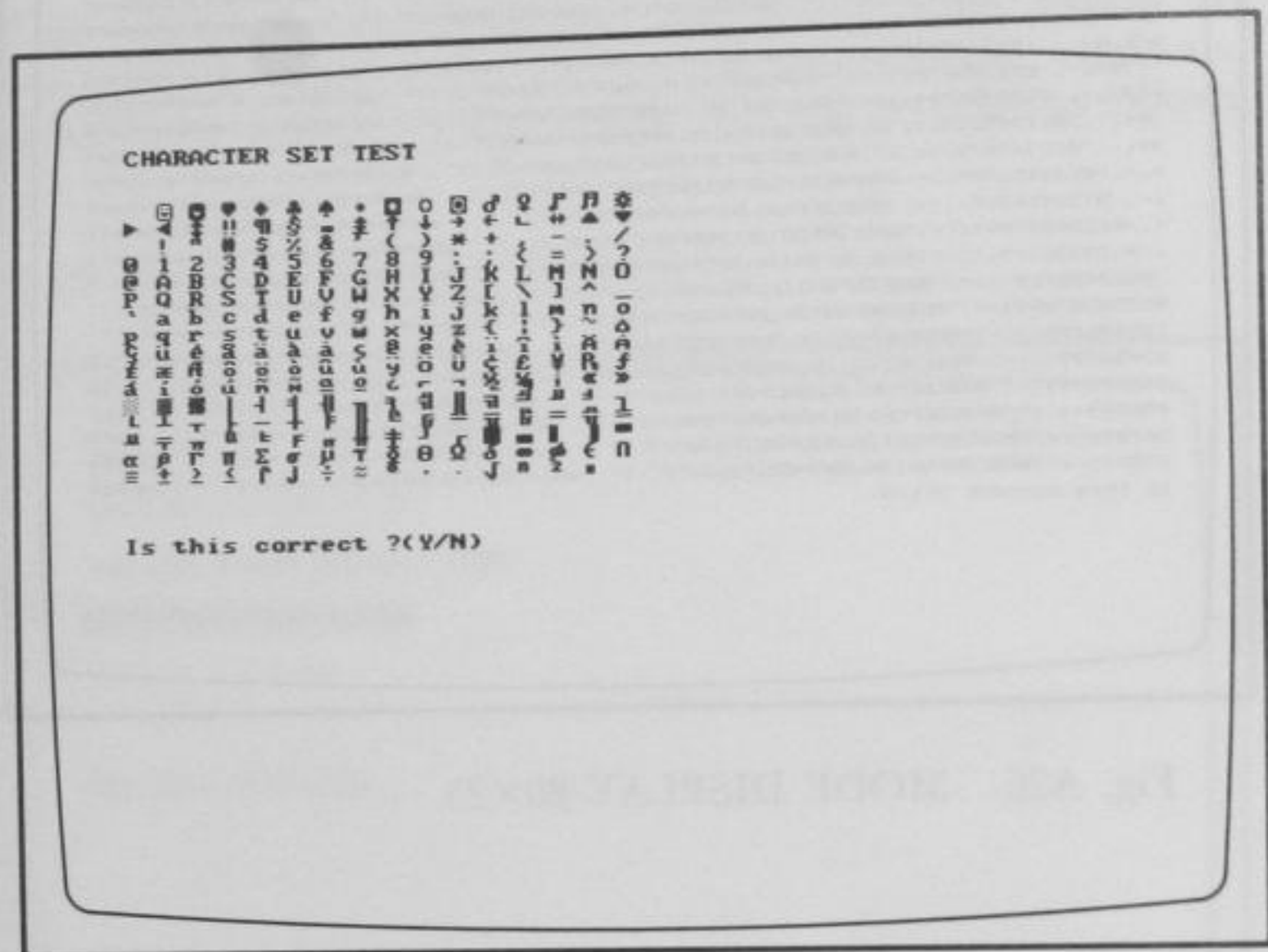


Fig. A-27. CHARACTER SET SCREEN

If your screen matches this display, press y and proceed to next step or System Diagnostics Menu.

If your screen does not look like this, press n and write down your procedure, any error messages displayed, and call your Panasonic dealer for service.

MEMO

MEMO



Fig. A-10 CHARACTER SET SCREEN

MEMO

DIAGNOSTICS MESSAGES

Various messages or error messages appeared in User Diagnostics are listed below.

To display not only the messages but also the code numbers, type in:

`diag/s <ENTER>`

The detailed status of a device can then be determined by looking up the code number in the code table.

hexadecimal Nibble consists of 4 bits as follows:

$$n = D_3D_2D_1D_0$$

For example,

$$n_1 = 5(0101) \text{ then } D_3=0, D_2=1, D_1=0, D_0=1$$

$$n_1 = D(1101) \text{ then } D_3=1, D_2=1, D_1=0, D_0=1$$

MAIN BOARD TEST

(1) LED TEST

LED ON/OFF TEST		
ALL	LEDs	OFF
Caps	Lock	ON
Num	Lock	ON
Fast	Speed	ON
Ext.	Printer	ON
All	LEDs	OFF

(2) DIP SWITCH STATUS

DIP SWITCH 1 ON... N/A
 2 ON... 64 K DRAM chip
 3 ON... Color/Graphics card
 4 ON... 4.77 MHz default
 5 ON... N/A
 6 ON... 8087 disable
 7 ON... 80 char. display
 8 ON... 2 FDD

(3) CPU CLOCK TEST

Fast Speed clock failed.

(4) OTHER MESSAGES

SYSTEM ERROR

System area cannot be saved into VRAM area.

PPI register failed

Port B register of Programmable Peripheral Interface (8255) has failed.

PIT register failed

Register of Programmable Interval Timer (8253) has failed.

PIT count test failed

Count of Programmable Internal Timer (8253) has failed.

DMAC register failed

Register of Direct Memory Access Controller (8237) has failed

DMACDRAM refresh failed

Refresh counter of Direct Memory Access Controller (8237) has failed.

DRAM failed

Dynamic RAM data has failed.

DRAM address failed

Dynamic RAM address has failed.

DRAM refresh failed

Dynamic RAM refresh has failed.

Page register failed

Page register (LS670) has failed.

PIC register failed

Register of Programmable Interrupt Controller (8259) has failed.

PIC handling failed

Handling of Programmable Interrupt Controller (8259) has failed.

KEYBOARD TEST

(1) KEYBOARD

KEY Keyboard failed

(2) KEY

KEY Key has been hit or broken

SCAN code: XX

VIDEO DISPLAY TEST

(1) VRAM

VIDEO VRAM failed

VRAM data has failed. Data bus or VRAM chip has failed.

Address: BXXX:XXXX Write:XXXX Read:YYYY

(2) VRAM ADDRESS BUS

VIDEO VRAM Address Bus failed

VRAM address has failed. Address bus or VRAM chip has failed.

Address: BXXX:XXXX

(3) OTHER MESSAGES

CRTC register failed

CRTC register has failed.

Display timing signal failed

Status has failed. Horizontal Sync. signal is not detected at address 3DAH.

FLOPPY DISK DRIVE TEST

(1) FDC FAILED

DISK FDC failed

Floppy Disk Controller (FDC) master status has failed. Data bus or FDC chip has failed.

Master status:XX

(2) FDC INIT. FAILED

DISK FDC Init. failed

Floppy Disk Controller (FDC) initialization has failed. Data bus or FDC chip has failed.

BIOS status:XX

XX is BIOS disk error status listed below.

BIOS disk error status

Status	Meaning
80	Time out
40	Seek error
20	FDC failed
10	CRC error
09	DMA boundary
08	DMA overrun
04	No record
03	Write protected
02	No address mark
01	Invalid command

(3) SEEK FAILED

DISK Seek failed

Seek has failed. Floppy Disk Controller (FDC) cable or FDC chip has failed.

FDC result status: $n_1n_2n_3n_4$

n_1n_2 is failed FDC status register #0 listed below.

nibble	bit	meaning
n_1	D_3, D_2	Interrupt Code 00: Normal termination 01: Abnormal termination 10: Invalid command 11: Attention interrupt
	D_1	Seek end
	D_0	Equipment check
n_2	D_3	Not ready
	D_2	Head address
	D_1	Unit select 1
	D_0	Unit select 0

n_3n_4 is a track number.

(4) READ/WRITE/FORMAT/VERIFY COMMAND FAILED

READ command failed

Read error. FDC, Floppy disk drive (FDD) or data separator (9420) has failed.

WRITE command failed

Write error. FDC, FDD or data separator (9420) has failed.

FORMAT command failed

Format error. FDC or FDD has failed.

VERIFY command failed

Verify error. FDC, FDD or data separator (9420) has failed.

TIME OUT
SEEK ERROR
FDC FAILED
CRC ERROR
DMA OVERRUN
NO RECORD
WRITE PROTECTED
NO ADDRESS MARK
INVALID COMMAND

HEAD=XX TRACK=XX SECTOR=XX
STATUS= $n_1, n_2, n_3, n_4, n_5, n_6, n_7$
 $n_1 \dots n_7$ is FDC status. This value
is only for service station.

(5) ROTATION

Rotation speed can't be measured

Drive-X DISK rotation speed is out of range

Rate of rotation period is out of valid range. (less than 96% or more than 104%)

Rotation speed=XXXX ms

*** WARNING ***

This status sometimes depends on a condition of a disk or disk drive. If the errors occurred, try to clean a head of the disk drive by a cleaning disk etc.

PARALLEL PORT TEST

(1) DATA REGISTER FAILED

PRT Int. }
PRT Ext. } parallel port failed
PRT Opt. }

Write:XX Read:YY

(2) ERROR STATUS

TIME OUT }
PAPER END } occurs with { Int. printer
I/O ERROR } { Ext. printer
 } { Opt. printer

BIOS status: n_1n_2 Character: n_3n_4

n_1n_2 is printer BIOS error status listed below

nibble	bit	meaning
n_1	D_3	0: Busy
	D_2	1: Acknowledge
	D_1	1: Paper end
	D_0	1: Printer is selected
n_2	D_3	1: I/O error
	D_0	1; Time Out

n_3n_4 is the character sent to a printer

SERIAL PORT TEST

(1) RS232 INTERNAL SERIAL PORT

RS232 Int. Serial Port failed

Serial port controller (8250) could not be found.

Write:XX Read:YY

(2) RS232 INTERNAL/EXTERNAL SERIAL PORT

RS232 Int. }
RS232 Ext. } Serial port failed

Divisor Write:XXXX Read:YYYY

Divisor register has failed

Signal loopback Modem Status: n_1n_2 Expected: n_3n_4

Loopback status has failed
 n_1 or n_3 : Refer to the following table

nibble	bit	meaning
n_1 or n_3	D_3	Carrier detect
	D_2	Ring indicator
	D_1	Data set ready
	D_0	Clear to send

Data loopback X mit:XX Rcvd:YY

Loopback data has failed

Data loopback Line & Modem Status: n_1n_2 , n_3n_4

Loopback status has failed
 $n_1n_2n_3n_4$: Refer to the following table.

nibble	bit	meaning
n_1	D_3	1: Time out
	D_2	1: Transmit shift register is empty
	D_1	1: Transmit holding register is empty
	D_0	1: Break detect
n_2	D_3	1: Framming error
	D_2	1: Parity error
	D_1	1: Overrun error
n_3	D_0	1: Data ready
	D_3	1: Carrier detect
	D_2	1: Ring indicator
	D_1	1: Data set ready
n_4	D_0	1: Clear to send
	D_3	1: Delta carrier detect
	D_2	1: Trailing edge ring indicator
	D_1	1: Delta data set ready
	D_0	1: Delta clear to send

Data real loopback X mit: XX Rcvd: YY

Data real loopback Line & Modem status: n_1n_2, n_3n_4

The controller and/or signal drivers/receivers have failed.

$n_1n_2n_3n_4$: Refer to the before table.

Signal real loopback Modem status: n_1n_2 Expected: n_3n_4

The controller and/or signal drivers/receivers have failed
 n_1 or n_3 : Refer to the following table.

nibble	bit	meaning
n_1 or n_3	D_3	Carrier detect
	D_2	Ring indicator
	D_1	Data set ready
	D_0	Clear to send

REAL TIME CLOCK TEST

Realtime Clock failed

Realtime Clock count failed

Current Setting: Date=DD-MM-YYYY Time=XX:XX:XX

HARD DISK DRIVE TEST

If your Exec. Partner has a hard disk drive, these messages appeared in User Diagnostic are listed below.

HDC BIOS checksum failed
Hard Disk BIOS checksum has failed

BIOS ROM ID could not be found

HDC BIOS checksum failed

Failed checksum: XX

HDC reset failed

BIOS status: XX HDC status: YY

HDC internal diagnostics failed
Controller failed

BIOS status: XX HDC status: YY

HDC RAM diagnostics failed
Controller RAM failed

BIOS status: XX HDC status: YY

HDC RAM write/verify failed
Controller RAM write/verify failed

BIOS status: XX HDC status: YY
If XX is zero, YY will be address on RAM.
If XX is not zero, XX will be BIOS error which is listed in Table A-1 status and YY will be controller error status.

HDISK diagnostics failed
Drive failed

BIOS status: XX HDISK status: YY

HDISK drive not ready
Drive not ready

BIOS status: XX HDISK status: YY

HDISK seek & recalibrate failed
Seek or recalibrate failed

BIOS status: XX HDISK status: YY

HDISK Init. failed
Initialization of drive characteristics failed

BIOS status: XX HDISK status: YY

NOTE: BIOS status is listed in Table A-1.

Read command failed
Write command failed
Read long command failed
Write long command failed
Verify command failed

SENSE FAILED
UNDEFINED ERROR
TIME OUT
SEEK ERROR
HDC FAILED
DATA CORRECTED
ECC ERROR
BAD TRACK
DMA BOUNDARY
INIT FAILED
RESET FAILED
NO RECORD
NO ADDRESS MARK
INVALID COMMAND

HEAD=XX TRACK=XXXX SECTOR=XX
STATUS=XX, XX, XX, XX, XX XX

HEAD=XX: XX is head number.
 TRACK=XXXX: XXXX is track number.
 SECTOR=XX: XX is sector number.
 STATUS=XX, XX...XX: XX, XX...XX is function code.

COMPARE ERROR

HEAD=XX TRACK=XXXX SECTOR=XX STATUS=XX, XX...XX

HEAD=XX: XX is head number.
 TRACK=XXXX: XXXX is track number.
 SECTOR=XX: XX is sector number.
 STATUS=XX, XX...XX: XX, XX...XX is function code.

Status	Meaning
FF	Sense status failed
BB	Undefined error
80	Time out
40	Seek error
20	Controller failed
11	ECC corrected data error
10	ECC error
0B	Bad track
09	DMA boundary
07	Drive parameter error
05	Reset failed
04	No record
02	No Address mark
01	Invalid command

Table A-1

APPENDIX B

ERROR MESSAGES

1. INTRODUCTION B-2
2. DEVICE ERROR MESSAGES B-3
3. OTHER MESSAGES B-6

INTRODUCTION

APP. B

There are two parts to this chapter.

Device Error Messages

Messages written by DOS to indicate errors while reading from or writing to devices.

Other Messages

The other DOS messages are written in bold type with a description following the message and are presented in alphabetic order.

The program name or command which created the message appears as the first word of the description.

DEVICE ERROR MESSAGES

APP. B

A message will be displayed on your system when an error is detected while reading from or writing to any of the system devices, such as disk drives, printer, etc. The following format is used:

type error writing *device* Abort, Retry, Ignore?
or
type error reading *device* Abort, Retry, Ignore?

device is the name of the device in error, such as CON, or drive A, drive B etc.

type is one of the following error types:

Bad call format

The length request header passed to a device driver is wrong. Contact your dealer for assistance.

Bad command

A command issued to a *device* by a device driver is incorrect.

Bad unit

The sub-unit number passed to a device driver is invalid. Contact your dealer for assistance.

Data

Data was written or read incorrectly. Your disk may have developed a defective spot.

Disk

A disk error has occurred which was not previously described.

No paper

The named printer is either not turned on or lacks paper.

Non-DOS disk

There is invalid information in the file allocation table.

Not ready

The indicated device is not ready. It cannot receive or send data.

Read Fault

The data was not successfully read.

Sector not found

The sector which is indicated as containing the data could not be located. The disk may have developed a defective spot.

Seek

The proper track on the disk could not be located.

Write fault

The data to the device was not successfully written from the device.

Write protect

An attempt was made to write on a write protected disk.

***** WARNING *****

If any of these messages appear concerning a disk drive, DO NOT remove or insert disks until you respond to the following device error message.

Abort, Retry, Ignore?

What To Do When A Device Error Message Appears:

The system will wait for one of the following responses:

A for Abort. The system ends the program that requested the device read or write.

R for Retry. The system will try again to do the device read or write operation.

I for Ignore. The system processes the program as if the error had not happened.

The following order is suggested in responding:

R Retry the operation. The problem may not happen again.

A Abort the program.

I Ignore the problem and continue processing. Be aware that data is lost when you use the Ignore response.

OTHER MESSAGES

APP. B

A

About to generate .EXE file
Change disks <hit ENTER>

LINK. When you specify the /PAUSE parameter, this is displayed. Put the Runfile disk into the appropriate drive and press <ENTER>.

All files canceled by operator

PRINT. This message appears when you cancel the printing of all queued files using the /T parameter. It is printed out.

All specified file(s) are contiguous

CHKDSK. All the files you requested to write are on the disk sequentially.

Allocation error, size adjusted

CHKDSK. An invalid sector number was in the file allocation table. The file whose name appears before this message, was truncated at the end of the last good sector.

Attempt to access data outside of segment bounds, possibly bad object module

LINK. An object file is most likely incorrect.

Attempted write-protect violation

FORMAT. An attempt was made to format a write-protected disk. Insert a new disk and press a key to begin formatting.

B

Bad command or file name

DOS. You have entered a DOS command incorrectly. Check your syntax and re-try the command. If the command is correct, be sure that the default drive contains either the external command or batch file you are trying to execute.

Command Interpreter
Bad or missing

DOS. The file COMMAND.COM was not contained in the disk being booted, or an error was found during loading the disk.

Bad numeric parameter

LINK. An invalid numeric constant was specified with the /STACK parameter.

BF Error

DEBUG. Bad Flag. An incorrect flag code setting was entered. See the Register command for the valid codes.

APP. B

BP Error

DEBUG. Too many breakpoints. No more than ten breakpoints may be specified for the Go command. Retype the Go command with ten or fewer breakpoints.

BR Error

DEBUG. Bad register. An incorrect register name was specified. See the Register command for the valid register names.

C

Cannot do binary reads from a device

COPY. /B switch was used with a device name during copying from the device.

Cannot edit .BAK file--rename file

EDLIN. .BAK files shouldn't be edited because these files are assumed to be backup files, with more up-to-date versions of the files.

To edit the .BAK file either rename the file, or copy it and give the copy another name.

Cannot find file xxxx.OBJ
Change disk <hit ENTER>

LINK. The specified object module was not found on the drive by linker. Insert the proper disk with the named module on it and press <ENTER>.

Cannot find library xxxx.LIB
Enter new drive letter:

LINK. The named library could not be located on the drive. Enter the proper letter for the drive the library is on.

Cannot open temporary file

LINK. The temporary file cannot be opened because the disk or the directory is full.

COMn: bbbb, p, d, s, t

MODE. The Serial (RS232C) Interface was initialized. The parameters are:

n COM1 or COM2

bbbb baud rate

- p parity
- n none
- e even
- o odd
- s stop bits (1 or 2)
- t type of serial device
 - p serial printer (serial timeouts are retried)
 - other serial device (serial timeouts will not be retried)

Compare error(s) on Track xx, Side y

DISKCOMP. Contents on the track number xx on the side y were different.

Contains xxx non-contiguous blocks

CHKDSK. The file named has been written in sections on different areas of the disk rather than sequentially. However, since badly fragmented file take longer to read, copying the file sequentially could improve run time and enhance system performance.

DF Error

DEBUG. Double flag. There were conflicting codes denoted for a single flag. A flag may be altered only once per Register command.

Disk boot error

DOS. This disk was not for booting DOS. Insert the System Disk or other disk for booting and restart.

Disk error writing FAT X

CHKDSK. A disk error occurred while CHKDSK was trying to update the file allocation table (FAT) on the named drive. X (1 or 2) depends on which of the 2 copies of the file allocation table could not be written. If the message is for FAT 1 and 2, the disk is not usable.

Disk full—write not completed

EDLIN. The disk does not have enough free space to save the entire file; therefore an End Edit command terminated abnormally. Part of the file may be on the disk, but the file will be incomplete.

Disk unsuitable for system disk

FORMAT. The system program could not be written onto the disk being formatted.

Duplicate filename or File not found

RENAME. While attempting to rename a file, the new filename either already existed on the disk, or the file to be renamed was not found on the specified (or default) drive.

E

Entry error

EDLIN. There is a syntax error on the last command.

Error: Dup record too complex

LINK. An object module created from an assembler program has a DUP record which is too complex. To correct this problem you must debug the assembler program and then LINK it again.

Error found, F parameter not specified.
Corrections will not be written to disk.

CHKDSK. Even though the /F parameter was not used, CHKDSK will perform its analysis so that you can see the results, but the corrections will not actually be written on the disk.

Error writing to device

Commands. DOS could not write data to the device because the data was more than the device was expecting.

EXE and HEX files cannot be written

DEBUG. DEBUG cannot convert data which requires a backwards conversion.

F

File cannot be copied onto itself

DOS. When a request is made to COPY a file a different name should be given to the copy or it should be put in a different directory or on another disk.

File creation error

DOS and commands. You tried to add a new filename to the directory or to replace a file that was already there. If the file already existed, it was marked read-only and could not be replaced. If you run CHKDSK, you will see if the directory is full, or is something else caused to error.

File is cross-linked: on cluster XX

CHKDSK. This message renames the two files in error and appears twice for each cross-linked cluster number. Both files are allocated to the same data block. Since no corrective action is taken automatically, you must correct the problem by:

1. Copy both files using the COPY command.
2. Delete the original files using the ERASE command.
3. Review the files and edit.

File not found

DOS and commands. A file listed in either a command parameter or command could not be found in the directory on the specified (or default) drive.

First cluster number is invalid,
entry truncated.

CHKDSK. There is an invalid pointer to the data area on the file named before this message. If the /F parameter is specified, the file is truncated to a zero-length file.

Fixup offset exceeds filed width

LINK. A NEAR attribute instead of a FAR attribute was indicated in an assembler instruction address. You must edit the assembler source code and reprocess again.

Fix-ups needed—base segment (hex):

EXE2BIN. A load segment was indicated as required for the file in the source (.EXE) file. You must indicate the absolute segment address for loading the finished module.

*** WARNING ***

Because the program will be dependent upon being loaded at a specific memory location, we do not recommend using such a program as a .COM file.

FOR cannot be nested

BATCH. There are more than one FOR command on the command line.

Format failure

FORMAT. While creating the target disk, a disk error occurred. This disk is unusable, or a write-protected disk.

I

Illegal device name

MODE. The named printer must be LPT1:, LPT2:, or LPT3:; the named RS232C Interface must exist and be COM1: or COM2:. No more than one blank is allowed between MODE and its parameters.

Incompatible disk or drive types

DISKCOMP. The number of sectors or sides of disks being compared is different.

Incompatible system size

SYS. A system transfer did not occur because the disk being copied contains a larger copy of DOS than the target disk. One solution is to format a blank disk using the **FORMAT /S** command and then copy the files you want to the new disk.

Incorrect DOS version

Commands: In order to use the command you just entered, a different version of DOS is needed than the one you are running.

Input file read error

LINK. An object file is probably bad.

Insufficient disk space

DOS and commands. There is not enough free space on the disk to hold the file being written. If you think there is enough space, run **CHKDSK** to determine the status of the disk.

Insufficient memory

Commands. These commands need more memory than is available to function. Change the **BUFFERS=**parameter in the **CONFIG.SYS** file to a smaller value (if you have specified **BUFFERS=**), restart the system and retry the command. If the message appears again, your system does not contain enough memory to execute this command.

Insufficient room in root directory

Erase files from root and repeat **CHKDSK**.

CHKDSK. While creating files from the "lost" data blocks, **CHKDSK** found the root directory full; therefore all of the lost chains could not be recovered into files. Copy some of the recovered files to another disk for review, delete them from the disk you are checking, and rerun **CHKDSK** to recover the rest of the lost data.

Insufficient space on disk

DEBUG. There is not enough free space to hold the data being written on disk. You may insert a disk that does have enough free space and re-issue the Write command. Otherwise, erase the files from the disk and rerun **DEBUG**.

Invalid baud rate specified

MODE. The only valid baud rates are 110, 150, 300, 600, 1200, 2400, 4800, or 9600. You may even enter just the first two characters of the number.

Invalid characters in volume label

FORMAT. It is not a valid filename character.

Invalid date

DATE. Either you entered an invalid date or the delimiter was not a hyphen (-) or slash (/).

Invalid directory

DOS and commands. There isn't the directory in the path you specified.

Invalid drive in search path

DOS. There was an invalid drive specification in the path you specified in the PATH command.

Invalid drive or file name

EDLIN. When starting EDLIN you did not specify a valid drive or filename.

Invalid drive specification

DOS and commands. The drive specified in a command or command parameter is invalid.

Invalid number of parameters

Commands. The number of parameters you specified does not agree with the number of parameters the command requires.

Invalid object module

LINK. The object module is either incomplete or it is incorrectly formed. This can occur when a language processor is stopped while it is processing.

Invalid parameters

DOS and Commands. A command request was made with an incorrect parameter. One of the things to check is that the drive specifier includes a colon. Check all parameters for invalid characters.

Invalid path, not directory,
or directory not empty

RMDIR. The directory you specified did not exist, or the directory you wish to remove still has files or other sub-directories.

Invalid sub-directory entry

CHKDSK. There is invalid information in the sub-directory. **CHKDSK** will try to correct the error. Run **CHKDSK** with the **/V** parameter for more information about the error.

Invalid time

TIME. There is an invalid time or delimiter. A colon between the hours and minutes, and the minutes and seconds; and a period between the seconds and hundredths of a second are the only valid parameters.

L

Label not found

BATCH. The label to which **GOTO** command jumps does not exist.

Line too long

EDLIN. The Replace Text command has ended abnormally because the replacement causes the line to expand beyond the 253-character limit. Break the long line into shorter lines, and reissue Replace Text command.

List output is not assigned to a device

PRINT. The device named as the **PRINT** list device is invalid. Re-issue the **PRINT** command and give a valid list device name when prompted.

M

Must specify destination line number

EDLIN. A destination line number must be entered with a Move or Copy command. Enter the command again but enter a valid destination line number.

N

Name of list device [PRN]:

PRINT. When you start print after DOS has been restarted, reply with the reserved device name which is to receive the printed output, or press **<ENTER>** to use the first parallel printer [PRN].

No Path

PATH. There is no path after DOS searches your working directory.

No room for system on destination disk

SYS. The system cannot be transferred because the destination disk does not contain the required reserved space for DOS. One solution is to format a blank disk using the `FORMAT /S` command, and copy your other files to the new disk.

No room in directory for file

EDLIN. The file directory is full.

Non-System disk or disk error
Replace and strike any key when ready

Startup. The directory contains no entries for system files or there was a disk read error while starting up the system. Place a DOS disk in drive A: and restart your system.

Not enough room to merge the entire file

EDLIN. The entire contents of the specified file was not merged while issuing a Transfer command because there is insufficient memory. Only part of the file was merged.

Not found

EDLIN. While using a Replace text or Search text command the range of lines indicated does not contain the string being searched for or no further occurrences of the string were found after replying "N" to the "OK?" prompt.

Out of environment space

DOS. DOS could not execute the SET command since the area of the environment information could not be expanded.

P

PRINT queue is empty

PRINT. No files are being processed by PRINT.

PRINT queue is full

PRINT. There is a limit of 10 files that can be added to the print queue. After a file is printed, you can add another file to the print queue.

Printer error

MODE. Printer is not connected, or printer paper is empty etc.

Probable non-DOS disk.
Continue (Y/N)?

CHKDSK. There is invalid information in the file allocation table identification byte. The disk is either not formatted by DOS or has become damaged. CHKDSK will indicate its possible corrective actions without actually changing the disk if you have not used the /F parameter and a "Y" reply. It is recommended that you do this first, before you try using the /F switch and replying Y.

Program too big to fit in memory

DOS. The file for the command cannot be transferred in the memory since the file is too big to fit in it.

Program size exceeds capacity of LINK

LINK. Load module is too large for execution.

R

Read error in: <filename>

DOS and commands. Commands could not read the entire file.

Resident part of PRINT installed

PRINT. Resident part of the PRINT command were transferred into memory for printing. This message is displayed. When you use the PRINT command for the first time.

S

Segment size exceeds 64K

LINK. The addressing limit is 64K bytes. The LINK attempted to combine segments with the same name which resulted in a segment requirement greater than 64K bytes.

Stack size exceeds 65535 bytes

LINK. A stack size specified with /STACK must be less than 65535 bytes.

Symbol defined more than once

LINK. A single symbol name may not be defined in more than one module.

Symbol table capacity exceeded

LINK. The names exceeded 50K bytes. Use either fewer and/or shorter names.

Syntax error

DOS. An illegal string was typed.

T

Target disk may be unusable

DISKCOPY. The target disk may be unusable because of unrecoverable read, write or verify error.

Terminate batch job (Y/N)?

DOS. If you press Ctrl-Break while DOS is processing a batch file, this message appears. To stop processing the batch file press Y. If you press N, only the command that was executing when Ctrl-Break was pressed will stop and processing will continue with the next command in the batch file.

Too many external symbols in one module

LINK. There is a limit of 256 external symbols per module.

Too many groups

LINK. There is a limit of 10 groups, including DGROUP.

Too many libraries specified

LINK. There is a limit of eight libraries.

Too many public symbols

LINK. There is a limit of 1024 public symbols.

Too many segments or classes

LINK. There is a limit of 247 segments and classes, taken together.

Track 0 bad—disk unusable

FORMAT. The boot record, FAT and directory could not be written on the disk being formatted.

Tree past this point not processed

CHKDSK. Because track 0 is bad, CHKDSK cannot continue processing the directory path being examined.

U

Unable to create directory

MKDIR. The directory you specified already exists, the directory path name could not be found or there was no room on the disk to create the directory.

Unable to write BOOT

FORMAT. The boot record could not be written on the track 0 of the disk.

Unable to write FAT

FORMAT. The FAT (File Allocation Table) could not be written on the track 0 of the disk.

Unexpected end-of-file on VM.TMP

LINK. The LINK could not be completed because the diskette with VM.TMP was removed.

Unrecognized command in CONFIG.SYS

Startup. An illegal command was found in the file CONFIG.SYS. Correct the illegal command by EDLIN.

Unrecoverable read error on source (target)-Track xx, side y

DISKCOPY, DISKCOMP. DISKCOPY or DISKCOMP could not read contents on the track XX on the side Y of the drive source (target) disk.

Unresolved Externals:

LINK. The modules or library files specified did not contain the definitions of the external symbols. When this error occurs, do not run the executable file created by the linker.

W

Warning: no STACK segment

LINK. A statement allocating stack space was not specified in any of the object modules.

WARNING—Read error on EXE file.

EXE2BIN. An error was detected during reading the input file. The result file may be unusable.

APPENDIX C

PIN CONFIGURATIONS

PARALLEL PORT (Centronics)	C-2
SERIAL PORT (RS232C)	C-4
EXPANSION BUS PORT	C-6
I/O CHANNEL DIAGRAM	C-10
I/O ADDRESS MAP	C-14

PARALLEL PORT (Centronics)

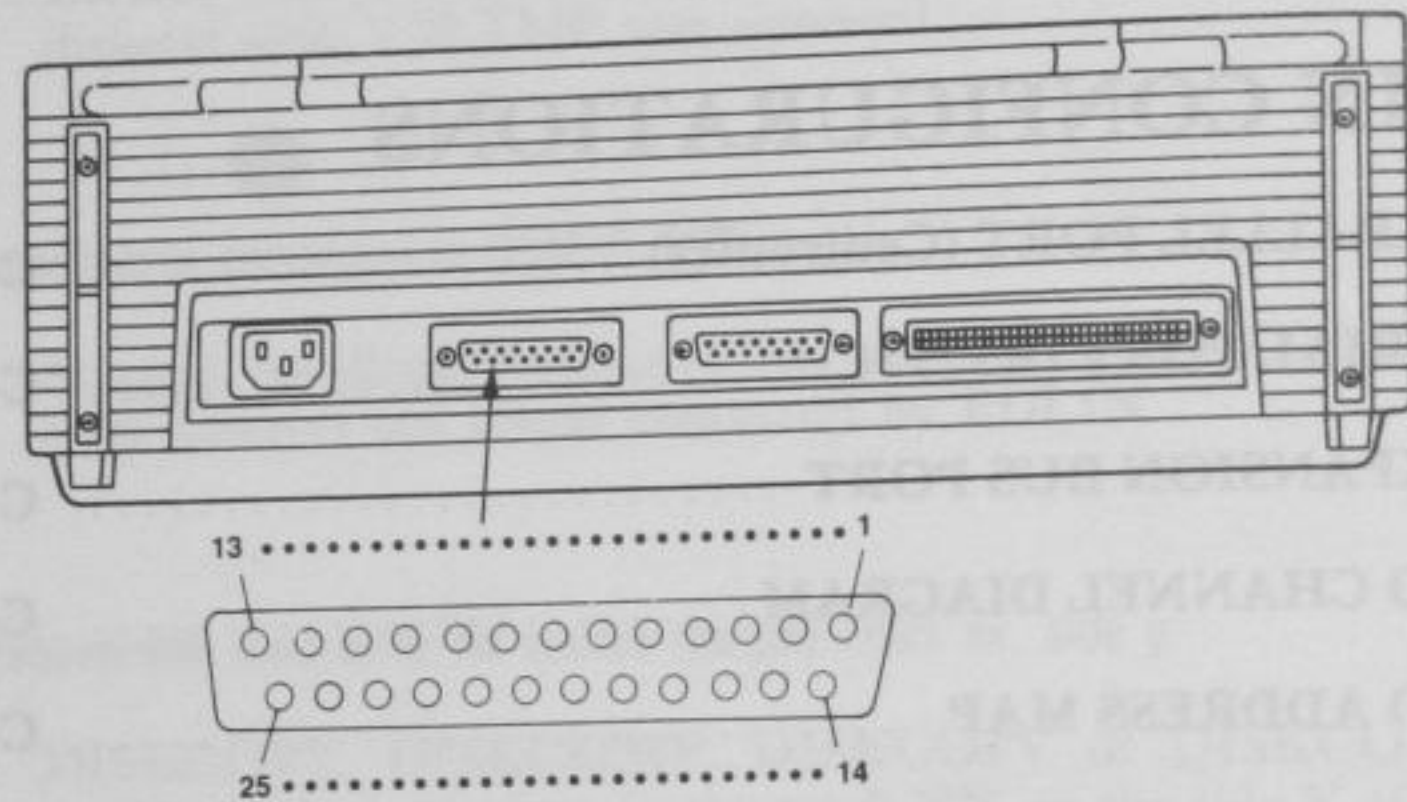


Fig. C-3. PIN ARRANGEMENT FOR PRINTER

Pin Number	Signal/Description	Direction
1	$\overline{\text{STROBE}}$ (STROBE Pulse to read data in)	Out
2	DATA 0	Out
3	DATA 1	Out
4	DATA 2	Out
5	DATA 3	Out
6	DATA 4	Out
7	DATA 5	Out
8	DATA 6	Out
9	DATA 7	Out
10	$\overline{\text{ACKNLG}}$ (Acknowledge data reception)	In

Pin Number	Signal/Description	Direction
11	BUSY (Printer cannot receive data)	In
12	PE (out of paper)	In
13	SLCT (Printer in Selected state)	In
14	AUTO FEED (Paper feed automatically)	Out
15	$\overline{\text{ERROR}}$ (Error State)	In
16	INIT (Clear the print buffer)	Out
17	SLCT IN (Data Entry is Possible)	Out
18 to 25	GND	—

The parallel port in Exec. Partner uses the address 378-37F. If you wish to use other parallel printer interface boards, refer to "I/O ADDRESS MAP" in this APPENDIX.

*Use a **shielded** cable to connect the parallel port and your equipment with a parallel port.

SERIAL PORT (RS232C)

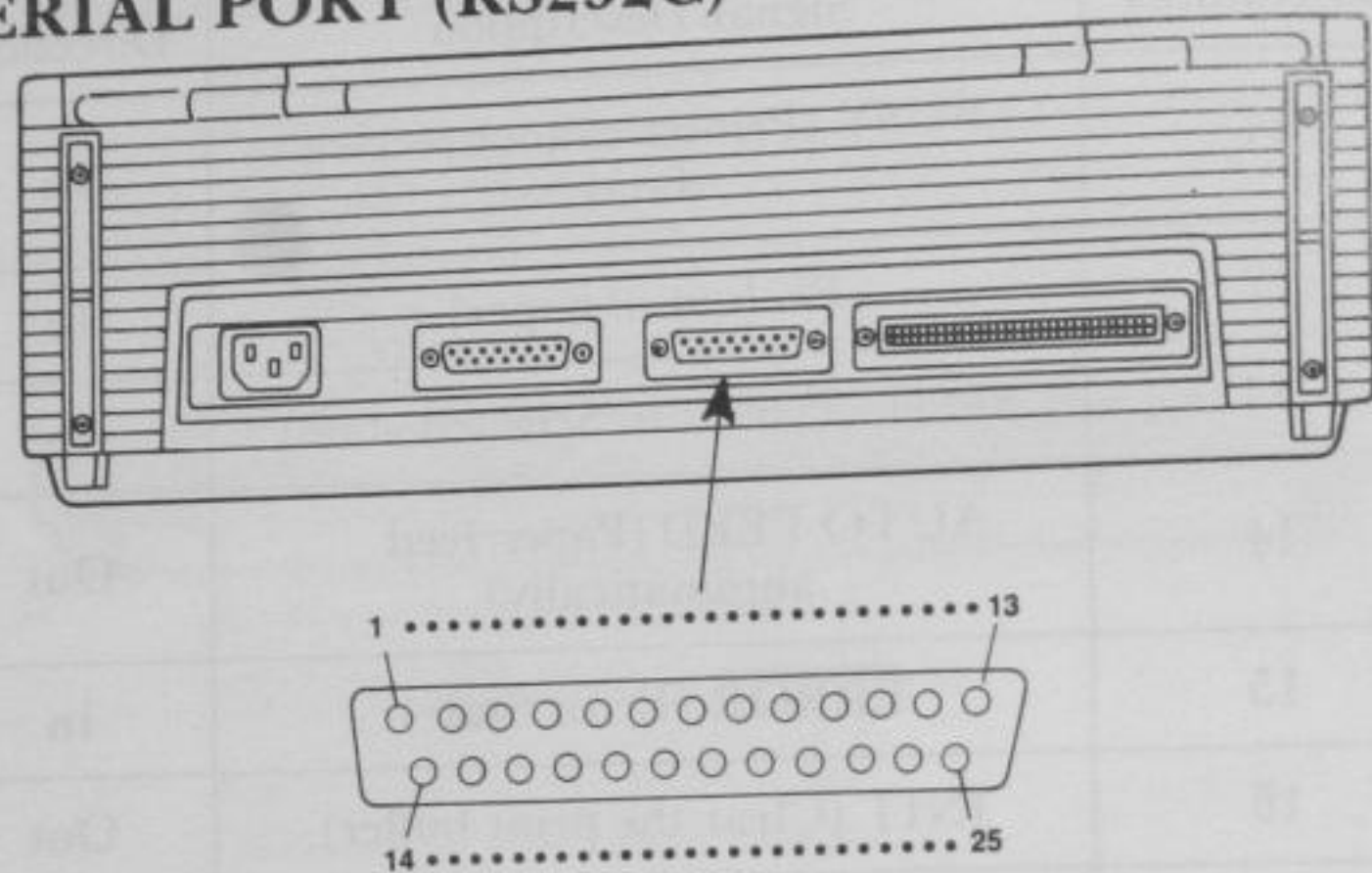


Fig. C-2. PIN ARRANGEMENT FOR SERIAL INTERFACE

Pin Number	Signal Name/Description	Direction
1	NC	---
2	Transmitted Data	Out
3	Received Data	In
4	Request to Send	Out
5	Clear to Send	In
6	Data Set Ready	In
7	Signal Ground	---
8	Carrier Detect	In
9	NC	---

Pin Number	Signal/Description	Direction
10	NC	---
11	NC	---
12	NC	---
13	NC	---
14	NC	---
15	NC	---
16	NC	---
17	NC	---
18	NC	---
19	NC	---
20	Data Terminal Ready	Out
21	NC	---
22	Ring Indicator	In
23	NC	---
24	NC	---
25	NC	---

The serial port in Exec. Partner uses the address 3F8-3FF. If you wish to use other asynchronous adapter boards, refer to "I/O ADDRESS MAP" in this APPENDIX.

*Use a **shielded** cable to connect the serial port and your equipment with a serial interface.

EXPANSION BUS PORT

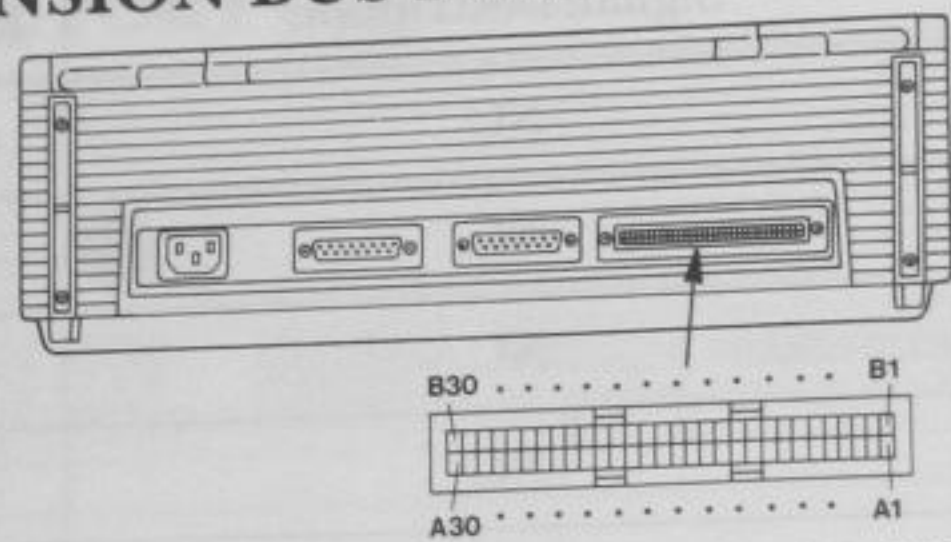


Fig. C-3. PIN ARRANGEMENT FOR EXPANSION BUS PORT

Pin Number	Signal/Description	Direction
B1	SG	—
B2	RESET DRV	Out
B3	SG	—
B4	IREQ 2	In
B5	DATA 4	In/Out
B6	DATA 3	In/Out
B7	DATA 2	In/Out
B8	DATA 1	In/Out
B9	DATA 0	In/Out
B10	$\overline{\text{MEMW}}$	Out
B11	$\overline{\text{MEMR}}$	Out
B12	$\overline{\text{IOW}}$	Out
B13	$\overline{\text{IOR}}$	Out
B14	$\overline{\text{DACK 3}}$	Out
B15	DRQ 3	In

Pin Number	Signal/Description	Direction
B16	$\overline{\text{DACK 1}}$	Out
B17	DRQ 1	In
B18	$\overline{\text{DACK 0}}$	Out
B19	CLOCK	Out
B20	IRQ 7	In
B21	IRQ 6	In
B22	IRQ 5	In
B23	IRQ 4	In
B24	IRQ 3	In
B25	$\overline{\text{DACK 2}}$	Out
B26	T/C	Out
B27	ALE	Out
B28	SG	—
B29	SG	—
B30	SG	—

Pin Number	Signal/Description	Direction
A1	SG	——
A2	DATA 7	In/Out
A3	DATA 6	In/Out
A4	DATA 5	In/Out
A5	DRQ 2	In
A6	SG	——
A7	HRQ I/O CH	In
A8	SG	——
A9	I/O CH RDY	In
A10	AEN	Out
A11	A19	Out
A12	A18	Out
A13	A17	Out
A14	A16	Out
A15	A15	Out

Pin Number	Signal/Description	Direction
A16	A14	Out
A17	A13	Out
A18	A12	Out
A19	A11	Out
A20	A10	Out
A21	A9	Out
A22	A8	Out
A23	A7	Out
A24	A6	Out
A25	A5	Out
A26	A4	Out
A27	A3	Out
A28	A2	Out
A29	A1	Out
A30	A0	Out

I/O CHANNEL DIAGRAM

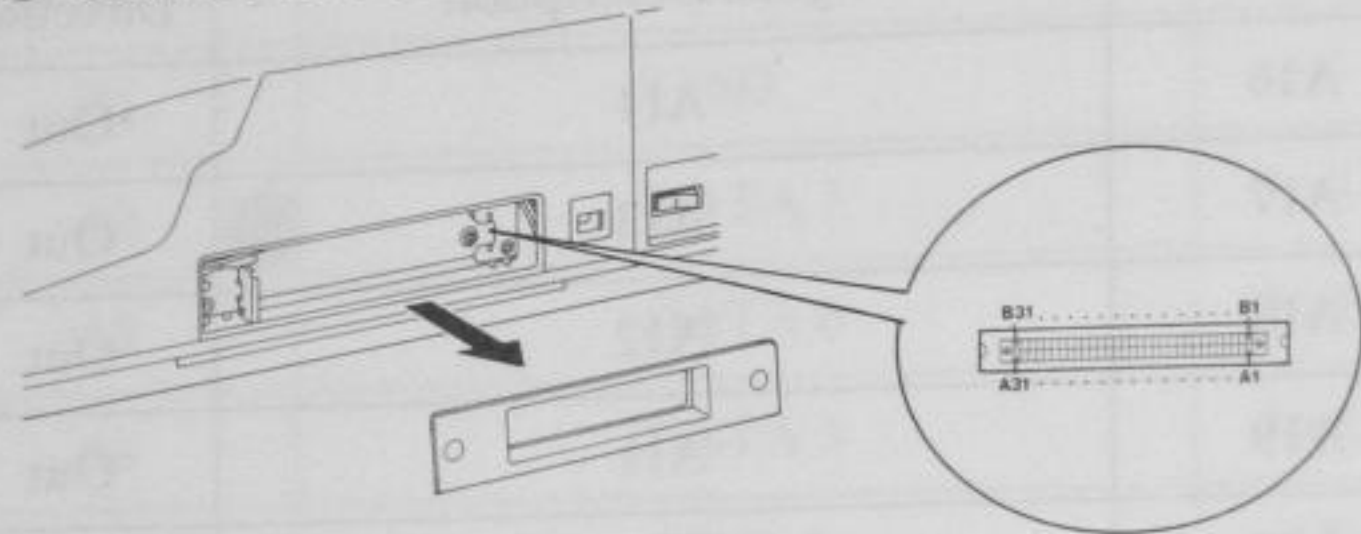


Fig. C-4. PIN ARRANGEMENT OF I/O CHANNEL

Pin Number	Signal/Description	Direction
B1	SG	—
B2	RESET DRV	Out
B3	+5 V	—
B4	IRQ 2	In
B5	(NC)	—
B6	DRQ 2	In
B7	-12 V	—
B8	HRQ I/O CH	In
B9	+12 V	—
B10	SG	—
B11	$\overline{\text{MEMW}}$	Out
B12	$\overline{\text{MEMR}}$	Out
B13	$\overline{\text{IOW}}$	Out
B14	$\overline{\text{IOR}}$	Out
B15	$\overline{\text{DACK 3}}$	Out

Pin Number	Signal/Description	Direction
B16	DRQ 3	In
B17	$\overline{\text{DACK 1}}$	Out
B18	DRQ 1	In
B19	$\overline{\text{DACK 0}}$	Out
B20	CLOCK	Out
B21	IRQ 7	In
B22	IRQ 6	In
B23	IRQ 5	In
B24	IRQ 4	In
B25	IRQ 3	In
B26	$\overline{\text{DACK 2}}$	Out
B27	T/C	Out
B28	ALE	Out
B29	+5 V	—
B30	OSC	Out
B31	SG	—

Pin Number	Signal/Description	Direction
A1	(NC)	—
A2	DATA 7	In/Out
A3	DATA 6	In/Out
A4	DATA 5	In/Out
A5	DATA 4	In/Out
A6	DATA 3	In/Out
A7	DATA 2	In/Out
A8	DATA 1	In/Out
A9	DATA 0	In/Out
A10	I/O CH RDY	In
A11	AEN	Out
A12	A19	Out
A13	A18	Out
A14	A17	Out
A15	A16	Out
A16	A15	Out

Pin Number	Signal/Description	Direction
A17	A14	Out
A18	A13	Out
A19	A12	Out
A20	A11	Out
A21	A10	Out
A22	A9	Out
A23	A8	Out
A24	A7	Out
A25	A6	Out
A26	A5	Out
A27	A4	Out
A28	A3	Out
A29	A2	Out
A30	A1	Out
A31	A0	Out

I/O ADDRESS MAP

Address (HEX)	Function
278-27F	Reserved for Parallel Printer Interface.
2F8-2FF	Reserved for Serial (RS232C) Interface.
378-37F	Used by Parallel Interface.
3B0-3BF	Reserved for Monochrome Display/Parallel Printer Interface.
3D0-3DF	Used by Color/Graphics Interface.
3F0-3F7	Used by Disk Drive Interface.
3F8-3FF	Used by Serial (RS232C) Interface.
350-35F	Used by Real Time Clock
E0	Used by LED

If you use an other serial interface board (Asynchronous Adapter), parallel interface board or other boards, you must select reserved address in above table. Refer to the user's manual of each interface board.

APPENDIX D

INTERRUPTS AND FUNCTION CALLS

1. INTERRUPTS	D-2
2. FUNCTION CALLS	D-8

INTERRUPTS

Interrupt types hex 20 to hex 3F are reserved for use by DOS. (Absolute memory locations hex 80 to hex FF are reserved.)

Interrupts follow with all values in hexadecimal.

- APP. D
- 20 Program terminate.
- Call
CS: Segment address of program Segment Prefix
- Return
None
- This is the common way to exit from a program. Control is transferred to DOS. The directory entries for unclosed files are not correct (be sure files are closed).
- 21 Function Request
- Call
AH: Function number
Other registers as specified in individual function
- Return As specified in individual function
- 22 Terminate Address
- When a program terminates, control transfers to the address at offset 0AH of the Program Segment Prefix. This address is copied into the Program Segment Prefix, from the Interrupt 22H vector, when the segment is created.

23 CONTROL-C Exit Address

If the user types <Ctrl> <C> during keyboard input or display output, control transfers to the INT 23H vector in the interrupt table. This address is copied into the Program Segment Prefix, from the Interrupt 23H vector, when the segment is created.

If the <Ctrl> <C> routine preserves all registers, it can end with an IRET instruction (return from interrupt) to continue program execution. When the interrupt occurs, all registers are set to the value they had when the original call to DOS was made. There are no restrictions on what a <Ctrl> <C> handler can do—including DOS function calls—so long as the registers are unchanged if IRET is used.

If Function 09H or 0AH (Display String or Buffered Keyboard Input) is interrupted by <Ctrl> <C>, the three-byte sequence 03H-0DH-0AH (EXT-CR-LF) is sent to the display and the function resumes at the beginning of the next line.

If the program creates a new segment and loads a second program that changes the <Ctrl> <C> address, termination of the second program restores the <Ctrl> <C> address to its value before execution of the second program.

24 Fatal Error Abort Address

If a fatal disk error occurs during execution of one of the disk I/O function calls, control transfers to the INT 24H vector in the vector table. This address is copied into the Program Segment Prefix, from the Interrupt 24H vector, when the segment is created.

BP:SI contains the address of a Device Header Control Block from which additional information can be retrieved.

Error Codes

When an error-handling program gains control from Interrupt 24H, the AX and DI registers can contain codes that describe the error. If Bit 7 of AH is 1, the error is either a bad image of the File Allocation Table or an error occurred on a character device. The device header passed in BP:SI can be examined to determine which case exists. If the attribute byte high order bit indicates a block device, then the error was a bad FAT. Otherwise, the error is on a character device.

The following are error codes for Interrupt 24H:

Error Code	Description
0	Attempt to write on write-protected disk
1	Unknown unit
2	Drive not ready
3	Unknown command
4	Data error
5	Bad request structure length
6	Seek error
7	Unknown media type
8	Sector not found
9	Printer out of paper
A	Write fault
B	Read fault
C	General failure

The user stack will be in effect (the first item described below is at the top of the stack), and will contain the following from top to bottom:

IP DOS registers from issuing INT 24H
CS
FLAGS

AX User registers at time of original INT 21H request

BX

CX

DX

SI

DI

BP

DS

ES

IP From the original INT 21H from the user to DOS

CS

FLAGS

The registers are set such that if an IRET is executed, DOS will respond according to (AL) as follows:

(AL)=0 ignore the error
=1 retry the operation
=2 terminate the program via INT

23H

Notes:

1. Before giving this routine control for disk errors, DOS performs five retries.
2. For disk errors, this exit is taken only for errors occurring during an Interrupt 21H. It is not used for errors during Interrupts 25H or 26H.
3. This routine is entered in a disabled state.
4. The SS, SP, DS, ES, BX, CX, and DX registers must be preserved.
5. This interrupt handler should refrain from using DOS function calls. If necessary, it may use calls 01H through 0CH. Use of any other call will destroy the DOS stack and will leave DOS in an unpredictable state.
6. The interrupt handler must not change the contents of the device header.
7. If the interrupt handler will handle errors rather than returning to DOS, it should restore the application program's registers from the stack, remove all but the last three words on the stack, then issue an IRET. This will return to the program immediately after the INT 21H that experienced the error. Note that if this is done, DOS will be in an unstable state until a function call higher than 0CH is issued.

25 Absolute disk read.

Call

AL: Drive number (0: Drive A, 1: Drive B)
 DS:BX: Disk transfer address
 CX: Number of sectors
 DX: Beginning relative sector

Return

AX: Error code if CF=1
 FLAGSL:CF=0 if successful
 =1 if not successful

This transfer control to the DOS BIOS. This number of sectors(CX) is read from the disk to the disk transfer address.

26 Absolute disk write. This vector is issued exactly as in 25 above but applies to a write operation.

27 Terminate but stay resident.

Call

CS:DX: First byte following last byte of code

Return

None

This vector is used if the program is to remain resident when COMMAND gets control.

28-3F Reserved for DOS.

FUNCTION CALLS

Function calls are provided by DOS for:

- Character device I/O.
- File management.
- Memory management.
- Date and time functions.
- Execution of other programs.

Error Return Table

When an error is found, some function calls set the carry flag, and put an error return code in AX. Refer to each function call. The error return codes are:

Code	Condition
1	Invalid function number
2	File not found
3	Path not found
4	Too many open files (no handles left)
5	Access denied
6	Invalid handle
7	Memory control blocks destroyed
8	Insufficient memory
9	Invalid memory block address
10	Invalid environment
11	Invalid format
12	Invalid access code
13	Invalid data
15	Invalid drive was specified
16	Attempted to remove the current directory
17	Not same device
18	No more files

Invoking DOS Functions

A function is requested of DOS by placing it in the AH register, setting other registers as appropriate, and then issuing a type 21H interrupt.

File handle

The calls which support files or devices use a 16-bit binary value, file handle, which is returned in AX. The file handle is used in referring to the file after it's been opened. The following values are set by DOS as the file handle and can be used by your program without opening the files:

0000 Standard input device. Input can be redirected.

0001 Standard output device. Output can be redirected.

0002 Standard error output device. Output cannot be redirected.

0003 Standard auxiliary device.

0004 Standard printer device.

Functions follow with all values in hexadecimal.

0 Program terminate. All closed files are written to disk. The directory entries for files left open will not be correct. CS must contain the segment address of program segment prefix.

1 Keyboard input. Waits for a character to be read from the keyboard (the character is echoed back to the standard output device and <Ctrl> <C> is checked). Returns the character in AL.

2 Display output. The character in DL is output to the standard output device.

3 Auxiliary input. Waits for a character from the standard auxiliary device [Internal RS232C interface (Serial Port)]. Returns the character in AL.

- 4 Auxiliary output. The character in DL is output to the standard auxiliary device [Internal RS232C interface (Serial Port)].
- 5 Printer output. The character in DL is output to the printer.
- 6 Direct console I/O. Sets flag to check if character is ready or not. When DL is 0 FFH (255)—If a character has been typed at the keyboard, it is returned in AL and the Zero flag is 0; if a character has not been typed, the Zero flag is 1. When DL is not 0FFH—The character in DL is displayed.
- 7 Direct console input. The character read from the keyboard is returned in AL. This function does not echo the character or check for <Ctrl> <C>.
- 8 Console input without echo. Function 8 is identical to function 1, except the key is not echoed. (It checks for <Ctrl> <C>)
- 9 Print string. Prints a character string in memory that is terminated by "\$" (hex 24). DX must contain the offset (from the segment address in DS) of a string that ends with "\$".
- A Buffered keyboard input. Fills the type-ahead buffer until the key <ENTER> is pressed. DX must contain the offset (from the segment address in DS) of an input buffer of the following form:

Byte	Contents
1	Maximum number of characters in buffer, including the CR (you must set this value).

- 2 Actual number of characters typed, not counting the CR (the function sets this value).
- 3-n Buffer; must be at least as long as the number in byte 1.
- B Check standard input status. Checks if a character is available from the keyboard and sets AL (AL=hex FF if a character is available, AL=00 if no character is available).
- C Clear standard input buffer and invoke a standard input function. The type-ahead buffer is cleared and a function specified in AL is performed. The value in AL must be 1, 6, 7, 8 or A.
- D Disk reset. Copies all files to disk. Files not properly closed will not be recorded correctly in the disk directory.
- E Select disk. The default drive is determined by the drive specified in DL (0=A, 1=B, etc.). The number of drives is returned in AL.
- F Open file. The indicated file is searched for in the directory. The file then is opened or made the current file. DX must contain the offset (from the segment address in DS) of an unopened File Control Block (FCB).

If a directory entry for the file is found, AL returns 0 and the FCB is filled as follows:

If the drive code was 0 (default disk), it is changed to the actual disk used (1=A:, 2=B:, etc.). This lets you change the default disk without interfering with subsequent operations on this file.

The Current Block field (offset 0CH) is set to zero.

The Record Size (offset 0EH) is set to the system default of 128.

The File Size (offset 10H), Data of Last Write (offset 14H), and Time of Last Write (offset 16H) are set from the directory entry.

Before performing a sequential disk operation on the file, you must set the Current Record field (offset 20H). Before performing a random disk operation on the file, you must set the Relative Record field (offset 21H). If the default record size (128 bytes) is not correct, set it to the correct length. If a directory entry for the file is not found, AL returns 0FFH (255).

- 10 Close file. After an opened file has been modified this function must be called. This lists the modified file correctly in the directory. DX must contain the offset (to the segment address in DS) of an opened FCB. If a directory entry for the file is found, the location of the file is compared with the corresponding entries in the FCB. The directory entry is updated, if necessary, to match the FCB, and AL returns 0.

If a directory entry for the file is not found, AL returns 0FFH (255).

- 11 Search for the first entry. This function is used to search for the first occurrence of the specified filename in the directory. DX must contain the offset (from the segment address in DS) of an unopened FCB. If a directory entry for the filename in the FCB is found, AL returns 0 and an unopened FCB of the same type (normal or extended) is created at the Disk Transfer Address.

If a directory entry for the filename in the FCB is not found, AL returns 0FFH (255).

Notes:

If an extended FCB is used, the following search pattern is used:

1. If the FCB attribute is zero, only normal file entries are found. Entries for volume label, sub-directories, hidden, and system files will not be returned.
2. If the attribute field is set for hidden or system files, or directory entries, it is to be considered as an inclusive search. All normal file entries plus all entries matching the specified attributes are returned. To look at all directory entries except the volume label, the attribute byte may be set to hidden+system+directory (all 3 bits on).

- 12 Search for the next entry. This function searches for the next occurrence of the match found with function 11. DX must contain the offset (from the segment address in DS) of an FCB previously specified in function 11H. If a directory entry for the filename in the FCB is found, AL returns 0 and an unopened FCB of the same type (normal or extended) is created at the Disk Transfer Address.

If a directory entry for the filename in the FCB is not found, AL returns 0FFH (255).

- 13 Delete file. Deletes the indicated file from the directory. DX must contain the offset (from the segment address in DS) of an unopened FCB. If a matching directory entry is found, it is deleted from the directory and AL returns 0.

If no matching directory entry is found, AL returns 0FFH (255).

- 14 Sequential read. The indicated record is read. DX must contain the offset (from the segment address in DS) of an opened FCB. The record pointed to by the current block (offset 0CH) and Current Record (offset 20H) fields is loaded at the Disk Transfer Address, then the Current Block and Current Record fields are incremented.

This record size is set to the value at offset 0EH in the FCB.

AL returns a code that describes the processing:

Code	Meaning
0	Read completed successfully.
1	End-of-file, no data in the record.
2	Not enough room at the Disk Transfer Address to read one record; read canceled.
3	End-of-file; a partial record was read and padded to the record length with zeros.

- 15 Sequential write. The indicated record is written. DX must contain the offset (from the segment address in DS) of an opened FCB. The record pointed to by Current Block (offset 0CH) and Current Record (offset 20H) fields is written from the Disk Transfer Address, then the current block and current record fields are incremented.

The record size is set to the value at offset 0EH in the FCB. If the Record Size is less than a sector, the data at the Disk Transfer Address is written to a buffer; the buffer is written to disk when it contains a full sector of data, or the file is closed, or a Reset Disk system call (Function 0DH) is issued.

AL returns a code that describes the processing:

Code	Meaning
0	Transfer completed successfully.
1	Disk full; write canceled.
2	Not enough room at the Disk Transfer Address to write one record; write canceled

- 16 Create file. DX must contain the offset (from the segment address in DS) of an unopened FCB. The directory is searched for an empty entry or an existing entry for the specified filename.

If an empty directory entry is found, it is initialized to a zero-length file, the Open File system call (Function 0FH) is called, and AL returns 0. You can create a hidden file by using an extended FCB with the attribute byte (offset FCB-1) set to 2.

If an entry is found for the specified filename, all data in the file is released, making a zero-length file, and the Open File system call (Function 0FH) is issued for the filename (in other words, if you try to create a file that already exists, the existing file is erased, and a new, empty file is created).

If an empty directory entry is not found and there is no entry for the specified filename, AL returns 0FFH (255).

- 17 Rename file. DX must contain the offset (from the segment address in DS) of an FCB with the drive number and filename filled in, followed by a second filename at offset 11H. The disk directory is searched for an entry that matches the first filename, which can contain the ? wild card character.

If a matching directory entry is found, the filename in the directory entry is changed to match the second filename in the modified FCB (the two filenames cannot be the same name). If the ? wild card character is used in the second filename, the corresponding characters in the filename of the directory entry are not changed. AL returns 0.

If a matching directory entry is not found or an entry is found for the second filename, AL returns 0FFH (255).

- 18 Used by DOS.
- 19 Current disk. The current default drive is returned in AL (0=A, 1=B, etc.).
- 1A Set disk transfer address. DX must contain the offset (from the segment address in DS) of the Disk Transfer Address.
- 1B-20 Used by DOS.

- 21 Random read. Reads a random record specified by file and sector. DX must contain the offset (from the segment address in DS) of an opened FCB. The Current Block (offset 0CH) and Current Record (offset 20H) fields are set to agree with the Relative Record field (offset 21H), then the record addressed by these fields is loaded at the Disk Transfer Address.

AL returns a code that describes the processing:

Code	Meaning
0	Read completed successfully.
1	End-of-file; no data in the record.
2	Not enough room at the Disk Transfer Address to read one record; read canceled.
3	End-of-file; a partial record was read and padded to the record length with zeros.

- 22 Random write. Writes a record to a random address. DX must contain the offset from the segment address in DS of an opened FCB. The Current Block (offset 0CH) and Current Record (offset 20H) fields are set to agree with the Relative Record field (offset 21H), then the record addressed by these fields is written from the Disk Transfer Address. If the record size is smaller than a sector (512 bytes), the records are buffered until a sector is ready to write.

AL returns a code that describes the processing:

Code	Meaning
0	Write completed successfully.
1	Disk is full.
2	Not enough room at the Disk Transfer Address to write one record; write canceled.

- 23 File size. This function call returns the number of records in the specified file. DX must contain the offset (from the segment address in DS) of an unopened FCB. You must set the Record Size field (offset 0EH) to the proper value before calling this function. The disk directory is searched for the first matching entry.

If a matching directory entry is found, the Relative Record field (offset 21H) is set to the number of records in the file, calculated from the total file size in the directory entry (offset 1CH) and the Record Size field of the FCB (offset 0EH). AL returns 00.

If no matching directory is found, AL returns 0FFH (255).

- 24 Set random record field. This function call sets the random record field to the current block and record fields. DX must contain the offset (from the segment address in DS) of an opened FCB.
- 25 Set interrupt vector. This function call uses the interrupt type specified in AL and sets the vector table to the address contained in DS:DX.

- 26 Create a new program segment. This function sets up a new program segment. It is best to use function call 4BH in place of this call.

- 27 Random block read. This function call reads the specified number of records. DX must contain the offset (to the segment address in DS) of an opened FCB. CX must contain the number of records to read; if it contains 0, the function returns without reading any records (no operation). The specified number of records—calculated from the Record Size field (offset 0EH)—is read starting at the record specified by the Relative Record field (offset 21H). The records are placed at the Disk Transfer Address.

AL returns a code that describes the processing:

Code	Meaning
0	Read completed successfully.
1	End-of-file; no data in the record.
2	Not enough room at the Disk Transfer Address to read one record; read canceled.
3	End-of-file, a partial record was read and padded to the record length with zeros.

28

Random block write. This function call writes the specified number of records. DX must contain the offset (to the segment address in DS) of an opened FCB; CX must contain either the number of records to write or 0. The specified number of records (calculated from the Record Size field, offset 0EH) is written from the Disk Transfer Address. The records are written to the file starting at the record specified in the Relative Record field (offset 21H) of the FCB. If CX is 0, no records are written, but the File Size field of the directory entry (offset 1CH) is set to the number of records specified by the Relative Record field of the FCB (offset 21H); allocation units are allocated or released, as required.

AL returns a code that describes the processing:

Code	Meaning
0	Write completed successfully.
1	Disk full. No records written.
2	Not enough room at the Disk Transfer Address to read one record; read canceled.

CX returns the number of records written; the Current Block (offset 0CH), Current Record (offset 20H), and Relative Record (offset 21H) fields are set to address the next record.

29

Parse filename. This function call checks the command line for a filename of the form d:filename.ext.

SI must contain the offset (to the segment address in DS) of a string (command line) to parse; DI must contain the offset (to the segment address in ES) of an unopened FCB. The string is parsed for a filename of the form d:filename. ext; if one is found, a corresponding unopened FCB is created at ES:DI.

Bits 0-3 of AL control the parsing and processing. Bits 4-7 are ignored:

Bit	Value	Meaning
0	0	All parsing stops if a file separator is encountered.
	1	Leading separators are ignored.
1	0	The drive number in the FCB is set to 0 (default drive) if the string does not contain a drive number.
	1	The drive number in the FCB is not changed if the string does not contain a drive number.
2	1	The filename in the FCB is not changed if the string does not contain a filename.
	0	The filename in the FCB is set to 8 blanks if the string does not contain a filename.
3	1	The extension in the FCB is not changed if the string does not contain an extension.
	0	The extension in the FCB is set to 3 blanks if the string does not contain an extension.

If the filename or extension includes an asterisk (*), all remaining characters in the name or extension are set to question mark (?).

Filename separators:

: . ; , = + / " [] \ < > | space tab

Filename terminators include all the filename separators plus any control character. A filename cannot contain a filename terminator; if one is encountered, parsing stops.

If the string contains a valid filename:

1. AL returns 1 if the filename or extension contains a wild card character (* or ?); AL returns 0 if neither the filename nor extension contains a wild card character.
2. DS:SI point to the first character following the string that was parsed.
3. ES:DI point to the first byte of the unopened FCB.

If the drive letter is invalid, AL returns 0FFH (255). If the string does not contain a valid filename, ES:DI+1 points to a blank (ASCII 32).

- 2A Get date. This function call returns the date in the following registers. The year (1980–2099 in binary) is in CH. The month (1=Jan, 2=Feb, etc.) is in DH. The day is in DL. The day of week (0=Sunday, 1=Monday, etc) is in AL.

- 2B Set date. This function call sets the date if CX has a year (1980–2099), DH has a month (1=Jan, 2=Feb,...) and DL has a day. AL returns 00 if the date was valid and the set operation was successful.
- 2C Get time. This function call returns the time-of-day in the following registers. The hours (0–23) are in CH. The minutes (0–59) are in CL. The seconds (0–59) are in DH. The 1/100 seconds (0–99) are in DL.
- 2D Set time. This function call sets the time if CH has hour, CL has minutes, DH has seconds and DL has 1/100 seconds. AL returns 00 if the time was valid.
- 2E Set/reset verify switch. This function call sets verify ON (AL=1) or OFF (AL=0). This is useful if you wish to verify the recording of critical data.
- 2F Get Disk Transfer Address (DTA). The current DTA transfer address is specified in ES:BX.
- 30 Set DOS version number. AL will contain the major version number and AH will contain the minor version number.
- 31 Terminates process. This function call terminates the current process. AL must contain the exit code and DX must contain the memory size in paragraph.
- 32 Used by DOS.
- 33 Ctrl-Break check. This function call either sets [AL=1, DL=1 (Set), DL=0 (Reset)] or checks [AL=0, Returns 0 (Off) or 1 (On) in DL] the state of control-break checking.

- 34 Used by DOS.
- 35 Get vector. The interrupt vector is returned in ES:BX.
- 36 Get disk free space. This function call returns information on the free space on the disk. DL must contain the drive designator (0=default, 1=A, etc.). This call returns available clusters in BX, clusters per drive in DX, bytes per sector in CX. If 0FFFFH is in AX drive number is invalid, otherwise sectors per cluster.
- 37 Used by DOS.
- 38 Return country dependent information. This function call is used to set up information to be used internationally. Error return code is 2.
- 39 Create a sub-directory. This function call sets up a sub-directory. If any part of the directory path does not exist the path is not changed. DX:DS must contain the pointer to pathname. If carry is set and 3 is returned in AX, path is not found, or 5 in AX, access is not denied. If carry is not set, no error.
- 3A Remove a sub-directory entry. This function call removes a sub-directory. Using this function call is same as "Create a sub-directory". Error return codes are 3, 5 and 16.
- 3B Change the current directory. This function call changes the current directory. If any part of the directory path does not exist the path is not changed. Using this function call is same as "Create a sub-directory". Error return code is 3.

- 3C Create a file. This function call creates a new file or sets an old file to a length of zero so that it can be written. DS:DX must contain the pointer to pathname and CX must contain the file attribute. Error return codes are 3, 4 and 5.
- 3D Open a file. This function call opens a file and sets the access code in AL. The following are valid access codes:
- 0=open for reading.
 - 1=open for writing.
 - 2=open for both reading and writing.
- DS:DX must contain the pointer of a filename. Error return codes are 2, 4, 5 and 14.
- 3E Close a file. This function call closes the file. All internal buffers are cleared. BX must contain the file handle. Error return code is 6.
- 3F Read from a file or device. This function call reads from a file or device however, the number of bytes read will vary from device to device. For instance, reading from the keyboard will read one line of text. DS:DX must contain pointer to buffer, CX: bytes to read and BX:file handle. Error return codes are 5 and 6.
- 40 Write to a file or device. This function writes to a file or device and returns the number of bytes written in CX. Using this function is same as "Read from a file or device" except CX. Error return codes are 5 and 6.
- 41 Delete a directory entry. This function call removes a directory entry associated with a file name. DS:DX must contain the pointer to path name. Error return codes are 2 and 5.

If ISDEV=0
 EOF=0 if channel has been written
 Bits 0-5 are the block device number
 for the channel (0=A:, 1=B:,...)

Bits 15, 8-13, 4 are reserved and should not be altered.

Calls 2..5:

These four calls allow arbitrary control strings to be sent or received from a device. The call syntax is the same as the read and write calls, except for 4 and 5, which take a drive number in BL instead of a handle in BX.

An invalid function error is returned if the CTRL bit (see above) is 0.

An access denied is returned by calls AL=4,5 if the drive number is invalid.

Calls 6, 7:

These two calls allow the user to check if a file handle is ready for input or output. Status of handles open to a device is the intended use of these calls, but status of a handle open to a disk file is allowed, and is defined as follows:

Input:

Always ready (AL=FF) until EOF reached, then always not ready (AL=0) unless current position changed via LSEEK.

Output

Always ready (even if disk full).

Error return codes are 1, 5, 6 and 13.

- 45 Duplicate a file identifier. This function call takes the identifier for an opened file and returns a new file identifier that refers to the same file at the same position. BX must contain the file handle.

- 46 Force a duplicate of a identifier. This function call sets the file handle in CX to the same stream as the handle in BX. Error return codes are 4 and 6.

- 47 Get Current directory. This function call places the full path name (starting from the root directory) of the current directory in the area of memory pointed to by DS:SI. DL must contain the drive number. Error return code is 15.

- 48 Allocate memory. BX must contain the size of memory to be allocated. If carry is not set, memory is allocated and the pointer to the allocated memory is in AX. Error return codes are 7 and 8.

- 49 Free allocated memory. A block of memory that was allocated by function call hex 48 is returned to the system pool. ES must contain the segment address of memory area to be freed. Error return codes are 7 and 9.

- 4A Modify allocated memory blocks. This function call will either enlarge or shrink the block depending on the block size request. ES must contain the segment address of memory area and BX must contain the requested memory area size. Error return codes are 7, 8 and 9.

- 4B Load or execute a program. This function call loads a program into memory and allows for program execution. DS:DX must contain the pointer to pathname and ES:BX must contain the pointer to parameter block.

A function must be in AL:

AL	Function
0	Load and execute the program. A program header is established for the program and the terminate and CONTROL-C addresses are set to the instruction after the EXEC system call.

- 3 Load (do not create) the program header, and do not begin execution. This is useful in loading program overlays.

For each value of AL, the block has the following format:

AL=0 → load/execute program

WORD segment address of environment
DWORD pointer to command line at 80H
DWORD pointer to default FCB to be passed at 5CH
DWORD pointer to default FCB to be passed at 6CH

AL=3 → load overlay

WORD segment address where file will be loaded.
WORD relocation factor to be applied to the image.

Note that all open files of a process are duplicated in the child process after an EXEC. This is extremely powerful; the parent process has control over the meanings of stdin, stdout, stderr, staux and stdprn. The parent could, for example, write a series of records to a file, open the file as standard input, open a listing file as standard output and then EXEC a sort program that takes its input from stdin and writes to standard output.

Also inherited (or passed from the parent) is an "environment." This is a block of text strings (less than 32K bytes total) that convey various configuration parameters. The format of the environment is as follows:

(paragraph boundary)

BYTE ASCIZ* string 1
BYTE ASCIZ string 2
...
BYTE ASCIZ string n
BYTE of zero

*ASCIZ string consists of ASCII string and code 00 at the end of the string.

Typically the environment strings have the form:

parameter=value

For example, COMMAND.COM might pass its execution search path as:

PATH=A: \ BIN;B: \ BASIC \ LIB

A zero value of the environment address causes the child process to inherit the parent's environment unchanged. Error return codes are 1, 2, 8, 10 and 11.

- 4C Terminate a process. This function call will end the current process and transfer control to the calling process. A return code (must be in AL) can be sent and interrogated by the batch subcommands IF and ERRORLEVEL and by the wait function call (4DH). All open files are closed.

- 4D Retrieve the return code of a sub-process. An Exit code which was specified by a function call hex 4C or function call hex 31 is returned in AX.
- 4E Find first matching file. This function call finds the first occurrence of the filename in the directory. Filespec must be in DS:DX and search attribute in CX. (Refer to function call 11)
- 4F Find next matching file. This function call will find the next directory entry matching the name that was specified on the previous Find First call. Error code 18 is returned in AX if there is no match on the filename.
- 50-53 Used by DOS.
- 54 Get verify state. This function call sets AL to 00 if verify is OFF. AL is set to 01 if verify is ON.
- 55 Used by DOS.
- 56 Rename a file. This function call renames a file allowing the file to be moved to another directory while it is being renamed. DS:DX must contain the pointer to filespec of existing file and ES:DI must contain the pointer to new filespec. Error return codes are 2, 5 and 17.
- 57 Get/Set a file's date and time. This function sets the date and time if AL=01 (CX: time to be set, DX=date to be set) and gets the date and time in CX and DX if AL=00. Bx must contain file handle. Error return codes are 1 and 6.

APPENDIX E

CONTROL BLOCKS

1. STANDARD FILE CONTROL BLOCK E-2
2. EXTENDED FILE CONTROL BLOCK E-4

STANDARD FILE CONTROL BLOCK

The Standard File Control Block (FCB) is defined as follows:
(The offsets are in decimal)

Byte	Function
0	Drive number. 0: default drive, 1: drive A, 2: drive B
1-8	Filename. This must be left-justified with trailing blanks.
9-11	Filename extension. This must be left-justified with trailing blanks.
12-13	Current block number. Each block contains 128 records. Zero indicates the first block of the file; one, the second and so on. Sequential READ or WRITE function requires the current block number as well as the current record field contained in byte 32.
14-15	Logical record size (bytes). OPEN and CREATE function calls set the logical record size to 80H.
16-19	File size (bytes). It may be read by a program but not changed. This is a 2-word field, the first word is the low-order part of the size.

20-21 Date of file creation or update. It may be used by a program but not changed. The format of the 16-bit field is :

0-4 dd
5-8 mm
9-15 yy (the range for year is 0 to 119.
This is year minus 1980.)

22-31 Reserved for use by DOS.

32 Current relative record number. It is within the current block for the current file. This indicates one of 128 records accessed by a sequential READ or WRITE. This field must be set before sequential READ or WRITE.

33-36 Random record field. This field must be set before doing random read/write operations.

Only the first three bytes are used if the record size is greater than or equal to 64 bytes. Four are used otherwise.

EXTENDED FILE CONTROL BLOCK

The Extended File Control Block is a 7 byte prefix used to search for files in the disk directory with special attributes.

The 7 byte prefix is formatted as follows:

Byte	Function
FCB-7	Flag byte. FFH indicates this is an extended FCB.
FCB-6 to FCB-2	Reserved.
FCB-1	Attribute byte. If bit #1 is set to 1 then searches will include hidden files. If bit #2 is set to 1 then searches will include system files.

APP. E

APPENDIX F

ESC SEQUENCES

To execute an escape sequence, files ANSI.SYS and CONFIG.SYS are required. ANSI.SYS is found on the System Disk. However, you must create CONFIG.SYS. We will use line editor EDLIN to create CONFIG.SYS.

First enter:

```
A> EDLIN CONFIG.SYS
```

Then enter:

```
*i Press <ENTER>
1:* DEVICE=ANSI.SYS Press <ENTER>
Press <Ctrl> + <Break>
*e Press <ENTER>
```

NOTE: Escape sequences cannot be executed in BASIC. This completes the CONFIG.SYS file. You must reboot the system to execute CONFIG.SYS.

NOTE: Now, CONFIG.SYS is automatically executed everytime you boot up the system. The system can now execute escape sequences. Since escape sequences cannot be directly entered through the keyboard, you must first create an escape sequence file and then execute it with the TYPE command.

We will describe 2 methods to create an escape sequence file. In our first method we use DEBUG program to enter the file. In our second method we use BASIC to enter the file.

Let's enter the escape sequence to execute "Erase Entire Display."

APP. F

Using DEBUG

First start the DEBUG program by typing:

```
A> DEBUG <ENTER>
— (“—” is the prompt for DEBUG)
```

After the code for ESC (hex 1B), enter the escape sequence by enclosing them in double quotation marks “ ”.

YOU TYPE:

```
—E100 1B “[2J” <ENTER>
—R CX <ENTER>
```

SCREEN DISPLAYS:

```
CX 0000
```

Enter a hexadecimal number for the number of characters in the escape sequence to be written plus 1. In our example, the escape sequence consists of the 3 characters [2J]. We add 1 to 3 and enter 4.

```
:4 <ENTER>
```

Name this file using the N command.

YOU TYPE:

```
—NTEST.TXT <ENTER>
```

Write this file to the disk using the W command.

YOU TYPE:

```
—W <ENTER>
```

SCREEN DISPLAYS:

```
Writing 00004 bytes
—
```

This confirms that 4 bytes were written to the disk. We now exit the DEBUG program.

YOU TYPE:

```
—Q <ENTER>
```

SCREEN DISPLAYS:

```
A>
```

To execute the escape sequence we have just written, enter the following:

```
A> TYPE TEST.TXT <ENTER>
```

This executes our escape sequence and erases the screen.

Using BASIC

Enter the following program:

```
10 OPEN"TEST.TXT" FOR OUTPUT AS #1
20 PRINT #1, CHR$(27);"[2J"
30 CLOSE
40 END
```

Now run this program.

SCREEN DISPLAYS:

```
RUN
OK_
```

Return to the system.

YOU TYPE:

```
SYSTEM <ENTER>
```

SCREEN DISPLAYS:

```
A>
```

To execute the escape sequence we have just written, enter the following:

```
A> TYPE TEST.TXT
```

This executes our escape sequence and erases the screen.

NOTE: If a double quotation mark is used within an escape sequence, use CHR\$(34) for the double quotation mark.

Application

Enter the following program. This program can be used to enter any number of escape sequences.

```
10 OPEN"TEST.TXT" FOR OUTPUT AS #1
20 INPUT "ENTER ESCAPE SEQUENCE:", CS
30 PRINT #1, CHR$(27);CS
40 CLOSE
50 END
```

Save this program by entering:

```
SAVE"ANSI
```

The following will be displayed when you run this program:

```
ENTER ESCAPE SEQUENCE:
```

Enter the escape sequence you wish to execute.

Let's enter an example where we reassign the F10 key on page 7-36 to the DIR command.

YOU TYPE:

```
[0;68;"dir";13P
```

```
OK_
```

Return to the system.

```
A>
```

To execute this escape sequence,

YOU TYPE:

```
A> TYPE TEST.TXT <ENTER>
```

```
A>
```

Now when you press the F10 key, the DIR command is executed and the directory will be displayed on the screen.

Mode Operation

Set Graphics Rendition

SGR

ESC [# ;...; # m

This sequence causes the character/screen #;...;#m attributes to change according to the numeric values as specified below (these values are enabled until a subsequent issuance of SGR):

Parameter	Function
0	All attributes OFF (normal white on black)
1	Bold On (high intensity)
4	Underscore On
5	Blink On
7	Reverse video On
8	Concealed On (invisible)
30	Black foreground
31	Red foreground
32	Green foreground
33	Yellow foreground
34	Blue foreground
35	Magenta foreground
36	Cyan foreground
37	White foreground
40	Black background
41	Red background
42	Green background
43	Yellow background
44	Blue background
45	Magenta background
46	Cyan background
47	White background

Set Mode

SM

ESC [= # h These sequences set screen width/display or according to the numeric values as specified below:
or ESC [= h
or ESC [= 0 h
or ESC [= ? 7 h

Parameter	Function
0	40×25 black and white
1	40×25 color
2	80×25 black and white
3	80×25 color
4	320×200 color
5	320×200 black and white
6	640×200 black and white
7	automatic wrap-around (at end of line)

Reset Mode

RM

ESC [= # 1 These sequences function in the same manner as SM (see above) except that parameter 7 will cause no-wrap at end of line (excess characters are ignored).
or ESC [= 1
or ESC [= 0 1
or ESC [? 7 1

Erasing

Erase Entire Display

ED

ESC [2 J

This sequence causes erasure of the entire screen. The cursor is left at the home position.

Erase Entire Line

EL

ESC [k

This sequence causes erasure of everything from the current cursor position to the end of the line (to the right).

Cursor Control

Cursor Position

CUP

ESC [# ; # H

This sequence causes the cursor to be moved to the screen position addressed by the two numeric values. The first gives the line number that the cursor is to be positioned at and the second gives the column number. The default value is one. If no values are specified, the cursor will move to the home position.

Cursor Up

CUU

ESC [# A

This sequence causes the cursor to be moved directly up according to the number of lines specified by the numeric value. The default value is one. If the cursor is already on the top of the screen then this sequence has no effect.

Cursor Down

CUD

ESC [# B

This sequence causes the cursor to be moved directly down according to the number of lines specified by the numeric value. The default value is one. If the cursor is already at the bottom of the screen then this sequence has no effect.

Cursor Forward

CUF

ESC [# C

This sequence causes the cursor to be moved directly to the right according to the number of columns specified by the numeric value. The default value is one. If the cursor is already at the right edge then this sequence has no effect.

Cursor Backward

CUB

ESC [# D This sequence causes the cursor to be moved directly to the left according to the number of columns specified by the numeric value. The default value is one. If the cursor is already at the left edge then this sequence has no effect.

Horizontal and Vertical Position

HVP

ESC [# ; # f This sequence causes the cursor to be moved to the screen position addressed by the two numeric values. The first gives the line number that the cursor is to be positioned at and the second gives the column number. The default value is one. If no values are specified the cursor will move to the home position.

Device Status Report

DSR

ESC [6 n This sequence causes the console driver to issue a CPR sequence. (see CPR)

Cursor Position Report

CPR

ESC [# ; # R This sequence causes the current cursor position to be reported through the standard input device. The first numeric value gives the current line and the second numeric value gives the current column.

Save Cursor Position

SCP

ESC [s This sequence causes the current cursor position to be saved.
(see RCP to restore the cursor position)

Restore Cursor Position

RCP

ESC [u This sequence restores the cursor to its position prior to the issuance of the SCP sequence.

Keyboard Key

Reassignment

ESC [# ; # ; ... # p
or ESC ["string" ; p
or ESC [# ; "string" ; # ; # ; "string" ; # p
or any other combination of strings and decimal numbers

These control sequences reassign keyboard keys. The numeric value is the ASCII code for a particular key. The key to be mapped is specified by the first numeric value. Subsequent numeric values specify the sequence of ASCII character codes to be generated when the mapped key is depressed. Any DOS command can be invoked or assigned to a key. Extended ASCII codes are indicated by specifying zero (NUL) as the first numeric value and the appropriate code as the second numeric value.

For example:

1. Reassign the A key to the B key.
Reassign the a key to the b key.
Reassign the B key to the A key.
Reassign the b key to the a key.
2. Reassign the F10 key to a DIR command followed by a carriage return:

ESC [6 5 ; 6 6 p	A becomes B
ESC [9 7 ; 9 8 p	a becomes b
ESC [6 6 ; 6 5 p	B becomes A
ESC [9 8 ; 9 7 p	b becomes a

ESC [0 ; 6 8 ; "dir" ; 1 3 p

(0;68 is the extended ASCII code for the F10 key. 13 decimal is a carriage return.)

MEMO

APP. F

APPENDIX G

INSTALLING POPULAR APPLICATION PROGRAMS

INSTALLING POPULAR APPLICATION PROGRAMS	G-2
Lotus 1-2-3	G-2
SuperCalc 3	G-3
Symphony	G-3
d BASE III	G-4
WordStar 2000	G-5
pfs:file	G-6
pfs:report	G-7
pfs:graph	G-8
Microsoft Word	G-8
HOW TO EXECUTE APPLICATION PROGRAMS ..	G-9

APP. G

Lotus 1-2-3 and Symphony are trademarks of Lotus Development Corporation.
WordStar™ is the trademark of MicroPro International Corporation.
d BASE III is trademark of Ashton-Tate.
SuperCalc® is trademark of Sorcim Corporation.
pfs:® file, report, and graph are the registered trademarks of Software Publishing Corporation.
Microsoft Word is the trademark of Microsoft Corporation.

INSTALLING POPULAR APPLICATION PROGRAMS

Before you run an application program, you must usually "install" it so that it runs properly with your particular computer system. In this appendix, we will outline how to install some popular application programs for your Exec. Partner. The installation procedures that follow will be based on the basic Exec. Partner hardware, that is the plasma display and built-in printer, and the hard disk drive if so equipped. If you are using another display or printer, install accordingly.

Lotus 1-2-3

STEP 1 Insert a copy of the Exec. Partner System Disk into drive A and start the system.

STEP 2 Follow the procedure described under "Two-Disk System" or "Hard-Disk System" in the booklet "Getting Started". These procedures transfer the DOS to each of the 1-2-3 disks or transfers the 1-2-3 files to the hard disk.

STEP 3 Follow the procedure described under "Installing 1-2-3 Drivers."

Under "Installation Choices" issue the command:

B&W x:

(where x: specifies the target drive)

NOTE: Select "IBM Graphics Printer, Triple Dens. Mode" for the graphics device in the GRAPH program.

SuperCalc 3

STEP 1 Insert a copy of the Exec. Partner System Disk into drive A and start the system.

STEP 2 YOU TYPE:

STARTUP.BAT <ENTER>

Read the messages and follow the instructions. Basically, you will run GOPC if you have a floppy disk drive system or HARDDISK if you have a hard disk system. See the User's Guide & Reference Manual for details.

STEP 3 Select IBM graphics printer for the graphics device.

Symphony

NOTE: Since this program requires at least 320K bytes of memory, you will have to expand the memory of the Exec. Partner to 640K bytes.

STEP 1 Insert a copy of the Exec. Partner System Disk into drive A and start the system.

STEP 2 Read the instructions in Chapter 3 of the Introduction manual, "Preparing the Symphony Disks." You will make backup copies of the Symphony disks if you have a floppy disk drive system or transfer the files on the Symphony disks onto your hard disk if you have a hard disk system.

STEP 3 Start the INSTALL program as instructed.

STEP 4 For the Text and Graph Display driver, select 3 (IBM Color Monitor+IBM Color Card...Share Mode)

- STEP 5 For the Text Printer driver, select 17 (IBM Graphics Printer).
- STEP 6 For the Graph Printer driver, select 37 (IBM Graphics Printer, density 3).
- STEP 7 If you select a modem, select 1 (COM1 8250) for the Asynchronous Port driver.
- STEP 8 Select a Collating driver according to your needs.
- STEP 9 Continue with the install procedure as instructed in the Introduction manual.

dBASE III

- STEP 1 Read and follow the instructions on backing up the dBase III system disk in the User Manual.
- STEP 2 Copy dbase.exe and config.sys to the backup disk.
- STEP 3 Insert the backup disk in drive A and original dBase III System Disk in drive B.
- STEP 4 At the A> prompt, type

DBASE <ENTER>

to start dBase III.

NOTE: Even if you use a hard disk, the copy-protection system prevents dBASE III from being run without the master disk in the disk drive. However, a feature that allows you to use dBASE III without using the SYSTEM disk is mentioned. Contact Ashton-Tate for further information.

WordStar 2000

- STEP 1 Read your Getting Started booklet and follow the instructions.
- NOTE:** If you are installing WordStar 2000 on your hard disk, you must have at least 2150K bytes of disk space.
- STEP 2 You will answer a series of questions on your Exec. Partner computer system in Phase 2 of the AutoInstall Procedure. To reinstall WordStar 2000, use the installation program (WS2INS).
- STEP 3 When asked if you are using a color monitor, press N.
- STEP 4 When asked if you are using a graphics card, press Y.
- STEP 5 When asked if you will be using WordStar 2000 with a hard disk, press Y for yes or N for no.
- STEP 6 To select a printer, go to menu 4. Then press K to select the IBM PC Graphics Printer.
- STEP 7 Press Y to confirm your printer selection.
- STEP 8 This completes your selections for the Exec. Partner. Press A to save your selections or D to continue to the Advanced Modification Menu. Follow the messages displayed on the screen.

pfs:file

NOTE: If you copy more than one PFS program onto your hard disk or onto the same disk, you can install all the PFS programs at the same time using the installation program SETUP. See "Running SETUP with More than One PFS Program" at the end of Appendix B for more details.

- STEP 1 Follow the instructions in Chapter 2, Getting Started, and make a backup copy of the PFS:FILE disk.
- STEP 2 Follow the instructions in Appendix B on using SETUP. (See the section on the hard disk if your Exec. Partner is equipped with one.)
- STEP 3 First select a printer by entering 1 in the main menu.
- STEP 4 Select 1 for the IBM Graphics printer.
- STEP 5 Select LPT1: for the printer port.
- STEP 6 Enter 3 in the main menu to set the display.
- STEP 7 Select N for no color.

pfs:report

NOTE: If you copy more than one PFS program onto your hard disk or onto the same disk, you can install all the PFS programs at the same time using the installation program SETUP. See "Running SETUP with More than One PFS Program" at the end of Appendix B for more details.

- STEP 1 Follow the instructions in Chapter 2, Getting Started, and make a backup copy of the PFS:FILE deskette.
- STEP 2 Follow the instructions in Appendix B on using SETUP. (See the section on the hard disk if your Exec. Partner is equipped with one.)
- STEP 3 First select a printer by entering 1 in the main menu.
- STEP 4 Select 1 for the IBM Graphics printer.
- STEP 5 Select LPT1: for the printer port.
- STEP 6 Enter 3 in the main menu to set the display.
- STEP 7 Select N for no color.

pfs:graph

NOTE: If you copy more than one PFS program onto your hard disk or onto the same floppy disk, you can install all the PFS programs at the same time using the installation program SETUP. See "Running SETUP with More than One PFS Program" at the end of Appendix B for more details.

- STEP 1 Follow the instructions in Chapter 2, Getting Started, and make a backup copy of the PFS:FILE disk:
- STEP 2 Follow the instructions in Appendix B on using SETUP. (See the section on the hard disk if your Exec. Partner is equipped with one.)
- STEP 3 First select a printer by entering 1 in the main menu.
- STEP 4 Select 1 for the IBM Graphics printer.
- STEP 5 Select LPT1: for the printer port.
- STEP 6 Enter 3 in the main menu to set the display.
- STEP 7 Select N for no color.

Microsoft Word

- STEP 1 Follow the instructions in the manual.
- STEP 2 Select IBMGRAPH for the printer.

HOW TO EXECUTE APPLICATION PROGRAMS

When executing an application program for the Exec. Partner, it may not be run as described in the application manual. In such cases, an error message will be displayed. This Appendix contains possible error messages and the procedure to recover from the error. Check the error message and execute the appropriate procedure described below.

1. "Insufficient disk space" was displayed when you executed "COPY BASICA.COM B:"

Copy BASICA.EXE to the disk instead of BASICA.COM.

2. "Insufficient disk space" was displayed when you executed the batch file for an initialization such as UPDATE, SETUP, INSTALL etc. (It depends on application program).

This error message is displayed because the batch file for an initialization tried to COPY BASICA.COM to the disk without enough space for it. After the DOS prompt appears, type COPY BASICA.EXE B: and execute the batch file (UPDATE, SETUP, INSTALL etc.) again. "Insufficient disk error" will be displayed again, but ignore it.

3. "Insert Panasonic System Disk" was displayed when you executed an application program.

This message is displayed when you execute BASIC or BASICA on your application program disk that does not contain BASICA.COM. It may contain BASIC.COM. or BASICA.EXE. Remove the application program disk and insert your back up Panasonic System Disk. Press any key to load BASICA.COM. When "Replace the disk" appears, reinsert your application program disk. Then the application program will be executed.

NOTE: When you copy BASIC.COM. on an application program disk, also copy BASICA.COM. if that disk has sufficient space. Then the procedure of removing and inserting disks can be omitted.

4. An application program was not executed though the memory (RAM) size was same as that designated by the application manual.

This is because of our BASIC's specific structure. Expand memory size.

APPENDIX H CHARACTER SET/PLASMA DISPLAY

Character and Graphic dot on Plasma display.

On Text Mode:

- Character 80 columns×25 lines/40 columns×25 lines
(8×16 dot for one character area to be used)
- Graphics 640 (H)×200 (V) dots
320 (H)×200 (V) dots for color shading use.
(8×8 dots for graphics character; but actual on the display, use 8 (H)×16 (V) dots of Plasma.)

Plasma Color Definition (Text Mode)

For making clear definition to the color software on the Plasma display, we propose following definition by character change. (For kinds of character type used.)

Roman
Bold Roman
Italic (Except white color)
Bold Italic

Color	Intensity	Red	Green	Blue	Character Type
Blue	0	0	0	1	<i>Italic</i>
Green	0	0	1	0	<i>Italic</i>
Cyan	0	0	1	1	<i>Italic</i>
Red	0	1	0	0	Bold Italic
Magenta	0	1	0	1	Bold Italic
Brown	0	1	1	0	Bold Italic
Light Gray	0	1	1	1	Roman
Dark Gray	1	0	0	0	Bold Roman
Light Blue	1	0	0	1	Bold Roman
Light Green	1	0	1	0	Bold Roman
Light Cyan	1	0	1	1	Bold Roman
Light Red	1	1	0	0	Bold Roman
Light Magenta	1	1	0	1	Bold Roman
Yellow	1	1	1	0	Bold Roman
White	1	1	1	1	Bold Roman

HEXA-DECIMAL	0	1	2	3	4	5	6	7
	0	16	32	48	64	80	96	112
0	BLANK (NULL)	▶	BLANK (SPACE)	0	@	P	‘	p
	1	17	33	49	65	81	97	113
1	☺	◀	!	1	A	Q	a	q
	2	18	34	50	66	82	98	114
2	☹		!"	2	B	R	b	r
	3	19	35	51	67	83	99	115
3	♥	!!	#	3	C	S	c	s
	4	20	36	52	68	84	100	116
4	♦	qT	\$	4	D	T	d	t
	5	21	37	53	69	85	101	117
5	♣	§	%	5	E	U	e	u
	6	22	38	54	70	86	102	118
6	♠	-	&	6	F	V	f	v
	7	23	39	55	71	87	103	119
7	•	⊥	’	7	G	W	g	w
	8	24	40	56	72	88	104	120
8	•	↑	(8	H	X	h	x
	9	25	41	57	73	89	105	121
9	○	↓)	9	I	Y	i	y
	10	26	42	58	74	90	106	122
A	○	→	*	:	J	Z	j	z
	11	27	43	59	75	91	107	123
B	♂	←	+	;	K	[k	{
	12	28	44	60	76	92	108	124
C	♀	└	,	<	L	\	l	!
	13	29	45	61	77	93	109	125
D	♪	—	—	=	M]	m	}
	14	30	46	62	78	94	110	126
E	♪	▲	•	>	N	^	n	~
	15	31	47	63	79	95	111	127
F	⚙	▼	/	?	O	—	o	△

HEXA-DECIMAL VALUE	8	9	A	B	C	D	E	F
	128	144	160	176	192	208	224	240
0	Ç	É	á	☐	☐	☐	α	≡
	129	145	161	177	193	209	225	241
1	ü	æ	í	☐	☐	☐	β	±
	130	146	162	178	194	210	226	242
2	é	Æ	ó	☐	☐	☐	Γ	≥
	131	147	163	179	195	211	227	243
3	â	ô	ú	☐	☐	☐	π	≤
	132	148	164	180	196	212	228	244
4	ä	ö	ñ	☐	☐	☐	Σ	∫
	133	149	165	181	197	213	229	245
5	à	ò	Ñ	☐	☐	☐	σ	∫
	134	150	166	182	198	214	230	246
6	å	û	ã	☐	☐	☐	μ	÷
	135	151	167	183	199	215	231	247
7	ç	ù	o	☐	☐	☐	τ	≈
	136	152	168	184	200	216	232	248
8	ê	ÿ	¿	☐	☐	☐	♀	◊
	137	153	169	185	201	217	233	249
9	ë	Ö	□	☐	☐	☐	θ	▪
	138	154	170	186	202	218	234	250
A	è	Ü	□	☐	☐	☐	Ω	•
	139	155	171	187	203	219	235	251
B	ï	ç	½	☐	☐	☐	δ	√
	140	156	172	188	204	220	236	252
C	î	£	¼	☐	☐	☐	∞	ⁿ
	141	157	173	189	205	221	237	253
D	ì	¥	ì	☐	☐	☐	φ	²
	142	158	174	190	206	222	238	254
E	Ä	Pt	«	☐	☐	☐	€	█
	143	159	175	191	207	223	239	255
F	Å	f	»	☐	☐	☐	∩	BLANK (FF)

APPENDIX I

DEMONSTRATION SOFTWARE

Your new Exec. Partner can help you in many different ways. In the previous section you learned how DOS can help you in your everyday operations.

In this section you will get a preview of some of the capabilities and applications programs which make your Exec. Partner a very powerful tool.

The demonstration programs are contained in a file called **DEMO.BAT**. This file is contained on your System Disk.

Using DEMO.BAT

Turn ON the main power switch.

Insert your backup System Disk in Drive A.

After you have entered the date and time the A> prompt will appear.

YOU TYPE:

demo

and press <ENTER>.

Now watch the screen!

For a moment, your screen will look like this:

Exec. Partner™

- 1 == Graph
- 2 == Spread Sheet
- 3 == Word Processor
- 4 == Specifications
- 5 == Automatic Demonstration
- Q == QUIT

Please select number

If you want to start the DEMO 1-4, press <1> for Graph, <2> for Spread Sheet, <3> for Word Processor or <4> for Specifications.

If you want to return to the system mode, press <Q> or <q> key.

If you want to start the DEMO 1 and continue DEMO 2, 3, 4, press <5>.

1 = introduces Graph

2 = introduces Spread Sheet

3 = introduces Word Processor

4 = introduces Specifications

5 = introduces Automatical DEMO

If you want to return to the menu, press <F5> during the demonstration.

APPENDIX K

INDEX

\ (backslash) 6-10
← (left arrow) 3-14
→ (right arrow) 3-14
↑ (up arrow) 3-13
↓ (down arrow) 3-14
(BACKSPACE) 3-9
(TAB) 3-10
(SHIFT) 3-11
* (wild card) 6-7
? (wild card) 6-7
+ (command character, LINK) 10-12
; (command character, LINK) 10-13
.BAT 8-9
.EXE (LINK) 10-5
.LIB (LINK) 10-4
.MAP (LINK) 10-5
.OBJ (LINK) 10-4
<F1> 9-7
<F2> 9-8
<F3> 9-9
<F4> 9-11
<F5> 9-15
 9-10
<Esc> 9-12
<Ins> 9-13

A

A> prompt 3-36, 3-74
abort B-5
AC power cord 2-15
Active partition 5-15

ALT key 3-11
alphanumeric keys 3-7
ASCII codes H-2
ASSIGN 7-16
asterisk (*) 6-7
AUTOEXEC.BAT file 8-4
automatic program
 execution 8-4
AUX: reserved device
 name 7-6, 7-7

B

B> prompt 3-36
backing up the system
 3-34, 3-72
backspace key 3-9
backslash (\) 6-10
BACKUP 7-18
back up 3-34, 3-72
BASIC 3-118
Basic/Basica 3-118
batch processing 8-1
booting 3-25
Break key (see Scroll Lock/
 Break) 3-10

C

cancel current operation 6-2
Caps Lock Indicator 3-5
Caps Lock key 3-11
cardboard retainer 2-10
carrying handle 2-5
CF-PR110 1-2
change directory 7-21
character codes, printer 4-38
checking electrical
connections A-4
CHDIR 7-21
CHKDSK 7-22
class (LINK) 10-6
CLS 7-24
connecting the AC power cord
socket 2-15
command characters
+ (LINK) 10-12
; (LINK) 10-13
command prompts
libraries (LINK) 10-8
list file (LINK) 10-8
object modules (LINK) 10-7
run file (LINK) 10-7
Commands- DOS
BREAK 7-20
CD 7-21
CHDIR 7-21
CHKDSK 7-22
CLS 7-24
COPY 7-25
CTTY 7-33
DATE 7-34
DEL 7-35
DIR 7-36
DISKCOMP 7-40
DISKCOPY 7-43
ECHO 8-9
ERASE 7-46
EXE2BIN 7-48
external command 8-4
EXTPRT 7-51

FAST 7-52
FC 7-53
FDISK 7-57
FIND 7-58
FOR 8-11
FORMAT 7-59
GRAPHICS 7-63
GOTO 8-12
IF 8-13
internal command 7-4
MKDIR 7-64
MODE 7-66
MORE 7-72
PATH 7-73
PAUSE 8-16
PRINT 7-75
PROMPT 7-79
RECOVER 7-82
REM 8-17
RENAME 7-84
RESTORE 7-85
RMDIR 7-87
SET 7-88
SHIFT 8-18
SORT 7-91
SYS 7-92
TIME 7-93
TREE 9-95
TYPE 7-96
VER 7-97
VERIFY 7-98
VOL 7-99
commands, summary of 7-10
CON:—reserved device
name 7-7
concatenation 7-30
control codes, printer 4-10
control keys 3-9
COPY command 7-25
copying files 3-61, 3-101, 7-25
C>prompt 3-74

CPU clock frequency 3-111
creating a file 3-40, 3-80
CTRL key 3-10
CTRL Z 3-48, 3-88
CTTY 7-33
current date 7-34
current directory 6-10
current time 7-93
cursor 3-25

D

DATE (command) 7-34
date, entering or changing
3-114
DEBUG
ASSEMBLE 11-10
COMPARE 11-13
DUMP 11-14
ENTER 11-16
FILL 11-18
GO 11-19
HEX 11-21
INPUT 11-22
LOAD 11-23
MOVE 11-25
NAME 11-26
OUTPUT 11-28
QUIT 11-29
REGISTER 11-30
SEARCH 11-32
TRACE 11-33
UNASSEMBLE 11-34
WRITE 11-36
default drive 3-36, 3-74
default drive, changing
3-36, 3-74
DEL (command) 7-35
DEL key 3-13
demonstration software I-1
density selector 2-17
device designations 3-35, 3-74

device names, reserved
6-6, 7-7
diagnostic tests A-1
DIP switches 2-42
DIR 7-36
directories 6-8
directory
creating 6-14
deleting 6-14
setting 6-15
disk errors
data B-3
disk B-3
not ready B-4
sector not found B-4
seek B-4
write fault B-4
write protect B-4
disk drive 2-9
disk
backup 3-34, 3-72
copying 3-61, 3-101
formatting 3-52, 3-92
purchasing 3-17
source 3-34, 3-72
target 3-34, 3-72
DISKCOPY 7-43
display panel 2-11
drive specifier 3-45, 3-85
dummy device 6-7, 7-7
DOS Partition 5-12

E

ECHO 8-9
EDIT 9-26
editing commands
 copy multiple characters 9-8
 copy cone character 9-7
 copy template 9-9
 insert mode 9-13
 new template 9-15
 quit input 9-12
 skip multiple characters 9-11
 skip one character 9-10
editing keys 9-6
EDLIN 9-1
EDLIN Commands
 Append Lines 9-21
 Copy Lines 9-22
 Delete Lines 9-24
 Edit Lines 9-26
 End Editing 9-28
 Insert Text 9-29
 List Text 9-32
 Move Text 9-35
 Page 9-36
 Quit 9-37
 Replace Text 9-38
 Search Text 9-41
 Transfer Text 9-44
 Write Lines 9-45

END key 3-12
enter date 3-31, 7-34
ENTER key 3-10
enter time 3-32, 7-93
entering a command 7-3
ERASE 3-69, 3-108
error indicator 2-18
ESC key 3-9
escape sequences, printer 4-10
EXE2BIN 7-48
execution, automatic program 8-4
expansion RAM 2-39
extensions, filename 3-44, 3-8
external commands 7-4
Ext.Printer Indicator 3-6
external printer 3-116, 4-3

F

fast speed 3-111
Fast Speed Indicator 3-6
FDISK 5-10
filename 3-41, 3-81
filename extension 3-42, 3-82
filename extensions—default
 .EXE (LINK) 10-7
 .LIB (LINK) 10-8
 .MAP (LINK) 10-8
 .OBJ (LINK) 10-7
format 7-59
valid characters 6-6
filename, valid characters 6-6

files
 creating 3-40, 3-80
 copying 4-61, 3-101
 deleting 3-69, 3-109
 listing 3-56, 3-96
 naming 3-67, 3-107
 renaming 3-67, 3-107
 typing 3-65, 3-106
filespec 7-6
filters 12-5
FIND 7-58
floppy disk—see disk 3-17
FOR 8-11
FORMAT 3-52, 3-92, 7-59
form feed button 2-24
function keys 3-16
fuse, replacing A-5

G

GOTO 8-12
group (LINK) 10-6

H

hard disk 5-2
hierarchical directory 6-9
Home key 3-12

I

ignore B-5
illegal filenames 3-43, 3-83
input redirections 12-2
insert mode 3-12
inside cover 2-20
INS key 3-12
internal commands 7-4
internal printer 4-2
In Use Indicator 3-29

K

keyboard 3-5
 using 3-5
keys
 alphabet 3-7
 ALT 3-11
 backspace 3-9
 break 3-10
 control 3-10
 cursor control 3-13
 delete 3-13
 end 3-12
 enter 3-10
 escape 3-9
 function 3-16
 insert 3-12
 minus 3-13
 number 3-12
 num lock 3-12
 page up 3-12
 page down 3-12
 plus 3-13
 print screen 3-11
 scroll lock 3-10
 shift 3-11
 spacebar 3-8
 tab 3-10
keypad, numeric 3-12

L

lever, disk drive load 3-20
line feed, printer 2-16
line feed button 2-24
LINK 10-1
lock button 2-26

M

main power switch 2-13
memory count 3-28
memory
 expansion 2-36
 RAM 2-36
messages, error
 diagnostic A-31
 DOS B-1
minus key 3-13
MKDIR 7-64
MODE 7-66
MORE 7-72

N

NUL:—reserved device
 name 6-7, 7-7
number lock key 3-12
numeric keypad 3-12
Num Lock Indicator 3-5

O

object modules 10-7
on, turning the computer 3-26
off, turning the computer
 2-35
ON/OFF power switch 2-13
opening screen 3-29
operating system 6-2
options 1-2
option board installation 2-45
output redirection 12-2

P

page down 3-12
page up 3-12

paper, removing from
 printer 2-22
parameters, replaceable 8-6
Partition 5-12
paper feed knob 2-19, 2-30
paper, inserting in printer
 2-17, 2-29
parallel ports C-5
PATH 7-73
path 6-12
PAUSE 8-16
PFS: File G-6
PFS: REPORT G-7
PFS: GRAPH G-8
pin configurations
 expansion bus port
 serial port C-4
 parallel port C-2
pipe 12-4
Plasma display 2-9
plastic cover 2-24
plus key 3-13
PRINT 7-75
print screen key 4-6, 6-4
printer
 ASCII codes 4-38
 control codes 4-10
 Printer control codes bit image
 graphics mode 4-31
 Printer control codes
 compressed mode 4-21
 Printer control codes CR
 4-12
 Printer control codes double
 strike mode 4-24
 Printer control codes double
 width mode 4-19
 Printer control codes
 emphasized mode 4-22
 Printer control codes HT
 4-19
 Printer control codes LF
 4-15

Printer control codes line
 feed 4-12
Printer control codes page
 length 4-17
Printer control codes skip
 perforation function 4-34
Printer control codes
 subscript mode 4-26
Printer control codes
 superscript mode 4-25
Printer control codes
 underline mode 4-27
Printer control codes
 unidirectional printing
 mode 4-30
Printer control codes VT
 4-18
printer DIP switches 4-8
printer head 2-29
printer ribbon 2-28
diagnostic test A-7
error indicator 2-18
escape sequences 4-8
inserting paper 2-20, 2-32
paper guide 2-23
paper release lever
 2-22, 2-32
paper rod 2-21
paper tab 2-28
removing paper 2-25
PRN:—reserved device
 names 6-6, 7-7
programs execution,
 automatic 8-4
PROMPT 7-79

R

RAM 2-36
RD-9442 1-2
RD-9655 1-2
RD-9671 1-2
Real Time Clock 3-31, 3-114
rear panel 2-14
RECOVER 7-82
redirecting output 12-2
REM 8-17
RENAME 7-84
replaceable parameters 8-6
reserved device names
 AUX 6-6, 7-7
 CON 6-6, 7-7
 LPT 6-7, 7-7
 NUL 6-7, 7-7
 PRN 6-7, 7-7
 RESTORE 7-85
retry B-5
RMDIR 7-87
root directory 6-8
RS232C port C-4

S

scroll lock/break key 3-10
sector 3-18
segments (LINK) 10-6
serial port C-4
SET 7-88
set date 3-31, 3-114
set time 3-32, 3-115
SHIFT (command) 8-18
Shift key 3-11
side panels 2-12

Sort 7-91
source disk 3-34, 3-72
spacebar 3-8
special keys for the printer 4-6
specifications J-1
standard speed 3-111
subdirectory 6-9
switches

/DSALLOCATE (LINK)
10-9

/HIGH (LINK) 10-10

/LINENUMBERS

(LINK) 10-10

/MAP (LINK) 10-10

/NO (LINK) 10-11

/PAUSE (LINK) 10-10

/STACK (LINK) 10-11

system

backing up 3-34, 3-72

bringing up 3-25

booting 3-25

System Disk 3-25

System Diagnostics Menu A-8

T

tab key 3-10

target disk 3-34, 3-72

The internal printer 2-17

thermal printing 2-20

thermal transfer paper 2-32

time, change 3-115

top panel 2-14

tracks 3-18

transporting system. 2-32

Transporting the Exec.

Partner 2-35

turning OFF system 2-35

turning ON system 3-26

TYPE 7-96

two ribbed-tabs 2-8

Two types printing

methodes 2-19

U

User Diagnostics

error messages A-47

testing procedures A-2

V

VER 7-97

VERIFY 7-98

VM.TMP (LINK) 10-5

VOL 7-99

voltage switch 2-13

W

wild cards

* 6-7

? 6-7

WORDSTAR G-8

write protect notch 3-24

write protect tab 3-24

USA

Panasonic Industrial Company
Division of Matsushita Electric Corporation of America
One Panasonic Way,
Secaucus, New Jersey 07094

Panasonic Hawaii Inc.
91-238 Kauhi St. Ewa Beach
P.O. Box 774
Honolulu, Hawaii 96808-0774

Panasonic Sales Company
Division of Matsushita Electric of Puerto Rico, Inc.
Ave. 65 De Infanteria, KM 9.7
Victoria Industrial Park
Carolina, Puerto Rico 00630

CANADA

Matsushita Electric of Canada Limited
5770 Ambler Drive, Mississauga,
Ontario L4W 2T3

OTHERS

Matsushita Electric Trading Co., Ltd.
32nd floor, World Trade Center Bldg.,
No. 4-1, Hamamatsu-Cho 2-Chome,
Minato-Ku, Tokyo 105, Japan
Tokyo Branch P.O. Box 18 Trade Center