

Operating Systems and Languages Library

PCOS *(Professional Computer Operating System)*

User Guide



**OLIVETTI
PERSONAL
COMPUTER
M24**

olivetti

... of the ...
... of the ...
... of the ...

AMMUS
1954

The ... of the ...
... of the ...
... of the ...

The ... of the ...
... of the ...
... of the ...

The ... of the ...
... of the ...
... of the ...

The ... of the ...
... of the ...
... of the ...

The ... of the ...
... of the ...
... of the ...

RECEIVED

- 1. ... (Code 400120)
- 2. ... (Code 400120)
- 3. ... (Code 400120)
- 4. ... (Code 400120)
- 5. ... (Code 400120)
- 6. ... (Code 400120)
- 7. ... (Code 400120)
- 8. ... (Code 400120)
- 9. ... (Code 400120)
- 10. ... (Code 400120)

DISTRIBUTION - General (C)

...

PUBLISHED BY:

...
...
...

...

vq : +prt

mv. Elenor

Notice:

Revision 1.0 of PCOS is subject to the following restrictions:

- The IEEE-488 interface is not yet supported.
- The DELTA and ARABIC keyboards are not yet implemented.
- The following printers are not supported:

PR19/B

PR38/B

PR342B / PR341B

ET111 / ET121

These restrictions must be taken into consideration when referring to PCOS documentation.

Notice:

Revision 1.0 of PCOS is subject to the following restrictions:

- The IEEE-488 interface is not yet supported.
- The DELTA and KIM-1 keyboards are not yet implemented.
- The following programs are not supported:

PCAT08

PCAT12

PCAT28 / PCAT32

PCAT48 / PCAT52

These restrictions may be lifted from time to time without notice. For more information, contact the author.

PREFACE

This manual describes the Professional Computer Operating System (PCOS) and may be used with an OLIVETTI M24 Personal Computer. It is directed at the user who has some experience of computer programming and is familiar with computing terminology.

SUMMARY

The manual is made-up of two parts. Part I comprises Chapters 1 to 13 and contains introductory and operational information. Part I should be read before attempting to use Part II which provides a command reference.

The first three chapters of Part I are introductory. They provide an overview of the M24.

Chapter 4 describes the notation used throughout the manual. It also describes the rules, and defines the terminology used, for entering PCOS commands.

Chapters 5 to 13 provide operational details of how to use PCOS.

Part II comprises Chapter 14 and should be used as a reference. It describes all the PCOS commands in alphabetical order. Each command description includes the command action, a syntax diagram, the characteristics of the command, and examples.

REFERENCE:

BASIC - 8000 User Guide - Code 4001550E (0)

BASIC & PCOS Pocket Reference - Code 4002000Y (0)

Installation and Operations Guide - Code 3986490W (1)

Serial Interface for I/O Peripherals User Guide - Code 4009750C (0)

MS-DOS User Guide - Code 400140G (0)

Concurrent DOS User Guide - Code 4001490Q (0)

DISTRIBUTION: General (G)

EDITION: September 1984

Olivetti is a trademark of Ing. C. Olivetti & C., S.p.A.
OLITERM is a trademark of Ing. C. Olivetti & C., S.p.A.
SORTP is a trademark of Ing. C. Olivetti & C., S.p.A.

PUBLICATION ISSUED BY:

Ing. C. Olivetti & C., S.p.A.
Direzione Documentazione
77, Via Jervis - 10015 Ivrea (Italy)

Copyright © 1984 by Olivetti
All rights reserved

TRADEMARKS NOTICE

- Concurrent CP/M-86 is a trademark of Digital Research
- Personal Basic is a trademark of Digital Research
- GSX-86 is a trademark of Digital Research
- CBASIC-86 is a trademark of Digital Research
- DR LOGO is a trademark of Digital Research
- DDT-86 is a trademark of Digital Research
- CP/M-86 is a trademark of Digital Research
- CB-86 is a trademark of Digital Research
- ASM-86 is a trademark of Digital Research
- The p-System is a trademark of Softech Microsystem, Inc.
- UCSD Pascal is a trademark of the Regents of the University of California
- PEACHPAK is a trademark of Peachtree Software International Ltd.
- UNIX is a trademark of Bell Laboratories
- PC-DOS is a trademark of International Business Machines Corp.
- Z80 is a registered trademark of Zilog Inc.
- Z8000 is a trademark of Zilog Inc.
- MS is a trademark of Microsoft Corporation
- GW is a trademark of Microsoft Corporation
- SID-86 is a trademark of Digital Research
- IBM is a registered trademark of International Business Machines Corp.
- Concurrent DOS is a trademark of Digital Research

CONTENTS

PAGE

1-1	1. <u>INTRODUCTION</u>
1-1	<u>INTRODUCTION</u>
1-2	COMMANDS
1-2	KEYBOARD
1-3	BASIC LINE EDIT
2-1	2. <u>GETTING STARTED WITH PCOS</u>
2-1	<u>THE KEYBOARD</u>
2-1	LINE TERMINATOR KEYS
2-2	THE CTRL KEY
2-2	LOCKING THE SHIFT KEY
2-3	SCREEN PRINT
2-3	LOGICAL RESET
2-3	ASSIGNING VALUES TO KEYS
2-3	KEYBOARD BUFFER WARNING BUZZER
2-3	<u>PHYSICAL RESET</u>
2-3	DISKETTE HANDLING
2-4	LABELING DISKETTES
2-4	WRITE-PROTECTION
2-5	INSERTING AND REMOVING DISKETTES
2-5	<u>TUTORIAL</u>
2-5	<u>SYSTEM STARTUP AND BOOT</u>
2-5	POWER ON

PAGE	
2-6	POWER-ON TESTS
2-6	BOOTSTRAP
2-6	CONFIGURATION
2-6	PCOS BANNER
2-6	PROMPT
2-6	<u>SYSTEM REBOOT</u>
2-7	PHYSICAL RESET
2-7	LOGICAL RESET
2-7	<u>THE COMMAND LINE</u>
2-7	COMMAND TYPES
2-8	LOADING
2-8	<u>ENTERING COMMANDS</u>
2-8	DCONFIGURE
2-8	REEXECUTION
2-8	<u>DIRECTORY LISTING (QUICK LIST)</u>
2-8	LISTING A VOLUME
2-9	LISTING FILES
2-9	EXAMPLES
2-10	<u>DIRECTORY LISTING (LONG LIST)</u>
2-10	SYSTEM DIRECTORY
2-10	EXAMPLES
2-11	<u>DEFAULT DRIVE</u>
2-11	COMMAND PROMPT
2-11	CHANGING DEFAULT
2-12	ERRORS
2-12	NAMING DISK
2-12	<u>COPYING DISKS</u>

PAGE

2-12	VCOPY
2-13	CHECKING DISK
2-13	<u>FCOPY</u>
2-13	<u>COPYING FILES</u>
2-14	EXAMPLES
2-14	OVERWRITE
2-14	PLOADING COMMANDS
2-14	PUNLOAD
2-15	<u>ERROR MESSAGES</u>
2-15	ENTRY ERRORS
2-15	EPRINT COMMAND
2-15	OPERATING ERRORS
3-1	3. <u>SOFTWARE COMPONENTS</u>
3-1	<u>INTRODUCTION</u>
3-3	<u>PCOS</u>
3-4	MEMORY OPTIMIZATION
3-4	RESIDENT AND TRANSIENT COMMANDS
3-4	PROGRAMMABLE KEYS
3-5	PROTECTION MECHANISMS
3-5	LINE EDITOR FUNCTIONS
3-5	REAL-TIME CLOCK
3-5	ROUTING INPUT/OUTPUT
3-6	USER-DEFINED FONTS
3-6	CONTROL CHARACTER DISPLAY
3-6	INITIALIZATION FILES
3-6	USER-CONFIGURABLE OPERATING SYSTEM
3-7	<u>BASIC INTERPRETER</u>

PAGE	
3-7	<u>VIDEO FILE EDITOR</u>
3-8	<u>ASSEMBLER</u>
3-9	<u>PASCAL</u>
3-9	<u>LINKER</u>
3-9	<u>PROGRAM DEBUGGER</u>
3-9	<u>STANDARD INTERFACE HANDLERS</u>
3-11	<u>PCOS COMMAND LIBRARY</u>
3-11	CHANGING ENVIRONMENT COMMANDS
3-11	PCOS CONFIGURING COMMANDS
3-12	SET SYSTEM GLOBAL COMMANDS
3-12	KEYBOARD-RELATED COMMANDS
3-13	VOLUME HANDLING COMMANDS
3-14	FILE HANDLING COMMANDS
3-15	STANDARD INTERFACE HANDLING COMMANDS
3-15	PCOS GRAPHIC FACILITY COMMANDS
3-16	ALTERNATIVE OPERATING SYSTEM RELATED COMMANDS
3-16	USER AIDS
4-1	4. <u>ENTERING A COMMAND</u>
4-1	<u>NOTATION CONVENTION</u>
4-2	<u>COMMAND SYNTAX</u>
4-3	<u>COMMAND NAMES AND KEYWORDS</u>
4-3	<u>PARAMETERS</u>
4-5	DEFAULT VALUES
4-5	<u>RESIDENT AND TRANSIENT COMMANDS</u>
4-6	<u>COMMAND SEARCH PROCEDURE</u>
4-6	<u>FILE AND VOLUME IDENTIFIERS</u>
4-10	<u>WILD CARDS</u>

PAGE

4-10	<u>NO-INTERACTION FLAG</u>
4-11	OTHER FLAGS
5-1	<u>5. INITIALIZATION AND CHANGING ENVIRONMENTS</u>
5-1	<u>M24 ENVIRONMENTS</u>
5-1	PCOS
5-1	BASIC
5-1	VIDEO FILE EDITOR
5-2	CHANGING ENVIRONMENTS
5-5	<u>MODES OF PCOS</u>
5-5	COMMAND MODE
5-5	EXECUTION MODE
5-6	CHANGING MODES
5-6	<u>STANDARD INITIALIZATION</u>
5-7	INITIALIZATION FILES
5-8	<u>NONSTANDARD INITIALIZATION</u>
5-9	INITIALIZATION FOLLOWING A PSAVE COMMAND
5-9	INITIALIZATION FOLLOWING A PRUN COMMAND
5-9	<u>THE INITIALIZATION FLOWCHART</u>
5-13	<u>BEGINNING AND ENDING A WORKING SESSION</u>
6-1	<u>6. CONFIGURING PCOS</u>
6-1	<u>INTRODUCTION</u>
6-2	<u>MAKING TRANSIENT COMMANDS RESIDENT</u>
6-6	<u>ASSIGNING STRINGS TO KEYS</u>
6-8	KEYBOARD CHANGES
6-8	<u>CHANGING SYSTEM GLOBAL PARAMETERS</u>
6-9	SETTING THE SYSTEM GLOBAL ENVIRONMENT
6-13	DISPLAYING AND MODIFYING DEVICE NAMES

PAGE	
6-14	MODIFYING THE BASIC ENVIRONMENT
6-16	SETTING THE ENVIRONMENT FOR AN RS-232-C COMMUNICATIONS PORT
6-16	SETTING THE PRINTING ENVIRONMENT
6-19	RECONFIGURING THE KEYBOARD LANGUAGE
6-20	ENABLING TRANSLATION OF 7-BIT CHARACTER CODES TO 8-BIT
6-23	<u>SAVING A USER-CONFIGURED PCOS</u>
7-1	<u>7. DEVICE REROUTING</u>
7-1	<u>INTRODUCTION</u>
7-2	<u>LOCAL DEVICE REROUTING</u>
7-5	<u>GLOBAL DEVICE REROUTING</u>
7-7	<u>DEVICE REROUTING FROM A BASIC PROGRAM</u>
8-1	<u>8. PROTECTION TOOLS</u>
8-1	<u>INTRODUCTION</u>
8-1	<u>VOLUME PASSWORDS</u>
8-2	<u>FILE PASSWORDS</u>
8-4	<u>WRITE-PROTECTION</u>
8-4	<u>COPY-PROTECTION</u>
8-5	<u>BASIC PROGRAM SECURITY</u>
9-1	<u>9. VOLUME HANDLING</u>
9-1	<u>PARTITIONING AND INITIALIZING NEW VOLUMES</u>
9-1	INITIALIZING A VOLUME
9-2	<u>LISTING A VOLUME</u>
9-2	LISTING A VOLUME USING THE VLIST COMMAND
9-3	LISTING A VOLUME USING THE VQUICK COMMAND
9-4	<u>COPYING VOLUMES</u>
9-5	COPYING VOLUMES ON A DUAL-DRIVE SYSTEM (VOLUMES OF EQUAL SIZE)
9-6	COPYING DISKETTES USING ONE DRIVE (VOLUMES OF EQUAL SIZE)

PAGE

9-7	COPYING VOLUMES (OF DIFFERENT SIZES)
9-7	BACKING UP AND RESTORING THE HARD DISK
9-9	<u>NAMING AND PROTECTING A VOLUME</u>
9-9	WRITE-PROTECTION
9-10	PASSWORD PROTECTION
9-11	NAMING A VOLUME
9-11	<u>ALPHABETIZING A VOLUME</u>
10-1	<u>10. FILE HANDLING</u>
10-1	<u>CREATING FILES</u>
10-1	CREATING AN EMPTY FILE
10-2	CREATING A FILE BY COPYING
10-2	<u>COPYING FILES</u>
10-3	COPYING FILES ON A DUAL DRIVE SYSTEM
10-5	COPYING FILES USING ONE DRIVE
10-6	BACKING UP HARD DISK FILES
10-8	<u>LISTING FILES</u>
10-10	<u>PROTECTING FILES</u>
10-10	PASSWORD PROTECTION
10-11	WRITE-PROTECTION
10-12	<u>FREEDING UNUSED FILE BLOCKS</u>
10-15	<u>DELETING AND RECOVERING FILES</u>
10-15	DELETING FILES
10-16	RECOVERING DELETED FILES
10-17	<u>RENAMING FILES</u>
11-1	<u>11. PCOS GRAPHIC AND CONSOLE-RELATED FACILITIES</u>
11-1	<u>INTRODUCTION</u>
11-1	<u>DISPLAYING LABELS</u>

PAGE	
11-3	<u>PRINTING THE SCREEN IMAGE</u>
11-4	USING THE SPRINT AND LSCREEN COMMANDS
11-5	<u>ENTERING CHARACTERS AT THE KEYBOARD</u>
11-5	RAW KEY CODES
11-6	ASCII TABLES
11-8	INTERPRETATION OF ASCII TABLE OUTPUT
11-10	<u>CREATING USER-DEFINED FONTS</u>
11-11	CREATING A FONT MATRIX FILE USING THE RFONT COMMAND
11-11	THE FONT MATRIX FILE
11-14	USING THE WFONT COMMAND
11-15	<u>DISPLAYING CONTROL CHARACTERS</u>
11-19	<u>THE LINE TERMINATION KEYS</u>
12-1	<u>12. FACILITIES RELATED TO ALTERNATIVE OPERATING SYSTEMS</u>
12-1	<u>INTRODUCTION</u>
12-1	<u>HOW TO PREPARE THE HARD DISK FOR USE WITH ALTERNATIVE OPERATING SYSTEMS</u>
12-2	HOW TO PARTITION THE HARD DISK
12-4	HOW TO INITIALIZE EACH PARTITION
12-4	HOW TO CHECK THE BAD BLOCK LIST
12-5	HOW TO UPDATE THE BAD BLOCK LIST
12-5	<u>HOW TO COPY THE OPERATING SYSTEM TO THE HARD DISK</u>
13-1	<u>13. VIDEO FILE EDITOR</u>
13-1	<u>INTRODUCTION</u>
13-1	THE DISPLAY
13-2	THE KEYBOARD
13-5	<u>HOW TO INVOKE THE VIDEO FILE EDITOR</u>
13-5	EDIT.COMD
13-7	<u>GENERAL EDITING FUNCTION KEYS</u>

PAGE

13-14	<u>WINDOW MOVING FUNCTION KEYS</u>
13-15	<u>EXITING AND SAVING FUNCTION KEYS</u>
13-16	<u>COMMANDS AND SEARCHING</u>
13-16	STRING SEARCHES
13-18	COMMANDS
13-18	GOTO
13-19	DELETE
13-19	FILE
14-1	14. <u>PCOS COMMANDS</u>
14-1	BASIC.CMD
14-3	BVOLUME.SAV
14-4	BVOLUME search
14-6	BVOLUME diskfree
14-7	BVOLUME getvolname
14-9	CI.SAV
14-9	CKEY.CMD
14-14	DCONFIG.CMD
14-17	EDIT.CMD
14-17	EPRINT.SAV
14-19	FCOPY.CMD
14-24	FDEPASS.CMD
14-26	FFREE.CMD
14-28	FKILL.CMD
14-29	FLIST.CMD
14-31	FMOVE.CMD
14-33	FNEW.CMD
14-35	FPASS.CMD

PAGE	
14-37	FRENAME.COMD
14-38	FUNPROT.COMD
14-40	FWPROT.COMD
14-41	HBACKUP.COMD
14-44	HDISK.COMD
14-45	HRESTORE.COMD
14-47	IEEE.SAV
14-48	LABEL.COMD
14-53	LSCREEN.COMD
14-55	LTERM (ALWAYS RESIDENT)
14-56	PKEY.COMD
14-60	PLOAD (ALWAYS RESIDENT)
14-62	PLOT.COMD
14-67	PRUN.COMD
14-69	PSAVE.COMD
14-71	PUNLOAD (ALWAYS RESIDENT)
14-73	RFONT.COMD
14-75	RKILL.COMD
14-76	RS232.SAV
14-76	SBASIC.COMD
14-80	SCOMM.COMD
14-80	SDEVICE.COMD
14-82	SFORM.COMD
14-85	SLANG.COMD
14-87	SPRINT.COMD
14-89	SSYS.COMD
14-93	TLANG.SAV

PAGE

14-95 VALPHA.CMD
14-97 VCOPY.CMD
14-99 VDEPASS.CMD
14-100 VLIST.CMD
14-102 VMOVE.SAV
14-104 VNEW.CMD
14-107 VPASS.CMD
14-109 VQUICK.CMD
14-111 VRENAME.CMD
14-113 VVERIFY.CMD
14-116 WFONT.CMD

A-1 A. ASCII CODE
B-1 B. NATIONAL KEYBOARDS LAYOUTS AND CODES
C-1 C. HARD DISK AND DISKETTE CHARACTERISTICS
C-1 THE HARD DISK UNIT
C-2 CHARACTERISTICS
C-3 DISKETTES
C-4 CHARACTERISTICS
D-1 D. DIAGNOSTIC/BOOTSTRAP ERROR MESSAGES
D-1 DIAGNOSTIC ERROR MESSAGES
D-4 BOOTSTRAP ERROR MESSAGES
E-1 E. PCOS AND BASIC ERROR MESSAGES
E-1 INTRODUCTION
E-1 PCOS AND BASIC ERRORS
F-1 F. PCOS 4-0 ENHANCEMENTS
F-1 PCOS ENHANCEMENTS -- RELEASE 3-0 TO RELEASE 4-0
F-1 NEW COMMANDS
F-2 ENHANCEMENTS TO EXISTING COMMANDS

PAGE

G-1	G. <u>GLOSSARY OF TERMS</u>
G-1	<u>GLOSSARY OF TERMS</u>
H-1	H. <u>KEYBOARD CHANGES</u>
H-1	<u>INTRODUCTION</u>
H-1	<u>FUNCTION KEYS</u>
H-2	<u>NEW KEYS</u>
H-3	<u>NEW ACTIONS PERFORMED</u>
H-3	<u>NEW MAPPINGS</u>

1. INTRODUCTION

ABOUT THIS CHAPTER

This chapter provides a general introduction to the M24.

CONTENTS

<u>INTRODUCTION</u>	1-1
COMMANDS	1-2
KEYBOARD	1-2
BASIC LINE EDIT	1-3

INTRODUCTION

The Olivetti M24 is a stand-alone system designed for professional use as a problem-solving tool. It has the versatility to help the businessman, the scientist, the student, and the technician to process information quickly and accurately.

The Professional Computer Operating System (PCOS) runs on the M24 equipped with the Z8000 softcard. PCOS provides an environment for a set of programming tools that enable you to develop and run application programs. Programming facilities include: an extensive BASIC-8000 Interpreter, an (optional) Assembler package comprising assembler and program debugger, an (optional) Pascal compiler, and a linker. Moreover, the Video File Editor enables programs written in BASIC-8000, Z8000 Assembler, or Pascal to be created and modified.

N.B. PCOS, Pascal under PCOS, BASIC-8000 and Z8000 Assembler imply the use of the Z8001 alternate processor unit. For the sake of brevity, all further references to BASIC-8000 and Z8000 Assembler will be written simply as BASIC and Assembler.

The PCOS command library comprises resident and transient commands. Optimum use of memory is assured by the use of transient commands which are automatically removed from memory when no longer required. However, the PCOS command library contains a group of commands that enable transient commands to be made resident. A further group of commands enables you to set global parameters as required. Using these two groups, you can tailor the operating system to suit your specific needs.

Further functional groups of commands facilitate volume and file handling (including protection mechanisms), keyboard handling, standard interface handling, graphic facilities, and a set of user aids.

PCOS on the M24 has several enhancements, made possible by the improved software and hardware.

Note:

From now on we shall use:

- diskette instead of 5 1/4 in. floppy disk for brevity
- disk instead of either a diskette or the hard disk.

COMMANDS

PCOS on the M24 includes the following new or enhanced commands and utilities:

- TLANG.SAV enhances translation between PCOS 7-bit national-equivalent ASCII character codes and those systems and devices using an 8-bit ASCII code.
- PLOT.CMD allows the user to direct a plotter by means of PCOS commands from the terminal.
- HBACKUP.CMD and HRESTORE.CMD allow files on the hard disk to be backed up onto a series of diskettes, and then to be restored to the hard disk.
- SFORM.CMD has been enhanced for the RS-232-C expansion port.
- SPRINT.CMD has been enhanced to permit color printing. Once the SPRINT command is PLOADED, the new /SCR PRT/ key also performs the SPRINT function.
- VVERIFY.CMD has been modified. PCOS now maintains its own bad sector list within its own partition. VVERIFY checks bad block addresses to ensure that they are within the PCOS partition. The final message displays the number of bad blocks, and the beginning and ending cylinder numbers of the PCOS partition. See Chapters 12 and 14.

Diskettes and the hard disk are now factory-formatted, so VFORMAT.CMD is no longer necessary. Use VNEW.CMD to initialize volumes. HDISK.CMD creates the PCOS partition on the hard disk. See Chapter 12 for full instructions on preparing the hard disk.

KEYBOARD

The M24 keyboard has several new groups of keys:

- The F-keys are function keys, for ease and speed. They are mapped to the same functions as the top-row key-combinations on the M20, plus other key-combinations on the extended keyboard. Users accustomed to the M20 keyboard can continue to use the same key-combinations on the M24.
- Four cursor-movement keys are in easy reach from the main keyboard. The shifted number-key combinations for cursor movement are still available, as on the M20 numeric keypad.
- The following keys have been added: /CLEAR/, /BREAK/, /SCROLL ON/, /HELP/, /CAPS LOCK/, and an additional /CR/ on the numeric keypad. Reprogramming of keys by means of CKEY and PKEY is available for these extra keys as well as all the others previously available.

INTRODUCTION

BASIC LINE EDIT

The /FUNCT.LOCK/ /CR/ key-combination is used to invoke Line Edit Mode when in BASIC. The BASIC Line Editor is useful for correcting long or complicated lines without having to reenter them completely.

For convenience, it is suggested that you use the PKEY command to, in effect, make the carriage return after /FUNCT.LOCK/ unnecessary. This lets you invoke edit mode by simply pressing /FUNCT.LOCK/ when you are in the BASIC interpreter.

You can reprogram the /FUNCT.LOCK/ key as follows:

```
pkey %FE,'edit'
```

Now, when you enter /FUNCT.LOCK/, the screen displays EDIT in the upper left corner; the keys with BASIC editor function names on their front now provide line-edit functions, as follows:

- INS = "i"
- DEL = "d"
- END = "x"
- S1 = space
- /SHIFT/ S1 = backspace
- HOME = "1" (list line and go to beginning)

Enter the BASIC program line number, /CR/, and the line you wish to edit appears on the screen. After you make the editing changes, strike /CR/ to exit edit mode.

To edit additional lines, follow the same procedure, beginning with /FUNCT.LOCK/. To exit BASIC, type "system."

PCOS can be configured for a wide range of applications, thus maximizing usefulness of your Olivetti M24.

1945

The following information was obtained from the records of the Department of the Interior, Bureau of Land Management, on the subject of the land in question.

The land in question was acquired by the United States Government in 1864, and was then conveyed to the State of California by the Act of March 3, 1850.

The land in question was then conveyed to the State of California by the Act of March 3, 1850.

1945

The land in question was then conveyed to the State of California by the Act of March 3, 1850.

1945

1945

1945

1945

1945

1945

The land in question was then conveyed to the State of California by the Act of March 3, 1850.

The land in question was then conveyed to the State of California by the Act of March 3, 1850.

The land in question was then conveyed to the State of California by the Act of March 3, 1850.

1945

1945

1945

1945

1945

1945

... ..
... ..
... ..

2. GETTING STARTED WITH PCOS

...

...

...

...

...

...

...

...

...

...

...

...

...

ABOUT THIS CHAPTER

This chapter describes the various hardware components that make up the M24. They are discussed from the viewpoint of the function performed, the options available and physical controls.

CONTENTS

<u>THE KEYBOARD</u>	2-1
LINE TERMINATOR KEYS	2-1
THE CTRL KEY	2-2
LOCKING THE SHIFT KEY	2-2
SCREEN PRINT	2-3
LOGICAL RESET	2-3
ASSIGNING VALUES TO KEYS	2-3
KEYBOARD BUFFER WARNING BUZZER	2-3
<u>PHYSICAL RESET</u>	2-3
DISKETTE HANDLING	2-3
LABELLING DISKETTES	2-4
WRITE-PROTECTION	2-4

INSERTING AND REMOVING DISKETTES 2-5

TUTORIAL 2-5

SYSTEM STARTUP AND BOOT 2-5

POWER ON 2-5

POWER-ON TESTS 2-5

BOOTSTRAP 2-6

CONFIGURATION 2-6

PCOS BANNER 2-6

PROMPT 2-6

SYSTEM REBOOT 2-6

PHYSICAL RESET 2-7

LOGICAL RESET 2-7

THE COMMAND LINE 2-7

COMMAND TYPES 2-7

PLOADING 2-8

ENTERING COMMANDS 2-8

DCONFIGURE 2-8

REEXECUTION 2-8

<u>DIRECTORY LISTING (QUICK LIST)</u>	2-8
LISTING A VOLUME	2-8
LISTING FILES	2-9
<u>DIRECTORY LISTING (LONG LIST)</u>	2-10
SYSTEM DIRECTORY	2-10
<u>DEFAULT DRIVE</u>	2-11
COMMAND PROMPT	2-11
CHANGING DEFAULT	2-11
ERRORS	2-12
NAMING DISK	2-12
<u>COPYING DISKS</u>	2-12
VCOPY	2-12
CHECKING DISK	2-13
<u>FCOPY</u>	2-13
<u>COPYING FILES</u>	2-13
OVERWRITE	2-14
<u>PLOADING COMMANDS</u>	2-14
PUNLOAD	2-14

<u>ERROR MESSAGES</u>	2-15
ENTRY ERRORS	2-15
EPRINT COMMAND	2-15
OPERATING ERRORS	2-15

- Exit and save functions.
- Search functions to search the file for a specified string.
- A subset of "high-level" commands which enable you to:
 - . move the window to a specified line in the file
 - . delete blocks of text
 - . suspend processing of the current file and invoke the editor on another file.

For further details see Chapter 13.

ASSEMBLER

The (optional) PCOS Assembler processes an Assembly Language source file of ASCII text and produces an object file containing Z8000 machine code. Optionally, a listing file can be produced. This displays the source file program lines along with the generated code. The Assembler package also contains a number of commands to enable you to examine and manipulate your program files.

For details refer to the "Z8000 Assembler User Guide".

THE KEYBOARD

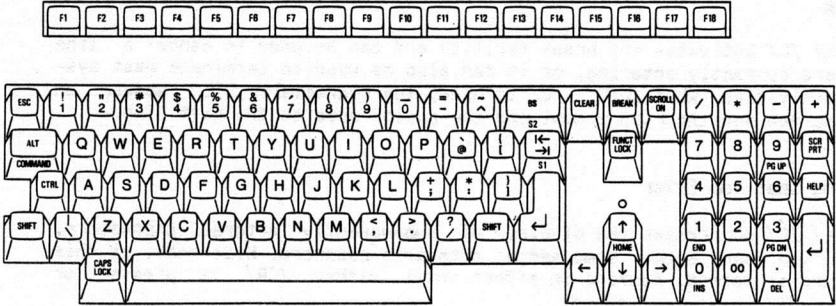


Fig. 2-1 The Keyboard

The keyboard shown in Figure 2-1 is the USA ASCII keyboard. Your keyboard may be that of another country, in which case the keyboard layout will be different. If so, you will find it described in Appendix B. The following description, however, applies to all keyboards.

The keyboard is divided into two sections: one for the entry of alphabetic, numeric, and control characters; the other - the numeric keypad - for rapid entry of numeric data. A set of programmable function keys is also provided.

LINE TERMINATOR KEYS

The /↵/, /S1/ and /S2/ keys often serve the same function. Any one of these keys can be used to complete a command, a program statement, a data entry request, or a request for an immediate calculation.

For brevity, these keys will subsequently be referred to as /CR/.

In a BASIC program, however, you can test to determine which of the three /CR/ keys was used in response to a data entry request. For some applications this is a valuable feature - see the LTERM command.

For details refer to the "Serial Interface for I/O Peripherals User Guide".

SCREEN PRINT

The /SCR PRT/ key will print the contents of your screen display on your printer, if you have one attached. Using this key is equivalent to entering the SPrint command, described in Chapter 14.

LOGICAL RESET

The key combination /CTRL/ /ESC/ causes a logical reset of the system. This reinitializes the system as described in Chapter 5.

ASSIGNING VALUES TO KEYS

All of the keys on your computer are "programmable", with the exception of the /CTRL/, /COMMAND/ and /SHIFT/ keys, and can be assigned values other than those shown on the keytops. These keys can be assigned a command name, an arithmetic expression, a numeric value, an algorithm, or any string of characters you may find useful to have at a single key-stroke. Assignment is made by means of the PKEY or CKEY commands.

KEYBOARD BUFFER WARNING BUZZER

When data is entered at the keyboard it is stored in a buffer which is subsequently read. In rare situations, however, it is possible that the buffer will become full; that is, data has been entered faster than it is read. In such situations a buzzer will be heard (when the buffer contains 56 characters), warning you that input may be lost if you continue to key-in data.

PHYSICAL RESET

Physical reset has the effect of switching the power off and on again. All system parameters are reset to their default values and the system is reinitialized, including a set of diagnostic tests. (Note that logical reset, as explained above, does not reset all system parameters, nor does it cause diagnostic tests to be performed.) See Chapter 5 for details.

DISKETTE HANDLING

Although diskettes are generally durable, damage to diskettes will be minimized if you take the following precautions:

- Never bend diskettes.
- Do not touch the exposed surface of the diskette.
- Always keep the diskette in its cardboard envelope when not in use and store it in the diskette carton.

SET SYSTEM GLOBAL COMMANDS

KEYWORD		
SHORT FORM	FULL MNEMONIC	COMMAND FUNCTION
sb	SBASIC.CMD	Sets the BASIC environment
sc	SCOMM.CMD	Sets the RS-232-C communications Port Environment
sd	SDEVICE.CMD	Changes device names
sf	SFORM.CMD	Sets the printer environment
sl	SLANG.CMD	Sets the national keyboard language
ss	SSYS.CMD	Sets the system environment

Table 3-3 Set System Global Commands

KEYBOARD-RELATED COMMANDS

KEYWORD		
SHORT FORM	FULL MNEMONIC	COMMAND FUNCTION
ck	CKEY.CMD	Changes the value of a key
lt	LTERM (resident)	Returns an integer (0, 1 or 2) depending on which of the three carriage return keys (↵, S1 or S2) was last used.
pk	PKEY.CMD	Assigns a string to a key

Table 3-4 Keyboard-Related Commands

INSERTING AND REMOVING DISKETTES

To insert a diskette you do the following:

- Open the drive cover.
- Insert the diskette into the slot with its label facing upward and nearest you.
- Push the diskette gently into the drive until you feel it click into position. Do not attempt to force it in; if it will not go, withdraw the diskette and reinsert it.
- Once the diskette has clicked into position, close the drive cover.

To remove a diskette you merely open the drive cover. This automatically pushes the diskette out of the drive so you can withdraw it easily.

You can insert and remove diskettes while the computer is powered-up or with the power off.

DO NOT ATTEMPT TO WITHDRAW THE DISKETTE WHILE IT IS BEING ACCESSED (DRIVE INDICATOR LIGHT ON). ATTEMPTING TO DO SO WILL CAUSE AN ERROR CONDITION AND IT MAY DESTROY THE INFORMATION ON THE DISKETTE.

TUTORIAL

The following is a short tutorial to give you hands-on experience in using PCOS on your system. In addition to your PCOS system diskette, you will also need a scratch (or new, Olivetti-formatted) diskette.

SYSTEM STARTUP AND BOOT

To start up (boot) the operating system, insert the PCOS system diskette into drive 0:. Turn on the power.

POWER ON

The display will respond with:

Select Alternate CPU (y/n)?

Answer "y" (no /CR/ required).

FILE HANDLING COMMANDS

KEYWORD		
SHORT FORM	FULL MNEMONIC	COMMAND FUNCTION
fc	FCOPY.CMD	Copies a file
fd	FDEPASS.CMD	Removes a password from a file
ff	FFREE.CMD	Frees unused file sectors
fk	FKILL.CMD	Deletes a file
fl	FLIST.CMD	Lists ASCII files
fm	FMOVE.CMD	Copies a file (diskette to diskette on a single-drive system)
fn	FNEW.CMD	Creates a new file
fp	FPASS.CMD	Assigns a password to a file
fr	FRENAME.CMD	Renames a file
fu	FUNPROT.CMD	Removes write protection from a file
fw	FWPROT.CMD	Assigns write protection to a file
rk	RKILL.CMD	Recovers a killed file

Table 3-6 File Handling Commands

PHYSICAL RESET

The physical reset allows the user to change the operating system by switching diskettes.

To perform a physical reset, press the reset button.

The system then returns to the initial prompt:

Select Alternate CPU (y/n)?

To select PCOS, place the PCOS System Diskette in drive 0: and answer 'y'.

Note: To change to Concurrent CP/M-86 or MS-DOS, place the relevant diskette in drive 0: and answer 'n'.

LOGICAL RESET

A logical reset reruns the power-on tests and then reboots the PCOS system diskette. To perform a logical reset, press /CTRL/ /ESC/.

THE COMMAND LINE

PCOS performs tasks in response to commands entered on the command line.

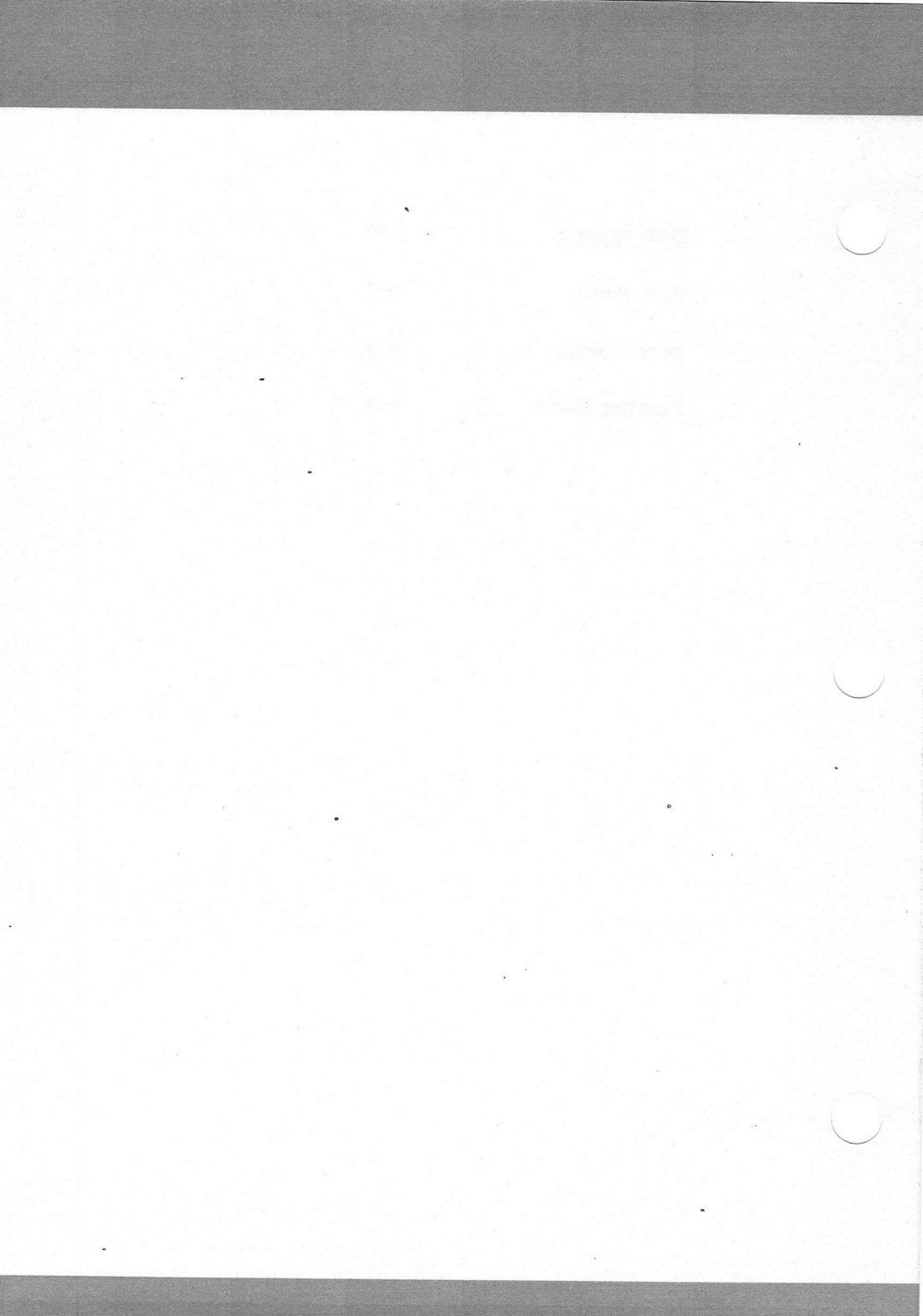
A command line consists of a command word (and its arguments, if any) entered at the keyboard and followed by a carriage return. The command word identifies the program or file to be loaded and/or executed by the CPU.

COMMAND TYPES

As in most systems, PCOS has two different types of commands: resident and transient.

RESIDENT Resident commands are loaded with PCOS, reside in memory, and can be executed immediately. The ONLY resident commands are PLOAD, PUNLOAD, and LTERM.

TRANSIENT The transient commands are stored on the disk as files and must be loaded into memory before they can be executed. Examples of this type are FCOPY, VCOPY, and VLIST. These take longer to execute, because of the loading time.



```

VOLUME: 0          Free Disk Blocks = 157
PCOS.SAV          basic.abs          basic.cmd          bvolume.sav
ci.sav           ckey.cmd           dconfig.cmd       eprint.sav
fcopy.cmd        fdepass.cmd        ffree.cmd         fkill.cmd
.               .               .               .
.               .               .               .
.               .               .               .
vrename.cmd     vverify.cmd       wfont.cmd
    
```

The information in the heading includes the drive (volume) number, the volume name (if any), and the number of free blocks.

LISTING FILES

To check whether a given file or files are on the disk, enter the name(s):

```
0: ffree.cmd
```

PCOS will display the banner again and, if present, the name of the file:

```

Volume Quick List Rev. 1.00
VOLUME: 0          Free Disk Blocks = 157
ffree.cmd
    
```

EXAMPLES

If you enter ...	Then ...
vq	The system lists all files on the current (last accessed) volume
vq 1:	The system lists the files on the disk in drive 1:
vq 0:*.cmd	The system lists all files on drive 0: that have the suffix '*.cmd'

THE CTRL KEY

The /CTRL/ key is always used in conjunction with other keys to add other meanings to existing keys.

Break

/CTRL/ /C/ activates the break facility and can be used to cancel a line you are currently entering, or it can also be used to terminate most system activities. When /CTRL/ /C/ is used, the characters "^C" appear on the screen and the PCOS prompt and the cursor move to the next line.

Hiding What You Enter

/CTRL/ /G/ suppresses the display of subsequently entered characters. Thus you can enter some secret data or a password. Hide mode, as this feature is termed, remains in effect until either /CR/ is pressed or /CTRL/ /G/ is pressed again.

Deleting Characters and Correcting Errors

/CTRL/ /H/ performs a backspace function. That is, it deletes the last character entered and moves the cursor one position to the left. In PCOS, you use this facility to correct any error you spot in a line before you have pressed /CR/. Simply delete the characters back to the point of the error, and then reenter the rest of the line correctly.

8-Character Tab

/CTRL/ /I/ advances the cursor to the next eight-character tab position on the screen.

Stopping and Starting a Listing

/CTRL/ /S/ suspends the display of a text listing. To resume the listing after scanning the screen for the information you need, press any key.

LOCKING THE SHIFT KEY

The /COMMAND/ key is used with the bottom rightmost key (/?) on the USA ASCII keyboard) to provide a "shift lock" for the letters A-Z. After you press /COMMAND/ with /?, all letters subsequently keyed-in appear as upper-case letters. Furthermore, when the /SHIFT/ key is used, subsequently keyed characters appear in lower-case. The shift lock stays effective until you press /COMMAND/ /?/ again. The alphabetic keys can also be locked using the /CAPS LOCK/ key.

v1 fk

v1 fk	The system returns an empty list (full name must be used for files)
-------	---

DEFAULT DRIVE

The default drive is the last drive accessed, and the command line prompt always identifies the default drive. This is the drive on which the system will always search, unless you give it different instructions.

COMMAND PROMPT

For example, following bootstrap the display

0>

identifies drive 0: as the current or default drive.

When a command is entered, and unless the drive is specified with the command, the system will always search first on the diskette on drive 0:. If the command is not found, it then searches the diskette on drive 1:.

CHANGING DEFAULT

The default drive can be changed by preceding the command or file name with the number for the other drive. For example, if 0 were the default drive, but you wished to check for MYFILE on the scratch disk (on drive 1:), you would enter

1:vq myfile

The system will load VQUICK from drive 0: and then search the disk on drive 1: for myfile.cmd. However, after the command is executed, and whether or not MYFILE is found, drive 1: will remain the default drive. This will be indicated by the prompt

1>

The default drive can also be changed simply by accessing a command on the other drive. For example, if you switch the system diskette to drive 1: and then enter

vq

the system will search drive 0: for the command, then search drive 1: and load and execute the command. Following the listing of the directory, the prompt becomes

1>

Drive 1: is now the default drive.

- Keep dust out of the diskette drives by keeping the drive covers closed when not in use.

LABELING DISKETTES

Every carton of diskettes contains a supply of self-adhesive labels for identifying diskettes. It is good practice to write all relevant details on the label before attaching it to the diskette. But if you do find it necessary to write on the label after sticking it to the diskette, you should avoid using sharp pencils or ball-point pens as this may damage the surface of the diskette. In this case a felt-tipped pen is recommended.

WRITE-PROTECTION

A sheet of aluminized write-protect labels is provided with every carton of diskettes. To apply write-protection simply fix an aluminized label over the write-protect notch as indicated in figure 2-2.

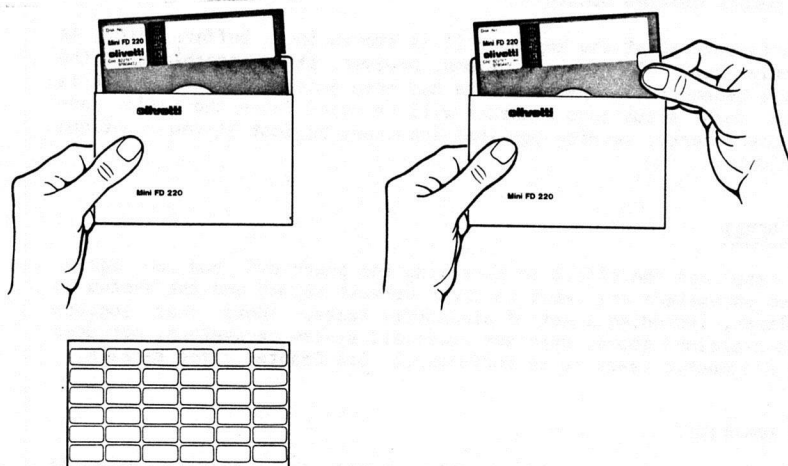


Fig. 2-2 Diskette Write-Protection

To remove write-protection simply peel off the aluminized label.

CHECKING DISK

You can use the VQUICK command to make a quick comparison of your new system diskette with the original.

An even better check is to remove the system disk from drive 0: and press /CTRL/ /ESC/ (to boot the system with your new copy).

COPYING FILES

The FCOPY command is used for copying selected files. When copying a single file from one disk to another, if the destination name is not specified, the file is given the same name. The command line

```
fc 0:fkil.cmd 1:
```

copies the file fkill.cmd from the disk on drive 0: to the disk on drive 1 and names it 'fkill.cmd'. But

```
fc 0:prun.cmd 1:myfile
```

copies the file and renames it.

EXAMPLES

If you enter ...	Then ...
fc 0:*.cmd 1:	The system copies all files ending in .cmd
fc 0:*.abs mydisk:	The system copies all files ending in .abs to 'mydisk' (if it exists)

FCOPY

The FCOPY command copies one file at a time, but the asterisk (*) can be used as a wild card to represent any number of characters in a file name. This makes it possible to copy an entire disk with one entry. This may be faster, since VCOPY copies all blocks, including blank ones, while FCOPY copies only the files.

Note: If your scratch diskette is a copy of the system disk, first use the VNEW command to clear the directory. Otherwise the system will pause each time to ask if you wish to overwrite the existing file.

POWER-ON TESTS

A program stored in ROM then tests the processor, memory, various chips, and the two drives.

BOOTSTRAP

If all diagnostics pass, a message identifying the Bootstrap Loader is displayed.

CONFIGURATION

If the boot is successful, the System Configuration data are displayed.

PCOS BANNER

Lastly, the PCOS Banner is displayed, showing the version of PCOS that has been booted.

PROMPT

The system then goes to the command line prompt:

```
0>
```

The prompt indicates that the system is ready to receive a command.

SYSTEM REBOOT

There are several cases in which it may be necessary or desirable to reboot (restart or reset) the system. Some of these are:

- . to terminate a working session.
- . to reconfigure the system (change operating systems).
- . to clear a program hangup or bug.

There are two ways to reset the system: physical reset and logical reset.

ERROR MESSAGES

ENTRY ERRORS

If an error is made in entry, an error message is returned. For example, if you enter

```
df
```

a search is made for the 'df' file, and then, when it is not found, the system returns

```
ERROR 92
```

EPRINT COMMAND

The EPRINT command is a user's aid. It converts the error number into a message. To use it, enter the command, followed by a space and the error number:

```
ep 92
```

which returns

```
ERROR 92 --- command not found
```

Note that the **eprint.sav** command becomes resident once loaded. Reenter the command; it will be executed immediately from memory.

OPERATING ERRORS

An error in operation will also return an error message. For example, your system diskette has (or should have) a write-protect tab. Put the system diskette in drive 0: and a scratch disk in drive 1:.. Now enter:

```
vc 1: 0:
```

The system, after displaying the warning, will read block 0 from drive 1:, but when it attempts to write to drive 0:, the following error message is displayed:

```
ERROR 57 -- disk i/o error
```

alerting you to the fact that the command you specified cannot be executed because something about the disk is not as it should be (in this case, because the disk is write-protected).

PLOADING

One reason PCOS has so few resident commands is its unique ability to turn any transient command into a resident command. This is done by means of the PLOAD command. Examples of its use will be given later.

ENTERING COMMANDS

To enter a command, type the command name at the keyboard (to save time, only the first two letters need be entered). Either upper- or lower-case letters are allowed. The DCONFIGURE command is used below as an example.

DCONFIGURE

The DCONFIGURE command displays data on the system configuration. Enter

dc

and note that the red light for drive 0: goes on as the system searches for and loads the **dconfig.cmd** file. It then checks the second drive.

When the DCONFIGURE utility is executed, the display will show the system configuration.

REEXECUTION

Note that the DCONFIGURE command does not stay in memory after execution. Try reentering the command:

dc

Note that the system again searches for the file and loads it before executing it.

DIRECTORY LISTING (QUICK LIST)

The VQUICK command provides a fast way to list the files on a disk. It displays a short list (names only) and gives no data on the files.

LISTING A VOLUME

To list the files on your system diskette, enter

vq 0:

The system will read the disk and respond as in the example below:

Volume Quick List Rev. 1.00

3. SOFTWARE COMPONENTS

DIRECTORY LISTING (LONG LIST)

The VLIST command provides a "long" list of the files on a specified disk, including data on file sizes and sectors.

SYSTEM DIRECTORY

To check the files on your system diskette, enter

```
vl 0:
```

The system will read the disk and respond as in the example below.


```
Volume List Rev. 1.00
VOLUME: 0
          BYTES  USED  SECTORS  EXTENTS  WRITE PROT/
          .      .      .         .         PASSWORD
PCOS.SAV 37666  148   149       1
basic.abs 38269  150   151       1
basic.cmd 1341    6     7         1
.         .      .      .         .
.         .      .      .         .
fkill.cmd 2611   11    12        1
SUBTOTALS      324   360       12
12 files      <HIT ANY KEY TO CONTINUE>
```

Press any key to continue the listing.

The information includes the drive number, a volume name, if any, and data on the bytes and sectors used by each file. Note that password protection ('PW') is noted for files, but not for volumes. Write protection for files is indicated by 'WP'.

EXAMPLES

If you enter ...	Then ...
vl	The system lists all files on the current (last accessed) volume
vl 0:*.cmd	The system lists all files on drive 0: having the suffix '.cmd'
vl fkill.cmd	The system lists the data on the file FKill



USER-CONFIGURABLE OPERATING SYSTEM	3-6
<u>BASIC INTERPRETER</u>	3-7
<u>VIDEO FILE EDITOR</u>	3-7
<u>ASSEMBLER</u>	3-8
<u>PASCAL</u>	3-9
<u>LINKER</u>	3-9
<u>PROGRAM DEBUGGER</u>	3-9
<u>STANDARD INTERFACE HANDLERS</u>	3-9
<u>PCOS COMMAND LIBRARY</u>	3-11
CHANGING ENVIRONMENT COMMANDS	3-11
PCOS CONFIGURING COMMANDS	3-11
SET SYSTEM GLOBAL COMMANDS	3-12
KEYBOARD-RELATED COMMANDS	3-12
VOLUME HANDLING COMMANDS	3-13
FILE HANDLING COMMANDS	3-14
STANDARD INTERFACE HANDLING COMMANDS	3-15
PCOS GRAPHIC FACILITY COMMANDS	3-15

ERRORS

If the command is not on the current disk or the disk specified, an error message is returned. A complete list of PCOS and BASIC error messages may be found in an appendix to this manual.

NAMING DISK

To name the disk, use the VNEW command. Enter a space after the drive number, followed by a name, such as "mydisk":

```
vn 1: mydisk
```

Then, after the disk is initialized, check it with

```
vq mydisk:
```

COPYING DISKS

A diskette may be copied using either the VCOPY (volume copy) command or the FCOPY (file copy) command. The latter will copy only those files the user specifies, but by using wild card characters the user may copy an entire diskette with one command line. The asterisk (*) is one such character.

VCOPY

Put the system diskette in drive 0:, and the destination diskette in drive 1:. Now enter

```
vc 0: 1:
```

The system will display the banner and then respond

```
Warning - vcopy deletes all files. Copy disk? (y/n)
```

Enter 'y', and the system responds

```
Read block N to X
```

followed by

```
Write block N to X
```

where N and X are numbers ranging from 0 to 1103 (copying is usually done in groups of approximately 210 blocks at a time). When the copying is completed, the response is

```
VCopy complete
```

INTRODUCTION

The PCOS software comprises a number of independent but closely related functional components. These can be considered to be at three levels: user, language, and operating system (see Figure 3-1).

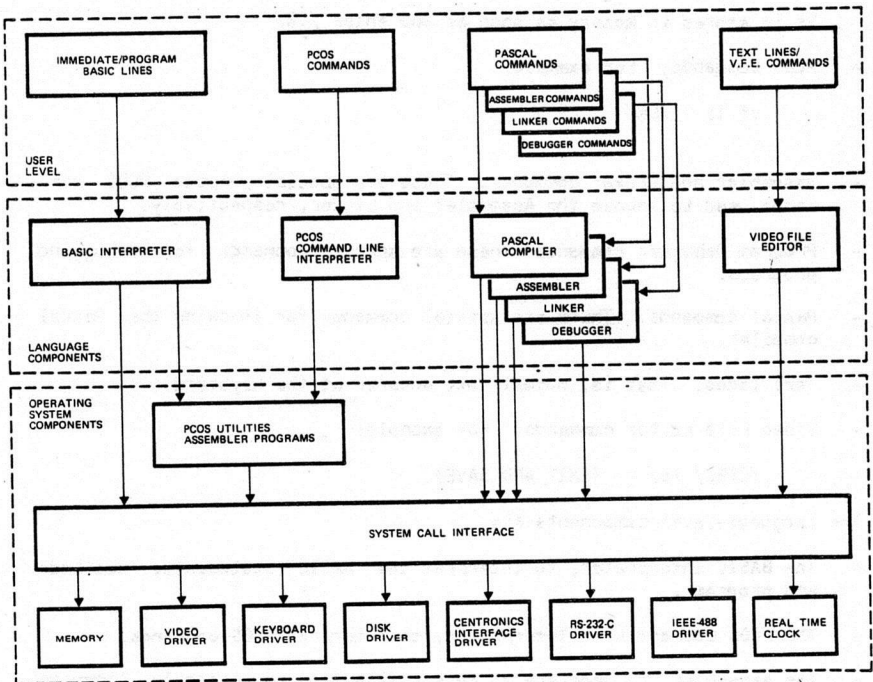


Fig. 3-1 PCOS Software Components

To copy files with FCOPY, put the system disk in drive 0: and the destination disk in drive 1:. Enter

```
fc 0:* 1:
```

The system then displays the name of each file while it is being copied, as in the following example:

```
COPY FILE 0:PCOS.SAV TO 1:PCOS.SAV
```

```
COPY FILE 0:basic.abs TO 1:basic.abs
```

Again, to check your copy against the original, use VQUICK or VLIST, or try booting with it.

OVERWRITE

Note that the system protects you from an accidental overwrite. If any of these files already exist on your scratch disk, the system pauses and displays

```
File already exists. Do you wish to overwrite? (y/n)
```

LOADING COMMANDS

To speed up the process, you can first PLOAD the command. Enter the following:

```
pl fc vq
```

This will PLOAD both the FCOPY and the VQUICK commands (store them in memory). Now try

```
vq 1:
```

and observe how it executes immediately (without a search or loading operation).

PUNLOAD

To remove a PLOADED command from memory (for example, the FCOPY which you have just PLOADED), use the PUNLOAD command:

```
pu fc
```

- The Video File Editor, to create and modify text files in response to text and Video File Editor commands entered at the keyboard.

The operating-system-level components are:

- PCOS utilities (such as BASIC.COM, BVOLUME.SAV, ...WFONT.COM) and Assembler or Pascal programs. These are executable routines to which are passed user-specified parameters. These in turn generate system calls.
- The system calls. These are system routines which allow access to the drivers of the hardware components, and perform operations such as moving strings of bits into memory, activating pixels on the video, reading characters from the keyboard, opening or closing a file, writing strings of characters to a file, etc.

PCOS

Your computer has its Professional Computer Operating System (PCOS) to manage:

- Interaction with the CPU, memory, keyboard, diskette (and/or hard disk) drives and the VDU
- Interaction with any connected peripherals through the four possible interfaces
 - . Centronics-like parallel interface suitable for a range of printers
 - . EIA RS-232-C serial interface suitable for access to peripherals or computers
 - . EIA RS-232-C serial interface expansion (as an option)
 - . IEEE-488 parallel interface (as an option) suitable for access to other talkers and listeners such as counters, heat sensors, signal generators and measuring instruments
- Handling of the real-time clock for all timing functions including date and time

The Command Line Interpreter enables you to communicate with PCOS using a library of over 50 commands.

1944

1945

1946

1947

1948

1949

1950

1951

1952

1953

1954

1955

1956

1957

1958

1959

1960

1961

PROTECTION MECHANISMS

PCOS offers the following protection mechanisms:

- Volume password protection using the VPASS command such that the protected volume cannot be accessed without knowledge of the password.
- File password protection using the FPASS command such that the corresponding file cannot be accessed without knowing the password.
- File write-protection using the FWPROT command to inhibit writing to the specified file(s).

For further details, and for information about other protection mechanisms, see Chapter 8.

LINE EDITOR FUNCTIONS

PCOS offers line editor functions to:

- Backspace (by pressing /CTRL/ /H/ simultaneously).
- Cancel the current line (by pressing /CTRL/ /C/ simultaneously).
- Hide what you enter (by pressing /CTRL/ /G/ simultaneously).

REAL-TIME CLOCK

The CPU includes an oscillator that generates a clock pulse every 50 ms used to strobe the real-time clock. The real-time clock provides the user with the local time in the ISO 24-hour format of hours:minutes:seconds (for example, 23:59:59 for one second to midnight) and the date in the format of month/day/year (for example, 12/01/82 for December 1, 1982). Note that the date format is month/day/year only for the USA keyboard versions; for all other national keyboards the date format is day/month/year. The internal calendar keeps track of days, months and years provided you set the time and date at switch-on using the SSYS command. The real-time clock stops at switch-off or physical reset, but not on logical reset.

ROUTING INPUT/OUTPUT

Your computer normally expects to receive input from the keyboard and sends output to the screen. However, PCOS enables both input and output to be redirected to other devices connected to your computer by specifying "device rerouting parameters". This can be done in two ways:

- In a command line; where they will only be effective for the command in question.
- By themselves; thus remaining in effect for all subsequent commands until they are changed, or until the system is reinitialized.

ABOUT THIS CHAPTER

This chapter describes the PCOS-related components of the M24.

CONTENTS

<u>INTRODUCTION</u>	3-1
<u>PCOS</u>	3-3
MEMORY OPTIMIZATION	3-4
RESIDENT AND TRANSIENT COMMANDS	3-4
PROGRAMMABLE KEYS	3-4
PROTECTION MECHANISMS	3-5
LINE EDITOR FUNCTIONS	3-5
REAL-TIME CLOCK	3-5
ROUTING INPUT/OUTPUT	3-5
USER-DEFINED FONTS	3-6
CONTROL CHARACTER DISPLAY	3-6
INITIALIZATION FILES	3-6

BASIC INTERPRETER

The BASIC Interpreter allows you to create, debug, and execute BASIC Language programs. The comprehensive instruction set includes sophisticated graphics facilities and special features for logical control of the IEEE-488 interface.

The instruction set is composed of commands and statements.

Statements are preceded by line numbers and grouped to form BASIC programs, which can then be executed. The user can halt the execution of a program, issue BASIC commands, and return to program execution without destroying the program variables. The BASIC statements include the following features:

- Program segmentation through CHAINING and COMMON areas.
- Ability to CALL and EXECUTE Assembly Language routines and PCOS commands.
- Simple but powerful control statements (FOR/NEXT, GOTO, IF/GOTO, IF/THEN, ON/GOTO, WHILE/WEND, IF/THEN/ELSE, IF/GOTO/ELSE, ON ERROR/RESUME NEXT, GOSUB, ON/GOSUB).
- Effective character string handling.
- Powerful print/display formatting statements.
- Predictable error handling (ON ERROR etc.).
- IEEE-488 control statements.
- Sophisticated graphics statements.

For details refer to the "BASIC-8000 User Guide".

VIDEO FILE EDITOR

The Video File Editor enables you to create and edit files of text, where a text file can be a file of normal text or a program written in any programming language.

The VDU displays a 21 line window of text which can be moved up or down within the file. Editing functions are entered from the keyboard and include :

- Line and general editing functions.
- Window moving functions. These enable you to move the window up or down the file.

ALTERNATIVE OPERATING SYSTEM 3-16
RELATED COMMANDS

USER AIDS

3-16

PASCAL

The (optional) Pascal compiler processes a source program file written in Pascal - a high-level programming language suitable for structured programming - and produces the corresponding object code. This Pascal is an extended version of the language in that it can call run-time routines including graphics features.

For details refer to the "Pascal under PCOS User Guide".

LINKER

The (optional) Link utility creates an executable file from one or more object files.

For details refer to either the "Z8000 Assembler User Guide" or the "Pascal under PCOS User Guide".

PROGRAM DEBUGGER

The (optional) Program Debug utility enables you to enter a range of commands for debugging and testing programs.

For details refer to the "Z8000 Assembler User Guide".

STANDARD INTERFACE HANDLERS

PCOS contains two communications packages to manage input/output with peripherals and/or computers via the built-in RS-232-C interface, the (optional) single RS-232-C interface expansion, and the (optional) IEEE-488 parallel interface. User interaction is via a group of PCOS commands that enable BASIC programs to communicate via these interfaces. These commands are:

- IEEE which loads the IEEE-488 extension package
- RS232 which loads the RS-232-C interface package
- SCOMM which sets the protocol for an RS-232-C port
- CI which provides a BASIC interface with the RS-232-C driver

The user-level components are:

- Immediate BASIC lines (that is, one or more BASIC statements or commands separated by colons). For example:

```
RUN "1:NEWFILE" /CR/
```

It is executed as soon as you enter /CR/

- BASIC program lines (that is, a line number followed by one or more BASIC statements or commands separated by colons). For example:

```
100 PRINT "SIN of X is"; SIN (X) : IF X>2 THEN 1000 /CR/
```

It is stored in memory as soon as you enter /CR/

- PCOS commands. For example:

```
vf 1: /CR/
```

- Assembler and Linker commands. These are special purpose PCOS commands used to invoke the Assembler and Linker, respectively.
- Program Debugger commands. These are special commands for debugging programs.
- Pascal commands. These are special commands for invoking the Pascal compiler.
- Text lines. That is, normal text entered at the keyboard.
- Video File Editor commands. For example:

```
/CTRL/ /6/ (EXIT AND SAVE)
```

The language-level components are:

- The BASIC Interpreter, to interpret the BASIC statements, commands and programs.
- The PCOS Command Line Interpreter, to interpret PCOS commands.
- The Assembler, to generate an object file from an Assembly language source file.
- The Pascal compiler, to compile object code from Pascal source files specified by the corresponding Pascal command.
- The Linker, to create an executable file from the object file(s) generated by the Assembler or Pascal compiler and specified by the corresponding linker command. (Executable files are in turn treated exactly like PCOS utilities).

PCOS COMMAND LIBRARY

This section lists the PCOS commands in functional groups. It lists both the full mnemonic and the two-character short form. The latter is the shortest form that PCOS will recognize as a keyword. This is explained in Chapter 4.

For details of a particular command see Chapter 14.

CHANGING ENVIRONMENT COMMANDS

KEYWORD		
SHORT FORM	FULL MNEMONIC	COMMAND FUNCTION
ba	BASIC.CMD	Loads the BASIC Interpreter
ed	EDIT.CMD	Loads the Video File Editor

Table 3-1 Changing Environment Commands

PCOS CONFIGURING COMMANDS

KEYWORD		
SHORT FORM	FULL MNEMONIC	COMMAND FUNCTION
pl	PLOAD (resident)	Loads commands
pr	PRUN.CMD	Reloads an operating system
ps	PSAVE.CMD	Saves PCOS
pu	PUNLOAD (resident)	Unloads commands

Table 3-2 PCOS Configuring Commands

MEMORY OPTIMIZATION

Your computer has a system of memory optimization which is handled dynamically by allocating memory according to need and usage.

Memory optimization is achieved by:

- using transient commands which are executed and then removed
- creating and then purging all temporary PCOS tables
- using the global command SBASIC to set resource level according to requirements of the application package and user program

For details of how to examine your particular memory configuration refer to the DCONFIG command in Chapter 14.

RESIDENT AND TRANSIENT COMMANDS

In an attempt to maximize user memory space, only three commands are always loaded into memory when the system is initialized. They cannot be removed from memory. These commands are:

PLOAD - used to load transient commands into memory

PUNLOAD - used to remove PLOADed commands from memory

LTERM - used to differentiate among the three line-terminator keys
↵ /, /S1/ and /S2/.

The remaining commands are transient and can be executed then removed from memory. However, these commands can also remain in memory by means of the PLOAD command, and can become permanently resident by use of the PLOAD and PSAVE commands (see Chapter 6).

PROGRAMMABLE KEYS

Any key, or key struck in combination with the /CTRL/, /COMMAND/ or /SHIFT/ key, can have a special meaning assigned to it. This may be a BASIC or PCOS command, an expression, a constant, or any group of characters that may be found useful to have at the touch of a single keystroke or key combination. Assignment is made using the PKEY command, and can be made a permanent feature by means of the PSAVE command (see Chapter 6).

VOLUME HANDLING COMMANDS

KEYWORD		
SHORT FORM	FULL MNEMONIC	COMMAND FUNCTION
bv	BVOLUME.CMD	Searches the volume directory, returns free disk space, or returns the name of the current volume (from BASIC only)
hb	HBACKUP.CMD	Backs up the hard disk onto a series of diskettes
hr	HRESTORE.CMD	Restores a backup to the hard disk
va	VALPHA.CMD	Alphabetizes a directory
vc	VCOPY.CMD	Copies a volume (drive to drive)
vd	VDEPASS.CMD	Removes a password from a volume
vl	VLIST.CMD	Lists a volume directory (full form)
vm	VMOVE.SAV	Copies a volume (using one drive)
vn	VNEW.CMD	Initializes a volume
vp	VPASS.CMD	Assigns a password to a volume
vq	VQUICK.CMD	Lists a volume directory (filename only)
vr	VRENAME.CMD	Renames a volume
vv	VVERIFY.CMD	Checks the hard disk for faulty blocks

Table 3-5 Volume Handling Commands

For more details about device rerouting see Chapter 7.

USER-DEFINED FONTS

Characters are displayed on the VDU with a shape defined by the system font tables. But PCOS enables you to define and implement your own character font sets using the RFONT and WFONT commands. See Chapter 11 for details.

CONTROL CHARACTER DISPLAY

ASCII control characters are normally unprintable. However, PCOS contains a facility whereby each of these control characters is assigned a unique font, which will be displayed on the occurrence of the corresponding control character if control character display is specified. This can be done in one of two ways:

- In a command line; where control characters will only be displayed for the command in question.
- By itself; thus remaining in effect until either actively cancelled, or the system is reinitialized.

For more details about control character display refer to Chapter 11.

INITIALIZATION FILES

PCOS enables you to create an initialization file that will be executed automatically every time the system is initialized. Any sequence of commands, programs, BASIC statements, etc., can be executed in this way to initialize the system to suit your own needs. See Chapter 5 for details.

USER-CONFIGURABLE OPERATING SYSTEM

Resident and transient commands, programmable keys, routing input/output, user-defined fonts and control character display are all features of PCOS that enable you to define your working environment. In addition, a subset of PCOS commands enables you to change the global parameters of your system, such as the amount of memory available to BASIC, the system date and time, etc. Moreover, the current state of the system, as defined by these features, can be saved at any time using the PSAVE command. By subsequently reinitializing the system from the PSAVEd file, you will then restore the system to the state it was in at the time it was saved.

For details on how to configure PCOS, refer to Chapter 6.

STANDARD INTERFACE HANDLING COMMANDS

KEYWORD		
SHORT FORM	FULL MNEMONIC	COMMAND FUNCTION
ci	CI.SAV	Provides the BASIC interface to the RS-232-C driver
ie	IEEE488.SAV	Loads the IEEE-488 package
rs	RS232.SAV	Loads the RS-232-C package
tl	TLANG.SAV	Allows the translation of SLANG character codes between 9-bit devices and systems

Table 3-7 Standard Interface Handling Commands

PCOS GRAPHIC FACILITY COMMANDS

KEYWORD		
SHORT FORM	FULL MNEMONIC	COMMAND FUNCTION
la	LABEL.CMD	Displays a label string
ls	LSCREEN.CMD	Prints the displayed text
pl	PLOT.CMD	Controls the plotter through the Operating System
rf	RFONT.CMD	Creates an ASCII font matrix file from the currently active font
sp	SPRINT.CMD	Prints the text and graphic contents of a specified window
wf	WFONT.CMD	Makes a font matrix file active

Table 3-8 PCOS Graphic Facility Commands

ALTERNATIVE OPERATING SYSTEM RELATED COMMANDS

KEYWORD		
SHORT FORM	FULL MNEMONIC	COMMAND FUNCTION
hd	HDISK.CMD	Partitions the hard disk

Table 3-9 Alternative Operating System Related Commands

USER AIDS

KEYWORD		
SHORT FORM	FULL MNEMONIC	COMMAND FUNCTION
dc	DCONFIG.CMD	Displays the hardware and/or memory configuration
ep	EPRINT.SAV	Displays error messages

Table 3-10 User Aids

ABOUT THIS CHAPTER

This chapter describes the format of a PCOS command, and the syntax used by this manual to describe the PCOS commands. For detailed description of the command mentioned in this chapter refer to Chapter 14.

CONTENTS

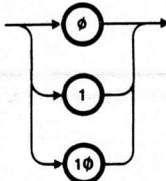
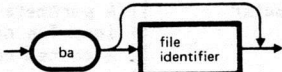
<u>NOTATION CONVENTION</u>	4-1
<u>COMMAND SYNTAX</u>	4-2
<u>COMMAND NAMES AND KEYWORDS</u>	4-3
<u>PARAMETERS</u>	4-3
DEFAULT VALUES	4-5
<u>RESIDENT AND TRANSIENT COMMANDS</u>	4-5
<u>COMMAND SEARCH PROCEDURE</u>	4-6
<u>FILE AND VOLUME IDENTIFIERS</u>	4-6
<u>WILD CARDS</u>	4-10
<u>NO-INTERACTION FLAG</u>	4-10
OTHER FLAGS	4-11

NOTATION CONVENTION

PCOS commands are represented using syntax diagrams, in which:

- the keyword, written in its short form (using the first two characters) in lower case letters, is enclosed in an oval
- parameters are enclosed in rectangles
- punctuation, single characters, and drive numbers are enclosed in circles
- flow lines connecting the above elements indicate any options which can be taken. Furthermore, a loop indicates any elements that may be repeated.

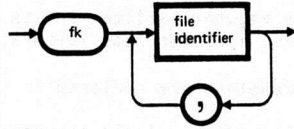
The following table shows some examples of the use of syntax diagrams. Note in particular the use of the flow lines.

No.	RULE	EXAMPLE
1	<p>A fork indicates a choice. One of the two paths must be followed in the direction of the arrow.</p> <p>For example, when a drive number has to be specified you can enter one of</p> <p>0 - drive 0</p> <p>1 - drive 1</p> <p>or</p> <p>10 - if you have a hard disk</p>	
2	<p>An empty branch or a bypass indicates an optional element.</p> <p>In the example the file identifier in the PCOS command BASIC is optional</p>	

3

A loop indicates that the parameter may be repeated at least once.

For example, in the FKILL command the file identifier may be repeated several times using a comma as a separator



COMMAND SYNTAX

The general format is defined as follows:

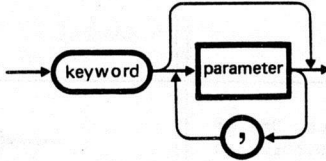


Fig. 4-1 General Format of a PCOS Command

Where

SYNTAX ELEMENT	MEANING
keyword	A mnemonic that specifies the command to be executed
parameter	A parameter to the command defining the command action. The number of parameters depends on the command executed but must be in the range 0 to 20

Remarks

A command may be entered over one or more lines (up to 255 characters) and terminates with the first occurrence of a /CR/.

A space may alternatively be used as the parameter separator except where nil parameters (see the section on "Parameters") are specified.

COMMAND NAMES AND KEYWORDS

A command name comprises a mnemonic string of up to 14 alphanumeric characters (the first of which must be a letter), optionally including a period and an extension. PCOS commands can have one of the following extensions: CMD, SAV, or BAS. For example, BASIC.CMD. These extensions serve a purpose that will become evident in the following sections.

The keyword can be entered in either upper or lower case and must comprise at least the first two characters of the command name. For example

```
ba /CR/
```

```
basic /CR/
```

```
basic.cmd /CR/
```

will execute the command named BASIC.CMD.

Standard PCOS commands, as well as user-written assembly language and Pascal programs, are all executed in this way. For details on executing assembly language programs refer to the "Z8000 Assembler User Guide." For details of executing Pascal programs refer to the "Pascal under PCOS User Guide."

PARAMETERS

Parameters are user-selected strings of alphabetic characters and of integers which can be optional. They are recognized by their position in the command line. The different types of parameter are described below:

PARAMETER TYPE	MEANING												
Integer parameter	<p>A decimal integer, or a hexadecimal integer (up to four characters) preceded by an "&" (ampersand).</p> <p>For example 10 &A</p>												
String parameter	<p>A string of alphabetic characters. Upper and lower case are interpreted differently. Leading and trailing quotation marks (either single or double) are optional except where the string contains any of the following:</p> <table border="0" data-bbox="327 666 889 744"> <tr> <td>+</td> <td>(plus)</td> <td>"</td> <td>(double quotation mark)</td> </tr> <tr> <td>&</td> <td>(ampersand)</td> <td>'</td> <td>(single quotation mark)</td> </tr> <tr> <td>SPACE</td> <td></td> <td>,</td> <td>(comma)</td> </tr> </table> <p>In this case the string must be enclosed in quotation marks. If the string contains a single quotation mark, then it must be enclosed in double quotation marks, and vice versa. The syntax diagrams indicate when it is necessary or useful to include quotation marks. For brevity single quotation marks are always indicated</p>	+	(plus)	"	(double quotation mark)	&	(ampersand)	'	(single quotation mark)	SPACE		,	(comma)
+	(plus)	"	(double quotation mark)										
&	(ampersand)	'	(single quotation mark)										
SPACE		,	(comma)										
Nil parameter	<p>A parameter that does not have a value specified in the command line. Such parameters assume default values. Nil parameters can be designated in one of two ways depending on their position in the command line:</p> <ul style="list-style-type: none"> - Before the last specified parameter. Such nil parameters are designated by a comma, with no preceding information. For example <p style="margin-left: 40px;">la 'title',,,5,2 /CR/</p> <p>has two nil parameters after 'title'. These will assume their default values.</p> - After the last specified parameter. In this case nil parameters are not entered. For example <p style="margin-left: 40px;">la 'title' /CR/</p> <p>has four nil parameters after 'title'. All of these will assume their default values</p> 												

<p>Device rerouting parameter</p>	<p>A parameter that reroutes input/output from/to specified devices or files. These are recognized by a "+" or a "-" sign as the first character.</p> <p>For example +dpvt:</p> <p>For details refer to Chapter 7</p>
<p>Control character display</p>	<p>A special parameter that enables or disables the display of control characters (unprintable ASCII characters 00 to 1F, hexadecimal).</p> <p>For example +cc</p> <p>For details refer to Chapter 11</p>

DEFAULT VALUES

Default values are automatically assumed when a nil parameter is detected.

Parameters which are governed by global commands (SBASIC, SSYS, SFORM, SCOMM, SLANG, and SDEVICE), assume default values in the absence of any command. If a global command has been executed then the values specified by it are assumed by the system from then on until a new global command changes those values or until the system is reinitialized. In the latter case, the default values are again assumed.

RESIDENT AND TRANSIENT COMMANDS

There are only three resident commands: PLOAD, PUNLOAD, and LTERM. These commands can never become transient. All the other commands are transient. Any transient command can be made resident, but the technique for doing this depends on the command file extension; that is, whether the command has a CMD or SAV extension. The former require the PLOAD command to make them resident, while the latter become resident simply by executing the command. Once a command has been made resident it remains so until the end of the current working session (that is, when the system is switched off or a physical or logical reset is performed), whereupon it becomes transient once again. Such commands, however, can be made permanently resident by means of the PSAVE command (see Chapter 6).

COMMAND SEARCH PROCEDURE

When a command is entered, PCOS will first search RAM for the first resident command that matches the characters entered. If found, the command is executed.

If no command is found in RAM then both drives are searched starting with the last drive selected, for

1. A transient command with a CMD extension. If such a command is found, it is loaded into RAM, executed, and subsequently removed from RAM
2. A transient command with a SAV extension. If such a command is found, it is loaded into RAM and executed. However, it is not removed from RAM. This means that the command can be used again even if the diskette it resides on is removed from its drive
3. A transient command with a BAS extension. If found, the system will load the BASIC Interpreter, enter into BASIC execution mode, and run the file with the BAS extension

If a command cannot be found an error message (ERROR 92) is issued.

Note: You may alternatively specify a drive number before the command keyword, thereby limiting the search to the specified drive. For example

```
1:v1 0: /CR/
```

will search only drive 1 for the VLIST command.

FILE AND VOLUME IDENTIFIERS

A disk may contain one or more program and/or data files. A single file, however, may not extend beyond one disk.

A group of files stored on the same diskette or disk forms a "volume." Each file and each volume has an identifier. Each file name must be unique on any one volume. Saving a program file which already exists on a volume causes the original file to be overwritten.

You may assign an identifier to a file either by an OPEN statement (data files), or by a SAVE command (program files), or by an FNEW, FCOPY, FMOVE, FRENAME, or EDIT command.

You may assign an identifier to a volume by a VNEW, VCOPY, VMOVE, or a VRENAME command.

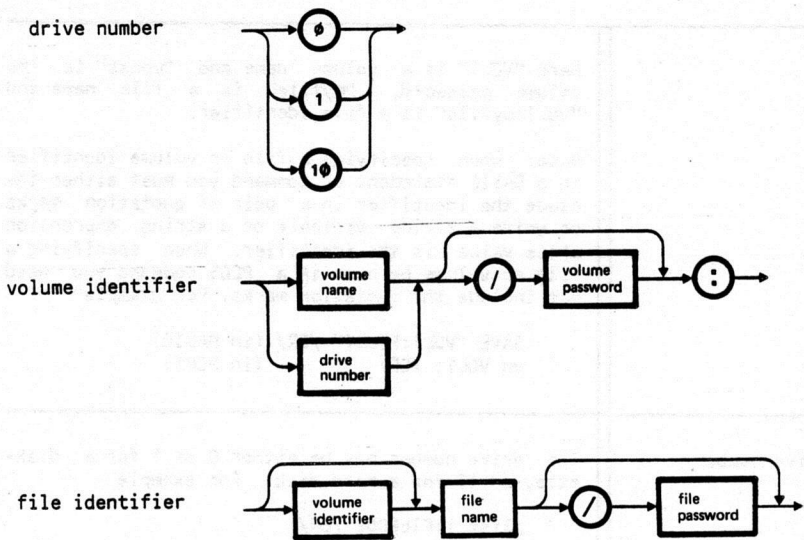


Fig. 4-2 File and Volume Identifier

SYNTAX ELEMENT	MEANING
volume name	<p>The name of a volume. This name must be a string of up to 14 printable ASCII characters (for illegal characters see below). The first character cannot be numeric.</p> <p>To select a specific volume in a PCOS or BASIC command or in an OPEN statement you must specify a volume name or a drive number.</p> <p>The volume name (or the drive number) may be followed by a volume password. At the end of a volume identifier a colon must be entered. For example</p> <pre>fp VOL1/vpass:myfile,newpass /CR/</pre>

	<p>Here "VOL1" is a volume name and "vpass" is the volume password, "myfile" is a file name and "VOL1:myfile" is a file identifier.</p> <p>Note: When specifying a file or volume identifier in a BASIC statement or command you must either include the identifier in a pair of quotation marks or write a string variable or a string expression whose value is the identifier. When specifying a file or volume by name in a PCOS command you need not include the quotation marks. For example</p> <pre>SAVE "VOL1:FILE1" /CR/ (in BASIC) vn VOL1: /CR/ (in PCOS)</pre>
drive number	<p>The drive number may be either 0 or 1 for a diskette, or 10 for a hard disk. For example</p> <pre>flist 1:FILE002 /CR/</pre> <p>Here "1:" indicates that file "FILE002" resides on the disk inserted in drive 1</p>
file name	<p>The name of a file, which must be a string of up to 14 printable ASCII characters (for illegal characters see below), optionally including a file extension. It must include at least one non numeric character or be enclosed in quotation marks.</p> <p>To select a file in a PCOS or BASIC command or in an OPEN statement you must specify the file name. The file name may be preceded by a volume identifier and followed by a (file) password. For example</p> <pre>ba 1:MYPROG/MYPASS /CR/</pre> <p>IF YOU DO NOT SPECIFY ANY VOLUME IDENTIFIER BEFORE THE FILE NAME, THE SEARCH IS LIMITED TO THE LAST SELECTED DRIVE.</p> <p>The file extension is a string of up to 12 printable ASCII characters, preceded by a period (.). (For illegal characters see below.)</p> <p>Notes: 1. filename.extension cannot exceed 14 characters</p>

	<p>2. The extensions BAS, CMD, and SAV have special meanings</p>
<p>file password or volume password</p>	<p>The password to the file or volume. It must be a string of up to 14 printable ASCII characters (for illegal characters see below).</p> <p>Passwords give the user protection at volume or file level. They may be entered after a volume name, a drive number, or a file name and preceded by a slash. For example</p> <p style="padding-left: 40px;">fl 0:myfile/newpass /CR/</p> <p>However, when assigning a password (using VPASS or FPASS) it is preceded by a comma as it does not yet form part of the identifier parameter. For example</p> <p style="padding-left: 40px;">fp 1:FILE111,NEWPASS /CR/</p> <p>In this case the password "NEWPASS" is assigned to the file "FILE111" resident on the diskette in drive 1</p>

Illegal Characters

The following table indicates the characters that may not be included in a volume name, file name, or password.

= (equals)	- (minus sign)	+ (plus sign)
, (comma)	: (colon)	# (hash or pound)
\ (backslash)	/ (slash)	' (single quotation mark)
* (asterisk)	? (question mark)	" (double quotation mark)
/SPACE/		
or any control character		

Table 4-1 Illegal Characters

Note: The asterisk (*) and the question mark (?) may also be used in a file name in certain commands but with a special meaning (see the section "Wild Cards").

WILD CARDS

PCOS supports two "wild card" characters, the asterisk (*), and the question mark (?); which can be used in a file name to specify a group of file names.

An asterisk (*) Represents any string of characters of any length (including no characters)

A question mark (?) Represents any character. That is, it must match one, and only one, character.

Examples

IF you enter...	THEN...
v*.cmd	All file names starting with "v" and with the extension "cmd" are specified
????.*	All files with a four-character file name, with an extension of any length, are specified

Wild cards can be used in file identifiers with the following commands:

FCOPY	FDEPASS	FFREE	FKILL
FLIST	FPASS	FUNPROT	FWPROT
VLIST	VQUICK		

NO-INTERACTION FLAG

The execution of PCOS commands often involves interaction with the user after the command has been entered (that is, after /CR/ is pressed). In some cases the video displays the result of the command. For example

if you enter

```
sb /CR/
```

then the system displays the current values of the SBASIC command parameters. That is, no further interaction takes place. But in other cases

ENTERING A COMMAND

interactive messages ask the user whether the process is to continue in one way or another. For example

if you enter

```
fk 1:V* /CR/
```

then the system will display all the file names starting with "V" one by one, asking the user whether the file is to be deleted or not. In each case the user must enter "y" for yes or "n" for no.

All this interaction and message display can be suppressed by specifying the "no-interaction" flag (%n) immediately after the command keyword. For example

if you enter

```
fk %n,1:V* /CR/
```

then all the file names beginning with "V" are deleted from the diskette inserted in drive 1, and no messages are displayed.

This facility allows PCOS commands, which normally display interactive messages, to be called and executed from a BASIC program without the need for interaction with the user. It can also be very useful to preserve the screen image while executing various commands.

OTHER FLAGS

Certain commands include other flag options which affect only those specific commands. Detailed explanation is provided in the command descriptions in Chapter 14. For example, FCOPY may have %a, %f, and %u flags. In the command line, a flag entry is always preceded by a percent sign, %.

Administrative matters are the first and most important of these. It is the duty of the Secretary to see that the work of the office is carried out in an efficient and economical manner.

It is the duty of the Secretary to see that the work of the office is carried out in an efficient and economical manner.

It is the duty of the Secretary to see that the work of the office is carried out in an efficient and economical manner.

The Secretary is also responsible for the maintenance of the office records and for the preparation of reports and documents. It is the duty of the Secretary to see that the work of the office is carried out in an efficient and economical manner.

The Secretary is also responsible for the maintenance of the office records and for the preparation of reports and documents. It is the duty of the Secretary to see that the work of the office is carried out in an efficient and economical manner.

It is the duty of the Secretary to see that the work of the office is carried out in an efficient and economical manner.

It is the duty of the Secretary to see that the work of the office is carried out in an efficient and economical manner.

The Secretary is also responsible for the maintenance of the office records and for the preparation of reports and documents. It is the duty of the Secretary to see that the work of the office is carried out in an efficient and economical manner.

The Secretary is also responsible for the maintenance of the office records and for the preparation of reports and documents. It is the duty of the Secretary to see that the work of the office is carried out in an efficient and economical manner.

It is the duty of the Secretary to see that the work of the office is carried out in an efficient and economical manner.

The Secretary is also responsible for the maintenance of the office records and for the preparation of reports and documents. It is the duty of the Secretary to see that the work of the office is carried out in an efficient and economical manner.

It is the duty of the Secretary to see that the work of the office is carried out in an efficient and economical manner.

5. INITIALIZATION AND CHANGING ENVIRONMENTS

ABOUT THIS CHAPTER

This chapter describes the operational modes of the M24 and the initialization process.

CONTENTS

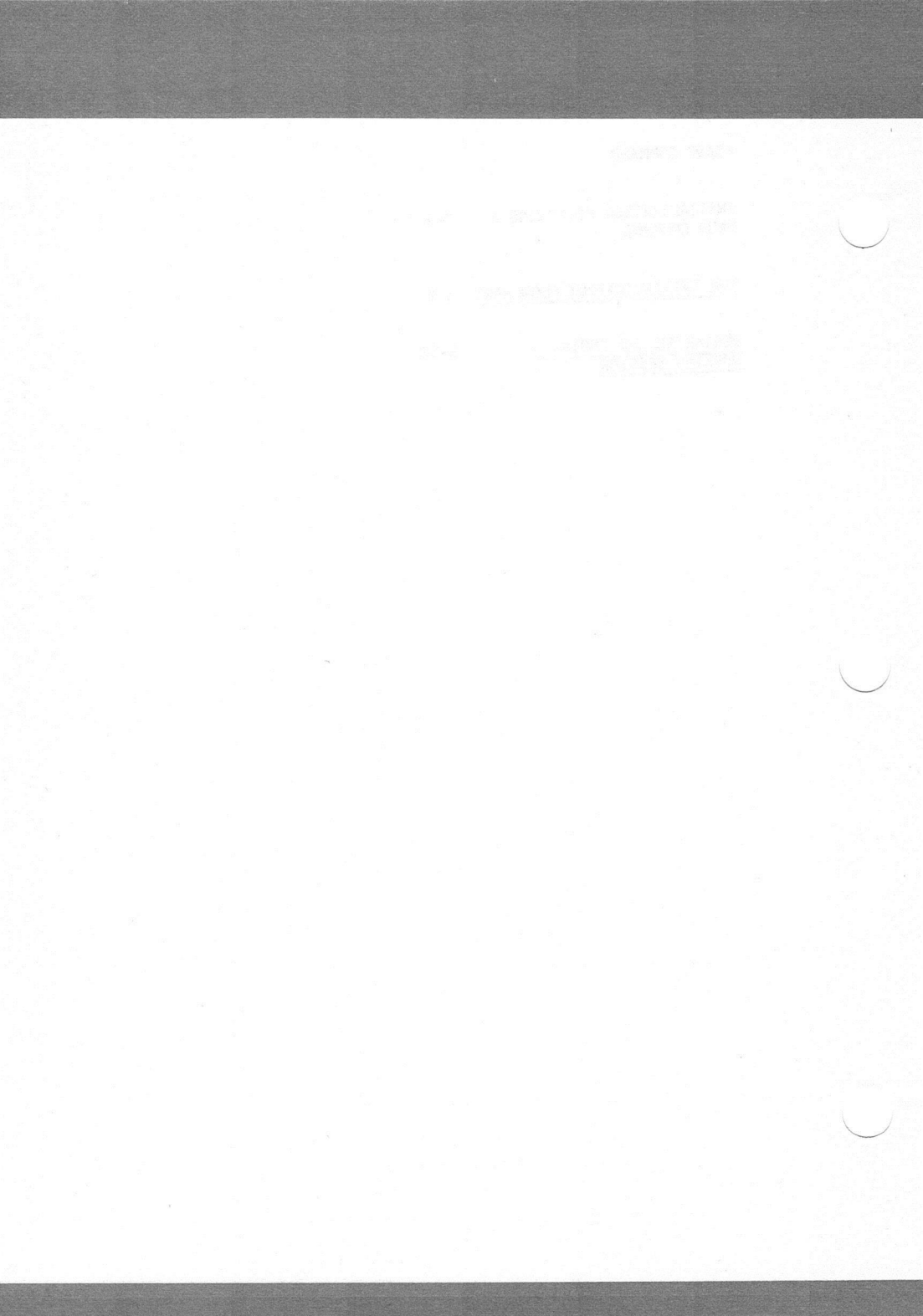
<u>M24 ENVIRONMENTS</u>	5-1
PCOS	5-1
BASIC	5-1
VIDEO FILE EDITOR	5-1
CHANGING ENVIRONMENTS	5-2
<u>MODES OF PCOS</u>	5-5
COMMAND MODE	5-5
EXECUTION MODE	5-5
CHANGING MODES	5-6
<u>STANDARD INITIALIZATION</u>	5-6
INITIALIZATION FILES	5-7
<u>NONSTANDARD INITIALIZATION</u>	5-8
INITIALIZATION FOLLOWING A	5-9

PSAVE COMMAND

INITIALIZATION FOLLOWING A PRUN COMMAND 5-9

THE INITIALIZATION FLOWCHART 5-9

BEGINNING AND ENDING A WORKING SESSION 5-13



M24 ENVIRONMENTS

The M24 can be operated in several distinct environments, such as alternative operating systems, editors and application programs. However, three such environments fall within the scope of this manual:

- PCOS
- BASIC
- Video File Editor

In each environment your computer responds to the keyboard in a different way.

This section outlines the modes of operation in each of the environments and how you pass from one environment to another. The modes of PCOS are described in detail in the next section as are the techniques for passing from one PCOS mode to another. BASIC modes are fully described in the "BASIC-8000 User Guide." The modes of operation within the Video File Editor are described in Chapter 13.

PCOS

In this environment you can:

- enter PCOS commands
- execute PCOS commands.

BASIC

In this environment you can:

- enter and edit BASIC immediate and program lines
- execute BASIC immediate and program lines
- invoke PCOS commands and Assembler subroutines using CALL or EXEC statements.

VIDEO FILE EDITOR

In this environment you can:

- create and edit text files (including programs written in Assembly Language or Pascal)
- create and edit BASIC program files.

CHANGING ENVIRONMENTS

The following are the only possible changes of environment:

- From PCOS to BASIC.
- From BASIC to PCOS.
- From PCOS to the Video File Editor.
- From the Video File Editor to PCOS.

That is, you cannot enter the Video File Editor from BASIC, nor can you enter BASIC from the Video File Editor.

The following figure illustrates the possible changes of environment.

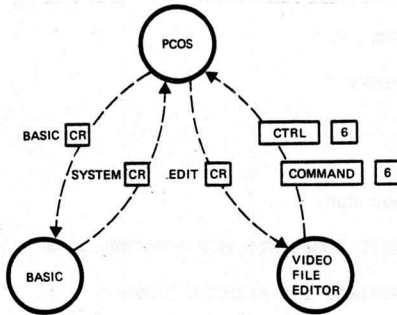


Fig. 5-1 M24 Environments

INITIALIZATION AND CHANGING ENVIRONMENTS

From PCOS to BASIC and Vice Versa

IF the computer is in...	AND you enter...	THEN...
PCOS	ba /CR/	The computer enters the BASIC environment
PCOS	<p>A file name with the extension BAS</p> <p>OR</p> <p>The BASIC command with a file identifier as a parameter</p>	The computer enters the BASIC environment and executes the specified program
PCOS or BASIC	/CTRL/ /ESC/ simultaneously (logical reset)	The system is reinitialized but without running diagnostics
PCOS or BASIC	Physical reset	The system is reinitialized and diagnostic tests are performed
BASIC	<p>SYSTEM /CR/</p> <p>OR</p> <p>The SYSTEM command is encountered during execution of a BASIC program</p>	The system returns to the PCOS environment

From PCOS to Video File Editor and Vice Versa

IF computer is in...	AND you enter...	THEN...
PCOS	ed fileid /CR/	The computer enters the Video File Editor environment for working on the specified file. If the file already exists on the volume specified by the file identifier (or on the last accessed drive if no volume is specified in the file identifier), then it is loaded into user memory. If not, the file is created
Video File Editor	/CTRL/ /6/ simultaneously. Note: /6/ must be entered from the top row alphanumeric section of the keyboard	The computer saves the current text file and exits the Video File Editor (see note below.)
Video File Editor	/COMMAND/ /6/ simultaneously. Note: /6/ must be entered from the top row alphanumeric section of the keyboard	If text has been changed the editor will prompt you to confirm the abort. To do this you must press /COMMAND/ /6/ again, upon which the computer exits the Video File Editor without saving the current text file (see note below).

Note: It is possible to work on more than one file without exiting the Video File Editor. If you want to exit the editor using /CTRL/ /6/ or /COMMAND/ /6/, you must first return to the file that was specified when the editor was invoked (by using /CTRL/ /6/ or /COMMAND/ /6/).

MODES OF PCOS

PCOS has two modes:

- Command mode (for entering PCOS commands).
- Execution mode (for executing PCOS commands).

COMMAND MODE

When the computer enters this mode it displays the PCOS prompt (n> - where n refers to the drive number) and the cursor (█).

In this mode you can enter PCOS commands. For example

```
fc 0:MYFILE,1:YOURFILE /CR/
```

While entering a PCOS command there are two line edit functions available. These can only be performed before /CR/ has been entered. If you

- press /CTRL/ /H/ (simultaneously) then the last character is deleted
- press /CTRL/ /C/ (simultaneously) then the entire line is deleted.

Furthermore, subsequent characters can be rendered invisible by pressing /CTRL/ /G/. To return to visual mode you must press /CTRL/ /G/ again or /CR/.

EXECUTION MODE

In this mode PCOS executes the command that you have just entered. The mode is entered when /CR/ is pressed.

The user can interrupt some activities and return to command mode by pressing /CTRL/ /C/ simultaneously. Other activities, however, once started cannot be interrupted; for example, copying volumes. Having interrupted a command in this way it is not possible to resume execution of that command. You must reenter the command.

On completion of command execution PCOS returns to command mode.

CHANGING MODES

The following table summarizes how to pass from command mode to execution mode and vice versa.

IF computer is in...	AND...	THEN...
PCOS command mode	You enter a PCOS command (terminating with /CR/)	PCOS passes into execution mode
PCOS execution mode	You enter /CTRL/ /C/	Where possible the command being executed is aborted and PCOS returns to command mode
PCOS execution mode	The command execution is completed	PCOS returns to command mode

STANDARD INITIALIZATION

Initialization takes place when you switch the machine on, or perform a physical reset. First of all diagnostics are run. These diagnostic tests check that the hardware is functioning correctly. Any faults are indicated by messages on the screen (see Appendix D). The diagnostics take a few seconds to run. A number is displayed in the upper left corner of the screen, indicating which test is in progress. If, after the '8' is displayed, and the bell has sounded twice, you press any of the keys /L/, /D/, /F/, /S/, or /B/, a nonstandard initialization takes place (see next section). Otherwise, the initialization is standard and proceeds as described below.

A logical reset (/CTRL/ /ESC/) also causes initialization, but without diagnostics.

After diagnostics the bootstrap loader searches the drives for a "bootable" file. This file must be the first file on the volume and must also be of a certain format.

The bootstrap loader first checks the hard disk (drive 10 - if fitted) for a bootable file and loads it into RAM if found. If your system does

not have a hard disk, or if the search on the hard disk was not successful, an error message is issued and the bootstrap examines the diskette in drive 0 and again tries to load a bootable file into RAM. If still unsuccessful, the search is repeated on the second diskette drive (drive 1 - if fitted). The loaded file is then executed.

If the above procedure has been unsuccessful in finding a bootable file, then the following bootstrap error message is displayed on the screen

Insert system disk and type any key

You must then insert the system diskette into an available drive, hit any key and the first file will be loaded into memory and executed. If it is a standard system diskette then the loaded file is named PCOS.SAV.

Note: Initialization resets global parameters to their default values (these are the parameters managed by the Set System global Commands). An exception to this, however, is that the system date and time parameters are not reset on logical reset.

INITIALIZATION FILES

Once the keyboard is initialized, PCOS starts a search on both drives for an initialization file. This file can be named INIT.CMD, INIT.SAV, or INIT.BAS (in upper or lower case).

The file names are searched for in the following order:

- INIT.CMD
- INIT.SAV
- INIT.BAS

If no initialization file is found then the system enters PCOS command mode. The effect of an initialization file, however, is as follows:

- INIT.CMD
This file can contain any program in machine language (for example a PCOS command) which is to be executed at initialization. It can be created by saving the program in question in a file called INIT.CMD. If it exists in any of the two drives, the file is automatically loaded into RAM by PCOS at initialization, executed and then purged from RAM. Subsequently, the system remains in the PCOS environment and waits for you to enter a PCOS command at the keyboard.
- INIT.SAV
This file has the same characteristics as INIT.CMD. It is only loaded into RAM if INIT.CMD does not exist. Moreover, when it is loaded, the program is executed but not purged. It remains in RAM for the rest of the current session. After INIT.SAV is executed, the system remains in the PCOS environment and waits for you to enter a PCOS command at the keyboard.

- INIT.BAS

This file can contain any BASIC program to be run at initialization. To create an INIT.BAS file you can save the program in question specifying the file name to be INIT.BAS. (This can only be done from the BASIC environment.) If INIT.BAS exists, and neither INIT.COM nor INIT.SAV exist, then the BASIC Interpreter and INIT.BAS are loaded into RAM by PCOS at initialization. On encountering the extension BAS, the computer enters the BASIC environment and the program file INIT.BAS is run. What happens after INIT.BAS is run depends on the program itself. Note that INIT.BAS must reside on the same volume as BASIC.COM and BASIC.ABS.

NONSTANDARD INITIALIZATION

The initialization process can be modified by pressing one of the keys /L/, /D/, /F/, /B/, or /S/ after an '8' has appeared in the upper left corner of the screen, and there have been two 'beeps'. Each of these options is described in the following table:

IF, during power-up diagnostics you press	THEN...
/L/	The system loops on diagnostics indefinitely. Its effect can be canceled only by a hard reset
/D/	The system loops indefinitely on disk drive diagnostics, repeatedly reading track zero, side zero of the first ready drive
/F/	The bootstrap loader examines the diskette drive before the hard disk drive when looking for a bootable file

/B/	The system passes directly into BASIC command mode without attempting to execute any initialization file
/S/	The system passes directly into PCOS command mode without executing any initialization file

INITIALIZATION FOLLOWING A PSAVE COMMAND

The PSAVE command saves the current operating system then reboots the system as described in the section "Standard Initialization", but without performing diagnostic tests.

INITIALIZATION FOLLOWING A PRUN COMMAND

A file identifier must be provided as a parameter to the PRUN command. This file must contain the operating system to be initialized.

If a volume identifier is included in the file identifier, then the file must exist on that volume, otherwise an error message is returned and the current operating system remains active. If no volume identifier is specified, then the file must exist on the default drive. If the file exists but is not bootable, then the message

Invalid File Error (xx) on drive (x)

is displayed and the system must be reinitialized.

The specified file may be anywhere on diskette or hard disk; that is, it need not be the last PSAVEd file on the volume.

No diagnostic tests are performed.

THE INITIALIZATION FLOWCHART

The following figure summarizes the initialization process - both standard and nonstandard - by means of a flowchart.

INITIALIZATION AND CHANGING ENVIRONMENTS

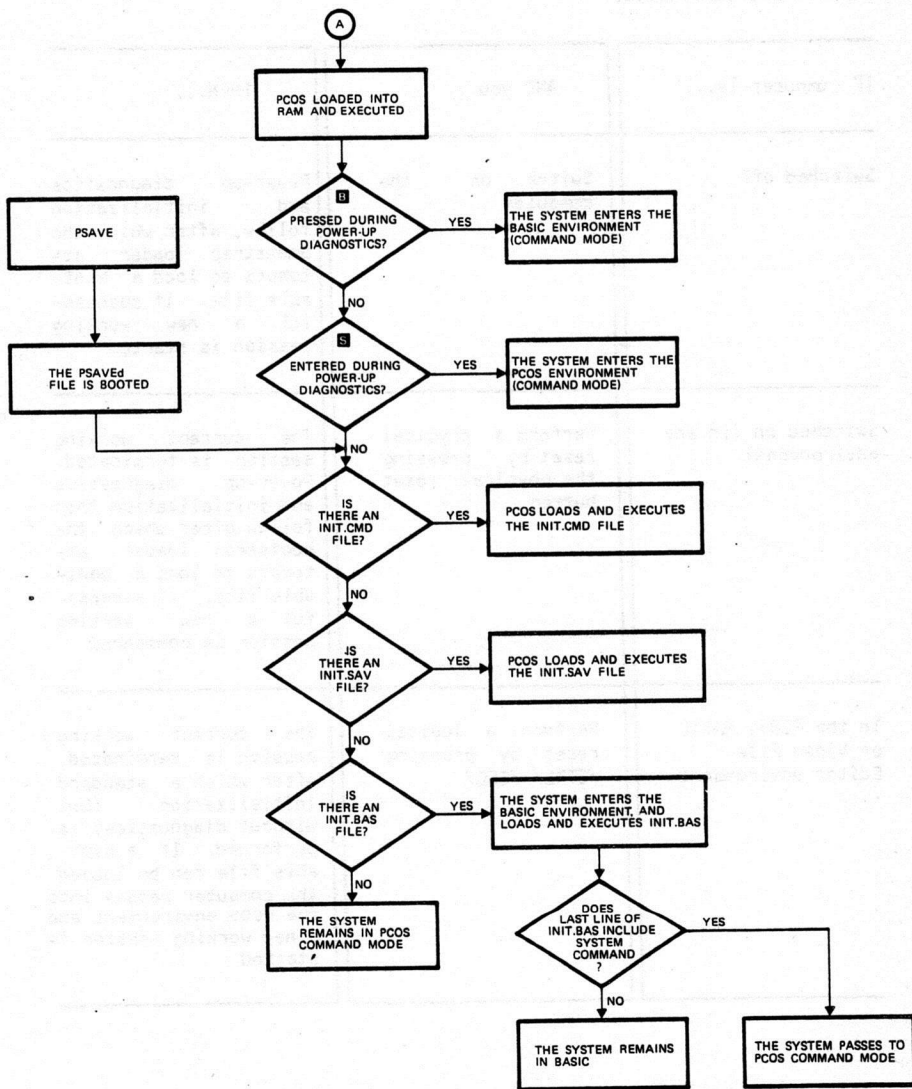


Fig 5-2 The Initialization Flowchart (cont.)

BEGINNING AND ENDING A WORKING SESSION

The following table summarizes the ways in which a working session can be started and terminated.

IF computer is...	AND you...	THEN...
Switched off	Switch on the computer	Power-up diagnostics and initialization follow, after which the bootstrap loader attempts to load a bootable file. If successful a new working session is started
Switched on (in any environment)	Perform a physical reset by pressing the physical reset button	The current working session is terminated. Power-up diagnostics and initialization then follow after which the bootstrap loader attempts to load a bootable file. If successful a new working session is commenced
In the PCOS, BASIC or Video File Editor environment	Perform a logical reset by pressing /CTRL/ /ESC/	The current working session is terminated, after which a standard initialization (but without diagnostics) is performed. If a bootable file can be loaded the computer passes into the PCOS environment and a new working session is started

6. CONFIGURING PCOS

ABOUT THIS CHAPTER

This chapter describes how to modify PCOS to meet your own needs. That is, how to make transient commands resident, how to assign functions to keys and how to change the system global parameters. For further details of the commands mentioned in this chapter refer to Chapter 14.

CONTENTS

<u>INTRODUCTION</u>	6-1
<u>MAKING TRANSIENT COMMANDS RESIDENT</u>	6-2
<u>ASSIGNING STRINGS TO KEYS</u>	6-6
KEYBOARD CHANGES	6-8
<u>CHANGING SYSTEM GLOBAL PARAMETERS</u>	6-8
SETTING THE SYSTEM GLOBAL ENVIRONMENT	6-9
DISPLAYING AND MODIFYING DEVICE NAMES	6-13
MODIFYING THE BASIC ENVIRONMENT	6-14
SETTING THE ENVIRONMENT FOR AN RS-232-C COMMUNICATIONS PORT	6-16
SETTING THE PRINTING ENVIRONMENT	6-16

RECONFIGURING THE KEYBOARD 6-19
LANGUAGE

ENABLING TRANSLATION OF 7-BIT 6-21
CHARACTER CODES TO 8-BIT

SAVING A USER-CONFIGURED PCOS 6-23

1950-1951

1952-1953

1954-1955

INTRODUCTION

PCOS contains facilities that enable you to modify the standard PCOS configuration supplied by Olivetti on your system diskette. The possible changes fall into three categories:

- Making transient commands resident: that is, transient commands that you expect to use frequently, or that you will require after removing the system diskette. These may be standard PCOS commands, or commands that you have written yourself using the Assembler package. (See the "Z8000 Assembler User Guide").
- Assigning functions to keys: for example a frequently input sequence of instructions can be assigned to a key, thus enabling a complex function to be performed with a single keystroke.
- Changing system global parameters: for example to change the system date and time, increase BASIC memory, accommodate a different type of printer, etc.

In this way you can create an operating system that is tailored to your specific needs.

The changes you make to your operating system become semi-permanent - that is, for the duration of the current working session (until you switch off the system or perform either a logical or physical reset) - unless you choose to make them permanent by saving your newly-configured operating system. How to do this is described in the section "Saving a User-Configured PCOS".

Note: A further way of altering the state of your system at initialization is by means of an initialization file. If such a file is present on an enabled volume, it will be loaded and executed at initialization to automatically perform functions that you always require at initialization (see Chapter 5).

Note: This chapter first tells you how to make transient commands resident, then how to assign functions to keys, and afterwards how to modify the system global parameters. These may, however, be performed in any order.

MAKING TRANSIENT COMMANDS RESIDENT

Once the standard PCOS is booted, three commands are loaded with PCOS. These are:

PLOAD used to load transient commands into memory

PUNLOAD used to remove a command from memory; that is, one that has previously been made memory resident by use of the PLOAD command. This can only be used for commands that are unloadable; that is, all except C1, RS232, EPRINT, VMOVE, PDEBUG (not included on the system diskette), PLOAD, PUNLOAD, TLANG, and LTERM

LTERM used in BASIC to distinguish between the use of the three line-terminator keys: / \downarrow /, /S1/ and /S2/

The computer's memory now comprises the PCOS Nucleus which includes the resident commands (PLOAD, PUNLOAD and LTERM) and user memory as shown in Figure 6-1.

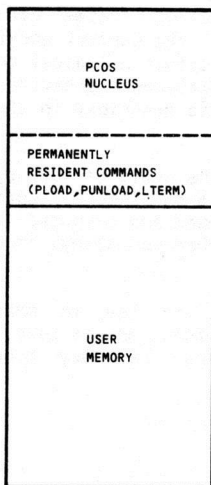


Fig. 6-1 Computer Memory after Initialization of a Standard PCOS

All commands other than PLOAD, PUNLOAD AND LTERM are transient and fall into two categories:

- Those with extension CMD : When you enter such a command it is loaded from the system diskette or hard disk, executed, then removed from

memory. Using a transient command in this way takes much longer than using a command that is already in the computer's memory. Furthermore, if PCOS resides on diskette, this must be inserted. It is often necessary to have some such commands memory resident. To do this, you require the PLOAD command.

- Those with the extension SAV : When you enter such a command it is loaded from the system diskette or hard disk into memory, executed, then remains in memory for the remainder of the current working session. Subsequent access will not require the command to be reloaded. Furthermore, once executed the command can be used again after the system diskette has been removed. It is, therefore, not necessary to use the PLOAD command unless you need to remove the system diskette before the command is first executed.

Commands with the SAV extension are:

- BVOLUME - for enabling a BASIC program to use the "Search" and "DiskFree" system calls or to obtain the name of the current volume
- CI - for programming the RS-232-C and current loop interface from BASIC
- EPRINT - for displaying PCOS error messages
- PDEBUG - for entering the program debug environment (not included on the system diskette)
- RS232 - for loading the RS-232-C interface package
- VMOVE - for copying diskettes on a single drive system
- TLANG - for enhancing the transfer of national-equivalent character codes (as set by SLANG) between PCOS and other systems or devices that use an 8-bit rather than a 7-bit character code

Example

If you enter

```
pl vc,px /CR/
```

then PCOS searches the drives, starting with the last volume accessed, for a command file with the short form "vc". VCOPY.COM is found, loaded into user memory, and information concerning the command is displayed on the screen. For example:

Disk file name = vcopy.cmd
Program name = Volume Copy Rev. 1.00
Operation Mode = Segmented / System
Main entry = <0A>%F7C0; Init entry = --None--
Memory allocated:
Block No. %00; Starting address = <0A>%F7BE; Size = %05C6

PCOS then looks for the command file with the short form "px". Since this does not exist, the error message

Error 92 in parameter 2

is displayed.

If you then enter

ep 92 /CR/

PCOS searches the drives for the corresponding command, finds EPRINT.SAV, loads it into memory and executes it. The result is a display of the PCOS error message 92 as follows

ERROR 92 --- command not found

That is, the command short form "px" does not exist.

Since the EPRINT command has the SAV extension it remains in semi-permanent memory after execution and therefore further reduces user memory.

The effect on memory caused by the addition of these two commands is illustrated in Figure 6-2.

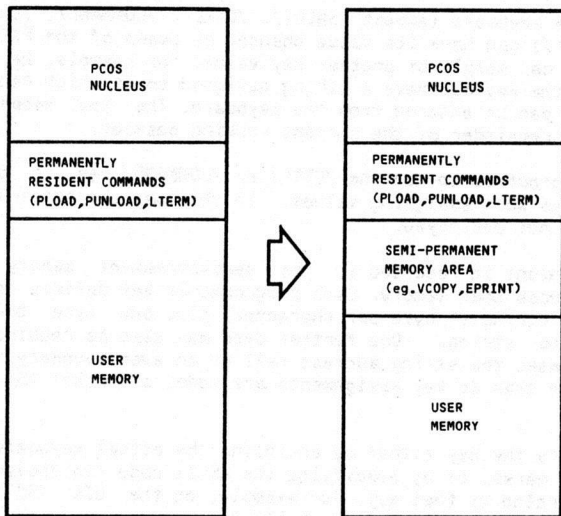


Fig. 6-2 Computer Memory after Making two Transient Commands Semi-Permanent

The exact amount by which user memory is reduced by the addition of the VCOPY and EPRINT commands can be determined using the DCONFIG command.

For each command that is loaded, user memory is reduced by the size of the command file (see the VLIST command) plus 40 bytes.

Making Commands Permanently Resident

Commands that are loaded into semi-permanent memory can be made permanent by means of the PSAVE command as described in the section "Saving a User-Configured PCOS".

ASSIGNING STRINGS TO KEYS

Any key on the keyboard (except /SHIFT/, /CTRL/, /COMMAND/, /ESC/, /S1/, /S2/, AND /↵/) can have its value changed by means of the PKEY command. The new value can simply be another key value, for example, key A becomes key B, or the key can have a string assigned to it which can perform a function that can be entered from the keyboard. The key keeps its new value for the remainder of the current working session.

It is normal practice to use the /CTRL/ or /COMMAND/ key in conjunction with other keys when assigning values. In this way the original function of the key is not destroyed.

The key assignment is recorded in the semi-permanent memory area and therefore reduces user memory. Each programmable key defined requires the space for the string (1 byte per character) plus one byte to hold the length of the string. One further byte may also be required if it is necessary to make the string address fall on an even boundary. Furthermore, if more than 26 key assignments are made, a further 156 bytes will be required.

You can specify the key either by enclosing the actual keyboard character in quotation marks, or by specifying the ASCII code (in decimal or hexadecimal) generated by that key. For example, on the USA ASCII keyboard 'B' can be specified by any of the following:

'B'

66

&42

all refer to the same key.

Similarly, the string(s) to be assigned to the key can be specified as either actual key values enclosed within quotation marks, or the ASCII code of each character, or a combination of the two. For example:

'ba',13,&A

is a valid string representing 'ba' followed by a carriage return and a line-feed.

Example

Suppose that you want to enter the BASIC Interpreter and execute the statement FILES simply by pressing key combination /CTRL/ /!1/ (on the USA ASCII keyboard) simultaneously. Do this by entering:

pk &E1,'ba',13,10,'files',13,10 /CR/

Where E1 is the (hexadecimal) ASCII code normally generated by pressing /CTRL/ /!1/.

CONFIGURING PCOS

On subsequently pressing /CTRL/ /!1/, PCOS will display 'ba', execute it by entering a carriage return (13) and a line feed (10), display 'files', then execute the BASIC command FILES by entering another carriage return / line feed pair.

The assignment will require the following memory space:

$$1 + 11 = 12 \text{ bytes}$$

and the effect on computer memory will be as shown in Figure 6-3.

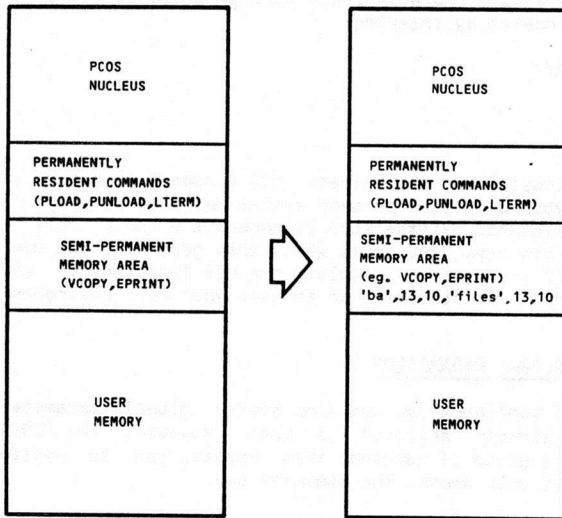


Fig. 6-3 The Effect of a PKEY command on Computer Memory

String assignments made in this manner are valid up to the end of the current working session. To make such assignments a permanent feature of the operating system you must use the PSAVE command as described in the section "Saving a User-Configured PCOS".

You can display a list of programmed keys, along with their string assignments by entering:

```
pk /CR/
```

A key assignment may be canceled by entering the ASCII code as a single parameter to the PKEY command, thereby releasing the memory occupied by the corresponding string. Moreover, all strings assigned using the PKEY command can be canceled by entering:

```
pk %c /CR/
```

KEYBOARD CHANGES

If you have previously used the Olivetti M20 personal computer, you will note that the keyboard on your present system has some new keys that were not on the M20 keyboard. Please turn to Appendix H for a full explanation of these new keys, the codes which they generate, and how you may reprogram them, if you wish, to simulate the M20 functions to which you are accustomed and/or to customize or enhance your PCOS environment.

CHANGING SYSTEM GLOBAL PARAMETERS

The standard PCOS configuration contains system global parameters with default values already assigned to them. However, the PCOS command library contains a group of commands that enable you to modify these parameters to suit your needs. The commands are:

- SSYS.CMD - for setting the system global environment. For example, setting the system date and time
- SDEVICE.CMD - for displaying or modifying the device name table; that is, a table that lists the current names assigned to the hardware configured devices
- SBASIC.CMD - for setting the BASIC environment
- SFORM.CMD - for specifying the printer type and printing style
- SCOMM.CMD - for specifying the environment for devices connected via the RS-232-C interface
- SLANG.CMD - for simulating the keyboard of another country
- TLANG.SAV - for enabling translation of 7-bit ASCII code to 8-bit

SETTING THE SYSTEM GLOBAL ENVIRONMENT

If, after initialization from a standard PCOS, you enter the SSYS command, the screen will display the default values of the system global environment parameters. That is, if you enter

```
ss /CR/
```

then the system will respond with a display as follows

```
Date = 01/01/1984 Time = 00:00:01 Disk Verify = 0 (Off)
Extent size = 8 Display = 0 (64 x 16) Disk time = 2
```

Each parameter is described in turn below.

Date

This parameter specifies the system date. Its value can be any actual date since 1899. It comprises month, day and year separated from each other by a date separator, where:

- Month is any integer value in the range 01 to 12
- Day is any integer value in the range 01 to 28, 29, 30 or 31 depending on the month
- Year is any integer value in the range 1900 (or 00) to 1999 (or 99)
- Date separator is any non-numeric printable character

(On keyboards other than the USA ASCII and USA ASCII with BASIC the order is day, month, year.)

The default value is 01/01/1984.

The system date is incremented by PCOS whenever the value of the Time parameter is incremented from 23:59:59 to 00:00:00.

The Date parameter is used in date-dependent programs. That is, it can be interrogated from BASIC using the BASIC DATE\$ function. For example:

DISPLAY	COMMENTS
<pre>100 IF DATE\$="04/30/82" THEN 3000 . . .</pre>	<p>statement 100 checks the date.</p>
<pre>500 DATE\$="05:06:82" . . .</pre>	<p>statement 500 sets the date and also changes the date separator.</p> <p>Note: it is not necessary to specify the year in full. The last two digits are sufficient</p>

Time

This parameter specifies the system time. Its value is maintained in hours, minutes and seconds separated by a time separator, where:

- Hours is any integer value in the range 00 to 23
- Minutes is any integer value in the range 00 to 59
- Seconds is any integer value in the range 00 to 59
- Time separator is any non-numeric printable character

The time parameter assumes its default value (00:00:01) at system switch-on and upon physical reset, but not upon logical reset. It is then incremented until you either switch the system off or perform a physical reset.

The time parameter can be interrogated from BASIC using the BASIC TIME\$ function for use within time-dependent programs. For example:

DISPLAY	COMMENTS
<pre>600 PRINT TIME\$. . .</pre>	<p>statement 600 displays the time.</p>

700 TIME\$="07:40:15" : : :	statement 700 sets the time
--------------------------------------	-----------------------------

Disk Verify

This parameter determines whether or not verification is to be performed following a diskette (or hard disk) I/O operation. Its possible values are:

- 1 - verification on
- 0 - verification off

When verification is on, data that is written to a diskette (or hard disk) is subsequently read and checked. If an error is detected then the message

ERROR 57 --- disk i/o error

is displayed.

The default is "0" - no verification.

Note: With Disk Verify on, the disk access time is slowed down significantly.

Extent Size

This parameter specifies the number of sectors (blocks) to be allocated to a file when an output operation requires more space. Its value must be an integer in the range 1 to 1087. A lesser value wastes less space at the end of the last extent to be allocated but may require more extents to be allocated. Conversely, a larger value will result in more wasted sectors at the end of the last extent allocated, but will reduce the number of extents that need to be allocated.

The default is "8".

Display

This parameter determines the number of characters per display line and the number of lines per screen. Its value must be one of :

0 - for a 64 character by 16 line display

1 - for an 80 character by 25 line display

The default is "0" - 64 by 16.

Disk time

This parameter specifies the number of seconds that the diskette drive motor will remain on after the last access to a diskette in that drive. This feature enables you to set this period such that start-up delays are minimized when, for example, an application performs communication or printer access between diskette accesses.

The Disk time parameter must be an integer in the range 1 to 30. The default is "2".

Using the SSYS Command to Modify the System Global Parameters

For example, if you enter

```
ss 12/12/84,09:30:00,,,1 /CR/
```

then the system date is changed to December 12, 1984, the system time is started from 9:30 a.m., the Disk Verify and Extent Size parameters remain unaltered, the Display parameter is changed to generate an 80 by 25 display, and the Disk time parameter remains unaltered. The computer responds by displaying the new parameter values as follows:

```
Date = 12/12/1984   Time = 09:30:00   Disk Verify = 0 (off)
Extent size = 8     Display = 1 (80 x 25)   Disk time = 2
```

Note that when entering the Date parameter it is not necessary to enter the year in full; the last two digits are sufficient. When examining the parameter value, however, all four digits will always be displayed.

Making the Changed Parameters Permanent

Any changes made to the Disk Verify, Extent Size, Display, and Disk time parameters remain until either they are respecified using another SSYS command, or the current working session is terminated. In the latter case these parameters revert to their default values. The Date and Time parameters are continually incremented until the system is either physically reset or switched off, upon which they too revert to their default values. Note, however, that logical reset has no effect on the Time and Date parameters.

All modified parameters can be made permanent by means of the PSAVE command as described in the section entitled "Saving a User-Configured PCOS".

DISPLAYING AND MODIFYING DEVICE NAMES

After initialization from a standard PCOS you can examine the device name default values via the SDEVICE command. That is, if you enter

```
sd /CR/
```

then the device name table will be displayed as follows:

Device Name	Type
cons:	R/W
prt:	W
com:	R/W
com1:	R/W

The device names are listed along with their device type. The latter is an indication as to whether the device can only be written to, only read from, or whether both read and write operations are possible.

The entries in the table have the following meanings:

cons:

This is the default device name for the PCOS console driver (VDU and keyboard). It is a read and write type of device.

prt:

This is the default device name for the printer driver. It is a write-only type of device.

com:

This is the default device name for the built-in RS-232-C communications port. This value is displayed only if the RS-232-C extension package has already been loaded by means of the RS232 command.

com1:

This is the default device name for the expansion RS-232-C communications port. This entry will not be displayed unless the RS-232-C extension package has already been loaded by means of the RS232 command.

Using Device Names

The device names are used for directing output to a specific device or for receiving input from a specific device. For example, when rerouting output from the console to the printer you would type

```
-dcons:,+dprt: /CR/
```

See Chapter 7 for further details on device rerouting.

Changing Device Names

To change the name of a device you must again use the SDEVICE command but this time specifying as parameters first the old device name, then the new name for the same device. For example, if you enter

```
sd prt:,printer: /CR/
```

then the printer driver will be renamed accordingly. PCOS will respond with the display

```
Device Name    Type  
prt:           W
```

changed to

```
Device Name    Type  
printer:      W
```

The new name will remain until the end of the current working session. If you subsequently examine the device name table by again entering

```
sd /CR/
```

then the updated device name table will be displayed.

Making Changed Device Names Permanent

The new device names entered using the SDEVICE command remain in effect until either they are changed again by a subsequent SDEVICE command, or the current working session is terminated. In the latter case the device names revert to their default values. The changed names, however, can be made permanent by means of the PSAVE command as described in the section "Saving a User-Configured PCOS".

MODIFYING THE BASIC ENVIRONMENT

If, on initializing the system from a standard PCOS, you subsequently enter the SBASIC command, the system will respond with the default values for the BASIC environment parameters. That is, if you enter

```
sb /CR/
```

then PCOS will respond with the default parameters as follows:

```
Files = 3 Memory = 36000 Windows = 1 Record size = 256
```

Each parameter is described in turn below.

Files

This parameter indicates the maximum number of files that can be open concurrently under BASIC. Its value must be in the range 0 to 15. The default is 3.

Memory

This parameter specifies how much memory (in bytes) is available to BASIC. All of BASIC memory resides within user memory and the maximum value of this parameter is therefore determined by the size of user memory. The default is 36,000.

Windows

This parameter indicates the number of windows for which BASIC memory is guaranteed. That is, if you open more windows than stated by this parameter you cannot be sure that the necessary space will be available. Its value must be in the range 1 to 16. The default is 1.

Record size

This parameter specifies the maximum record size in bytes. Its value must be in the range 1 to 4096. The default is 256.

The Effect on BASIC User Memory

The actual amount of memory that is available to BASIC programs is dependent upon all the parameters to the SBASIC command. That is, the Record Size and Windows parameters reduce the available memory by the following amount:

$$829 + (578 + R)F \text{ bytes}$$

where R is the value of the Record size parameter and F is the value of the Files parameter.

The Windows parameter reduces the amount of available memory by 108 bytes for each window - except the first - that has space guaranteed.

Example

If you enter

```
sb 4,40000,7,64 /CR/
```

then the maximum number of files that can be open simultaneously is defined as 4, BASIC user memory is set to 40000 bytes, the number of windows for which space is guaranteed is 7, and the maximum record size is defined as 64 bytes.

The amount of memory available to the BASIC program is therefore

$$\begin{aligned} & 40000 - [(829 + (578 + 64)4) + 108] \\ & = 35955 \text{ bytes} \end{aligned}$$

SETTING THE ENVIRONMENT FOR AN RS-232-C COMMUNICATIONS PORT

This procedure uses the SCOMM command which is described in full in the "Serial Interface for I/O Peripherals User Guide".

SETTING THE PRINTING ENVIRONMENT

If, upon initializing PCOS, you invoke the SFORM command without specifying any parameters, a display of the default values for the printing environment will result. That is, if you enter

```
sf /CR/
```

then the following display will appear.

```
auto ..... off, ptype ... pr15B, lines ..... 0060,  
spacing ... 0001, compress .. ne, interface ..parallel, .  
title .....
```

Each parameter is described below.

auto

This parameter determines the source of the nil parameters for this command. It can have one of two values:

- OFF - indicates that nil parameters will always take their values from the default values specified upon loading a standard PCOS, irrespective of any previous SFORM command.
- ON - indicates that the nil parameters will take their values from those set by the last SFORM command of the current working session. Alternatively, if there was no such command, the nil parameters take their values from the parameters that were made permanent during a previous working session (see section on "Saving a User-Configured PCOS").

A value of ON makes this command behave like any other Set System Global command; but note that OFF is the default value, and in this respect the SFORM command behaves differently from the other Set System Global Commands.

On entering the SFORM command immediately after initialization from a standard PCOS, the nil parameters will take their default values irrespective of the value of the "auto" parameter.

ptype

This parameter specifies the type of printer. It is entered as a valid printer number (in lower case, with hyphen, such as pr-15B) or as **transp**. The default value is pr-15B.

The parameter **transp** invokes transparent mode.

This is a free format facility whereby file contents are printed exactly as specified in the file irrespective of the type of printer. That is, no additional end-of-line characters or form feed characters are added.

lines

This parameter specifies the number of lines that are to be printed on each page before automatic form feed. Zero implies that no form feed will be issued.

The default value is 60.

spacing

This parameter specifies the number of inter-line spaces between printed lines. Its value can be :

1 - single spacing

2 - double spacing

etc.

The default is 1.

compress

This parameter specifies the style of the character. It is made up of two characters, the first of which must be one of:

w - wide (bold) character

n - normal width

and the second character specifies the width of character and must be one of:

c - compressed; that is, 16.6 characters per inch (at normal width)

e - elite; that is, 12.5 characters per inch (at normal width)

p - pica; that is, 10 characters per inch (at normal width)

The default value is ne.

interface

This parameter specifies whether the printer is to be connected to the serial or parallel interface. Its value must be one of:

- se - the RS-232-C (serial) built-in interface
- pa - the Centronics-like (parallel) interface
- s1 - the port on the single RS-232-C interface expansion (on the system motherboard)

The default value is pa.

title

This parameter defines the title which is to be printed at the top of each page. It can comprise up to 24 characters and must be enclosed in quotation marks. Entering a value of '' deletes the current title.

The default value is no title.

Using the SFORM command

If you enter

```
sf ,pr15B,30,2,,, 'PCOS USER GUIDE' /CR/
```

then the printing environment parameters will have the following values:

```
auto -          OFF - by default
ptype -         pr15B
lines -         30
spacing -       2
compress-       ne - by default
interface -     parallel - by default
title -         'PCOS USER GUIDE'
```

All subsequent printer output will be directed to a PR15B printer connected to the computer via the parallel interface. The output will be printed 30 lines per page at double spacing in normal/elite type-face. At the top of each page will be the heading "PCOS USER GUIDE".

If, at a later time during the same working session, you enter

sf on,,50,1 /CR/

then the printing environment parameters will take the following values:

```

auto -          ON
ptype -        pr15B - set by previous SFORM command
lines -        50
spacing -      1
compress -     ne - set by previous SFORM command
interface -    pa - set by previous SFORM command
title -        'PCOS USER GUIDE' - set by previous SFORM command
    
```

All subsequent output to the printer is to be printed 50 lines to a page at single spacing.

The above values will remain until either another SFORM command is issued, or until the end of the current working session. The values can be made permanent by use of the PSAVE command and thereby invoked during a future working session by entering an SFORM command with the auto parameter having a value of ON. See "Saving a User-Configured PCOS".

RECONFIGURING THE KEYBOARD LANGUAGE

The SLANG command enables you to reconfigure your keyboard to behave like any one of those defined in Appendix B.

If you enter

s1 /CR/

then the computer displays a menu as follows:

Available Country Configurations:

Italy	0	Denmark	8
West Germany	1	Yugoslavia	10
France	2	Norway	11
Great Britain	3	Greece.....	12
United States	4	Switzerland/France...	13
Spain	5	Switzerland/Germany...	14
Portugal	6	Germany (original)...	15
Sweden/Finland	7	Datev	16

Enter Your Selection by Number (or q to quit) -->

The last line of the display prompts you to enter the number corresponding to the country you require. For example, if you enter

2 /CR/

then the computer responds with the message

Language requested : French (2)

and the individual keys on your keyboard subsequently correspond to those of a French keyboard.

Alternatively, you can quit the menu without changing the country number by entering

q /CR/

If you already know the number of the keyboard you require it is not necessary to examine the menu. You can simply include the number as a parameter in the command line. For example, if you enter

s1 5 /CR/

then the system responds with the message

Language requested : Spanish (5)

and the Spanish keyboard is invoked.

The new keyboard setting will remain until either another SLANG command is entered, or the current working session is terminated. On rebooting the system the original language is restored, unless the current selection is made permanent by use of the PSAVE command as described in the section "Saving a User-Configured PCOS".

ENABLING TRANSLATION OF 7-BIT CHARACTER CODES TO 8-BIT

The TLANG command is provided to enhance the transfer of national-equivalent character codes (as set by SLANG, described above) between PCOS and other systems or devices which use an 8-bit ASCII code rather than the 7-bit code used by PCOS.

If you enter

t1 /CR

the system will display the default values as follows:

Port Name	Translate
prt:	w

Indicating that the device enabled is the printer and that the direction of translation is "write on output."

The full name of this utility is **tlang.sav**. As with other PCOS commands, only the first two letters need be entered. Its arguments are **portname**

and **direction**. These are explained below.

Portname

This is a device name as defaulted by the system or as set by the SDEVICE command, described previously. Its possible values are **prt**: (referring to the printer), **com**: (referring to the first RS-232 communications port), or **com1**: (referring to the expansion RS-232 port on the system motherboard).

Direction

This is the direction of translation, and may only be one of the following values:

n	--do not translate
w	--translate to 8-bit codes during output operations
r	--translate from 8-bit codes during input operations
r/w	--simultaneously do translations w and r

Examples

If you enter

```
tl prt: w /CR/
```

PCOS 7-bit codes will be translated to 8-bit codes during output to the system printer.

If you enter

```
tl com: r/w
```

this will enable translation in both directions during input from and output to the first serial port.

Note that before executing this operation, you must first load the RS-232 device driver by issuing the RS232 command. If you have not loaded this driver, entering the TLANG command as above will result in an Error 111 (invalid device name). This error message will occur:

- for an invalid port name
- if a translation direction is specified of which the device is not capable (for example, "r" for a printer)
- if a port name is specified for which the device driver has not been loaded

In general, a translation direction cannot be set until the corresponding device driver is loaded. However, most device drivers are already

permanently included in PCOS.

If you wish to invoke TLANG from within an application program, you are advised to do so by PLOADing it. For example, within a BASIC program you would include the statement

```
EXEC "pl tl"
```

which would set the default values as above but suppress the interactive display, which is not desired in an application program.

The TLANG utility is automatically PLOADed when first executed. During the PLOAD it will set the default for the printer device to "w". From then on, unless translation for the printer device is turned off, this value will remain in effect. TLANG may also be PSAVEd (as described under "Saving a User-Configured PCOS" so that it retains the selected translation directions when booted.

SAVING A USER-CONFIGURED PCOS

All temporarily resident commands, programmed key definitions and modified global parameters, along with any currently active user-defined font (see Chapter 11), and any specified global device rerouting parameters (see Chapter 7) can be made permanent features of the operating system by means of the PSAVE command. A user-configured PCOS can be created using this technique.

A new system diskette can be created by performing a sequence such as the following:

STEP	OPERATION
1.	Insert the system diskette in drive 0, and boot the system
2.	Use the PLOAD command to select all the individual utilities which you require to be permanently loaded
3.	Use the PKEY command to set up all the function key definitions required
4.	Use the global commands (SBASIC, SSYS, SFORM, SCOMM, SDEVICE, SLANG, and TLANG) to provide the system parameter values you require
5.	Insert a newly initialized diskette in drive 1

6.	<p>Enter</p> <p>ps 1: /CR/</p> <p>then the system responds</p> <p>Save system on "1:PCOS.SAV"? (y/n)</p> <p>A "yes" response causes the operating system to be saved on the specified file, after which the system is rebooted as for a standard initialization (see Chapter 5)</p>
7.	<p>Label your new system diskette with a suitable title to distinguish it from the standard PCOS. You should also write-protect it using an aluminized label and make a back-up (see Chapter 9)</p>

PCOS can also be PSAVEd on a volume that contains other files. Moreover, it may be given any valid file name. For example, entering

```
ps 1:newop /CR/
```

will save the operating system on the diskette inserted in drive 1. If the volume already contains a bootable file, but with a different name from the one you have just PSAVEd, the system will still reboot from the newly PSAVEd file. Moreover, any future attempt to reinitialize the system from this volume will cause the system to boot itself from the most recently PSAVEd file. To boot any other operating system on the volume requires the PRUN command. For example

```
pr 1:oldop /CR/
```

will boot the operating system saved in file "oldop".

Note: If your system has a hard disk, you can save the user-configured PCOS either on hard disk or on a diskette. If you choose to save it on the hard disk, you are advised to write-protect it using the FWPROT command.

UNITED STATES GOVERNMENT
OFFICE OF PERSONNEL MANAGEMENT
WASHINGTON, D. C. 20535

7. DEVICE REROUTING

1. PURPOSE

2. SCOPE

3. DEFINITIONS

4. PROCEDURE

5. REFERENCES

ABOUT THIS CHAPTER

This chapter describes how input and output can be re-routed from/to alternative devices or files.

For further details of the components mentioned in this chapter refer to Chapter 14.

CONTENTS

INTRODUCTION 7-1

LOCAL DEVICE REROUTING 7-2

GLOBAL DEVICE REROUTING 7-5

DEVICE REROUTING FROM A BASIC PROGRAM 7-7

INTRODUCTION

Normal operation of your computer involves entering information via the keyboard and receiving responses on the VDU. In some cases, however, you may wish to use additional or alternative input and/or output devices; for example a printer can be used to get a hard copy of what is displayed on the video. This can be done with PCOS using "device rerouting parameters."

Device rerouting can be specified on two levels:

- Local (for the duration of a single command only).
- Global (for all subsequent PCOS commands entered during the current working session, or until respecified by another global device rerouting operation).

The input device may be respecified as

- the keyboard (standard)
- a hard disk or diskette file
- an RS-232-C port

The output device may be specified as

- the VDU (standard)
- a disk or diskette file
- a printer
- an RS-232-C port

NOTE: The following Olivetti printers are available with the M24:

PR15/B
PR17/B
PR19/B
PR38/B
PR322B/PR321B
PR342B/PR321B
ET111/ET121

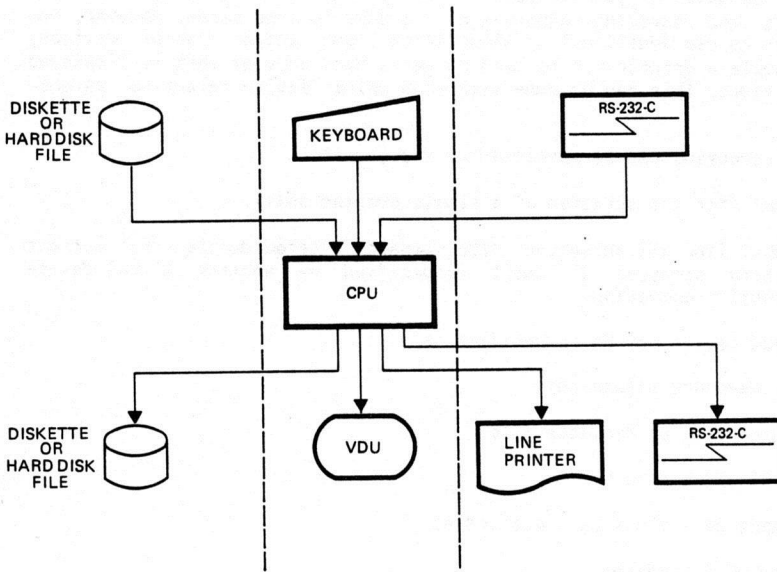


Fig. 7-1 Input/Output Devices

LOCAL DEVICE REROUTING

This facility enables you to change the input and output devices for the current PCOS command, so that input can be received from devices other than the keyboard, and output can be routed to devices other than the VDU. After the command has been executed the specified rerouting is canceled.

The implementation takes the form of a parameter to the command. This parameter specifies the name of the device whose I/O status is to be changed, preceded by two indicators. The first of these indicators specifies whether the device is to be canceled as an input/output device (indicated by a "-" sign), or enabled as an input/output device (indicated by a "+" sign). The second indicator specifies whether the device is to supply the input (indicated by "S" for source), or receive the output (indicated by "D" for destination).

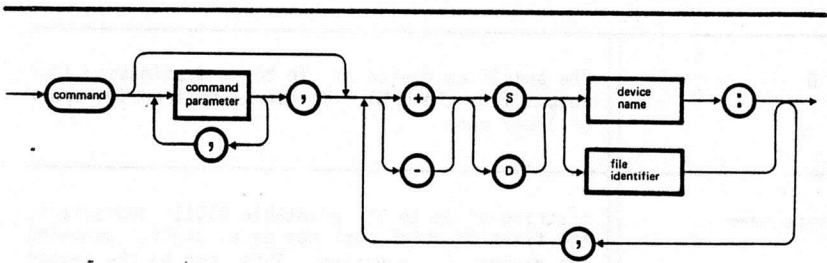


Fig. 7-2 Local Device Rerouting

Where

SYNTAX ELEMENT	MEANING
command	A keyword of any PCOS command to be executed via the devices specified in the device rerouting parameters
command parameter	A parameter to be passed to the PCOS command in question
+	The device or file specified is to be enabled
-	The device or file specified is to be canceled
S	The specified device is to be a source (for input to the CPU). It can be entered in upper or lower case

D	The specified device is to be a destination (for output from the CPU). It can be entered in upper or lower case
device name	A string of up to 13 printable ASCII characters, the first of which must not be a digit, denoting the device in question. This can be the device default name or any other name assigned to the device using the SDEVICE command. Note: A device name must be followed by a colon.
file identifier	Any file identifier. If the file does not exist it will be created on either the specified volume or the volume inserted in the last selected drive. Only one source and one destination file are permitted to be open at one time

Notes

- + or -, S or D and device name or file name follow each other without any spaces in between, and constitute one parameter.
- Each device rerouting parameter must be followed by a comma to separate it from the next parameter.
- The position of the device rerouting parameter in the command line is arbitrary. It is shown here at the end of the line merely for clarity.
- +DPRT: may also be written as +PRT (in upper- or lower-case letters) see last example below.
- If additional devices are enabled for input or output without disabling the currently active device(s) then devices become simultaneously active. The user must therefore exercise caution when rerouting input to avoid intermixing data from several devices.

Examples

IF you enter...	THEN...
<pre>v1 0:,-dcons:,+DPRT: /CR/</pre>	<p>The directory of the diskette inserted in drive 0 is printed (because of +DPRT:). However, it is not displayed on the VDU (because of -dcons:)</p>
<pre>ba +SBASIC.CMD /CR/</pre>	<p>The computer goes into the BASIC environment (see the BASIC command) and will take input from both the BASIC.CMD file and the keyboard. The CPU will no longer take input from the BASIC.CMD file when it goes back into the PCOS environment</p>
<pre>ss +PRT /CR/</pre>	<p>The global system environment parameters are not only displayed on the VDU, but also printed</p>

GLOBAL DEVICE REROUTING

Specification of global device rerouting causes the input and output of all PCOS commands executed thereafter to be rerouted. Any rerouting remains in operation until otherwise specified or until the system is reset. Furthermore, global device rerouting parameters can be made a permanent feature of the operating system by use of the PSAVE command.

Global device rerouting can be put through if device rerouting parameters, as described in the syntax diagram for local device rerouting (see Figure 7-2), are specified by themselves; that is, in the absence of a command keyword.

Notes:

- + or -, S or D, and device name follow each other without any spaces in between.
- Device rerouting parameters must be separated from each other by a comma.
- +DPRT: may also be written as +PRT (in upper- or lower-case letters).
- If additional devices are enabled for input or output without disabling the currently active device(s) then devices become simultaneously active. The user must therefore exercise caution when rerouting input to avoid intermixing data from several devices.
- No more than one file for output and one file for input may be specified at any one time.
- The keyboard can be disabled by specifying "-SCONS:". But, if this is done at the global level, control cannot be regained unless an external active device issues a "+SCONS:", or the system is reset.

Examples

IF you enter...	THEN...
+scom:,+dcom: /CR/	The CPU will receive input from both the keyboard and the built-in RS-232-C communications port (assuming the latter has already been initialized). Output will be displayed on the VDU and rerouted to the RS-232-C communications port
-dcom:,+dprt: /CR/	The RS-232-C communications port (previously enabled) will be canceled as a destination (for output) and the printer activated. Any other devices previously allocated as a source or a destination will remain active

DEVICE REROUTING

<pre>+d1:fileA /CR/</pre>	<p>"fileA" resident on the diskette inserted in drive 1 is enabled for output. If "fileA" does not exist on the volume it is created, and all the output will be both displayed on the screen and rerouted to "fileA"</p>
<pre>+dprt: /CR/ +d1:output /CR/ -dprt: /CR/ -d /CR/</pre>	<p>The first command enables the printer for output. The second enables the file named "output" resident on the diskette inserted in drive 1 to receive output. The third command cancels the printer and the fourth cancels the file</p> <p>Note: When a file is canceled the identifier is not checked and can therefore be omitted.</p>

DEVICE REROUTING FROM A BASIC PROGRAM

When local device rerouting is specified while in BASIC, I/O redirection will continue until either a command is executed to explicitly turn it off, or BASIC is exited. Conversely, global device rerouting will remain active after BASIC is exited.

Examples

IF you enter...	THEN...
<pre>ba /CR/ EXEC "v1 1:,+D1:OUT" /CR/ . . EXEC "-D1:OUT" /CR/</pre>	<p>The first command enters the BASIC environment. The first EXEC statement performs a VLIST command on the diskette inserted in drive 1 and routes the volume list to the file named "OUT" on the same diskette as well as to the VDU. All subsequent output will also be routed to the file until it is canceled by the second EXEC statement</p>

<pre> ba /CR/ EXEC "v1 1:;+dprt:" /CR/ . . SYSTEM /CR/ </pre>	<p>The system enters the BASIC environment. The EXEC statement causes the directory of the diskette inserted in drive 1 to be printed. All subsequent output is also directed to the printer until the printer is canceled as an output device by the SYSTEM command</p>
<pre> ba /CR/ EXEC "+prt" /CR/ . . SYSTEM /CR/ </pre>	<p>The system enters the BASIC environment. The EXEC statement specifies the printer as an output device, but because "+prt" is specified globally, the following SYSTEM command will not cancel it</p>

8. PROTECTION TOOLS

ABOUT THIS CHAPTER

This chapter describes the mechanism by which a file or volume can have protection applied. For details about the commands mentioned in this chapter refer to Chapter 14.

CONTENTS

<u>INTRODUCTION</u>	8-1
<u>VOLUME PASSWORDS</u>	8-1
<u>FILE PASSWORDS</u>	8-2
<u>WRITE-PROTECTION</u>	8-4
<u>COPY-PROTECTION</u>	8-4
<u>BASIC PROGRAM SECURITY</u>	8-5

INTRODUCTION

Your personal computer offers password protection at both volume and file level. Write-protection may be applied to a diskette or a file, but not to the hard disk. Moreover, a BASIC program can be protected so that it cannot be listed, edited or saved again.

The following sections summarize the various protection mechanisms.

VOLUME PASSWORDS

IF you want to...	THEN...
<p>Assign a password to a volume</p>	<p>Issue a VPASS command, specifying the password. For example</p> <pre>vp MYVOL:,MYPASS /CR/</pre> <p>OR</p> <p>If the volume already has a password, this must be specified by the VPASS command, which, in this case, will change the password. For example</p> <pre>vp VOL1/OLDPASS:,NEWPASS /CR/</pre>
<p>Access a volume that has a password (or a file saved on a volume that has a password)</p>	<p>Enable that volume by specifying the volume password after the volume name or the drive number. This can be done in a BASIC or PCOS command or in an OPEN statement.</p> <p>Note: Once a diskette volume password has been specified it need not be respecified until the diskette has been removed and another diskette has been referenced in the drive in which the diskette was inserted, or the current working session is terminated. Moreover, once a hard disk volume password has been specified, the hard disk will remain enabled until the end of the current working session</p>

<p>Remove a volume password</p>	<p>Issue a VDEPASS command, for example</p> <pre>vd MYVOL/MYPASS: /CR/</pre> <p>Note: You must know the password to use the VDEPASS command</p>
<p>Hide a volume password</p>	<p>Press /CTRL/ /G/.</p> <p>The cursor changes its shape and blink rate and the display of entered characters is suppressed (Hide mode). To return to normal display mode you must press /CTRL/ /G/ again or /CR/</p>

FILE PASSWORDS

IF you want to....	THEN...
<p>Assign a password to an existing file (that has no password)</p>	<p>Issue an FPASS command, specifying the password. For example</p> <pre>fp V1:MYFILE,PASS001 /CR/</pre>
<p>Create a new file and assign a password to it</p>	<p>Issue an FNEW command, specifying the password. For example</p> <pre>fn 1:newfile/pass002,4 /CR/</pre>
<p>Change the password to an existing file</p>	<p>Issue an FPASS command, specifying the old password within the file identifier, and the new password as the second parameter. For example</p> <pre>fp 1:newfile/pass002,pass102 /CR/</pre>

PROTECTION TOOLS

<p>Assign a password to a group of files</p>	<p>Issue an FPASS command, specifying the group using wild card characters. The same password will be assigned to all files in the group. For example</p> <pre>fp 1:my*,mine /CR/</pre>
<p>Assign a password to a list of files</p>	<p>Issue an FPASS command, specifying a list of files and, as the last parameter, the common password. For example</p> <pre>fp 1:myfile,yourfile,hisfile,herfile,ours /CR/</pre>
<p>Assign a password to a program file (that has none)</p>	<p>An FPASS command can be issued or, if in BASIC, the program can be stored on a volume using the SAVE command specifying the password. For example</p> <pre>SAVE "FILEABC/PASSABC" /CR/</pre>
<p>Access a file that has a password</p>	<p>Specify that password after the file name. For example</p> <pre>fw FILE002/PASS002 /CR/</pre>
<p>Remove a file password</p>	<p>Issue an FDEPASS command, specifying the password. For example</p> <pre>fd V1:MYFILE/PASS001 /CR/</pre>
<p>Hide a file password</p>	<p>Press /CTRL/ /G/ simultaneously.</p> <p>The cursor changes its shape and blink rate and the display of entered characters is suppressed (Hide mode).</p> <p>To return to normal display mode you must press /CTRL/ /G/ again, or /CR/</p>

WRITE-PROTECTION

You can apply write-protection to a diskette or a file.

IF you want to...	THEN...
Write-protect a diskette (that is, to prevent any writing to that diskette)	Cover the write-protect notch with an aluminized label
Remove write-protection from a diskette	Remove the aluminized label
Write-protect a file	Issue an FWPROT command, specifying the file identifier. For example fw 1:myfile /CR/
Remove write-protection from a file	Issue a FUNPROT command, specifying the file identifier. For example fu 1:myfile /CR/

COPY-PROTECTION

Copy-protection can only be assigned by the supplier. It can be assigned at file or volume level. A copy-protected file can be copied only a specified number of times. A copy-protected volume, however, cannot be copied at all.

BASIC PROGRAM SECURITY

Besides password protection and write-protection, your computer offers a further level of security. In the BASIC environment a program can be saved using the SAVE command with the P option. In such a case, the saved program can no longer be:

- listed
- modified
- saved again.

For example

SAVE "1:FILEAPROT", P /CR/

saves the file program FILEAPROT with the P option on the diskette inserted in drive 1.

THE UNIVERSITY OF CHICAGO
DEPARTMENT OF CHEMISTRY
5800 S. DICKINSON DRIVE
CHICAGO, ILLINOIS 60637
TEL: 773-936-3700

RESEARCH ASSISTANT
SARAH J. BROWN
5800 S. DICKINSON DRIVE
CHICAGO, ILLINOIS 60637
TEL: 773-936-3700

RESEARCH ASSISTANT
JAMES M. SMITH
5800 S. DICKINSON DRIVE
CHICAGO, ILLINOIS 60637
TEL: 773-936-3700

Volume handling is a critical factor in the design of a system. It is necessary to determine the volume of work that will be handled by the system and to design the system to handle this volume.

The volume of work is determined by the number of users, the number of transactions per user, and the number of transactions per second. The system must be designed to handle the maximum volume of work that will be handled by the system.

The volume of work is determined by the number of users, the number of transactions per user, and the number of transactions per second.

9. VOLUME HANDLING

The volume of work is determined by the number of users, the number of transactions per user, and the number of transactions per second.

The volume of work is determined by the number of users, the number of transactions per user, and the number of transactions per second.

The volume of work is determined by the number of users, the number of transactions per user, and the number of transactions per second.

The volume of work is determined by the number of users, the number of transactions per user, and the number of transactions per second.

The volume of work is determined by the number of users, the number of transactions per user, and the number of transactions per second.

The volume of work is determined by the number of users, the number of transactions per user, and the number of transactions per second.

The volume of work is determined by the number of users, the number of transactions per user, and the number of transactions per second.

The volume of work is determined by the number of users, the number of transactions per user, and the number of transactions per second.

The volume of work is determined by the number of users, the number of transactions per user, and the number of transactions per second.

ABOUT THIS CHAPTER

This chapter provides an operational guide to the use of the volume directed commands.

Throughout this chapter the availability of commands is always assumed. That is, it is assumed that either a volume containing the command is present in one of the drives, or that the command in question is already resident in memory.

Additional information about the commands mentioned in this chapter can be found in Chapter 14.

CONTENTS

PARTITIONING AND INITIALIZING 9-1 NEW VOLUMES

INITIALIZING A VOLUME 9-1

LISTING A VOLUME 9-2

LISTING A VOLUME USING THE 9-2
VLIST COMMAND

LISTING A VOLUME USING THE 9-3
VQUICK COMMAND

COPYING VOLUMES 9-4

COPYING VOLUMES ON A DUAL 9-5
DRIVE SYSTEM (VOLUMES OF EQUAL
SIZE)

COPYING DISKETTES USING ONE 9-6
DRIVE (VOLUMES OF EQUAL SIZE)

COPYING VOLUMES (OF DIFFERENT 9-7
SIZES)

BACKING UP AND RESTORING THE HARD DISK 9-7

NAMING AND PROTECTING A VOLUME 9-9

WRITE-PROTECTION 9-9

PASSWORD PROTECTION 9-10

NAMING A VOLUME 9-11

ALPHABETIZING A VOLUME 9-11

1. The first part of the document is a list of names and addresses of the members of the committee.

2. The second part of the document is a list of names and addresses of the members of the committee.

3. The third part of the document is a list of names and addresses of the members of the committee.

4. The fourth part of the document is a list of names and addresses of the members of the committee.

5. The fifth part of the document is a list of names and addresses of the members of the committee.

6. The sixth part of the document is a list of names and addresses of the members of the committee.

PARTITIONING AND INITIALIZING NEW VOLUMES

Before a hard disk can be accessed by PCOS, it must be partitioned by the HDISK command, and initialized by the VNEW command.

Hard Disks

Physical reformatting of the hard disk, hardly ever necessary, must be done under MS/DOS, with the FDISK command.

The PCOS partition of the hard disk is created by the HDISK command, as described in Chapter 12, and initialized with VNEW, described in the section "INITIALIZING A VOLUME."

Bad (unusable) sectors within the PCOS partition are detected and marked unavailable with the VVERIFY command.

Diskettes

Diskettes supplied by Olivetti for use with PCOS are preformatted, but not initialized, and therefore require the VNEW command before they can be used.

Used diskettes need not be reformatted, but may be reinitialized using the VNEW command. The same is true for hard disks.

INITIALIZING A VOLUME

To initialize a volume you must enter the VNEW command along with the volume identifier and, if the disk or diskette is not enabled, the volume password. For example

vn mydisk/pass: /CR/ - for a diskette having the name "mydisk"
and password "pass"

OR

vn mydisk: /CR/ - for a diskette having the name "mydisk"

OR

vn 10: /CR/ - for a hard disk

The system will respond with the message

Warning - vnew deletes all files. Initialise disk? (y/n)

Respond by entering

y /CR/

The initialization process then begins. Its completion is indicated by the PCOS prompt.

The initialization process removes any volume name.

Note that the above dialogue can be avoided by use of the no-interaction flag (%n). For example

```
vn %n mydisk/pass: /CR/
```

will perform the same function but without dialogue.

If you have forgotten the password of the diskette you will not be able to use the VNEW command.

LISTING A VOLUME

There are two commands available for listing the files contained in a volume. These are VLIST and VQUICK.

LISTING A VOLUME USING THE VLIST COMMAND

If you enter the VLIST command along with an appropriate volume identifier, a list of the files on the volume is generated. Displayed with each file name is information about that file: that is, the number of bytes occupied by the file, the number of sectors used, the number of sectors allocated, the number of extents used, and whether or not the file is write-protected or password-protected.

For example, to examine the contents of the system diskette you must insert it in an available drive (for instance drive 1) then enter

```
v1 1: /CR/
```

A display such as the one shown in Figure 9-1 will result.

VOLUME: 0	BYTES	SECTORS			WR-PROT/ PASSWORD
		USED	ALLOCATED	EXTENTS	
PC05.SAV	36912	145	146	1	
basic.abs	37611	147	148	1	
basic.cmd	1600	7	8	1	
bvolume.sav	1610	7	8	1	
ci.sav	1530	6	7	1	
ckey.cmd	806	4	5	1	
dconfig.cmd	2275	9	10	1	
eprint.sav	1573	7	8	1	
fcopy.cmd	4887	20	21	1	
fdepass.cmd	1145	5	6	1	
ffree.cmd	3626	15	16	1	
fkil.cmd	1263	5	6	1	
flist.cmd	2241	9	10	1	
fmove.cmd	2849	12	13	1	
fnew.cmd	1235	5	6	1	
font.all	15888	63	64	1	
fpass.cmd	1353	6	7	1	
frename.cmd	662	3	4	1	
funprot.cmd	1591	7	8	1	
fuprot.cmd	1581	7	8	1	
ieee.sav	2583	11	12	1	
SUBTOTALS		500	521	21	
21 FILES		(HIT ANY KEY TO CONTINUE)			

Fig. 9-1 Typical Volume List of the System Diskette

By repeatedly striking a key you will step through the list one screen at a time until you reach the end of the list. At this point the system will display some totals, after which the PC05 prompt will reappear.

Use of the no-interaction (%n) flag with this command causes the list to scroll continuously until the end of the list.

LISTING A VOLUME USING THE VQUICK COMMAND

Entering the VQUICK command along with a volume identifier also generates a list of the files on the volume but without any additional information about the files. For example, if you want to list the files contained on the system diskette using the VQUICK command, first insert the diskette in one of the drives (for instance drive 0), then enter

```
vq 0: /CR/
```

A display such as that shown in Figure 9-2 will result.

```
VOLUME: 0 yourdisk Free Disk Blocks = 206
PC05.SAV basic.abs basic.cmd bvolume.sav cl.sav
ckey.cmd dconfig.cmd eprint.sav fcopy.cmd fdepass.cmd
ffree.cmd fkill.cmd flist.cmd fmove.cmd fnew.cmd
font.all fpass.cmd frename.cmd funprot.cmd fuprot.cmd
ieee.sav kana.sav kb.all label.cmd lscreen.cmd
pkey.cmd prun.cmd psave.cmd rfont.cmd rkill.cmd
rs232.sav sbasic.cmd scomm.cmd sdevice.cmd storn.cmd
slang.cmd sprint.cmd ssys.cmd valpha.cmd vcopy.cmd
vdepass.cmd vformat.cmd vlist.cmd vmove.sav vnew.cmd
vpass.cmd vquick.cmd vrename.cmd vverify.cmd vfont.cmd
```

Fig. 9-2 Typical Volume Quick List of the System Disk

COPYING VOLUMES

There are two commands available for copying diskettes. These are

VCOPY - for copying a diskette from one drive to another. That is, for copying diskettes on a dual-drive system.

VMOVE - for copying diskettes on a system with only one diskette drive

Note that by using these commands it is not possible to copy a volume of a given capacity onto a volume of a different capacity. Any attempt to do so will cause an error message to be displayed: that is, you can only copy a 320 Kbyte diskette onto another 320 kbyte diskette, or a 640 Kbyte diskette onto another 640 Kbyte diskette. Copy operations between different diskette sizes can only be performed using the FCOPY command (or FMOVE command on a single drive system). To back up the hard disk you must use the HBACKUP command.

CAUTION: IT IS IMPORTANT THAT YOU WRITE-PROTECT YOUR SOURCE DISKETTE BEFORE COPYING IT TO AVOID ACCIDENTALLY OVERWRITING IT.

COPYING VOLUMES ON A DUAL-DRIVE SYSTEM (VOLUMES OF EQUAL SIZE)

Before copying a diskette you must first load the VCOPY command into memory using the PLOAD command (if you are copying the system diskette this is not necessary). That is, insert the system diskette into one of the drives, then enter

```
p1 vc /CR/
```

Remove the system diskette and insert the write-protected source diskette in one of the drives (for instance drive 0) and the target volume (which must not be write-protected) in the other drive. Then enter

```
vc 0:,1: /CR/
```

and the system will respond with

```
Warning- vcopy deletes all files. Copy disk? (y/n)
```

Respond by entering

```
y /CR/
```

A message such as the following will then be displayed

```
Read block 0 to 144
```

which indicates that the system is reading blocks 0 to 144 from the source volume. After a time this message changes to

```
Write block 0 to 144
```

which indicates that blocks 0 to 144 are being written to the target diskette. (The number of blocks read and written at once depends on the size of user memory.) When this operation is complete another "Read block" message appears for the next group of blocks. This process continues until all blocks have been copied to the target volume. At this point the message changes to

```
VCopy complete
```

and the PCOS prompt appears.

Note that in the command line the volume identifiers may alternatively be specified by name, in which case the target volume assumes the same name as the source volume once the copy operation is complete.

If the source volume is password-protected then the password need not be specified, but it will also be copied to the target volume. If the target volume is password-protected then its password need not be specified either.

COPYING DISKETTES USING ONE DRIVE (VOLUMES OF EQUAL SIZE)

If you intend to copy a diskette on a single-drive system you must first insert the system diskette, then make the VMOVE command memory resident via the PLOAD command (if you are copying the system diskette this is not necessary). That is, you must enter

```
pl vm /CR/
```

Now remove the system diskette.

The PLOAD step is unnecessary on a hard disk system, provided the VMOVE command resides on the hard disk.

Now insert the write-protected source diskette into the drive and enter the VMOVE command. If neither the source volume nor the target volume is password-protected it is only necessary to specify the command name. Simply enter

```
vm /CR/
```

If the source volume is password-protected then the full source volume identifier - including the password - must be specified. For example, if your source volume is named "mydisk" and has the password "pass" then you must enter

```
vm mydisk/pass: /CR/
```

Furthermore, if the target volume is password-protected then both the source and target volumes must be specified in full. For example, if your source volume is named "mydisk" and has the password "pass", and the target volume is named "yourdisk" and has the password "pass1", then it is necessary to enter

```
vm mydisk/pass:,yourdisk/pass1: /CR/
```

Once you have entered the command line the system responds with the following message

```
Warning- vmove deletes all files and PCOS. Vmove disk? (y/n)
```

This means that not only does the VMOVE command overwrite everything in the target volume, but in doing so it also overwrites your computer's memory, thereby deleting the operating system.

Respond by entering

```
y /CR/
```

At this point the computer will fill all its memory space with data from the source diskette. When memory is full, a message will be generated asking you to insert the target diskette and hit any key. Your computer will then transfer all the data from memory onto the target diskette.

This process will be repeated a number of times (depending on the

diskette capacity). At the end of the copy operation you can either make further copies or reboot PCOS.

COPYING VOLUMES (OF DIFFERENT SIZES)

Where the source and destination volumes are of different sizes, the copy operation can only be performed using a file copy command. If the target volume is to contain only the contents of the source volume it must be blank, and formatted. For example, to copy the entire contents of a 320 Kbyte diskette inserted in drive 0 onto a 640 Kbyte diskette inserted in drive 1 enter

```
fc 0:*,1 /CR/
```

The wild card character '*' specifies all the files on volume 0.

You must take care, however, when copying from a larger volume to a smaller volume that there is enough capacity on the diskette to accommodate all files.

This operation can be performed on a single drive system using the FMOVE command but note that wild card characters may not be used; each file must be specified in a separate FMOVE command.

For further details on the FCOPY and FMOVE commands refer to Chapter 10.

BACKING UP AND RESTORING THE HARD DISK

While it is possible to back up the hard disk onto a number of diskettes using the FCOPY command, this is cumbersome because:

- the amount of information may be very large and therefore require a lot of copy operations
- it is difficult to keep track of which files you have already copied.

Therefore the FCOPY command should be used only if you want to back up a small number of files. Moreover, FCOPY cannot copy a file that is larger than the target diskette.

Creating a Backup

The HBACKUP command solves the above problems and enables you to create a backup of the hard disk on a series of diskettes. When you change diskettes the program remembers which files it has already copied. Moreover, it reduces the number of diskettes required by compressing the data. The program creates separate backup files -- one per diskette -- which have a different format from the actual files on hard disk. It copies as many hard disk files (or part of a file) as it can into each of these "composite" files.

To back up the entire hard disk you must first insert a blank, formatted diskette in the drive, then enter the command line. However, the procedure differs depending on whether or not your hard disk contains password-protected files. If not, you simply enter

```
hb 10:,0: /CR/
```

If the hard disk does contain password-protected files, these files must be specified along with their passwords. Thus, if your hard disk contains two password-protected files named "myfile" and "yourfile", having passwords "mypass" and "yourpass", respectively, you need to enter

```
hb 10:myfile/mypass,yourfile/yourpass,* /CR/
```

where "*" backs up all files except password-protected ones.

After entering the command line the backup operation begins and a file named "Backup.000" will be created on the diskette. To avoid confusion you are advised not to use this file name for anything else. This composite file will occupy the entire diskette -- assuming the backup requires more than one diskette -- and will contain:

- a table of all the files to be copied from the hard disk. This provides a reference as to what the backup contains.
- as many of these files as will fit on the remainder of the diskette.

When this operation is complete you will be prompted

```
disk full - insert new disk and press RETURN
```

You must then insert another blank, formatted diskette and strike /CR/. The file on this diskette will be called "Backup.001". The backup will continue from where it was when the previous diskette became full. Note that only the first diskette contains the table of file names.

You must repeat the process until all the files have been copied from hard disk. At this point the operation will terminate and the PCOS prompt will reappear.

Succeeding diskettes will contain one composite file each named "Backup.002", "Backup.003", etc. It is advisable to label each diskette as you proceed, because when you restore the data to the hard disk you will need to restore the diskettes in the same order as they were made.

Note that copy-protected files will not be backed up.

Restoring a Backup

As the files backed up by the HBACKUP command are of a different format than that of the original files on hard disk, the FCOPY command cannot be used to copy them back to the hard disk. Instead you must use the HRESTORE command. This command restores the backup diskettes back to the hard disk, and returns the actual hard disk files to their original

format. Note that you must restore the diskettes in the correct sequence.

To restore a backup, insert the first of the backup diskettes -- the one containing the file "Backup.000" -- into the diskette drive. At this point you may check the contents of the backup by entering

```
hr %t /CR/
```

This will display the table of file names. If you are satisfied that this is the backup you wish to restore then enter

```
hr 0:,10: /CR/
```

The backed up files on this diskette are then restored to the hard disk in their original form. On completion you will be prompted to insert the next diskette.

You must then insert each backup diskette in turn and continue the restore operation for each diskette by striking /CR/. After the last diskette is restored control will be returned to PCOS.

Note that a write-protected file cannot be overwritten by the HRESTORE command.

Remark

The HBACKUP and HRESTORE commands can also be used to back up and restore specified files, or groups of files, from/to the hard disk. Moreover, the HRESTORE command can also restore a portion of a backup. Refer to Chapter 10 for details.

NAMING AND PROTECTING A VOLUME

Volumes can be write-protected and/or password-protected. Furthermore, it is often convenient to name a volume such that it can subsequently be identified by name rather than the number of the drive in which it resides.

WRITE-PROTECTION

Diskettes can be write-protected by fixing an aluminized label over the notch in the side of the diskette. Write-protection can be removed by simply removing the aluminized label. Hard disks cannot be write-protected.

PASSWORD PROTECTION

Password protection can be applied to a volume by means of the VPASS command. If you enter

```
vp 1:,pword /CR/
```

then the diskette inserted in drive 1 will be assigned the password "pword".

Similarly the (optional) hard disk can be password-protected by specifying either the volume name or the drive number. In the latter case the drive number is always 10. For example, if you enter

```
vp 10:,dpass
```

then the hard disk is assigned password "dpass".

Password protection has no effect on some commands, such as VCOPY, VLIST, and VQUICK, but most operations require the diskette to be enabled.

Enabling a diskette implies specifying the diskette password as part of its volume identifier as a parameter to a command. For example, if you terminate the current working session, then reboot the system and insert into drive 0 the diskette that you previously password-protected, then enter

```
v1 0/pword: /CR/
```

then the files contained on that volume are listed and the volume is enabled. The volume then remains enabled until either the volume is removed from its drive and another one inserted, or the current working session is terminated.

To remove password protection from a diskette use the VDEPASS command. For example, if you enter

```
vd 0/pword: /CR/
```

then password "pword" is removed from the diskette. But note that you must know the password to be able to delete it.

Do not assign a password to your system diskette since doing so prevents you from accessing any command on that diskette. Moreover, the VDEPASS command will also be disabled and hence you will not be able to enable the volume except from a backup.

NAMING A VOLUME

There are two commands that enable you to name a volume. These are VNEW and VRENAME.

The VNEW command enables you to name a volume at the same time as initializing by specifying the volume name as the second parameter to the command. For example, if you enter

```
vn 1:,datadisk /CR/
```

then the new diskette inserted in drive 1 will be initialized and assigned the name "datadisk".

Alternatively, a used diskette (or a hard disk) can be reinitialized and assigned a name by use of the VNEW command. For example, if you enter

```
vn 1:,newname /CR/
```

then the volume inserted in drive 1 is reinitialized and assigned the name "newname".

The second method of naming a diskette (or hard disk) simply assigns a name to an existing volume. This is done using the VRENAME command. If you enter

```
vr newname:,nextname /CR/
```

then the volume you previously named "newname" will have its name changed to "nextname". Note that it is not necessary to specify an old volume name in this command; it is also valid to specify the diskette (or hard disk) by the drive number. For example, if the last operation was performed with the diskette in drive 1 it would have been equally effective to enter

```
vr 1:,nextname /CR/
```

Note that if a volume is not enabled then its password must be specified if its name is to be changed using the VRENAME command.

ALPHABETIZING A VOLUME

PCOS contains a facility that enables you to sort the files contained in a volume into alphabetical order. Any unused directory blocks on the volume are removed, thereby improving access time whenever the directory is scanned. To do this you simply enter the VALPHA command along with the volume identifier. For example, if the volume named "datadisk" resides in one of the drives you can alphabetize it by entering

```
va datadisk: /CR/
```

Note that if the volume is not enabled then you must also specify the password. Furthermore, the volume must not be write-protected.

ABOUT THIS CHAPTER

This chapter provides an operational guide for the use of the file directed commands.

Throughout the chapter the availability of commands is always assumed. It is always that either a volume containing the command is inserted in one of the drives, or that the command in question is already resident in memory.

For a detailed description of the commands mentioned in this chapter, refer to Chapter 14.

CONTENTS

<u>CREATING FILES</u>	10-1
CREATING AN EMPTY FILE	10-1
CREATING A FILE BY COPYING	10-2
<u>COPYING FILES</u>	10-2
COPYING FILES ON A DUAL DRIVE SYSTEM	10-3
COPYING FILES USING ONE DRIVE	10-5
BACKING UP HARD DISK FILES	10-6
<u>LISTING FILES</u>	10-8
<u>PROTECTING FILES</u>	10-10
PASSWORD PROTECTION	10-10
WRITE-PROTECTION	10-11

FREEING UNUSED FILE BLOCKS 10-12

DELETING AND RECOVERING FILES 10-15

DELETING FILES 10-15

RECOVERING DELETED FILES 10-16

RENAMING FILES 10-17

1950
1951
1952
1953
1954



CREATING FILES

File creation within the PCOS environment can be performed either by the FNEW command which creates empty files, or simply by copying existing files using the FCOPY or FMOVE command. Text files can be created using the Video File Editor (see Chapter 13). BASIC files can be created via the Video File Editor and also from the BASIC environment (see "BASIC-8000 User Guide").

CREATING AN EMPTY FILE

Creating an empty file requires the use of the FNEW command. Before this command can be executed the volume on which the file is to be created must be enabled and must not be write-protected.

To create an empty file simply enter the FNEW command along with the file identifier and the file size (in terms of the number of sectors to be made available to the file). For example

```
fn 1:myfile,6 /CR/
```

causes an empty file named "myfile" to be created on the diskette inserted in drive 1, and 6 sectors (blocks) to be allocated to it.

If the file size parameter is specified as zero, or not specified at all, then the file system allocates the number of sectors specified in the "extent size" global parameter. For example, if you enter

```
ss ,,,4 /CR/
```

```
.  
.  
.
```

```
fn 1:newfile,0 /CR/
```

then the "extent size" parameter is set to 4 by the SSYS command, and an empty file named "newfile" is created on the diskette inserted in drive 1 and consequently allocated 4 sectors.

Note that the minimum number of sectors that can be allocated is 2. If you specify the file size as 1 then the file system issues an error message.

If there is insufficient space on the diskette to accommodate the new file then a message

```
ERROR 61 ----- disk filled
```

is displayed.

CREATING A FILE BY COPYING

Files are created with the file copy commands if the target file does not already exist. For example, if you enter

```
fc 0:myfile,1: /CR/
```

on a dual drive system then the file named "myfile" on the diskette inserted in drive 0 will be copied onto the diskette inserted in drive 1. That is, the file named "myfile" will be created there, unless a file of that name already exists, in which case it will be overwritten. Furthermore, if you enter

```
fc 0:myfile,yourfile /CR/
```

then file "yourfile" is created (if it does not already exist) and the contents of "myfile" are copied into it.

On single drive systems you must use the FMOVE command. In this case only one file can be specified. Assuming the source diskette is already inserted in the drive, the file to be copied is written into user memory. You must then remove the source diskette, insert the target diskette and copy the contents of user memory onto the target diskette, thereby creating a file of the same name. If the file is too large to fit into user memory then the operation requires a number of passes.

Further operational details are provided in the section "Copying Files."

COPYING FILES

The PCOS command library has two commands for copying files from diskettes. These are

FCOPY - for copying files on a dual drive system

FMOVE - for copying files from one diskette to another using one drive.

Hard disk files can be backed up onto a series of diskettes using the HBACKUP command, and subsequently restored using the HRESTORE command.

COPYING FILES ON A DUAL DRIVE SYSTEM

First the volume containing the file(s) to be copied must be inserted in one of the drives. To copy a file, enter the FCOPY command specifying the file to be copied (the source file) and the destination (the target volume or target file).

Copying One File at a Time

To copy "myfile" from a volume named "myvol" inserted in drive 1 to a volume named "copyvol" inserted in drive 0 enter

```
fc myvol:myfile,copyvol: /CR/
```

and, provided "myfile" does not already exist on "copyvol" the system will respond

```
COPY FILE 1:myfile TO 0:myfile
```

The file is copied and the new file is given the same name as the source file.

If the source file has a password this is also assigned to the new file. For example, to copy a file named "datafile" having password "dpass" from the diskette inserted in drive 0 to the hard disk you would enter

```
fc 0:datafile/dpass,10:
```

Assuming a file named "datafile" does not already exist on the hard disk, the system would display

```
COPY FILE 0:datafile TO 10:datafile
```

after which a copy of "datafile" with password "dpass" would be created on the hard disk.

The FCOPY command can also copy a file to an existing file thereby overwriting the target file. For example, to copy a file named "ifile1" having password "ipass" from volume "myvol" inserted in drive 0 to file "ifile2" with password "ind" on volume "yourvol", enter

```
fc myvol:ifile1/ipass,yourvol:ifile2/ind /CR/
```

The system will respond

```
COPY FILE 0:ifile1 TO 1:ifile2
File already exists. Do you wish to overwrite (y or n)?
```

If you then respond by entering

```
y /CR/
```

the target file will keep the file name "ifile2" and password "ind", but its contents will be overwritten with those of "ifile1". The PCOS prompt

appears when the copy operation is complete.

It is also possible to copy a file to another file within the same volume. For example, if you enter

```
fc myvol:ifile1/ipass,copyfile/cpass /CR/
```

and reply "y" to the subsequent prompt, the contents of "copyfile" are overwritten by those of "ifile1". However, "copyfile" still maintains its name and password.

Note: The dialogue in any of the above examples can be bypassed by using the no-interaction (%n) flag. For example

```
fc %n myvol:newfile,copyvol: /CR/
```

performs the copy operation without any intervening dialogue.

Copying Groups of Files Using Wild Cards

A group of related files can be copied from one volume to another using the wild card facility. To do this you must enter the FCOPY command along with the group of files specified using wild cards, and the target volume. For example, you can copy all the volume-directed commands from the system diskette to another volume using one command by inserting the system diskette in one of the drives (for instance, drive 0), having the target volume inserted in the other and entering

```
fc 0:v*,1: /CR/
```

The system will respond with the display shown in Figure 10-1.

```
COPY FILE 0:valpha.cmd TO 1:valpha.cmd
COPY FILE 0:vcopy.cmd TO 1:vcopy.cmd
COPY FILE 0:vdepass.cmd TO 1:vdepass.cmd
COPY FILE 0:vformat.cmd TO 1:vformat.cmd
COPY FILE 0:vlist.cmd TO 1:vlist.cmd
COPY FILE 0:vmove.sav TO 1:vmove.sav
COPY FILE 0:vnew.cmd TO 1:vnew.cmd
COPY FILE 0:vpass.cmd TO 1:vpass.cmd
COPY FILE 0:vquick.cmd TO 1:vquick.cmd
COPY FILE 0:vrename.cmd TO 1:vrename.cmd
COPY FILE 0:vverify.cmd TO 1:vverify.cmd
```

Fig. 10-1 Sample Display of a Copy Operation on a Group of Files

The copy operation is then complete. In this case none of the files already existed on the target volume. For each file that had existed on

the target volume the prompt

File already exists. Do you wish to overwrite (y or n)?

would be displayed.

Password-protected files will not be copied by this procedure, unless the password is specified in the command line and is common to all files in the group.

Note: The dialogue in all the above examples can be bypassed by entering the no-interaction (%n) flag in the command line. In this case the system simply returns the PCOS prompt when the copy operation is complete.

COPYING FILES USING ONE DRIVE

Copying Files Diskette to Diskette

To copy files from one diskette to another on a single drive or hard disk system you must enter the FMOVE command along with the file name. For example, to copy "myfile" from the diskette named "mydisk" to the diskette named "archive" you would first enter

```
fm myfile /CR/
```

and the system will respond with

```
Please put SOURCE disk in drive,  
then press any key (CTRL C to abort)
```

Respond by inserting the diskette named "mydisk" in the drive and hitting any key. A message such as the following will then be displayed:

```
File transfer will take two pass(s)  
Please put DESTINATION disk in drive,  
then press any key (CTRL C to abort) (pass 1 of 2):
```

Respond as instructed. Part of the file has been copied into user memory already.

The above message indicates that it will require two passes to transfer the file. The system must first copy it into user memory, then ask you to insert the target diskette, and finally transfer the file to the target volume. In this case, however, the file is too large to fit into user memory, therefore the transfer requires two passes.

Now insert the target diskette "archive" and hit any key as instructed. A file named "myfile" will be created on the target diskette (if it does not already exist) and the contents of the user memory copied into it. When this is completed the following message will be displayed:

```
Please put SOURCE disk in drive,  
then press any key (CTRL C to abort) (pass 2 of 2):
```

Respond by removing the target diskette, inserting the source diskette and hitting any key, as instructed; then the remainder of the file will be copied into user memory. When this is completed the following message will be displayed:

```
      Please put DESTINATION disk in drive,  
      then press any key (CTRL C to abort) (pass 2 of 2):
```

Respond as instructed and the remainder of the file will be copied from user memory onto the target diskette. The process is then complete. The system responds with the PCOS prompt.

Note that it is not possible to use the wild card facility with FMOVE.

Copying Files on the Same Diskette

This process is the same as for copying files on the same diskette on a dual drive system. For example, to overwrite the contents of a file named "oldfile" with the contents of "myfile" - both being resident on the same diskette - insert the diskette in the drive and enter

```
      fc myfile,oldfile /CR/
```

The system responds

```
      COPY FILE 0:myfile TO 0:oldfile  
      File already exists. Do you wish to overwrite (y or n)?
```

Respond by entering

```
      y /CR/
```

and the copy process is executed. The target file keeps its original name. Furthermore, a password-protected target file will also keep its own password. For example, if you enter

```
      fc %n dfile/passa,efile/passb /CR/
```

then "efile" is overwritten with the contents of "dfile", but the target file maintains its name "efile" and password "passb". Use of the no-interaction (%n) flag bypasses the dialogue.

BACKING UP HARD DISK FILES

To back up one or more files from the hard disk it will often be convenient to use the FCOPY command as described previously. However, if the backup requires several diskettes, this process becomes unwieldy. It is cumbersome to remember which files have already been copied. In such cases it is more convenient to use the HBACKUP command because this command remembers which files have already been copied. You simply change the disks when prompted. Moreover, a file that is larger than the target diskette cannot be copied using the FCOPY command. In this case you must use the HBACKUP command.

Making a Backup

Chapter 9 describes how you can use the HBACKUP command to make a backup of the entire hard disk. However, the procedure for backing up specific files is slightly different.

You must first insert a blank, formatted diskette in the drive and enter the command line.

If, for example you wish to back up a large file named "bigfile" from your hard disk, enter

```
hb 10:bigfile,0: /CR/
```

To back up a group of files you can use a wild card character. Moreover, a list of files may also be specified. For example

```
hb 10:*.cmd,myfile,yourfile/yourpass /CR/
```

will back up all files that have the "cmd" extension, and the files "myfile" and "yourfile". Note that any password must be specified.

Once you have entered the command line correctly the procedure continues as for backing up the entire hard disk as described in Chapter 9.

Note that if you specify a file that does not exist on the hard disk, the message

```
file not found
```

followed by the file name will be displayed, and the backup will continue.

Restoring a Backup

The backup can only be restored using the HRESTORE command. The procedure for restoring the entire backup is exactly the same as described in Chapter 9. However, if you wish to restore only a subset of the backed-up files, you must specify the files in the command line. You may specify a single file, a list of files separated by commas, and/or a group of files specified using wild card characters. For example

```
hr *.bas,yourfile,0:,10: /CR/
```

will restore from drive 0 all files that have the file name extension "bas", and the file "yourfile".

The procedure is otherwise identical to restoring an entire backup.

LISTING FILES

To list a text file simply enter the FLIST command followed by the file identifier. For example, to list the file named "workfile" that is resident on the diskette inserted in drive 1, enter

```
fl 1:workfile /CR/
```

and a display such as the following will be generated.

```
LISTING FILE 1:workfile
10 subname$="getvolname"
20 volname$="12345678901234"
30 call "bv"(subname$, @volname$)
40 print "the current volume is named"; volname$
50 end
```

Fig. 10-2 Sample Display of a Text File List

Moreover, any file can be listed in hexadecimal form, 16 bytes per line, by including the %h program flag in the command line. For example, if you enter

```
fl %h 1:workfile /CR/
```

then the following display will be generated (in 80 by 25 display mode):

```
LISTING FILE 1:workfile
0000: 31 30 20 73 75 62 6E 61 6D 65 24 3D 22 67 75 74 10 subname$=get
0010: 76 6F 6C 6E 61 6D 65 22 00 0A 32 30 20 76 6F 6C volname$..20 vol
0020: 6E 61 6D 65 24 3D 22 31 32 33 34 35 36 37 38 39 name$="123456789
0030: 30 31 32 33 34 22 00 0A 33 30 20 63 61 6C 6C 20 01234"..30 call
0040: 22 62 76 22 23 73 75 62 6E 61 6D 65 24 2C 40 76 "bv"(subname$,ev
0050: 6F 6C 6E 61 6D 65 24 29 00 0A 30 30 20 70 72 69 olname$)..40 pri
0060: 6E 74 20 22 74 68 65 20 63 75 72 72 65 6E 74 20 nt "the current
0070: 76 6F 6C 75 6D 65 20 69 73 20 6E 61 6D 65 64 22 volume is named"
0080: 38 20 70 6F 6C 6E 61 6D 65 24 00 0A 35 30 20 65 ; volname$..50 e
0090: 6E 64 0D 0A 0D 0A nd....
```

Fig. 10-3 Sample Display of a Hexadecimal File List

The first four columns specify the byte address relative to the beginning of the file (in hexadecimal) of the first character in the line. Each such entry is followed by the hexadecimal codes for the 16 bytes subsequent to the displayed address. The right-hand columns show the corresponding ASCII characters, or '.' where the character is nonprintable (octal characters 0 to 31 and 127).

The text or hexadecimal contents of password-protected files can be displayed, but the password must be specified to do so.

The contents of more than one file can be displayed by specifying a list of files in the command line. Moreover, groups of files can be listed by use of the wild card facility. However, password-protected files can be displayed in this manner only if the password is common to all the files in the group.

Remark

A text listing is useful only where the specified file is a text file, but a hexadecimal listing of any file can be obtained.

PROTECTING FILES

PCOS files can be password-protected to limit access to those who know the password. Furthermore, they can be write-protected to prevent accidental damage to the contents of a file.

PASSWORD PROTECTION

Assigning and Removing a File Password

To assign a password to a file, enter the FPASS command along with the file identifier and the password to be assigned. For example, to assign the password "pass" to the file named "mine" that resides on the hard disk, simply enter

```
fp 10:mine,pass /CR/
```

At some later time, you may wish to remove this password protection. To do so requires the FDEPASS command.

To use the FDEPASS command simply enter the command mnemonic along with the file identifier, including the password. In this case enter

```
fd 10:mine/pass /CR/
```

Note that you must know the password to be able to remove it.

Assigning and Removing a Password to a Group of Files

To assign a password to a group of files is the same as for a single file except that the wild card feature is used. For example, to assign the password "rst" to all files beginning with "data..." on a diskette named "datadisk" you would enter

```
fp datadisk:data*,rst /CR/
```

and the system would respond

```
Set Password on 0:data1?
```

Then a "y" response will set the password of the file named "data1" and corresponding interactive messages will be displayed for each subsequent file in the group. The PCOS prompt appears when the operation is complete.

To subsequently remove this password, enter

```
fd datadisk:data*/rst /CR/
```

and an interactive prompt for each file in the group appears thus:

```
Delete Password of 0:data1?
```


A response of "y" deletes the corresponding password and the prompt reappears for the next file in the group. The PCOS prompt appears when the operation is complete.

To remove passwords in this way the password must be common to all the files in the group.

Note: The dialogue in the above examples will be suppressed if the no-interaction (%n) flag is used.

WRITE-PROTECTION

Applying and Removing File Write-Protection

To apply write-protection to a file enter the FWPROT command along with the file identifier. For example, to write-protect a file named "masterfile" on the diskette inserted in drive 1 enter

```
fw 1:masterfile /CR/
```

To subsequently remove write-protection requires the FUNPROT command to be entered along with the file identifier. For example, to remove write-protection from "masterfile" enter

```
fu 1:masterfile /CR/
```

Applying and Removing Write-Protection to a Group of Files

To write-protect a group of files, for example, all files beginning with "index" on the diskette inserted in drive 1, enter

```
fw 1:index* /CR/
```

The system displays a prompt for each file in turn in the specified group asking you to confirm or decline write-protection

```
Set WP 1:index1?
```

Respond "y" and write-protection is set for the file named "index1"

```
Set WP 1:index3.5?
```

A "y" response sets write-protection for the file named "index 3.5", etc.

To remove write-protection from the above group of files, enter

```
fu 1:index* /CR/
```

The system responds with a sequence of interactive prompts requesting confirmation for each file in the group. The PCOS prompt is displayed when the operation is complete.

Note: The dialogue in the examples shown will be suppressed if the no-interaction flag (%n) is used.

FREEING UNUSED FILE BLOCKS

This section describes how to free blocks that are allocated to files but not used.

To free unused blocks of a file, group of files, or an entire volume requires the FFREE command. Before this command can be executed the volume containing the files to be worked on must be present in one of the drives. It must also be enabled and must not be write-protected.

Before performing the FFREE command take a look at the contents of the volume to be worked on. For example, if you enter

```
v1 1: /CR/
```

then a volume list of the contents of the diskette inserted in drive 1 will be displayed and will look something like the following:

VOLUME:	1 mydisk	SECTORS			WR-PROT/
	BYTES	USED	ALLOCATED	EXTENTS	PASSWORD
kfile	3589	15	16	1	WP
kfile1	3585	15	19	1	WP
asd	1687	7	14	1	/PW
sdf	1687	7	14	1	
qwe	1687	7	14	1	
basfile	32512	127	150	1	
TOTALS		178	227	6	
6 FILES		FREE SECTORS ON DISK = 861			

Fig. 10-4 Sample Volume List Before Freeing Unused Blocks

To free unused blocks in the entire volume perform the following:

Enter

```
ff 1: /CR/
```

And the system will respond

You may not change disks while FFree in progress. Continue?

Respond by entering

y /CR/

and the display shown in Figure 10-5 will result.

```

kfile . . . . . write-protected
kfile1 . . . . . write-protected
asd . . . . . ERROR 73 --- invalid password

sdf . . . . . 6 block(s) freed
qwe . . . . . 6 block(s) freed
basfile . . . . . 22 block(s) freed

FFree complete -- 34 block(s) freed
1>

```

Fig. 10-5 Example of a Display Produced by an FFREE Command

The unused blocks allocated to "kfile" and "kfile1" are not freed because these files are write-protected. "asd" did not have unused blocks freed because it is password-protected and did not have its password specified.

To free the unused blocks from file "asd" another FFREE command is necessary, specifying the complete file identifier including the password. That is, if you enter

```
ff 1:asd/pass /CR/
```

the system will display

You may not change disks while FFree in progress. Continue?

Respond by entering

y /CR/

The unused blocks will be freed and the system will display

```

asd . . . . . 6 block(s) freed

FFree complete -- 6 block(s) freed
1>

```

To free the unused blocks in files "kfile" and "kfile1" you must first

remove the write-protection by entering

```
fu 1:k* /CR/
```

After answering "y" to the subsequent confirmation prompts, free the unused blocks by entering

```
ff 1:k* /CR/
```

The system will respond

```
You may not change disks while FFree in progress. Continue?
```

Respond by entering

```
y /CR/
```

and the system will display

```
kfile . . . . . 0 block(s) freed
kfile1 . . . . . 3 block(s) freed

FFree complete -- 3 block(s) freed
1>
```

The total effect of freeing the unused blocks can be seen by examining a volume list and comparing it to the previous one. That is, if you enter

```
vl 1: /CR/
```

the following will be displayed:

VOLUME:	1 mydisk	SECTORS			WR-PROT/
	BYTES	USED	ALLOCATED	EXTENTS	PASSWORD
kfile	3589	15	16	1	.
kfile1	3585	15	16	1	.
asd	1687	7	8	1	/PW
sdf	1687	7	8	1	.
qwe	1687	7	8	1	.
basfile	32512	127	128	1	.
TOTALS		178	184	6	.
6 FILES	FREE SECTORS ON DISK =			984	.

Fig. 10-6 Sample Volume List After Freeing Unused Blocks

Finally, write-protection should be reapplied to files "kfile" and "kfile1".

DELETING AND RECOVERING FILES

Deleting a file implies freeing the disk space occupied by that file for use by other files. The file data, however, is not deleted until the space it occupied is actually overwritten. This makes it possible to recover a deleted file so long as the file has not yet been overwritten and the volume has not been alphabetized.

DELETING FILES

To delete a file or group of files requires the FKILL command. Before this command can be executed the following conditions must apply:

- The volume containing the file or files to be deleted must be inserted in one of the drives. It must also be enabled and not write-protected.
- The file or files to be deleted must not be write-protected.

To delete a file from a volume, enter the FKILL command along with the file identifier, including any necessary volume identifier and/or password. For example, to delete a file that has the name "myfile" and password "mine" from the diskette inserted in drive 1 enter

```
fk 1:myfile/mine /CR/
```

A list of files can be specified for deletion. For example, to delete the (unprotected) files "myfile" and "yourfile" from the diskette inserted in drive 0 enter

```
fk 0:myfile,yourfile /CR/
```

A killed file is not actually deleted from the diskette, but it can no longer be accessed by any command other than the RKILL command. Furthermore, its name will no longer appear in a volume list as it is erased from the directory, and the space it occupied will be available for other files.

Groups of files can be deleted using the wild card feature. For instance to delete from the diskette inserted in drive 0 all the files beginning with "k" you must enter

```
fk 0:k* /CR/
```

The system will respond

```
Delete 0:kfile?
```

which gives you the option whether or not to delete this particular file. If you enter

y

the file is deleted and the system responds

```
Delete 0:kfile1?
```

and so on through the complete list of files beginning with "k", after which the PCOS prompt appears. **Note:** The dialogue can be bypassed by specifying the no-interaction (%n) flag.

RECOVERING DELETED FILES

To recover a deleted file requires the RKILL command. For this command to be executed successfully the following conditions must apply:

- The volume containing the file to be recovered must be inserted in one of the drives.
- The deleted file to be recovered must still be intact. That is, it must not have been fully or partially overwritten.
- The volume must not have been alphabetized since the file was killed.

To recover a file requires you to enter the RKILL command along with the file identifier. It is not necessary to enter the password. For example, to attempt to recover the deleted file "myfile" on the diskette inserted in drive 1 enter

```
rk 1:myfile /CR/
```

If the recovery is successful, the system will respond

```
File Successfully Repaired
```

If the recovery was not successful then the following error message is displayed:

```
ERROR 53 --- file not found in parameter 1
```

Note: It is not possible to use the wild card facility. Neither is it possible to specify a list of files.

RENAMING FILES

Renaming a file requires the FRENAME command. Before this command can be executed the file to be renamed must reside on a volume that is currently inserted in one of the drives.

To rename a file requires you to enter the file identifier along with the new file name. The file must not be write-protected.

For example, to change to "oldfile" the name of a file named "newfile" with password "npass" that is resident on a disk named "datadisk" enter

```
fr datadisk:newfile/npass,oldfile /CR/
```

Note: FRENAME has no effect on the password.

SECRET

1. The purpose of this document is to provide information regarding the activities of the organization in the field of international relations.

2. The organization has been active in the field of international relations since its inception in 1945.

3. The organization has been active in the field of international relations since its inception in 1945.

4. The organization has been active in the field of international relations since its inception in 1945.

5. The organization has been active in the field of international relations since its inception in 1945.

1. The PCOS system is designed to provide a secure and reliable means of communication between the command and control elements of the fleet. It is capable of handling a large volume of data and is designed to be highly resistant to electronic attack.

11. PCOS GRAPHIC AND CONSOLE-RELATED FACILITIES

11.1. Overview

11.1.1. Introduction

11.1.1.1. Purpose

11.1.1.2. Scope

11.1.1.3. Objectives

11.1.1.4. Organization

11.1.1.5. Summary

11.1.2. System Architecture

11.1.2.1. Hardware

11.1.2.2. Software

11.1.2.3. Data Flow

ABOUT THIS CHAPTER

This chapter describes the graphic and console-related facilities that are available within the PCOS environment. For further details of the commands mentioned in this chapter, refer to Chapter 14.

CONTENTS

<u>INTRODUCTION</u>	11-1
<u>DISPLAYING LABELS</u>	11-1
<u>PRINTING THE SCREEN IMAGE</u>	11-3
USING THE SPRINT AND LSCREEN COMMANDS	11-4
<u>ENTERING CHARACTERS AT THE KEYBOARD</u>	11-5
RAW KEY CODES	11-5
ASCII TABLES	11-6
INTERPRETATION OF ASCII TABLE OUTPUT	11-8
<u>CREATING USER-DEFINED FONTS</u>	11-10
CREATING A FONT MATRIX FILE USING THE RFONT COMMAND	11-11
THE FONT MATRIX FILE	11-11

USING THE WFONT COMMAND 11-14

DISPLAYING CONTROL CHARACTERS 11-15

THE LINE TERMINATION KEYS 11-19

Faint, illegible text at the top of the page, possibly a header or title.

Faint, illegible text below the first line, possibly a subtitle or a second line of a header.



INTRODUCTION

PCOS allows you to set the display mode to either 256 x 512 pixels (64 columns x 16 lines) or 256 x 480 pixels, (80 columns x 25 lines) by means of the SSYS command, and to allocate space for a certain number of windows using the SBASIC command. Moreover, PCOS contains a number of graphic and console-related facilities that enable you to:

- Display a label. That is, to display a string of characters of a specified magnification and orientation at a specified position, and within a specified window (LABEL command).
- Print out the entire contents of the video display or a specified window (SPRINT command).
- Print out just the textual content of a video display or a specified window (LSCREEN command).
- Modify the ASCII code generated on striking a specific key or key combination (CKEY command).
- Reconfigure the keyboard to simulate another standard national layout (SLANG command).
- Assign a string to a key or key combination (PKEY command).
- Create user-defined fonts (RFONT and WFONT commands).
- Display control characters.
- Distinguish, in BASIC, among the three line termination keys /↵/, /S1/ and /S2/ (LTERM command).

Note: The LABEL, SPRINT, and LSCREEN commands are often called from BASIC by a CALL or EXEC statement, as windows cannot be opened within the PCOS environment.

DISPLAYING LABELS

Labels can be displayed by using the LABEL command. This command enables you to specify the following features of the label to be displayed.

FEATURE	MEANING
the string	the text to be displayed. This can be any string of printable characters, included in quotation marks.
position	the x/y co-ordinates of the bottom left-hand corner of the first character of the label string with respect to the bottom left-hand corner of the screen or window. This is measured in pixels.
magnification	the number of times that the font character is magnified. This can be up to 16 times.
orientation	the number (0, 1 or 2) specifying the direction of the label string; that is, parallel to the x-axis left to right (0), parallel to the y-axis bottom to top (1), or top to bottom (2).
color	the color number in the range 0 to 7 for an eight-color display, 0 to 3 for a four-color display, or 0 or 1 for a black and white display. If omitted, the foreground color is assumed.

Using the LABEL command

If you enter

```
la 'LABEL',100,150,4,2 /CR/
```

then the string "LABEL" is displayed in the foreground color, at (100,150) four times the normal size, with the text rotated as follows:

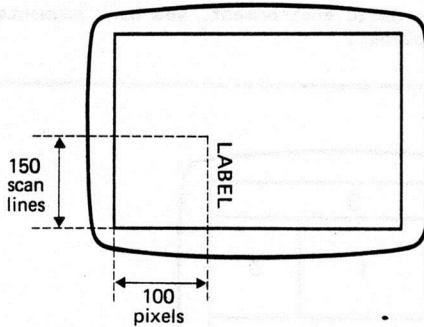


Fig. 11-1 Example of the LABEL command

PRINTING THE SCREEN IMAGE

There are two PCOS commands that perform this type of function. These are:

SPRINT This prints an image of all text and graphics displayed on the screen or within a specified window. With this command you are also able to specify the polarity of the printout, (that is, n for negative - black on white on printout for white on black on display, p for positive - white on black on printout for white on black on display, or c for color where the attached printer supports color printing - for example the PR 15B). You can also specify a title to appear at the top of the printout along with the date and time.

The SRRINT command can, however, only be used with printers that have graphic capabilities.

LSCREEN This prints just the text displayed on the screen or specified window. Graphic elements are ignored.

The LSCREEN command can be used with any PCOS-compatible printer

USING THE SPRINT AND LSCREEN COMMANDS

Supposing, when working in the BASIC environment, you have segmented the screen into five windows as follows:

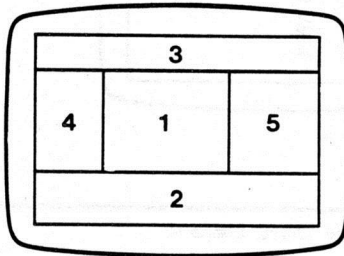


Fig. 11-2 Example of a Windowed display

IF you are in BASIC and you enter ...	THEN...
EXEC "sp 5,p,'SALES', dt" /CR/	the contents of window 5 will be printed with positive polarity beneath the heading "SALES" and the date and time (specified by dt)
EXEC "sp 0" /CR/	prints the contents of the entire screen
EXEC "ls 4" /CR/	prints the text contained in window 4. Any graphic elements are ignored

For details of how to define windows refer to the "BASIC-8000 User Guide".

ENTERING CHARACTERS AT THE KEYBOARD

Figure 11-3 illustrates what happens when you press a key.

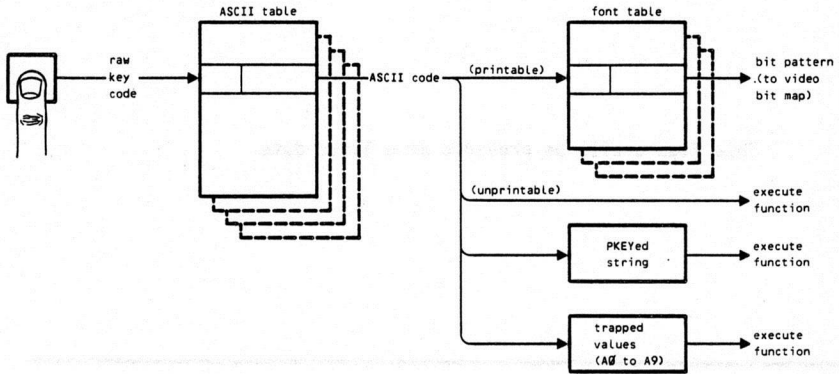


Fig. 11-3 Entering a Character at the keyboard

RAW KEY CODES

When a key is pressed it generates a raw key code (see Figure 11.4). This code depends on the physical position of the key on the keyboard; and on whether the key is pressed on its own, or in conjunction with the /SHIFT/ key, the /CTRL/ key, or the /COMMAND/ key. For example, if you press the /A/ key on the USA ASCII keyboard, then raw key code (hexadecimal) 02 will be generated. The same raw key code will be generated by pressing 'Q' on the French keyboard since the code depends on the physical position of the key, not the key inscription. Moreover, pressing this key in conjunction with the /SHIFT/ key will generate raw key code (hexadecimal) 32, with the /CTRL/ key will generate (hexadecimal) 62, and with the /COMMAND/ key will generate (hexadecimal) 92.

This figure will be provided at a later date.

Fig. 11-4 Raw Key Codes

ASCII TABLES

The raw key code points directly into an ASCII table that was loaded at initialization from the kb.all file. This table generates the appropriate ASCII code that corresponds to the inscription on the keytop. The ASCII table therefore varies from one national keyboard to another. For example, if you strike /A/ on the USA ASCII keyboard, raw key code 02 will be generated, which in turn will be converted by the USA ASCII table to (hexadecimal) 61 - the ASCII code for 'a'. Similarly, on the France keyboard, striking /Q/ will generate raw key code 02 which will be converted by the ASCII table corresponding to the French keyboard into the ASCII code for 'q' - (hexadecimal) 71. For a complete list see Appendix B.

The values of individual entries can be changed by use of the CKEY command. Moreover, the SLANG command can be used to replace the current ASCII table with one that corresponds to another national keyboard.

Using the CKEY command

Any entry in the current ASCII table can be redefined using the CKEY command. To do this you need to include two parameters in the CKEY command; the first defining the key or key combination to be redefined, and the second the ASCII code to be assigned to it. For example

```
ck &C3,8 /CR/
```

assigns ASCII code 8 (backspace) to the raw key code (hexadecimal) C3,

thereby enabling backspace to be performed by striking the S2 key. Note that values may be entered either in decimal (by entering the number on its own, or in hexadecimal) by preceding the number with &. Furthermore, the ASCII code can be specified simply by striking the corresponding key enclosed within quotation marks. For example

```
ck &72,'p' /CR/
```

will cause the ASCII code for 'p' to be generated when /CTRL/ /Q/ are pressed simultaneously.

The ASCII code assigned to a particular raw key code can be examined, again by use of the CKEY command by specifying just one parameter - the raw key code. For example, if you wish to examine the code assigned to the key combination /CTRL/ /C/, enter

```
ck &64 /CR/
```

then PCOS responds

```
KEY = 100
CODE = 162
```

The values displayed are decimal. That is, KEY specifies the raw key code (100 decimal being the equivalent of 64 hexadecimal), and CODE indicates the assigned code.

Using the SLANG Command

The ASCII table can be replaced at will by the ASCII table corresponding to another national keyboard by means of the SLANG command. Doing so also destroys any values assigned using the CKEY command. If you enter

```
sl /CR/
```

then a menu is displayed enabling you to select a national configuration by entering the appropriate number (see Chapter 6 for details). Alternatively, the selection can be made by entering the appropriate number as a parameter to the SLANG command. For example

```
sl 0 /CR/
```

selects the Italy keyboard. The result is that the ASCII table corresponding to the Italy keyboard is loaded from the kb.all file and overwrites the currently active table.

Making the Current ASCII Table Permanent

Any modifications made to the ASCII table by means of the CKEY command, or any ASCII table made active by means of the SLANG command remain active either until the current working session is terminated, or until further modified by a CKEY or SLANG command. However, such changes can be made a permanent feature of the operating system by means of the PSAVE

command (see Chapter 6); that is, any CKEYed values will become permanent, as will any ASCII table loaded by means of the SLANG command.

INTERPRETATION OF ASCII TABLE OUTPUT

The output from the ASCII table is treated as follows:

- Values A0 to A9 (hexadecimal) are special cases. They are never placed in the keyboard buffer but are always trapped by the keyboard handler to perform the following functions:
 - . A0 - Logical Reset
 - . A1 - (reserved)
 - . A2 - Break facility
 - . A3 - Halt Display
 - . A4 - Cursor lock
 - . A5 - Shift lock
 - . A6 - Two zeros
 - . A7 - End of line (CR in keyboard buffer, zero in LTERM buffer)
 - . A8 - End of line (CR in keyboard buffer, '1' in LTERM buffer)
 - . A9 - End of line (CR in keyboard buffer, '2' in LTERM buffer)
 - . AA - Special function for DATEV keyboard
 - . AB - Special function for DATEV keyboard
 - . AC - Special function for DATEV keyboard
 - . AD - Screen Print
 - . AE - (reserved)
 - . AF - No operation
- For values that have a PKEYed string assigned to them the value is placed in the buffer and the corresponding function is subsequently executed.
- For unprintable ASCII values (codes (hexadecimal) 00 to 1F and 7F) the code is placed in the keyboard buffer and the corresponding function is subsequently performed.
- For printable ASCII characters other than those that have strings assigned by means of the PKEY command (codes (hexadecimal) 20 to 7E plus any additional CKEYed values) the code is placed in the keyboard

buffer and the corresponding entry in the currently active font table is accessed and the bit pattern is written to the video bit map - see the section entitled "Creating User-Defined Fonts".

USING THE PKEY COMMAND

Any code output from the ASCII table (with the exception of the special cases A0 to AF) can have a string assigned to it by means of the PKEY command, thereby ensuring that the original function is not destroyed.

Assignment is made by passing parameters to the PKEY command, the first of which specifies the code as output from the ASCII table, and the second and subsequent parameters define the string to be assigned.

The ASCII code can be specified either as a decimal value on its own, a hexadecimal value preceded by an ampersand (&), or in the case of an ASCII code that already has a corresponding font defined in the currently active font table, the character can be entered directly from the keyboard, but enclosed within quotation marks. For example

```
'B'
```

```
66
```

```
&47
```

all refer to the same key.

Similarly, the strings to be assigned to the ASCII code can be specified as either the actual characters enclosed within quotation marks, or the ASCII code for each character, or a combination of the two. For example

```
'ba',13
```

is a valid string representing 'ba' followed by a carriage return.

Suppose that you wish to assign the string 'FILES "1:"',13 to the key combination /COMMAND/ /!1/ (ASCII code hexadecimal ED, decimal 237). Do this by entering

```
pk 237,'FILES "1:"',13 /CR/
```

Thus when in the BASIC environment, the key combination /COMMAND/ /!1/ can be used to list the directory on the diskette inserted in drive 1.

String assignments made in this manner are valid up to the end of the current working session. However, such assignments can be made a permanent feature of the operating system by means of the PSAVE command as described in Chapter 6.

You can display a list of programmed keys along with the string assignments by entering

```
pk /CR/
```


PCOS will typically respond:

Code	Char	String
35	#	ba, 13, 10, files, 13, 10
237		FILES *1:", 13
(Press any key to exit)		

Fig. 11-5 Sample Display of Programmed Keys

Note that the code is given in decimal.

An individual key assignment may be canceled by entering the ASCII code as a single parameter to the PKEY command. For example

```
pk 237 /CR/
```

Moreover, all assigned strings may be canceled by entering

```
pk %c /CR/
```

For details of the effect string assignments have on user memory, refer to Chapter 6.

CREATING USER-DEFINED FONTS

The RFONT command creates a graphic representation of the active font set and stores it in a font matrix file. This file can be edited using the Video File Editor (see Chapter 13) to redefine the shape of existing characters and add new characters. Once the editing session is complete you can invoke the WFONT command to cause the system to display character fonts as they appear in the edited file. Thus you can create customized font sets, each of which is stored on diskette (or hard disk) in a separate file, and can be selected to become the active font set.

CREATING A FONT MATRIX FILE USING THE RFONT COMMAND

The RFONT command is invoked by entering, for example

```
rf 1:myfont /CR/
```

The specified file will be created if it does not exist. If the file already exists, the following prompt will appear:

```
File Already Exists. Do You Wish to Overwrite? (y/n)
```

A "y /CR/" response causes the existing file to be overwritten.

THE FONT MATRIX FILE

A font matrix file must be structured as defined below. A file created by the RFONT command is of this structure.

At the beginning of a font matrix file is a four-line header. All four lines must be present, although only the fourth line is actually read by the WFONT command. The header is defined as follows:

- line 1: Any text that describes the file (for example, the national keyboard that the file corresponds to). It is for your reference and is ignored by the WFONT command.
- line 2: The country number that was active when the file was created by the RFONT command. This number corresponds to the particular national keyboard. For details refer to the SLANG command. The content is ignored by the WFONT command.
- line 3: The height (in lines) of a valid font matrix. This must always be 10.
- line 4: The character count followed by at least one other word (for example, "characters"). The count must match the number of font matrices that follow. The minimum is 95 characters and the maximum is 190.

Example:

```
USA
country 4
matrix height = 10
95 characters
```

Each matrix is defined as follows:

- line 1: A decimal code representing the character defined in the matrix. (For standard fonts this will be the ASCII code.) Its value is for your reference and is not read by the WFONT command, but it must be present.
- lines 2 to 11: A matrix, ten lines down and eight characters wide, made up of ' - ' and (upper case) 'X' characters.

Example:

```
50
-----
----XXX-
---X---X
-----X
-----X-
-----X-
-----X-
----X---
---XXXXX
-----
-----
```

The correspondence between a font matrix and the ASCII code generated by a particular key on the keyboard depends on the position of the font matrix within the file. That is, the first font matrix will correspond to /SPACE/ - the first printable ASCII character (ASCII code 32) - the second to /!/ (ASCII code 33) and so on up to the 95th entry which will correspond to /±/ - the last printable ASCII character (ASCII code 126).

Redrawing Existing Characters

To redraw a character, invoke the Video File Editor and place 'X's so that they show the intended appearance of the character.

In 64 x 16 display mode, the leftmost three columns are generally reserved for spacing between characters and should not be used except in special cases where regular spacing between characters is not desired. In 80 x 25 display mode, the leftmost two columns are not displayed at all, and the third column should be left blank unless joined characters are desired.

Once the edited file is saved and the Video File Editor exited, the new font can be made active by means of the WFONT command.

Defining Additional Characters

Characters can be added to a font matrix file by incrementing the character count (line 4 of the header) and adding matrices to the end of the file. However, matrices beyond the 95th, (that is, those corresponding to codes 127 to 222) do not necessarily have a corresponding key-generated ASCII code. You must therefore define a key combination for each additional matrix using the CKEY command.

Example:

To add a font matrix, defining the character ' ', to the standard font matrix file and assign key combination /COMMAND/ /Q/ to it:

1. Using the Video File Editor update the character count (line 4 of the header) to 96.
2. Using the Video File Editor add a font matrix to the end of the file, thus:

```

127
-----
---XXX-
---X---X
---X--XX
---X-X-X
---XX--X
---X---X
---XXX-
-----
-----

```

3. Assign ASCII code 127 to the key combination /COMMAND/ /Q/ using the CKEY command:

```
ck &A2, 127 /CR/
```

This assigns code 127 to raw key code (hexadecimal) A2, which is the raw key code generated when /COMMAND/ /Q/ are pressed simultaneously.

Once the edited font has been made active by means of the WFONT command, the key combination /COMMAND/ /Q/ will display ' '.

Note: It is possible to remove or insert font matrices within the first 95 characters, but this would offset the matrix/key correspondence following the first matrix to be removed/inserted.

USING THE WFONT COMMAND

The WFONT command takes one parameter which is the name of a font matrix file. After execution, the font defined by that file becomes active.

Invoke the WFONT command by entering, for example

```
wf 1:myfont /CR/
```

Assuming the specified font matrix file is on an active volume, and that enough memory is available, the font matrices will be read, converted to binary, and written into memory. Once execution is completed, the new fonts will replace those currently known to the system. The system will return to the fonts known at initialization when reinitialization occurs, or when the WFONT command is invoked with no parameter.

The WFONT command allocates user memory each time it is invoked with a valid font file. This memory is released either by reinitializing PCOS, or by invoking the WFONT command with no parameter. In order to save memory, it is advisable to release space allocated by the WFONT command before activating another user-defined font. The effect on memory can be determined using the DCONFIG command.

User-defined fonts can be made permanent by means of the PSAVE command.

Example

The following example assumes the existence of two user-defined font matrix files named "myfont" and "yourfont." Both files are located on the diskette inserted in drive 1.

If you enter...	THEN...
wf 1:myfont /CR/ . . .	the font defined by the font matrix file named "myfont" is made active
wf. /CR/ . . .	"myfont" is removed from memory and the font that was active at initialization is again made active

<pre>wf 1:yourfont /CR/ . . . wf 1:myfont /CR/ . . .</pre>	<p>the font defined by the font matrix file named "yourfont" is made active</p> <p>"myfont" is again made active, but data for "yourfont" is not removed from memory</p>
---	--

The effect on user memory is illustrated in Figure 11-6

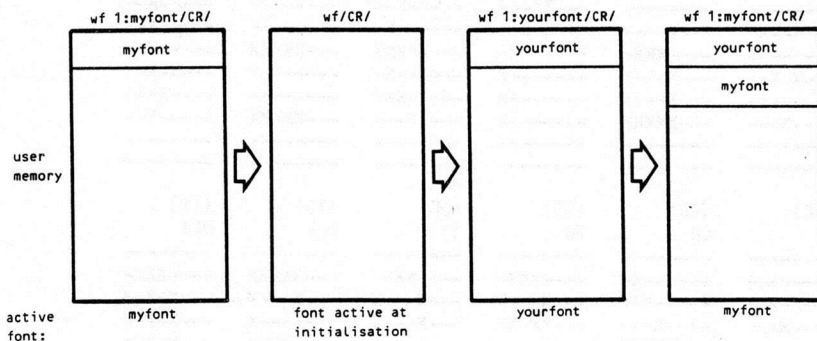


Fig. 11-6 Effect of the WFONT command on user memory

DISPLAYING CONTROL CHARACTERS

When displaying files or displaying data received from a communications line, you may wish to have a visual representation of the ASCII control characters (hexadecimal 00 to 0F). PCOS contains a facility whereby such characters can be represented by the character font definitions illustrated in Figure 11-7. Above each matrix in the figure is the corresponding hexadecimal ASCII code in parentheses, and the control function.

(00) NUL	(01) SOH	(02) STX	(03) ETX	(04) EOT	(05) ENQ
-----	-----	-----	-----	-----	-----
---XXXXX	---XXXXX	-----X-	-----X	---X---	---XXXXX
---XXXXX	---X---	-----X-	-----X	-----X--	---X---X
---XXXXX	---X---	-----X-	-----X	-----X-	---XX-XX
---XXXXX	---X---	-----X-	-----X	-----X-	---X-X-X
---XXXXX	---X---	-----X-	-----X	---X---	---XX-XX
---XXXXX	---X---	-----X-	-----X	-----X-	---X---X
---XXXXX	---X---	---XXXXX	---XXXXX	-----X-	---XXXXX
-----	-----	-----	-----	-----	-----

(06) ACK	(07) BEL	(08) BS	(09) HT	(0A) LF	(0B) VT
-----	-----	-----	-----	-----	-----
-----X	-----	---XXX-	---X---	---XXXXX	---X---
-----	-----	---XX-	---X---	-----	---X---
-----X-	-----	---X-X-	---X---	-----	---X---
-----	---XXX-	---X-	---XXXXX	---XXXXX	---X---
---X-X-	---X-X	-----X	---X---	-----	---X-X-X
-----	---X-X	-----X	---X---	-----	---XXX-
---X-	---XXXXX	-----X	---X---	---XXXXX	---X---
-----	-----	-----	-----	-----	-----

(0C) FF	(0D) CR	(0E) SO	(0F) S1	(10) DLE	(11) DC1
-----	-----	-----	-----	-----	-----
---X-	---X-	---XXX-	---XXX-	---XXXXX	---XXX-
---X-X-X	---X-	---X-X-X	---X-X-X	---X-X-X	---X-X-X
---XXX-	---X-	---XX-XX	---X-X-X	---X-X-X	---X-X-X
---X-	---XXXXX	---X-X-X	---X-X-X	---XXXXX	---X-XXX
---X-X-X	---X-	---XX-XX	---X-X-X	---X-X-X	---X-X-X
---XXX-	---X-	---X-X-X	---X-X-X	---X-X-X	---X-X-X
---X-	---XXX-	---XXX-	---XXX-	---XXXXX	---XXX-
-----	-----	-----	-----	-----	-----

(12) DC2	(13) DC3	(14) DC4	(15) NAK	(16) SYN	(17) ETB
-----	-----	-----	-----	-----	-----
---XXX-	---XXX-	---XXX-	-----X	---XXX-	-----
---X-X-X	---X-X-X	---X-X-X	-----	---X-X-	-----X
---X-X-X	---X-X-X	---X-X-X	---XXX	---X-X-	-----X
---X-XXX	---XXX-X	---XXX-X	-----	---X-X-	---XXX-
---X-X-X	---X-X-X	---X-X-X	---X-X-	---X-X-	---X-
---X-X-X	---X-X-X	---X-X-X	-----	---X-X-	---X-
---XXX-	---XXX-	---XXX-	---X-	---XX-XX	-----
-----	-----	-----	-----	-----	-----

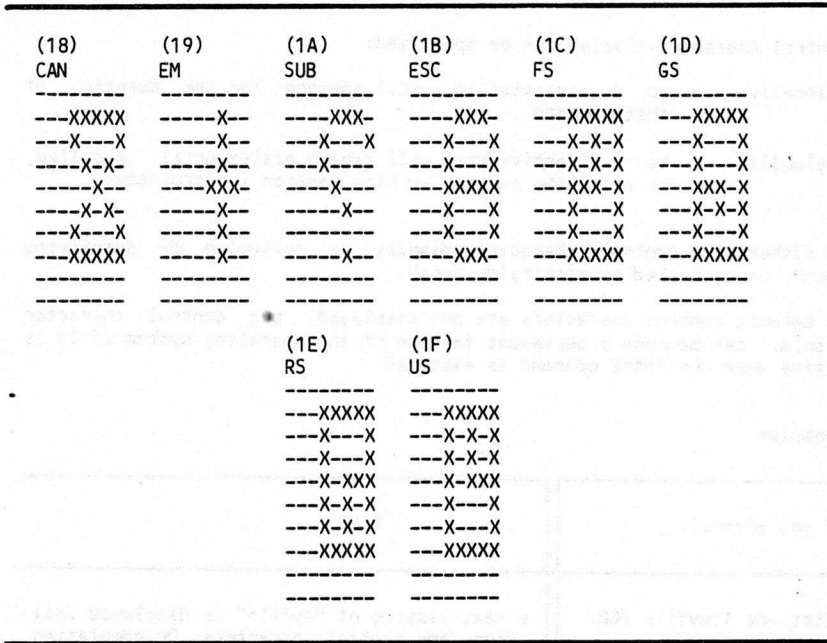


Fig. 11-7 Control Character Font Matrices

Control characters that produce a visible/audible effect require special explanation. These are described in the following table:

CONTROL CHARACTER	COMMENTS
TAB (09) LF (0A) FF (0C) CR (0D)	Any of these characters will simply appear as a single character on the screen without performing the corresponding visible effect. That is, characters will 'wrap around' the screen as a continuous line
BEL (07)	The bell will sound and the character will appear on the screen

Tab. 11-8 Control Characters that Produce a Visible/Audible Effect

Control character display can be specified:

- locally as a parameter to a PCOS command for the duration of that command
- globally as a directive that will remain active until cancelled, or until the current working session is terminated.

In either case control character display is activated by specifying "+cc", or cancelled by specifying "-cc".

By default control characters are not displayed, but control character display can be made a permanent feature of the operating system if it is active when the PSAVE command is executed.

Examples

IF you enter...	THEN...
<code>flist +cc 1:myfile /CR/</code>	a text listing of "myfile" is displayed including any control characters. On completion, control character display is cancelled
<code>+cc /CR/</code>	control character display is activated globally; that is, either until respecified, or until the end of the current working session
<code>vq -cc /CR/</code>	<ol style="list-style-type: none">1. If control character display is set globally, it will be cancelled for the duration of the VQUICK command. On completion of the command, control character display will be reactivated.2. If control character display is not set globally, "-cc" will have no effect.
<code>-cc /CR/</code>	control character display is cancelled until respecified

THE LINE TERMINATION KEYS

For most operations the three line termination keys perform the identical function of placing the ASCII code for carriage return (08) in the keyboard buffer, irrespective as to whether the key is struck on its own or in conjunction with one of the keys /SHIFT/, /CTRL/ or /COMMAND/. Alternative functions may be assigned by means of the CKEY command by assigning different ASCII codes to the raw key codes generated; but note that the PKEY command can only assign a string to the ASCII code placed in the keyboard buffer and therefore cannot assign unique strings to each of the keys (unless unique ASCII codes have first been assigned using the CKEY command).

Regardless of the CKEY command, PCOS contains a facility whereby the three line termination keys can be distinguished among the three from within a BASIC program. In addition to placing the ASCII code in the keyboard buffer, striking one of these keys also places a unique code in a special (LTERM) buffer (0, 1 or 2 for /↵/, /S1/ or /S2/, respectively). This buffer can then be interrogated from BASIC using the LTERM command to distinguish which of the three keys was last pressed. This can be useful in a situation where a BASIC program prompts the user for an entry of some sort. The LTERM command can subsequently be CALLED to return the current value of the LTERM buffer, in order to process the entry in one of three ways depending on which of the three line termination keys was used to terminate the entry.

THE UNIVERSITY OF CHICAGO

The first part of the book is devoted to a general introduction to the subject of the history of the United States. The author discusses the various factors which have influenced the development of the country, and the role of the individual states in the formation of the national government. He also touches upon the economic and social conditions of the early years of the Republic.

The second part of the book is devoted to a detailed study of the political and social changes which took place during the period of the American Revolution. The author examines the causes of the Revolution, the course of the war, and the establishment of the new government. He also discusses the impact of the Revolution on the development of the United States as a nation.

12. FACILITIES RELATED TO ALTERNATIVE- OPERATING SYSTEMS

ABOUT THIS CHAPTER

This chapter describes the PCOS commands that are related to the CP/M-86 and MS-DOS operating systems.

Throughout this chapter, the availability of commands is always assumed. That is, it is assumed that either a volume containing the command is present in one of the drives, or that the command in question is already resident in memory.

CONTENTS

<u>INTRODUCTION</u>	12-1
<u>HOW TO PREPARE THE HARD DISK FOR USE WITH ALTERNATIVE OPERATING SYSTEMS</u>	12-1
HOW TO PARTITION THE HARD DISK	12-2
HOW TO INITIALIZE EACH PARTITION	12-4
HOW TO CHECK THE BAD BLOCK LIST	12-4
HOW TO UPDATE THE BAD BLOCK LIST	12-5
<u>HOW TO COPY THE OPERATING SYSTEM TO THE HARD DISK</u>	12-5

INTRODUCTION

The CPU on the motherboard of your machine, as distinct from the Z8000 softcard which enables you to use PCOS, gives you the options of working with other operating systems such as Concurrent CP/M-86 and MS-DOS. These operating systems provide working environments that are independent of PCOS and are described in the "Concurrent CP/M-86 User Guide" and the "MS-DOS User Guide" respectively.

HOW TO PREPARE THE HARD DISK FOR USE WITH ALTERNATIVE OPERATING SYSTEMS

The hard disk, as it arrives from the factory, has been physically formatted. Before it can be used, a partition map must be created and the partition for each desired operating system must be logically formatted. The partition map can be created either under MS-DOS using the FDISK command, or under PCOS using the HDISK command. Logical formatting of the partition for a particular operating system must be done under that operating system. In addition, any bad blocks or bad sectors on the hard disk must be detected and marked as unavailable.

Instructions for preparing the hard disk for use under MS-DOS and Concurrent CP/M-86 are found in the respective User Guides for these operating systems. To prepare a new hard disk for use with PCOS, do the following:

1. Boot PCOS and run the PCOS HDISK command to create the PCOS partition. If a partition map has already been set up under MS/DOS with FDISK, HDISK will display it and can add an entry for PCOS.
2. Run VVERIFY "destructively", that is, using the %d option. This will disable any bad blocks which may have been on the disk when it came from the factory.
3. Run the PCOS VNEW command to initialize the PCOS partition as a PCOS volume with an empty directory.
4. Run the PCOS VVERIFY command, which will detect any bad blocks in the PCOS partition and mark them as unavailable.

Note: When running either HDISK or FDISK for the first time in order to create a partition map, be sure to leave sufficient free space for any other operating system(s) you plan to have resident on the HDU.

If errors are encountered during subsequent hard disk operation, the procedure for marking the new error blocks is as follows:

1. Run the PCOS HBACKUP command (described in Chapter 9). This will back up the files from the hard disk onto diskettes.
2. Run the PCOS VVERIFY command destructively to detect, mark, and disable the bad blocks.
3. Run the PCOS VNEW command to reinitialize the hard disk.

4. Run the PCOS HRESTORE command (also described in Chapter 9) to replace the hard disk data.

HOW TO PARTITION THE HARD DISK

The hard disk can be split into up to 4 partitions. You must now decide how many partitions you require and partition the hard disk accordingly. To use the hard disk under PCOS, MS-DOS and Concurrent CP/M-86 you will need to create partitions for MS-DOS and Concurrent CP/M-86 as well as PCOS. This requires limiting the size of the PCOS partition so as to allow sufficient space for these.

This is the procedure for creating a PCOS partition on the hard disk.

Insert the diskette containing the HDISK command into drive 0. Enter

```
hd /CR/
```

and the following menu will be displayed:

```
Hard Disk Setup Program 1.00  
HDISK Options
```

Choose one of the following

- 1 Create PCOS Partition
- 2 Delete PCOS Partition
- 3 Display Partition Information

Enter choice.....[1]

ESC to return to PCOS.....[]

The cursor will blink over the "1", which is the default choice. To create a partition, you may enter 1 followed by /CR/ or simply /CR/. The display will then show:

Partition Status Type Start End Size

No partition defined

Total disk space is 305 cylinders

Maximum available space is 306
cylinders at cylinder 0.

Do you wish to use total space
available for PCOS (Y/N).....[Y]

ESC to return to HDISK Options.....[]

The cursor will blink over the "Y", which is the default choice, but remember that if you wish to have partitions for other operating systems on the hard disk, you must limit the PCOS space.

If you enter "n", you will be prompted for partition size and starting cylinder number. After you have entered these numbers, this line will be displayed:

Clearing boot and control tracks

replaced shortly by

Creating bad block list

and finally

Partition created

The partition status information is then also displayed in the status line above. If you invoke HDISK subsequently, this status information will also be displayed.

Options 2 and 3 work similarly, with prompts to advise you what to do at each step.

If, after creating the PCOS partition, you hit /ESC/ to return to PCOS, this message will be displayed:

* YOU MUST RUN VNEW FOR PCOS TO OPERATE PROPERLY *

Run "VNEW 10:"
Exit HDISK
0>

HOW TO INITIALIZE EACH PARTITION

To initialize the MS-DOS partition of the hard disk you must boot MS-DOS from diskette, run FDISK to create an MS-DOS partition, and run the MS-DOS FORMAT command. Refer to the "MS-DOS User Guide" for details.

To initialize the Concurrent CP/M-86 partition you must boot CP/M-86 from diskette and run DSKMAINT to create a Concurrent CP/M-86 partition. Refer to the "Concurrent CP/M-86 User Guide" for details.

To initialize the PCOS partition use the VNEW command as follows

```
vn 10: /CR/
```

You will be prompted:

```
Warning- vnew deletes all files. Initialize disk? (y/n)
```

When you enter "y", VNEW will do its work silently without further advice, and then the hard disk prompt will return:

```
10>
```

HOW TO CHECK THE BAD BLOCK LIST

The bootstrap loader, system tables, directory, etc., of PCOS will reside in the first two cylinders of its partition. It is therefore necessary to check that there are no bad blocks in these areas. If you enter

```
vv %p 10: /CR/
```

then any bad blocks will be displayed in the following format:

```
Volume Verify Rev. 1.00  
PCOS partition begins at cylinder: 1  
PCOS partition ends at cylinder: 20
```

```
Bad Sector List cylinder(word) side(byte) sector(byte)
```

The bad sector information is in hexadecimal. The beginning and ending

cylinder numbers, of course, will correspond to what you specified when you partitioned the hard disk.

NOTE:

The M24 HDU has 17 blocks of 516 bytes each per track. PCOS addresses each physical block as 2 logical sectors of 256 bytes each. Since the VVERIFY command is searching for "physically" corrupt blocks on the disk, the number of "bad blocks in use" is the number of bad physical blocks in use. In actual fact, this figure is half the number of deallocated sectors, because when one physical block is deallocated, two PCOS logical sectors are deallocated. A VLIST.CMD or VQUICK.CMD, therefore, will show a reduction in the amount of free sectors available of 2 for each 1 block deallocated by VVERIFY.CMD.

HOW TO UPDATE THE BAD BLOCK LIST

If bad blocks should develop within files after you have been using the PCOS partition of the hard disk for a while, you will need to run VVERIFY "destructively" in order to add these bad blocks to the bad block list. It may or may not be possible to back these files up before running VVERIFY, depending on whether the floppy disk controller is able to successfully read the data from the bad sector during the backup phase. To perform this backup, it is safest to use HBACKUP, as you may have single files on the hard disk which are larger than the capacity of a diskette. FCOPY may be used when the files specified are small enough to fit on a single diskette.

If you enter

```
vv %d 10: /CR/
```

then any bad blocks on the PCOS partition will be added to the bad block list. This process takes several minutes and destroys all data on the PCOS partition, including the bitmap.

Note: Using the %d option makes it necessary for you to run VNEW subsequently in order to reinitialize the PCOS partition.

HOW TO COPY THE OPERATING SYSTEM TO THE HARD DISK

After partitioning the hard disk you will probably want to copy your operating systems to the appropriate partitions.

To copy PCOS you will either

- restore your back-up diskettes to the hard disk using the HRESTORE command
- or copy the PCOS system diskette by inserting it in drive 0 and entering the command

fc 0:*,10: /CR/

or the command

ps 10: /CR/

To copy the MS-DOS system diskette to the MS-DOS partition you need the MS-DOS FORMAT command. Refer to the "MS-DOS User Guide" for details.

To copy the Concurrent CP/M-86 system diskette to the CP/M-86 partition you need the Concurrent CP/M-86 PIP command. Refer to the "Concurrent CP/M-86 User Guide" for details.

13. VIDEO FILE EDITOR.

ABOUT THIS CHAPTER

This chapter describes how files containing text can be edited using the Video File Editor.

CONTENTS

<u>INTRODUCTION</u>	13-1
THE DISPLAY	13-1
THE KEYBOARD	13-2
<u>HOW TO INVOKE THE VIDEO FILE EDITOR</u>	13-5
EDIT.COMD	13-5
<u>GENERAL EDITING FUNCTION KEYS</u>	13-7
<u>WINDOW MOVING FUNCTION KEYS</u>	13-14
<u>EXITING AND SAVING FUNCTION KEYS</u>	13-15
<u>COMMANDS AND SEARCHING</u>	13-16
STRING SEARCHES	13-16
COMMANDS	13-18
GOTO	13-18

DELETE

13-19

FILE

13-19

INTRODUCTION

The Video File Editor enables you to create and edit files of text. A text file is a file of records containing printable ASCII characters, and each record is separated from the next either by a carriage-return/line-feed pair or by the record separator character RS (hexadecimal 1E).

The Video File Editor displays a 21-line "window" within which you can perform editing functions via the keyboard. A subset of these functions enables you to move the window to access any part of the file.

In addition to the functions mentioned above, the Video File Editor can also perform an extensive set of line-editing and cursor-moving functions and can operate in overstrike, insert text, or command mode. The latter enables a subset of high-level commands.

Each text line can contain up to 80 characters.

THE DISPLAY

Once the Video File Editor has been invoked the computer produces a display such as the one shown in Figure 13-1.

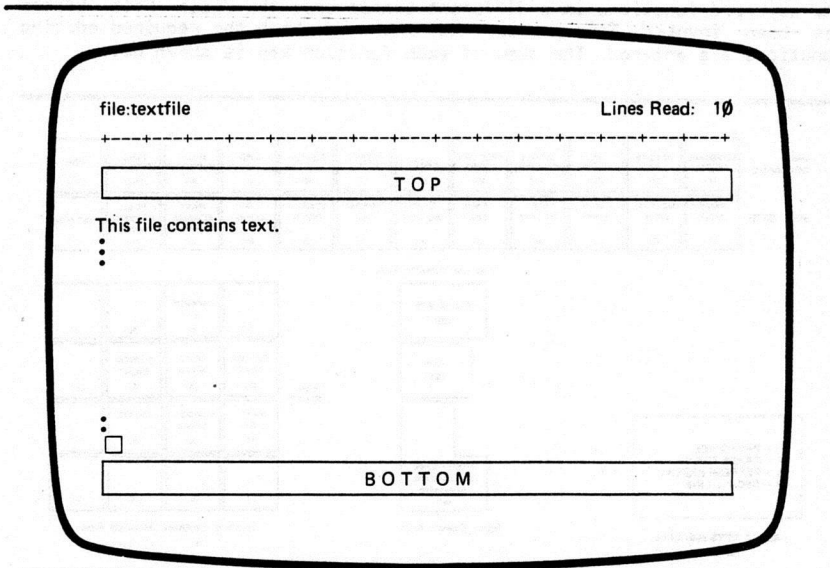


Fig. 13-1 Video File Editor Screen Layout

Line 1 indicates the file name and the current message.

Line 2 is used for high-level commands and search strings and is therefore used only when in command mode. Refer to the section entitled "Commands and Searching" for details.

Line 3 shows the tab stop settings (4 character positions per tab).

Lines 4 to 24 contain the text window.

Line 25 is not used.

On entering the Video File Editor, the beginning and end of the file are marked by two display lines containing the words TOP and BOTTOM, respectively. The former, known as the TOP bar, always appears immediately before the first line of text in the file. The BOTTOM bar always appears immediately after the last line of text. They are not actual lines of text and are there merely as markers. The cursor is positioned on the TOP bar.

The cursor changes shape when switching between certain modes of editing. It is represented here as underline.

THE KEYBOARD

The keyboard functions in a different manner once the Video File Editor has been invoked. This provides the means by which the required editing functions are entered. The name of each function key is shown below.

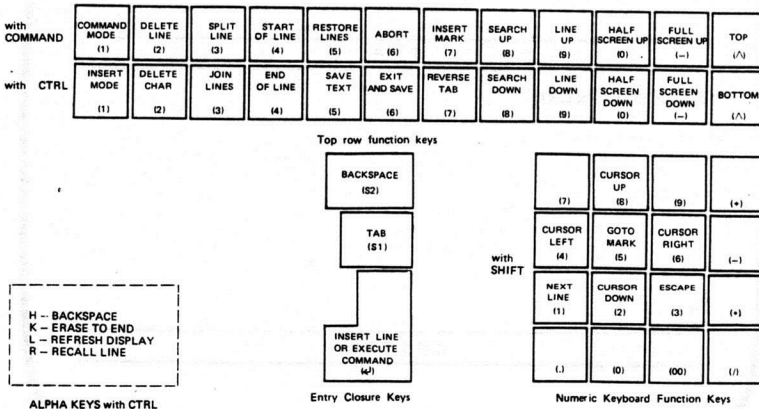


Fig. 13-2 Location of Video File Editor Function Keys

Note: Function keys F1 through F12 provide the same functions as the top row of typewriter-like keys on the USA ASCII keyboard. For example, /F1/ unshifted and /CTRL/ /1/ both enable insert mode; /SHIFT/ /F1/ and

/COMMAND/ /1/ both enable command mode. The values in parentheses shown in Figure 13-2 represent the appropriate key on the USA ASCII keyboard. Refer to Appendix B for other keyboard layouts.

The function keys are divided into six areas:

The Arrow Keys

Four arrow keys provide a set of functions for cursor motion. These functions are also available on the numeric keypad, for users accustomed to the M20 keyboard.

The Numeric Keypad

Pressing the SHIFT key in conjunction with keys on the numeric keypad provides a set of functions primarily for cursor motion, as shown in Figure 13-2. Keys 1, 3, and 5 have other functions and are described later. Note that the numeric keypad can be locked into shift mode by means of the /CTRL/ /?/ keys. Subsequently, the numeric values can be entered by means of the /SHIFT/ key. Return to unshifted mode can be made by entering /CTRL/ /?/.

The F-Key Row

Twelve F-keys F1 through F12 provide editing functions as shown in Figure 13-2. For users accustomed to the M20 keyboard, these same functions continue to be available on the top row of the typewriter-like keyboard.

The Top Row

Twelve of the top row of the typewriter-like keys are used in conjunction with the /COMMAND/ and /CTRL/ keys to provide 24 functions. These perform or enable most of the major editing operations such as moving the text window, saving text, inserting text, and switching between different modes of editing.

Alphabetic Keys

Some of the alphabetic keys when used with the /CTRL/ key provide additional functions.

Entry Closure Keys

These three keys are used for the most frequently used editing functions. They require no shift key.

Programming Function Keys

You are free to rearrange function keys at will (while in the PCOS environment) using either the CKEY or PKEY command. For example, if you prefer to have the INSERT MARK function assigned to the /CTRL/ /|/ key (bottom left on the USA ASCII keyboard) enter

```
pk &7F,&F3 /CR/
```

Moreover, frequently entered text strings can be assigned to a single key value by means of the PKEY command. However, YOU MUST TAKE CARE NOT TO DISABLE FUNCTIONS THAT YOU WILL REQUIRE AFTER ENTERING THE VIDEO FILE EDITOR. For example, if you assign some value to the key combination /CTRL/ /6/, then the corresponding edit function (EXIT AND SAVE) will be disabled. Also, all keys and key-combinations that share the same function will be affected by CKEY or PKEY changes to any of the keys. For example, if you change unshifted /F1/, the same change happens to /CTRL/ /1/.

HOW TO INVOKE THE VIDEO FILE EDITOR

EDIT.COMD

The EDIT command is used to enter the Video File Editor.

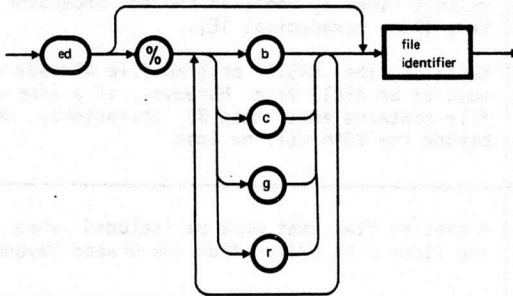


Fig. 13-3 EDIT

Where

SYNTAX ELEMENT	MEANING
b	A backup of the file is to be made when the Video File Editor is entered. This backup is named filename.bak where filename is the same as that specified in the command line

c	<p>The option which allows you to edit BASIC program files. This is necessary because BASIC files have a different format from standard text files. That is, they contain no tab characters (even if you enter TAB or REVERSE TAB) and have carriage-return/line-feed pairs between lines. Conversely, standard PCOS text files do have tab characters, moreover they delimit lines by means of record separator characters (RS = hexadecimal 1E).</p> <p>Note: if the BASIC program file already exists it must be an ASCII file. Moreover, if a line of such a file contains more than 80 characters, characters beyond the 80th will be lost</p>
g	<p>A special flag that must be included when invoking the Video File Editor from the Greece keyboard</p>
r	<p>The read-only option, used when you wish only to examine the contents of the file. This protects the file from accidental damage while examining it</p>
file identifier	<p>The name of the file to be edited plus any necessary password or volume identifier</p>

Characteristics

If the file does not already exist the prompt "OK to Create?" appears on the screen to which you must type "y" to create the file.

The Video File Editor is initially in "overstrike" mode. That is, you can enter text and overwrite whatever is already written on the file. The methods of entry into other modes of operation are described later.

The four optional flags (b, c, g, and r) can be specified in any order.

GENERAL EDITING FUNCTION KEYS

The keys whose functions are described below perform general editing functions such as moving the cursor and inserting and deleting text.

CLASS	FUNCTION KEY	MEANING
To move the cursor	/SHIFT/ /8/ (CURSOR UP) or /↑/	Moves the cursor one line up the screen but keeps the same position within the line. If the cursor was on the second line of the window then the window is moved one line up the file and the cursor remains on the second line
	/SHIFT/ /2/ (CURSOR DOWN) or /↓/	Moves the cursor one line down the screen but keeps the same position within the line. If the cursor was on the penultimate line of the window, it stays there and the window is moved down one line
	/SHIFT/ /4/ (CURSOR LEFT) or /←/	Moves the cursor one character position to the left within the same line
	/SHIFT/ /6/ (CURSOR RIGHT) or /→/	Moves the cursor one character position to the right within the same line
	/S1/ (TAB)	Moves the cursor one tab position (four characters) to the right
	/F7/ or /CTRL/ /7/ (REVERSE TAB)	Moves the cursor one tab position (four characters) to the left

	/SHIFT/ /4/ or /COMMAND/ /4/ (START OF LINE)	Moves the cursor to the start of the current line
	/F4/ or /CTRL/ /4/ (END OF LINE)	Moves the cursor to the character position immediately following the last nonspace character in the current line
To insert text	/F1/ or /CTRL/ /1/ (INSERT MODE)	Is entered from overstrike mode. The cursor changes its shape to show that a new mode has been entered. Any character which is subsequently entered is inserted immediately before the cursor position, and the remainder of the text in the line and the cursor are moved one character position to the right. Any character that was in the last character position in the line is discarded. Striking the INSERT MODE key a second time returns the Video File Editor to overstrike mode and the original cursor is restored
	/↵/ (INSERT LINE)	Inserts a blank line immediately after the current line and places the cursor at the beginning of that line. Subsequent text is pushed one line down the screen. If the cursor was already on the bottom line of the screen then the window is moved one line down the file and the blank line is inserted on the last line of the window
To delete text	/F13/ or /S2/ or /CTRL/ /H/ (BACKSPACE)	Moves the cursor one character position to the left and deletes the character under the cursor. Subsequent characters in the line do not move. The deleted characters are replaced with spaces.

		This function is usually used for correcting typing errors when entering new text
	/F2/ or /CTRL/ /2/ (DELETE CHAR)	Deletes the character under the cursor and shifts the subsequent characters in the line one position to the left
	/CTRL/ /K/ (ERASE TO END)	Deletes the contents of the current line from the current cursor position to the end of the line
	/SHIFT/ /F2/ or /COMMAND/ /2/ (DELETE LINE)	Deletes the current line and moves subsequent text one line up the screen. The position of the cursor is not changed; it remains in the same column position. The deleted line of text is placed in a holding area called the restore buffer. This action overwrites the previous contents of the restore buffer except where DELETE LINE functions immediately follow each other, in which case subsequent deleted lines are appended to the buffer. This enables you to move a block of text from the file into the buffer, from where it can be reinserted into the same or another file using the RESTORE LINES function
To restore text	/CTRL/ /R/ (RECALL LINE)	Restores the contents of the current line to its original state. The contents restored are those that existed before the cursor was moved to this line. Once the cursor is moved off a particular line the old contents of that line cannot be recalled using this function

	<p><code>/SHIFT/ /F5/ or</code> <code>/COMMAND/ /5/</code> (RESTORE LINES)</p>	<p>Inserts the contents of the restore buffer into the file starting at the line below the current cursor position. The cursor is moved to the start of the inserted line(s). The restore buffer itself is not changed. This function is used in conjunction with the DELETE LINE function to move and/or copy blocks of text</p>
<p>To split and join lines of text</p>	<p><code>/SHIFT/ /3/ or</code> <code>/COMMAND/ /3/</code> (SPLIT LINE)</p>	<p>Divides the current line into two by moving all text under and to the right of the cursor onto the next line. The cursor does not move. Text on subsequent lines is shifted one line down the screen</p>
	<p><code>/F3/ or</code> <code>/CTRL/ /3/</code> (JOIN LINES)</p>	<p>Combines two lines into one. The text on the subsequent line is placed immediately after the last nonspace character on the current line. The cursor does not move. If the current line cannot accommodate the entire text of the next line then only that amount which fits is moved and the remaining text stays on the same line but is moved to the left-hand edge of the screen</p>
<p>To insert a marker</p>	<p><code>/SHIFT/ /F7/ or</code> <code>/COMMAND/ /7/</code></p>	<p>Causes a marker to be inserted in the text immediately following the current line. The marker is a line of reverse video spaces containing the text "MARK". If the MARK line was previously located somewhere else in the text it is moved from where it was to the new position. Note that this is not an actual line of text and will never be written to the file. Its placement is therefore only significant during the current editing session. It is used in con-</p>

		<p>junction with the GOTO MARK function as a place marker (for details see the section entitled "Window Moving Function Keys"), and in conjunction with the high-level command DELETE (see the section entitled "Commands and Searching")</p>
to enter control characters	/SHIFT/ /3'/ (ESCAPE CODE)	<p>Inserts what the Video File Editor interprets as an Escape Code (not to be confused with the ASCII ESC character).</p> <p>The Video File Editor allows you to enter only the printable ASCII character set (hexadecimal codes 20 to 7E). To force the generation of "control" codes (hexadecimal 00 to 1F and 7F) the Escape Code character (not the ASCII ESC character or the /ESC/ key), must be used. When you type /SHIFT/ /3'/, the Escape Code character, reverse video (), is placed on the screen. The following character becomes a control character. Only the lower five bits of code are written to the file, generating a code in the range 00 to 1F. To generate a code of 7F, you must enter /ESCAPE/ /?/</p> <p>NOTE: 3' refers to the 3 key on the numeric keypad only.</p>

Examples

The following table shows some examples of how text can be modified using the functions discussed above.

STEP	IF you enter...	System displays...
		The purpose of this text is to act as an example of how to use the editing functions of the Video File Editor
1	DELETE LINE	as an example of how to use the editing functions of the Video File Editor
2	CURSOR UP	- as an example of how to use the editing functions of the Video File Editor
3	INSERT LINE INSERT MODE /T/ /h/ /i/ /s/ /SPACE/ /i/ /s/ /SPACE/	This is ▲ as an example of how to use the editing functions of the Video File Editor
4	JOIN LINES	This is as an example of how to use the editing functions of the Video File Editor
5	DELETE CHAR DELETE CHAR DELETE CHAR	This is an example of how to use the editing functions of the Video File Editor
6	NEXT LINE	This is an example of how to use the editing functions of the Video File Editor
7	DELETE LINE	This is an example of how to use the Video File Editor

VIDEO FILE EDITOR

8	RESTORE LINES NEXT LINE	This is an example of how to use the Video File Editor the editing functions of
9	INSERT MODE /T/	This is an example of how to use the Video File Editor The editing function of
10	END OF LINE	This is an example of how to use the Video File Editor The editing functions of _
11	BACKSPACE BACKSPACE	This is an example of how to use the Video File Editor The editing functions _
12	RECALL LINE	This is an example of how to use the Video File Editor the editing functions of
13	SPLIT LINE	This is an example of how to use the Video File Editor the editing functions _ of
14	CURSOR UP	This is an example of how to use the Video File Editor _ the editing functions of
15	INSERT LINE	This an example of how to use the Video File Editor _ the editing functions of

Note: To delete a character in the 80th column you should move the cursor to that position in overstrike mode and enter /SPACE/.

WINDOW MOVING FUNCTION KEYS

The function keys described in the following table enable you to move the window up and down the file.

FUNCTION KEY	MEANING
/SHIFT/ /F12/ or /COMMAND/ /~/ (TOP)	Moves the window to the top of the text file. The cursor is placed on the top bar of the file
/F12/ or /CTRL/ /~/ (BOTTOM)	Moves the window to the end of the file. The cursor is placed on the last line of text
/SHIFT/ /F11/ or /COMMAND/ /-/ (FULL SCREEN UP)	Causes the window to be moved up the file by 20 lines. This allows one line of overlap between the old and new displays. The cursor remains on the same screen line
/F11/ or /CTRL/ /-/ (FULL SCREEN DOWN)	Causes the window to be moved 20 lines down the file. This allows one line of overlap between the old and new displays. The cursor remains on the same screen line
/SHIFT/ /F10/ or /COMMAND/ /0/ (HALF SCREEN UP)	Causes the window to be moved half a screen (10 lines) up the file. The cursor remains on the same screen line
/F10/ or /CTRL/ /0/ (HALF SCREEN DOWN)	Causes the window to be moved half a screen (10 lines) down the file. The cursor remains on the same screen line

/SHIFT/ /F9/ or /COMMAND/ /9/ (LINE UP)	Causes the window to be moved one line up the file. The cursor remains on the same screen line
/F9/ or /CTRL/ /9/ (LINE DOWN)	Causes the window to be moved one line down the file. The cursor remains on the same screen line
/SHIFT/ /1'/ (NEXT LINE)	Moves the window one line down the file and places the cursor at the start of the next text line
/SHIFT/ /5'/ (GO TO MARK)	Moves the window up or down the file such that the cursor lies on the MARK line. The cursor remains on the same screen line

Note: The ' symbol after a number indicates a key on the numeric keypad.

EXITING AND SAVING FUNCTION KEYS

The function keys described in the following table enable you to exit from the Video File Editor and/or save the file you have been working on.

FUNCTION KEY	MEANING
EXIT AND SAVE /F6/ or /CTRL/ /6/	Causes the revised text to be written back to the file and the Video File Editor to be terminated. The screen is erased and control is returned to PCOS
SAVE TEXT /F5/ or /CTRL/ /5/	Causes the revised text to be written to the file. The Video File Editor does not terminate

ABORT /SHIFT/ /F6/ or /COMMAND/ /6/	Causes the Video File Editor to terminate without writing the revised text to the file.
	If text has been altered or added since starting the editor you are asked to "Abort?". Strike the ABORT key again to confirm. Any other action causes the Video File Editor to ignore the ABORT. Control is returned to PCOS

COMMANDS AND SEARCHING

The second line of the screen (above the scale line) is called the editor command line and is used for entering high level commands and search strings.

To enter text on the editor command line you must first press the COMMAND MODE function key /SHIFT/ /F1/. This moves the cursor to the second line. You can now enter text there. All line editing operations - such as INSERT MODE, BACKSPACE and DELETE CHAR - now apply to the editor command line. The RECALL LINE function when used in command mode restores the editor command line to its previous contents. The /CR/ key performs EXECUTE COMMAND when used in this mode.

Repeating the COMMAND MODE key returns the cursor to the text window without performing any command operation. The RECALL LINE function, when used in command mode, restores the command line to its previous contents.

STRING SEARCHES

This feature enables you to search the file for a particular combination of characters. Before searching for a text string you must enter command mode by striking the COMMAND MODE function key. Then type in the text to be searched for followed by the appropriate function key, as described in the following table:

FUNCTION KEY	MEANING
/F8/ or /CTRL/ /8/ (SEARCH DOWN)	Searches for the text string, starting from the current cursor position and moving down the file until the first occurrence of the string. If found, the window and cursor are moved to it

<p>/SHIFT/ /F8/ /COMMAND/ /8/ (SEARCH UP)</p>	<p>Searches for the text string starting from the cursor position and moving up the file. If the string is found, then the window and cursor are moved to it</p>
---	--

Examples

The following table exemplifies the use of the searching functions.

If you enter on the editor command line	Then strike function key...	System displays...
		<p>This is an example of how to use the search function keys of the Video File Editor to find a particular combination of characters</p>
<p>/f/ /u/ /n/ /c/</p>	<p>SEARCH DOWN</p>	<p>This is an example of how to use the search function keys of the Video File Editor to find a particular combination of characters</p>
<p>/e/ /SPACE/ /o/ /f/</p>	<p>SEARCH UP</p>	<p>This is an example of how to use the search function keys of the Video File Editor to find a particular combination of characters</p>

COMMANDS

The Video File Editor commands are a set of special commands that enable you to perform a number of high level functions. Before entering a command you must strike the COMMAND MODE function key. You can then type in the command which is subsequently displayed on the editor command line. To execute the command you must then strike the EXECUTE COMMAND key (carriage return).

GOTO

This command enables you to move the window to a specific line number in the file.

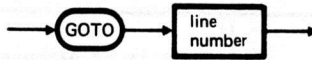


Fig. 13-4 GOTO

Where

SYNTAX ELEMENT	MEANING
line number	A decimal integer, the desired line number in the file. If this number is greater than the number of lines in the file, then the window is moved to the end of the file

Characteristics

Each line of the text file is automatically numbered. That is, the first line of the file is line 1, the TOP bar is line 0 and the MARK bar does not count.

DELETE

This command removes all text between the current line and the MARK line and places the removed text in the restore buffer from where it can be reinserted at will. If the MARK line does not exist an error message is given.



Fig. 13-5 DELETE

FILE

The FILE command allows you to suspend processing of the current file and invoke the editor on another file. When editing of the new file is terminated by a SAVE AND EXIT or ABQRT function, the old file is recalled at the point at which it was exited.

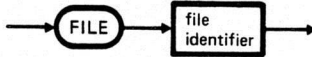


Fig. 13-6 FILE

Where

SYNTAX ELEMENT	MEANING
file identifier	The name of the new file to be edited, including any necessary password or volume identifier

Characteristics

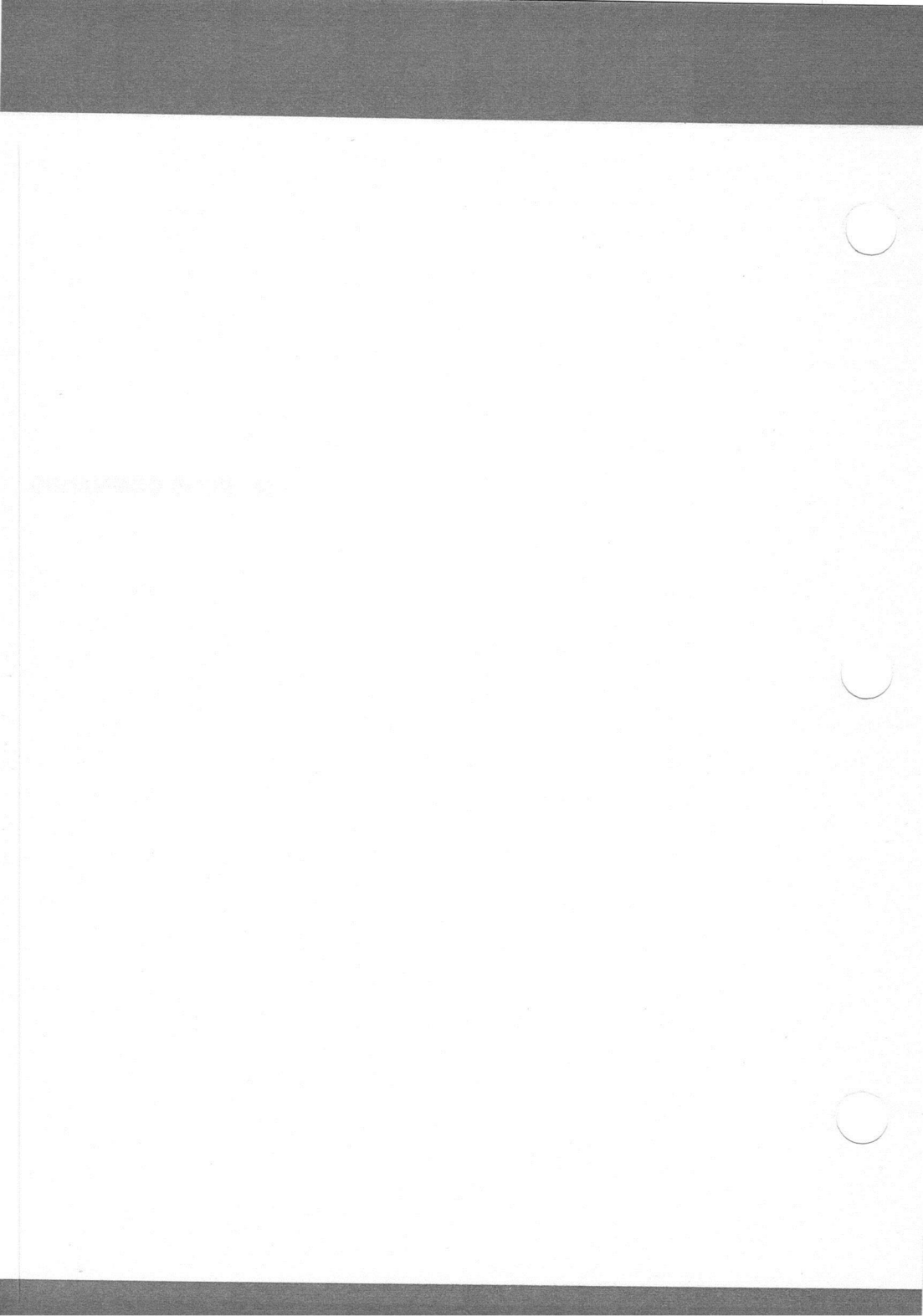
The command line option flags (b, c, g, or r) used by the old file remain the same for the new file.

Editing of each file is kept entirely independent except for the restore buffer, which enables the transfer of lines of text from one file to another.

Further files can be entered and edited from the new file using the FILE command. There is no limit to the number of levels that can be created in this way except that the text of all the files invoked must fit into user memory.

Files cannot be called recursively.

14. PCOS COMMANDS



BASIC.CMD

Loads the BASIC interpreter into memory and optionally runs a specified BASIC program.

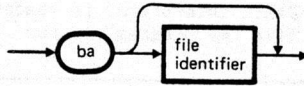


Fig. 14-1 BASIC

Where

SYNTAX ELEMENT	MEANING
file identifier	The file identifier of the BASIC program to be run

Characteristics

If the file is specified, the BASIC Interpreter is loaded, which in turn loads and executes the program stored under that file name. Following program execution the system remains in BASIC command mode.

The no-interaction (%n) flag cannot be used with this command.

Examples

IF you enter...	THEN...
ba my.program /CR/	The BASIC interpreter is loaded into memory, and the file "my.program" is run
ba /CR/	The BASIC Interpreter is loaded into memory and the computer enters the BASIC environment, command mode

BVOLUME.SAV

Enables a BASIC program to use the "Search" and "DiskFree" system calls as well as to obtain the name of the current volume.

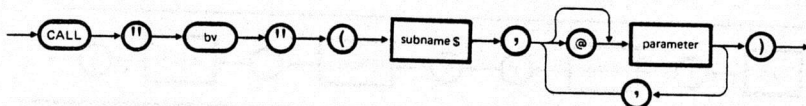


Fig. 14-2 BVOLUME

Where

SYNTAX ELEMENT	MEANING
subname	The name of the function to be used. This value may be one of - search To search the volume directory for a specified file name string - diskfree To return the number of free blocks remaining on the specified volume - getvolname To return the name of the current volume
parameter	A parameter to be passed to the command. This depends on the function. See below for a description of the corresponding function

Characteristics

This command can only be CALLED from BASIC. It cannot be executed directly from the PCOS environment.

BVOLUME search

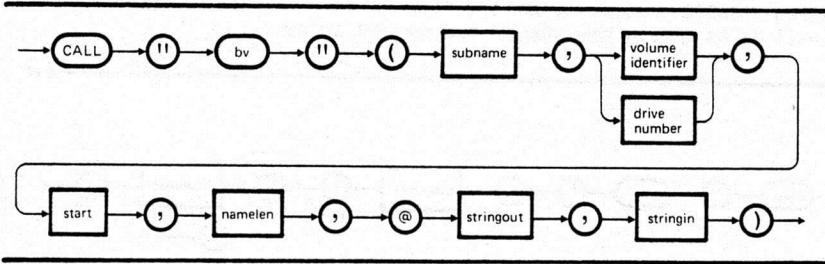


Fig. 14-3 BVOLUME search

Where

SYNTAX ELEMENT	MEANING
subname	The function name; in this case, 'search'
drive number	The drive number in which to search, or -1 if the last selected drive is desired
volume identifier	The volume in which to search
start	A flag that determines where to start the search. It has two possible values:

	<p>1 - Start searching from the start of the directory</p> <p>0 - Continue searching from the position established by the last BVOLUME call</p>
namelen	The length of the file name string to be searched for. This value can be set to zero if all names are to be returned
stringout	A dummy string variable in which the file name searched for will be returned. It must be initialized prior to calling BVOLUME by setting it to at least 14 characters
stringin	The name of the file to be searched for. A group of file names may be specified using wild cards

Example

DISPLAY	COMMENTS
<pre> 10 SUBNAME\$="search" 20 DRIVENUM%=0 30 NAMELEN%=5 40 STRINGIN\$="*.cmd" 50 STRINGOUT\$="12345678901234" 60 START%=1 70 CALL "BV"(SUBNAME\$, DRIVENUM%,START%,NAMELEN%, @STRINGOUT\$,STRINGIN\$) 80 PRINT"returned file name is:";STRINGOUT\$ 90 START%=0 100 GOTO 70 110 END </pre>	<p>Lines 10 to 60 set the parameters for a BVOLUME 'search' call on the volume inserted in drive 0. Line 30 sets NAMELEN% to the length of the string to be searched, while line 40 specifies the string; in this case it contains a wild card character. Line 50 sets the length of the output string to 14 characters.</p> <p>Line 70 CALLs the BVOLUME command and line 80 displays the returned file name. Line 90 sets the START% parameter to search for the next occurrence of a file whose name ends with ".cmd"</p>

BVOLUME diskfree

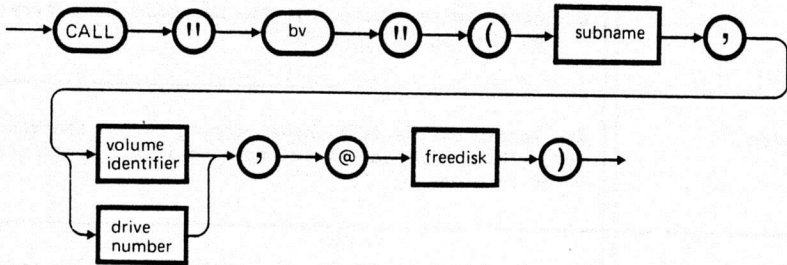


Fig. 14-4 BVOLUME diskfree

Where

SYNTAX ELEMENT	MEANING
subname	The function name; in this case, 'diskfree'
drive number	The drive number containing the volume whose number of free blocks is required. If a value of -1 is entered then the last drive accessed is assumed
volume identifier	The volume whose number of free blocks is to be returned
freedisk	An integer variable in which the number of free blocks will be returned

Example

DISPLAY	COMMENTS
<pre> 10 DEFINT F 20 SUBNAME\$="diskfree" 30 V ID\$="myvol:" 40 FREEDISK=0 50 CALL "bvolume" (SUBNAME\$,V ID\$, @FREEDISK) 60 PRINT "free blocks ..."; FREEDISK 70 END </pre>	<p>Lines 20 to 40 set the parameters for a BVOLUME "diskfree" CALL on the volume inserted in drive 0.</p> <p>Line 50 CALLs the BVOLUME command.</p> <p>Line 60 displays the number of free blocks</p>

Remark

BASIC always uses signed arithmetic with integer variables. Therefore, if the "diskfree" option of BVOLUME is used on the hard disk, the returned value must be converted to double-precision before being displayed or used in calculations.

BVOLUME getvolname



Fig. 14-5 BVOLUME getvolname

Where

SYNTAX ELEMENT	MEANING
subname	The function name; in this case, 'getvolname'
volume name	A dummy string variable in which the name of the last volume accessed will be returned

Example

DISPLAY	COMMENTS
10 SUBNAME\$="getvolname" 20 VOLNAME\$="12345678901234" 30 CALL "BV"(SUBNAME\$, @VOLNAME\$) 40 PRINT "the current volume is named"; VOLNAME\$ 50 END	Line 10 specifies 'getvolname' as the subname parameter. Line 20 defines the size of the string variable (14 characters) into which the volume name will be returned. Line 30 CALLs the BVOLUME command to perform the required function. Line 40 displays the returned volume name

CI.SAV

Allows a BASIC program to interface with the RS-232-C driver. It can be used only from BASIC.

For further details refer to "Serial Interface for I/O Peripherals User Guide."

CKEY.CMD

Enables the user to change the character codes assigned to the raw key codes generated at the keyboard, or to set the shift lock for the alphanumeric and/or numeric keypads.

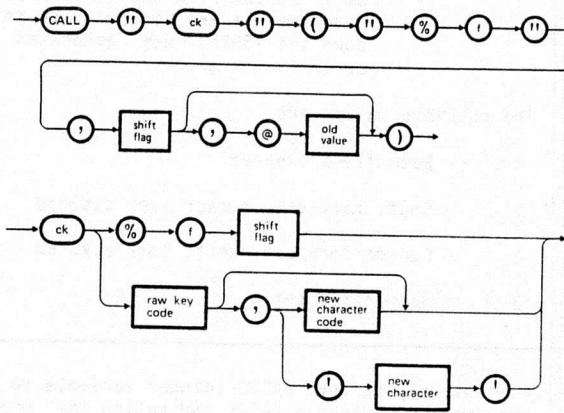


Fig. 14-6 CKEY

Where

SYNTAX ELEMENT	MEANING
f	When present, indicates that the numeric value that follows specifies the shift flag value

<p>shift flag</p>	<p>An integer in the range 0 to 3 that determines the setting of the two shift flags: shift lock and cursor lock</p> <p>shift lock - When set, all alpha keys on the alpha numeric keypad take on shifted values. That is, an alpha key struck alone generates an upper-case character, but an alpha key struck while holding down the SHIFT key generates a lower-case character</p> <p>cursor lock - When set, all keys on the numeric keypad take on shifted values. That is, a key struck alone generates code as if struck with the /SHIFT/ key. Likewise, a key struck while holding down the /SHIFT/ key generates code as if struck alone</p> <p>The possible values are:</p> <ul style="list-style-type: none"> 0 - Both flags cleared 1 - Shift lock set, cursor lock cleared 2 - Cursor lock set, shift lock cleared 3 - Both flags set
<p>old value</p>	<p>A previously defined BASIC integer variable to which the command assigns a value indicating the previous setting of the shift flag. This facility can be used only when the command is invoked from the BASIC environment using the BASIC CALL statement</p>
<p>raw key code</p>	<p>The code that is immediately generated by striking a particular key of the keyboard. This code is dependent only on the physical position of the key. That is, it is independent of the national keyboard and of any string assignments made by the PKEY command</p>

new character code	The code to be generated when the key or key combination specified by the raw key code is struck. The code may be specified as a decimal integer alone, or as a hexadecimal integer preceded by &
new character	Any character that can be entered from the keyboard

Characteristics

Figure 14-7 shows the raw key codes that are generated for every key whether struck alone, in conjunction with the SHIFT key, with the CTRL key, or with the COMMAND key. The keys shown correspond to the physical position of the keys on the keyboard.

This figure will be supplied at a later date.

Fig. 14-7 Raw Key Codes

Each raw key code points directly into a table which translates the raw key code into the ASCII code for the keyboard character, or into a function for the particular national keyboard. The translated raw key codes correspond to the standard ASCII codes (hexadecimal 0 to 80), while the remainder specify other keyboard functions. Where two keys or key-combinations have the same raw key code (e.g. /F1/ and /CTRL/ /1/), both translate to the same ASCII code or function. Thus, any changes made to one key affect both. But note the special cases shown in the following

table:

CODE (hexadecimal)	CORRESPONDING KEY COMBINATION	DESCRIPTION
A0	/CTRL/ /ESC/	Logical reset
A1	-	(reserved)
A2	/CTRL/ /C/	Break facility
A3	/CTRL/ /S/	Halt display
A4	/CTRL/ ///	Cursor lock
A5	/COMMAND/ ///	Shift lock
A6	/00/	Two zeros
A7	/CR/	End of line (CR in keyboard buffer, zero in LTERM buffer)
A8	/S1/	End of line (CR in keyboard buffer, '1' in LTERM buffer)
A9	/S2/	End of line (CR in keyboard buffer, '2' in LTERM buffer)
AA	/CR/ (DATEV keyboard)	End of line (CR in keyboard buffer, zero in LTERM buffer)
AB	/S1/ (DATEV keyboard)	End of line (CR in keyboard buffer, '1' in LTERM buffer)
AC	/S2/ (DATEV keyboard)	End of line (CR in keyboard buffer, '2' in LTERM buffer)
AD	/CTRL/ //	Screen Print
AE	-	(reserved)
AF	-	No operation

The raw key codes are shown in hexadecimal in Figure 14-7. They may, however, be entered either as hexadecimal values preceded by an & (ampersand), or alone as their decimal equivalents.

The user may enter the new character code either as a decimal integer, or as a hexadecimal integer preceded by an &; or the ASCII value can be entered simply by striking the corresponding key.

If, when entering a CKEY command, the raw key code is specified without also specifying a character code, then the current code corresponding to the raw key code is displayed.

The character codes assigned to the raw key codes remain either until they are changed by another CKEY command, until the entire conversion table is replaced by means of the SLANG command, or until the current working session is terminated. Alternatively, the changed values can be made permanent by use of the PSAVE command.

The no-interaction (%n) flag cannot be used with this command.

ANY CODE ASSIGNED TO A KEY USING THE CKEY COMMAND WILL BE EFFECTIVE EVEN AFTER ENTERING ANOTHER ENVIRONMENT OR APPLICATION PROGRAM. THIS ENABLES THE USER TO, FOR INSTANCE, REARRANGE FUNCTION KEYS AT WILL, BUT THE USER MUST TAKE CARE NOT TO DISABLE FUNCTIONS THAT WILL BE REQUIRED ON ENTERING THE NEW ENVIRONMENT/APPLICATION PROGRAM.

Examples

IF you enter...	THEN...
ck &C3,8 /CR/	The key S2 becomes backspace. That is, the decimal code for backspace (8) is assigned to the raw key code generated when S2 is struck (hexadecimal C3)
ck &C3 /CR/	<p>The raw key code and the corresponding ASCII code are displayed as follows:</p> <p>KEY = 195 (raw key code) CODE = 8 (ASCII code)</p> <p>(Note that values are displayed in decimal)</p>

Where

SYNTAX ELEMENT	MEANING
	The hardware configuration is to be displayed
m	The memory configuration is to be displayed

Characteristics

A typical hardware configuration would be displayed as follows :

```

Memory Configuration: 1
Floppy Disk Drive(s): 320 KByte.
Drive 0 Diskette:    320 KByte.
Drive 1 Diskette:    320 KByte.
Hard Disk Drive:     Not Present.
IEEE Board:          Not Present.
RS-232 (8251 - COM): Present.
RS-232 (8250 - COM1): Present.
Display Type:        Black and White.

```

CIU5TO

Note that on a 320 Kbyte drive, a 320 Kbyte diskette will be indicated even if there is none present. Furthermore, DCONFIG will not recognize a diskette change until the new diskette has been accessed.

The "Memory Configuration" value can be one of the following:

- 0 - main memory system with no expansion boards
- 1 - system with one or more 128K memory expansion boards

Typically, the memory configuration (for a system containing one 128k memory expansion board) is displayed as follows :

Memory Configuration:

Total memory size: 256 KBytes
 Free memory size: 227760 Bytes
 Basic memory size: 58906 Bytes

Address (Hex)	Block	Size	Filename	
Segment	Offset	Hex	Decimal	(Owner)

00	0004	3FFC	16380	PCOS.SAV
----	------	------	-------	----------

Address (Hex)	Block	Size	Filename	
Segment	Offset	Hex	Decimal	(Owner)

06	0004	7FFC	32764	FREE BLOCK
06	8004	3FFC	16380	PCOS.SAV

Address (Hex)	Block	Size	Filename	
Segment	Offset	Hex	Decimal	(Owner)

02	0084	0178	376	PCOS.SAV
02	0200	1480	5248	FREE BLOCK
02	1684	0158	344	PCOS.SAV
02	17E0	E61A	58906	FREE BLOCK

Address (Hex)	Block	Size	Filename	
Segment	Offset	Hex	Decimal	(Owner)

05	C004	200E	8206	PCOS.SAV
05	E016	0002	2	FREE BLOCK

Address (Hex)	Block	Size	Filename	
Segment	Offset	Hex	Decimal	(Owner)

0A	8004	76A2	30370	FREE BLOCK
0A	F6AA	0706	1798	dconfig.cmd
0A	FDB4	004C	76	FREE BLOCK
0A	FE04	0028	40	dconfig.cmd
0A	FE30	012C	300	FREE BLOCK
0A	FF60	004E	78	PCOS.SAV
0A	FFB2	004E	78	PCOS.SAV

Address (Hex)	Block	Size	Filename	
Segment	Offset	Hex	Decimal	(Owner)

09	8004	7FFC	32764	FREE BLOCK
----	------	------	-------	------------

Address (Hex)	Block	Size	Filename	
Segment	Offset	Hex	Decimal	(Owner)

0B	0004	FFFC	65532	FREE BLOCK
----	------	------	-------	------------

Note that this feature can be used to determine which programs have been PLOADED.

If both the %h and %m flags are specified, then both the hardware and memory configurations are displayed.

If no flag is specified, then a display such as the following is shown:

```
Total memory size: 256 KBytes.
Free memory size: 227760 Bytes.
Basic memory size: 58906 Bytes.
Display Type: Black and White.
Disk Drive(s): 1 ready.
```

The no-interaction (%n) flag cannot be used with this command.

Example

IF you enter...	THEN...
dc %hm /CR/	The hardware and memory configurations are displayed

EDIT.CMD

This command invokes the Video File Editor for editing a specified file.

Within the Video File Editor it is possible to perform a range of text editing functions. These are described in Chapter 13.

EPRINT.SAV

Lists any error or errors specified by their number, with a brief one-line description of each.

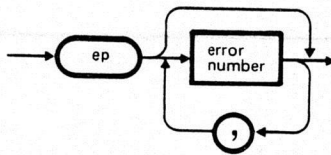


Fig. 14-9 EPRINT

Where

SYNTAX ELEMENT	MEANING
error number	Any existing PCOS error number. If no error number is specified, then no error is described, but the EPRINT command is loaded into memory

Characteristics

If an unprintable error or a BASIC error number is entered, no description is given.

Once the EPRINT command is resident in memory (via execution, PSAVE, or PLOAD), then any errors returned from PCOS will not only be displayed with the error number, but will also have the associated descriptive label with it. Note that the EPRINT command has a SAV extension and therefore becomes resident on execution.

Example

IF you enter...	THEN...
ep 58,59 /CR/	The system displays the two errors: ERROR 58... file already exists ERROR 59... disk type mismatch

- (1) - Copies the contents of one file into another.
- (2) - Copies one or more files (specified using wild cards) from one volume onto another

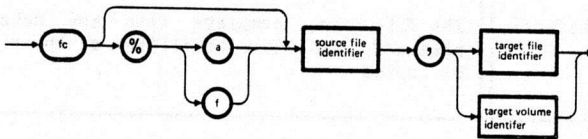


Fig. 14-10 FCOPY - (1)

Where

SYNTAX ELEMENT	MEANING
a	<p>The append flag.</p> <p>If the target file already exists and %a is specified, then the confirmation prompt will be suppressed and the contents of the source file will be appended to the end of the target file. If the target file does not exist, %a is ignored</p>

f	<p>The force copy flag.</p> <p>If the target file already exists and %f is specified, then the confirmation prompt will be suppressed and the contents of the target file will be overwritten with the contents of the source file. If the target file does not exist, %f is ignored</p>
source file identifier	<p>The file name, complete with any necessary password and volume identifier of the file to be copied</p>
target file identifier	<p>The name of a file on an unprotected volume, complete with any necessary password and volume identifier. If the file does not exist, it will be created and will have the same password as the source file</p>
target volume identifier	<p>The volume name or the drive number in which the target volume resides.</p> <p>If the volume is not enabled, the password has to be specified</p>

Characteristics

If the target file exists and no flag is specified, FCOPY will prompt the user for confirmation before overwriting it.

File already exists. Do you wish to overwrite? (y/n)

If %n, the no-interaction flag, is specified in the command line, then overwrite is automatically assumed.

If the target volume is specified the newly created file will have the same name and password as the source file.

Examples

IF you enter...	THEN...
fc %a dk1/dkpw1:FILE1, dk2:FILE2 /CR/	The contents of the file called "FILE1", on diskette "dk1", with password "dkpw1" are appended to the file called "FILE2", on diskette "dk2"
fc dk1:myfile, dk1:yourfile /CR/	A file may still be copied within the same volume. This example shows "myfile" is to be copied into "yourfile", both of which are on "dk1"
fc mydisk:*,10:	The entire contents of "mydisk" are copied onto the hard disk

Remarks

If there is sufficient space, a file can be copied or appended to another file on the same volume.

If the target file does not exist, it will be created and given the same password (if any) as the source file.

If the target file already exists, it will maintain its password.

The target file must not be write-protected.

If files have become extensively fragmented, the copying process may be used to gather all the scattered data into one contiguous target file - assuming there is enough space. This saves on I/O operation time.

At the end of an attempted FCOPY operation, error message 61 will result if there is insufficient space ("disk filled").

A file cannot be appended to itself.

If both %a and %f are specified, append will happen.

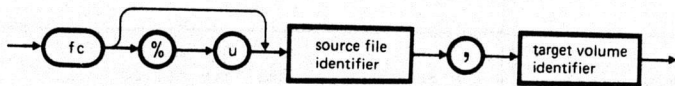


Fig. 14-11 FCOPY (2)

Where

SYNTAX ELEMENT	MEANING
u	<p>The unprotected copy flag.</p> <p>If "%u" is specified, then the copy process will skip any copy-protected files specified by the source file identifier parameter (see below). If not specified, then any copy-protected files will still be allocated file space on the target volume even though the contents of the files are not copied</p>
source file identifier	<p>The volume identifier of the volume containing the files to be copied followed by a string of characters, including at least one wild card, that specifies one or more file names complete with any password, if it is common to all files specified</p> <p>Note: If the volume identifier is not specified, the system will search for the files specified on the volume inserted in the last drive accessed</p>

<p>target volume identifier</p>	<p>The volume name or the drive number in which the target volume is inserted.</p> <p>If the volume is not enabled, then the password has to be specified</p>
---------------------------------	---

Characteristics

The newly created files will have the same names and password as the source files.

If a file specified within the group already exists on the target file, then the prompt

File already exists. Do you wish to overwrite? (y or n)

is displayed. If %n is specified in the command line, then overwrite is assumed.

Example

IF you enter...	THEN...
<p>fc %u 0:*.cmd,1: /CR/</p>	<p>All the files which have the extension CMD resident on the volume inserted in drive 0 are copied onto the volume inserted in drive 1. Any copy-protected files are skipped</p>

Remarks

Any target files that do not exist will be created. Each newly created file will have the same password as the corresponding source file (if the source file has one). On the other hand, if the target file already exists, it will maintain its password.

A target file must not be write-protected.

If files have become extensively fragmented, the copying process may be used to gather all the scattered data into one contiguous target file - assuming there is enough space. This saves on I/O operation time.

At the end of an attempted FCOPY operation, error message 61 will result if there is insufficient space ("disk filled").

FDEPASS.CMD

Deletes a previously assigned file password on an unprotected and enabled volume.

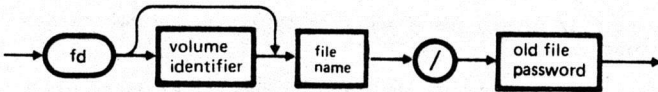


Fig. 14-12 FDEPASS

Where

SYNTAX ELEMENT	MEANING
volume identifier	The volume name and/or drive number, plus any necessary volume password
file name	EITHER the name of an existing file OR a string of characters, including wild cards, that specifies a group of file names
old file password	The existing file password which is to be deleted

Examples

IF you enter...	THEN...
fd 0:*/secret /CR/	The user can optionally remove the password "secret" from any file on the diskette inserted in drive 0 by answering "y" (yes) or "n" (no) to the messages displayed
fd 0: myfile/newpass /CR/	The password "newpass" to the file called "myfile" on the diskette inserted in drive 0 is deleted
fd 1:FILX/PASSX /CR/	The password called "PASSX" on the file called "FILX" on the diskette inserted in drive 1 is deleted

Remarks

Since the password must be known before it can be deleted, this command cannot be used to gain access to a file for which the password has been forgotten.

Wild card characters cannot be used in the password portion of a file identifier. Thus, even though more than one file can be specified, you can specify only one password, which in turn can be the same for more than one file.

FFREE.COMD

Frees any unused file sectors from a specified file or a group of files or an entire volume. The freed sectors are made available to the system for future use.

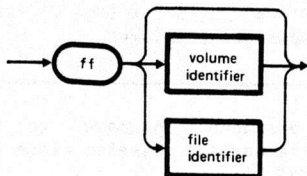


Fig. 14-13 FFREE

Where

SYNTAX ELEMENT	MEANING
volume identifier	The volume name or drive number in which the volume resides, plus any necessary volume password
file identifier	The name of the file plus any necessary password and volume identifier. The file name can be expressed using wild card characters to specify a group of files. If the file identifier is left out, then FFREE will be executed on all the files resident on the selected volume

Characteristics

After entering the command the following prompt appears:

You may not change disks while FFREE in progress. Continue?

If you reply "y /CR/", the operation continues; "n /CR/" aborts the operation.

If FFREE is used on the hard disk, any bad blocks that are allocated will not be freed.

Examples

IF you enter...	THEN...
ff 1:myfile /CR/	After you reply "y" to the subsequent prompt, the file "myfile" resident on the diskette inserted in drive 1 is cleared of any unused sectors
ff myvol: /CR/	After you reply "y" to the subsequent prompt, all the files on the volume "myvol" are cleared of any unused sectors. That is, the entire volume is cleaned up

Remarks

FFREE will not be executed if either the volume or the file is write-protected.

FKILL.COMD

Deletes a specified unprotected file or group of files from an unprotected or enabled volume.

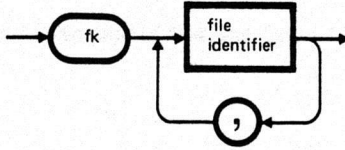


Fig. 14-14 FKILL

Where

SYNTAX ELEMENT	MEANING
file identifier	The name of the file to be deleted, plus any necessary password and volume identifier. The file name can be expressed using wild cards to specify a group of files

Characteristics

If the FKILL command is used on the hard disk, any bad blocks currently allocated to the specified file(s) will not be freed.

Examples

IF you enter...	THEN...
fk myvol: myfile/mypass /CR/	The file "myfile" with password "mypass" resident on the volume "myvol" is deleted
fk 1:myfile, yourfile /CR/	Files "myfile" and "yourfile" are deleted from the diskette inserted in drive 1
fk mydisk:* /CR/	You can optionally delete any file resident on "mydisk" by answering "y" (yes) or "n" (no) to the interactive messages displayed

Remarks

FKILL will not function if either the volume or the file is write-protected.

FLIST.CMD

Lists the contents of one or more files. The listing can be in either ASCII or hexadecimal form.

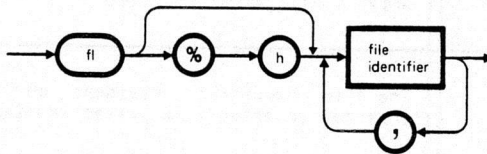


Fig. 14-15 FLIST

Where

SYNTAX ELEMENT	MEANING
file identifier	The name of the file, plus any necessary password and volume identifier. The file name can be expressed using wild card characters to specify a group of files
h	The specified files are to be listed in hexadecimal form

Example

IF you enter...	THEN...
<code>fl dk2/diskpwd: myfile/flpwd /CR/</code>	The file "myfile" which has the password "flpwd", and resides on volume "dk2", which in turn has the password "diskpwd", is listed
<code>fl 0:F*/pass,1:G.1 /CR/</code>	All files beginning with the letter F and residing on the diskette inserted in drive 0 will be listed, provided they have the password "pass" File G.1 residing on the diskette inserted in drive 1 will also be listed
<code>fl %h 1:yourfile /CR/</code>	The file "yourfile" resident on the diskette inserted in drive 1 is listed in hexadecimal form

Copies a file from one diskette to another using only one drive.

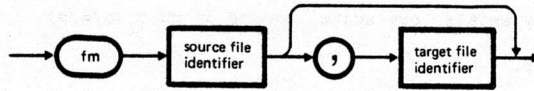


Fig. 14-16 FMOVE

Where

SYNTAX ELEMENT	MEANING
source file identifier	The name of the file to be copied plus any necessary password
target file identifier	The name of the file plus any necessary password. If the file does not exist it will be created. If the target file identifier is not specified then the source file is copied onto the target diskette into a newly created file of the same name as the source file

Characteristics

When FMOVE is called, the system displays a message asking the user to insert the source diskette in the drive. On hitting any key, the first part of the file is read into memory, and a message is displayed, specifying the number of passes necessary to make the copy. At this point the user is asked to insert the target diskette in the drive. On hitting any key, the data that was read into memory will be transferred to the target diskette. If the copy operation takes *n* passes, the system will display messages *n* times, asking the user to repeat the process described above. It is only after the last pass is completed that the PCOS prompt appears,

meaning that other commands can be entered.

FMOVE is useful on systems that have only one diskette drive. On dual diskette drive systems FCOPY would normally be used.

If the destination file already exists, then the message

File already exists: overwrite, append or quit (o/a/q)

will be displayed.

If you enter "o", then the contents of the target file will be overwritten with the contents of the source file.

If you enter "a", then the contents of the source file will be appended to the target file.

If you enter "q", then the command is aborted, and the PCOS prompt appears on the screen.

Example

IF you enter...	THEN...
fm myfile,myfile /CR/	The FMOVE process is started; if a diskette which has a file called "myfile" resident on it is inserted, then this file is read and stored in memory. If another diskette is now inserted, the file "myfile" will be written on it

Remark

FMOVE can do all that FCOPY can do except:

- FMOVE does not allow wild cards
- FCOPY deallocates blocks that are not used by the target file, while FMOVE does not. The number of blocks allocated to the target file after an FMOVE operation is always the same as that of the source file.

FNEW.COMD

Creates an empty file on an unprotected or enabled volume, assigns a name to the file and, optionally, a password. The file system allocates only one extent (a series of contiguous sectors) to the file.

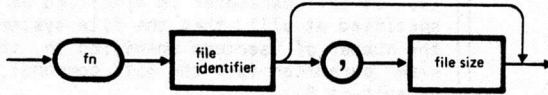


Fig. 14-17 FNEW

Where

SYNTAX ELEMENT	MEANING
file identifier	<p>The file name, optionally with a password and a volume identifier.</p> <p>If specified, the password is assigned to the file</p>

file size	<p>The number of sectors to be made available to the file. The minimum value is 2, and the maximum value depends on the number of contiguous free sectors available on the volume.</p> <p>There are two exceptions to the above:</p> <p>(i) If this parameter is specified as 0 (or not specified at all), then the file system allocates the number of sectors specified in the "extent size" parameter (see the SSYS command), which by default is 8</p> <p>(ii) If specified as 1, then the file system issues an error message</p>
-----------	--

Example

IF you enter...	THEN...
fn newfile/pass,12 /CR/	A 12 sector file called "newfile" is created on the volume in the last drive to be selected and is assigned the password "pass"

Remarks

If the volume has write-protection, the aluminized label must be removed before this utility can be used.

If there is not enough contiguous volume space to honor the request, a message

ERROR 61 ---- disk filled

will be displayed.

FPASS.COMD

Assigns a password to a file, or a number of files, on an unprotected volume, or changes an already existing file password to a new one.

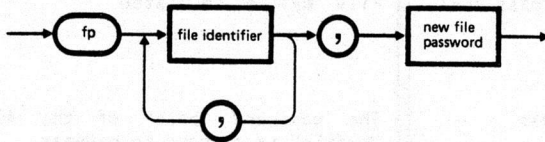


Fig. 14-18 FPASS

Where

SYNTAX ELEMENT	MEANING
file identifier	EITHER the file name, plus any necessary password and volume identifier OR a volume identifier (if necessary) followed by a string of characters, including wild cards, that specifies one or more file names plus any necessary password, if it already exists and is common to all the files specified
new file password	Is an alphanumeric string to be assigned to the file(s) as a password

Examples

IF you enter...	THEN...
<pre>flist 0:myfile/pass /CR/ . . fpass myfile/pass, newpass /CR/ . .</pre>	<p>File "myfile" is listed</p> <p>The password "pass" of the file called "myfile" is changed to newpass</p>
<pre>flist myfile /CR/ . .</pre>	<p>An error message</p> <p>ERROR 73 --- invalid password</p> <p>is displayed</p>
<pre>flist myfile/newpass /CR/</pre>	<p>File "myfile" is again listed</p>
<pre>fp 1:myfile,yourfile, ours /CR/</pre>	<p>The password "ours" is assigned to both "myfile" and "yourfile"</p>
<pre>fp 0:*,secret /CR/</pre>	<p>All the file names on the diskette inserted in drive 0 are displayed one by one, and an interactive message asks whether the password "secret" is to be set on the file. For each file, entering "y" (yes) assigns or "n" (no) does not assign the password</p>

Remarks

If the file is on a write-protected diskette, the write-protect label must be removed before issuing the command.

Like VPASS, the command FPASS can assign a new password directly to a file which already has one, but the user must specify the old password.

Once assigned, a password must be specified in all future references until it is deleted or changed. Otherwise, the file system will display

error code 73, corresponding to the message "invalid password."

File (and volume) security is a user responsibility. It is important to remember passwords, as neither PCOS nor BASIC will allow the user access to a file whose password has been forgotten.

To safeguard files from accidental changes by those who have access to them, use volume and file write-protection.

FRENAME.COMD

Changes the name of an existing file on an enabled or unprotected volume.

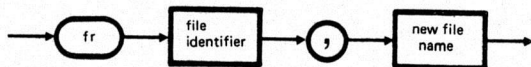


Fig. 14-19 FRENAM

Where

SYNTAX ELEMENT	MEANING
file identifier	The file name, plus any necessary password and volume identifier
new file name	Any file name not currently used on the volume where the file being renamed resides

Example

IF you enter...	THEN...
fr 1:OLDname, NEWname /CR/	The file name "OLDname" of the file resident on the diskette inserted in drive 1 is changed to "NEWname"

Remark

The FRENAME command has no effect on the file password.

FUNPROT.COMD

Removes write-protection from a file, or a number of files, on an unprotected or enabled volume.

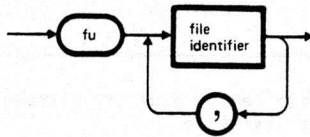


Fig. 14-20 FUNPROT

Where

SYNTAX ELEMENT	MEANING
file identifier	EITHER the name of the file, including any necessary extension, password, and volume identifier OR a volume identifier (if necessary) followed by a string of characters, including wild cards, that specifies one or more file names complete with any necessary password, if one exists and is common to all the files specified

Examples

IF you enter...	THEN...
fu mydisk:myfile /CR/	The write-protection is removed from "myfile" which resides on the volume "mydisk"
fu DKI:* /CR/	You can optionally remove write-protection from any file resident on the volume "DKI", by answering "y" (yes) or "n" (no) to the interactive messages displayed

Remarks

If the file is on a write-protected diskette, the write-protect label must be removed before, and replaced after, this operation.

Once write-protection is removed, the file can be deleted, overwritten, appended to, renamed, or in any other way modified (providing there is no write-protect label on the diskette).

If the file has no write-protection when this command is issued, this command will change nothing.

FWPROT.COM

Write-protects a file, or a number of files, on an unprotected or enabled volume.

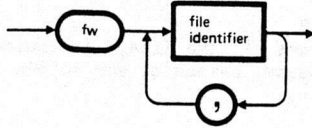


Fig. 14-21 FWPROT

Where

SYNTAX ELEMENT	MEANING
file identifier	EITHER the name of the file, plus any necessary password and volume identifier OR a volume identifier (if necessary) followed by a string of characters, including wild cards, that specifies one or more file names, complete with any necessary password if one exists and is common to all the files specified

Examples

IF you enter...	THEN...
fw mydisk/diskps: myfile /CR/	The file "myfile" which resides on volume "mydisk" and has a password of "diskps" is write-protected

<pre>fw 1:fileA, fileB /CR/</pre>	<p>Files "fileA" and "fileB" residing on the diskette inserted in drive 1 are both write-protected</p>
<pre>fw 1:prog.??? /CR/</pre>	<p>You can optionally apply write-protection to any file that has the file name "prog", with a three letter extension and which is resident on the diskette inserted in drive 1, by answering "y" (yes) or "n" (no) to the messages displayed</p>

Remarks

If the file is on a write-protected diskette, the write-protect label must be removed before this operation and replaced afterwards.

Once write-protected, the file may not be modified until write-protection is removed with a FUNPROT command.

If the file already has write-protection assigned, this command will change nothing.

A write-protected file can be listed or copied.

Write-protection can be assigned to any file, whether it has a password or not.

If all the files of a particular diskette are meant only to be read, then the diskette should be write-protected.

HBACKUP.CMD

The HBACKUP command allows the user to back up the PCOS partition of the hard disk (or the entire disk if it is all partitioned for PCOS) onto one or more diskettes. Moreover, it is the only way in which a file larger than the capacity of a diskette can be backed up.

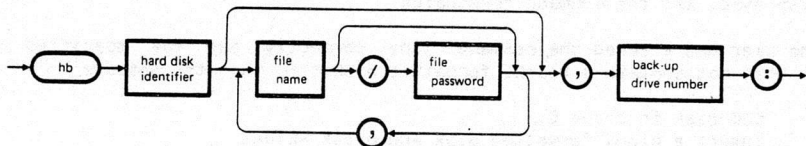


Fig. 14-22 HBACKUP

Where

SYNTAX ELEMENT	MEANING
hard disk identifier	The volume name or drive number to be backed up
file name	EITHER the name of a file to be backed up OR a file name containing wild cards to specify a group of files to be backed up
back-up drive number	The number of the diskette drive in which the backup will be made

Characteristics

If the specified diskette drive contains a blank, formatted diskette, then the backup commences immediately after the user enters the command line correctly.

If you specify less than two parameters in the command line, then the message

parameter missing

is displayed and the command terminates.

If the specified target volume identifier is not configured, then the message

destination volume missing

is displayed, and the command terminates.

If the user has entered the command line correctly but the specified drive does not contain a blank, formatted diskette, then the message

Bad disk in drive 0:

Insert a blank formatted disk and press RETURN

is displayed. Backing up will commence only after the user has inserted a blank, formatted diskette and strikes /CR/.

Once the backup has started, the HBACKUP command creates a file called "Backup.000" on the backup diskette. This is a composite file that will contain

- A table of all the names of the files on the hard disk to be backed up.
- As much of the data to be backed up as will fit onto the remainder of the diskette. This will comprise any number of the specified files, and/or a partial file. The data is held in a compressed form to save space and hence reduce the number of copies.

If all the specified files fit onto one diskette, then the backup operation terminates and control is returned to PCOS. If additional diskettes are required, however, the following prompt is displayed:

disk full - insert new disk and press RETURN

After the user inserts another blank, formatted diskette and strikes /CR/, the operation continues. The composite file on this diskette is automatically named "Backup.001." The user must then repeat this process until the backup is complete. Each subsequent diskette will contain one composite file automatically named in sequence "Backup.002," "Backup.003," etc., where each such file contains as many hard disk files and/or partial files as will fit on the diskette. Note that only the first diskette contains the table of file names.

Password-protected files may be backed up, but the password must be given. Write-protected files may be backed up, but will not be restored if a file of the same name exists on the hard disk and is still write-protected at restoration time. Copy-protected files will not be backed up by the HBACKUP command.

If the user specifies a file that does not exist on the hard disk, the message

file not found

is displayed, followed by the file name, and the backup continues.

Since the backup is saved in a format different from that of the original files on hard disk, a special command -- the HRESTORE command -- must be used to restore the backup to hard disk; that is, it is not possible to restore the backup using the FCOPY command. The backup can be copied (using the VCOPY command) **in the same format** to another set of diskettes, but cannot actually be used until it has been restored.

Examples

IF you enter...	THEN...
hb 10:,0: /CR/	The entire contents of the PCOS partition of the hard disk are backed up onto a series of diskettes in drive 0
hb 10:*.cmd,0: /CR/	All files with the file name extension ".cmd" on the PCOS partition of the hard disk will be backed up onto a series of diskettes in drive 0

Remarks

The FCOPY command can be used to back up hard disk files, provided no file to be copied is greater than the capacity of the target diskette.

If the hard disk is partitioned, only the PCOS partition can be backed up using the HBACKUP command. Other partitions must be backed up under their respective operating systems.

HDISK.COMD

Partitions the hard disk.

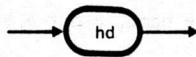


Fig. 14-23 HDISK

Characteristics

The HDISK command creates the PCOS partition on the hard disk. For operational details, refer to Chapter 12.

HRESTORE.CMD

Restores to hard disk all or part of a backup that was made using the HBACKUP command.

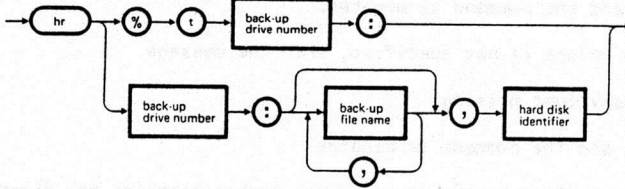


Fig. 14-24 HRESTORE

Where

SYNTAX ELEMENT	MEANING
t	A list of all files saved by the corresponding backup is displayed. No restore is performed. Only the first diskette of the corresponding backup operation is required for this operation
back-up drive number	The drive number of the diskette drive from which the backup copy is to be restored
back-up file name	EITHER the name of a backed up file to be restored OR a group of files specified using wild card characters. If this parameter is omitted, then all files from the corresponding backup are restored
hard disk identifier	The drive number of the hard disk to which the backup is to be restored

Characteristics

If the target volume is not specified and %t is not entered, the message
destination volume missing
is displayed and the command terminates.

If the source volume is not specified, then the message
backup volume missing
is displayed, and the command terminates.

If the drive specified by the source drive number contains the first disk of a backup session, then the restore operation begins as soon as the command line is entered. If not, then the message

please insert first back-up disk and press RETURN

will be displayed. In this case, the restore operation will begin after inserting the appropriate diskette and striking /CR/. The backup diskettes must be restored in the order in which they were saved; that is, the one containing backup file "Backup.000", followed by "Backup.001", and so on.

If the backup was made on more than one diskette, then the user is prompted to insert each diskette in turn.

If the no-interaction (%n) flag is used with this command, then files that already exist on the target volume will be overwritten without prompting.

If a requested file is not on the backup, the message
file not found

followed by the file name is displayed, and the restore operation continues.

Examples

IF you enter...	THEN...
hr 0:,10: /CR/	The hard disk will be restored from one or more backup diskettes inserted in series in drive 0
hr 0:*.cmd,10: /CR/	All files with the file name extension ".cmd" on a series of one or more backup diskettes inserted in drive 0 are restored to the hard disk
hr %t 0: /CR/	The files contained in the backup are displayed

IEEE.SAV

Loads and initialises the IEEE-488 extension package.

For more details about the IEEE-488 package refer to the "I/O with External Peripherals User Guide".

LABEL.CMD

Displays a string with a specified color, magnification, and orientation at a given position, within either the entire screen or the current window.

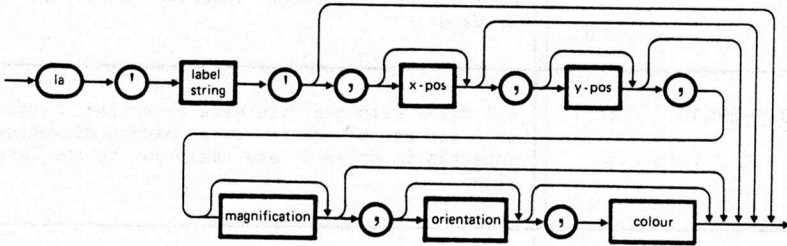


Fig. 14-25 LABEL

Where

SYNTAX ELEMENT	MEANING	RANGE	
		MIN	MAX
label string	<p>A valid string of printable characters that will be displayed at the specified position, magnification, and orientation.</p> <p>The string must be entered with leading/trailing quotation marks</p>		

<p>x - pos</p>	<p>A legal value in pixels, used by the LABEL command to determine the distance of the left side of the first string character from the left-hand edge of the screen or current window.</p> <p>The user can quote:</p> <ul style="list-style-type: none"> - The pixel value, if you are in PCOS or BASIC and the hardware coordinate system has been assumed, or - The corresponding SCALEX function, if you are in BASIC and a user coordinate system has been assumed (see "Remarks") 	<p>0</p>	<p>Window width, less the character width, less 1 (all measurements in pixels)</p>
<p>y - pos</p>	<p>A legal value in pixels, used by the LABEL command to determine the distance of the string base from the bottom of the screen or current window.</p> <p>The user can quote:</p> <ul style="list-style-type: none"> - The pixel value if you are in PCOS or BASIC and the hardware coordinate system has been assumed, or 	<p>0</p>	<p>Window height, less the character height, less 1 (all measurements in pixels)</p>

	- The corresponding SCALEY function, if you are in BASIC and a user coordinate system has been assumed (see "Remarks")		
magnification	A whole number n, where n is n times the normal character size	1	16
orientation	A value indicating the straight line aspect of the label string 0 - parallel to x-axis output left to right 1 - parallel to y-axis output bottom to top (+ 90 degrees) 2 - parallel to y-axis output top to bottom (- 90 degrees)	0	2
colour	The color number in the range 0 to 7 for eight color displays, 0 to 3 for four color displays, or 0 or 1 for black and white displays. If omitted, the foreground color is assumed	0	7

Defaults

SYNTAX ELEMENT	DEFAULT
x-pos	0
y-pos	0
magnification	1
orientation	0
colour	Current foreground color set from BASIC

The parameters x-pos and y-pos assume their default values if they are not specified. If the command is PLOADED and x-pos and y-pos are not specified they will assume their default values if the LABEL command is being entered for the first time after PLOADING.

For successive LABEL commands these unspecified parameters will assume the value corresponding to the character position following the last character of the preceding label string displayed.

Orientation

The following table illustrates orientation.

ORIENTATION	DISPLAY
0	

1	
2	

Examples

IF you enter...	THEN...
1a 'x-axis',,,8 /CR/ (in PCOS)	If the utility has not been PLOADED so that the given defaults apply, the "x axis" will be displayed starting at 0,0, with a magnification of 8 and orientation 0
SCALE 0,1,0,1 /CR/ CALL "1a" ("title", SCALEX(.25), 1,0) /CR/ (in BASIC)	The label string "title" is displayed at magnification one and orientation 0 starting at a point on the screen one quarter of the way along the x-axis (0.25) and three quarters of the way along the y-axis (0.75) Note: In BASIC, strings must always be enclosed in double quotation marks
1a 'name',,,1 /CR/	The string "name" will be displayed with normal orientation at coordinate 0,0, without magnification. The last parameter indicates the color in which the string is to be displayed

Remarks

LABEL is the only way that standard characters can be printed in graphics mode. It allows a string to be printed in various orientations and magnifications.

The string parameter must be specified.

If the parameters of a LABEL command are such that the label string does not entirely fit on the screen or within the specified window, the

command will still be executed, and the label string will be displayed with the part that falls outside the screen or window clipped.

Windows cannot be opened under PCOS.

The hardware coordinate system is set only if one of the following conditions applies:

- The system is in the PCOS environment
- The system is in the BASIC environment and
 - . The video has not been split into windows
 - . 512 x 256 display mode is in use

LSCREEN.COMD

Prints just the text displayed on the screen or within the specified window. Graphic elements are ignored.

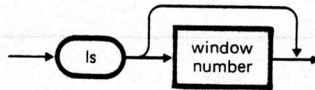


Fig. 14-26 LSCREEN.COMD

Where

SYNTAX ELEMENT	MEANING
window number	The number of the window to be printed. If this parameter is omitted, then the current window will be printed

Characteristics

The specified window may contain text with any spacing. Any screen data within the normal 5 by 10 character dot matrix that is not recognizable as text will not be printed.

On color systems, only screen plane 0 will be read. This means that if the background color is even, then the foreground color must be odd, and vice versa.

The command does not distinguish highlighted displays on the screen, but both normal and inverse video characters will be recognized and printed.

Example

IF you enter...	THEN...
EXEC "ls 3" /CR/ (in BASIC)	The text included within window number 3 will be printed

Remark

Windows cannot be opened in PCOS.

LTERM (ALWAYS RESIDENT)

Returns an integer (0, 1, or 2) corresponding to whichever one of the three carriage returns (/↵/, /S1/, or /S2/) was last used.

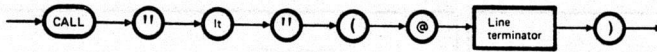


Fig. 14-27 LTERM

Where

SYNTAX ELEMENT	MEANING
line terminator	An integer variable which stores the value 0, 1, or 2, depending on whether the last line of input was terminated by /↵/, /S1/, or /S2/, respectively

Characteristics

The command is usable only in BASIC, using the CALL statement.

In a situation where a BASIC program prompts the user for an entry of some sort, the LTERM command can subsequently be CALLED to process the entry in one of three ways, depending on which of the three line terminator keys was used to close the entry.

Note: The integer variable used in a CALL "lt" command must already exist in the program. For this reason the user has to define the name being used by setting it equal to some value in a preceding statement. Subsequently, CALL "lt" assigns the variable a value depending on which carriage return key was last pressed.

Example

DISPLAY	COMMENTS
<pre> 100 Input A\$ 110 Let I%=0 120 Call "lt" (@I%) 130 If I%= 0 then 1000 else Goto 140 140 If I%=1 then 1100 or else Goto 1200 </pre>	<p>Line 110 defines the integer to be returned in line 120 which CALLs the LTERM command. Lines 130 and 140 determine whether the program is to continue at line 1000, 1100 or 1200 depending on whether /J/, /S1/, or /S2/ was used to terminate the data element retrieved by statement 100</p>

PKEY.CMD

Assigns a user-defined string to a particular key of the keyboard.

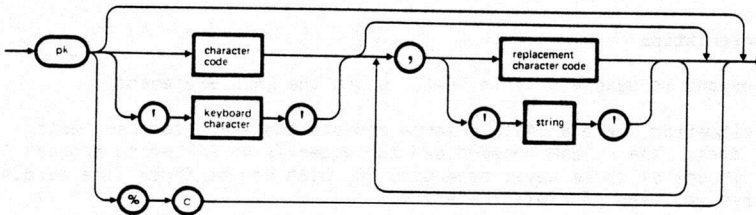


Fig. 14-28 PKEY

Where

SYNTAX ELEMENT	MEANING
character code	The decimal ASCII code, from 0 to 255, or hexadecimal code, from 0 to FF preceded by &, corresponding to a key or combination of keys to be replaced
keyboard character	Any of the characters on the keyboard, provided it has an ASCII equivalent (for example, A,a) excluding /ESC/, /CTRL/, /COMMAND/, /SHIFT/, /S1/, /S2/, and /↵/ to be assigned new representation
replacement character code	The ASCII code complying with the conditions described for "character code" which is to replace the original character
string	<p>Any string of characters the user wishes to have available at a single keystroke. The string can be up to 250 characters long.</p> <p>The string must be entered with leading and trailing quotation marks</p>
c	All strings previously assigned using the PKEY command are canceled

Characteristics

Note that certain codes correspond to more than one key or key-combination. (For example, the F-keys generate the same code as the top row of the typewriter-like keyboard.) Using PKEY on one of these keys causes the same change on any key or key-combination with the same code.

Any PKEY command is canceled by entering the character code parameter only. Since the entry of a character that has had a PKEY operation carried out on it causes the PKEYed string to be displayed, pk 'keyboard character' will not cancel the command. The character code itself must be used.

All assigned strings can be canceled by specifying the %c program flag in the command line.

A key that already has a string assigned to it can be redefined using another PKEY command specifying the character code and the new string. It is not necessary to cancel the first string.

If the PKEY command is entered without a parameter, then the defined keys are displayed along with their assigned strings. ANY VALUE ASSIGNED TO A KEY USING THE PKEY COMMAND WILL BE EFFECTIVE EVEN AFTER ENTERING ANOTHER ENVIRONMENT OR APPLICATION PROGRAM. THIS ENABLES THE USER TO, FOR INSTANCE, REARRANGE FUNCTION KEYS AT WILL, BUT THE USER MUST TAKE CARE NOT TO DISABLE FUNCTIONS THAT WILL BE REQUIRED ON ENTERING THE NEW ENVIRONMENT/APPLICATION PROGRAM.

Examples

IF you enter...	THEN...
pk &41,&42 /CR/	Each time "A" is pressed "B" appears. The ASCII character "A" is represented as hexadecimal &41, and "B" as &42
pk '#','ba',13,10, 'files',13,10 /CR/	This time the keyboard character '#' is used, and is therefore enclosed within quotation marks. 13 is the decimal code for "carriage-return," 10 is the code for "line-feed." The command thus reads: when # is entered, display "ba", do a carriage-return/line-feed, display "files" and execute a further carriage-return/line-feed. The result is that the BASIC interpreter is entered and the command FILES is then executed on the volume inserted in the last selected drive
pk 237,'FILES "1:"', 13,10, /CR/	This command assigns to the key combination /COMMAND/ /!1/ the string 'FILES "1:"',13,10 Thus, when in the BASIC environment, the key combination /COMMAND/ /!1/ can be used to list the directory on the diskette inserted in drive 1
pk &41 /CR/	The effect set up in the first example above is canceled
pk &23 /CR/	The effect set up in the second example above is canceled. (# is represented by hex code &23)

pk /CR/	All defined keys are displayed along with their assigned strings
pk %c /CR/	All defined keys are cleared

PLOAD (ALWAYS RESIDENT)

Loads a diskette-based or hard disk-based utility into memory.

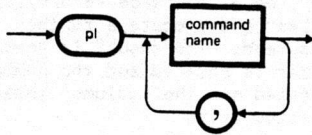


Fig. 14-29 PLOAD

Where

SYNTAX ELEMENT	MEANING
command name	The name of a transient command to be PLOADed

Characteristics

When a utility is PLOADed, it remains loaded in memory until either it is removed by the PUNLOAD command, or the current working session is terminated.

Once a utility has been PLOADed, the system displays the disk file name of the PLOADed utility (for example, VCOPY.COM), the program name (for example, Volume Copy Rev.4.x), the operation mode (for example, segmented/system), the entry points and memory allocated.

Example

IF you enter...	THEN...
pl vc /CR/	The VCOPY command is loaded into memory for the duration of the current working session

Remarks

The user may wish to PLOAD a utility for the following reasons:

- To use the utility after the diskette on which it resides has been removed
- To save the time lost in having to load the utility from diskette (or hard disk) every time it is used

However, PLOADing utilities has the adverse effect of reducing user memory.

None of the utilities have to be PLOADed in order to be executed. If they are invoked without being PLOADed, they will be automatically sought, loaded, and executed. Utilities having the CMD extension will then be removed from memory. But those having the SAV extension will remain until the end of the current working session.

The PLOAD command works only for transient commands. PLOAD, PUNLOAD, and LTERM are all resident and thus cannot be PLOADed.

PLOT.CMD

Controls the plotter through the computer operating system.

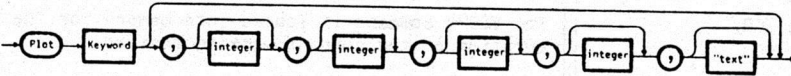


Fig. 14-30 PLOT

Note that it is necessary to enter the full word "plot" for PCOS to distinguish this command from PLOAD.

The number of integer parameters required is determined by the keywords preceding them. Please refer to the following table:

Where

SYNTAX ELEMENT	MEANING
keyword	A keyword string indicating the operation to be performed, from the following set: penup Raises pen pendn Lowers pen pen,n n = 0 - 8 Selects pen in stall n. n = 0, pen is deposited coords,x,y Transmits coordinates x,y move,x,y Moves pen to absolute coordinates x,y, with pen raised. If x, y not specified, pen is moved to most recent coordinates draw,x,y Moves pen to absolute coordinates x,y, with pen lowered. If x, y not specified, pen is moved to most recent coordinates ltype, n,l n = 0 - 4 Defines line type, as shown in User's Manual Plotter 281. l = length of

	basic line element
chsize,h,d,w	Defines character size and direction:
h	height (multiples of 0.1 mm)
d	direction, degrees from horizontal
w	width (multiples of 0.1 mm)
cslant,n	Defines character slant:
n = 0	normal character plotting, 90 degrees to horizontal
n = 1	75 degrees to horizontal
font,n	Defines character font:
n = 0	standard ASCII
1	German
2	Spanish
3	Swedish, Finnish
4	Danish, Norwegian
chars,,,, "string"	Enables character mode, skips next four parameters, transmits "string" for plotting at current pen position. Remains in character mode until you enter "plot cr"
lf	Issues line feed. Character plotting proceeds on next line
htab,n	Horizontally tabulate. Character plotting proceeds n characters from left margin of text
vtab,n	Vertically tabulate. Character plotting proceeds n lines from the upper margin.
n > 0	downwards
n < 0	upwards
cr	Issues carriage return. Disables character plot
point,n	Plots point mark.
n = 0 - 4	Specifies point symbol. (See manual.)
window,xmin, xmax,ymin, ymax	Sets the graphic limits for window plotting. Xmin, etc. must be specified with plotter coordinates (PC) (multiples of 0.1 mm with zero at lower left.)
vwindow	Verify window. Draws a frame according to the graphic limits
offset,x,y	Shifts plotter coordinates by an amount specified by parameters x and y
arc,r,a,b r	Draws circles and arcs. radius

a starting angle
b ending angle
Angles are specified in degrees, referred to the horizontal. Drawing begins, at the current pen position, at the 3 o'clock angle to the center, and proceeds counter clockwise. A circle is drawn if a and b are not specified, or if $a - b = 0$ or 360

circle,x,y,r Draws circles.
x,y coordinates of center
r radius

xaxis,l,d, Draws x-axis.
t1,t2
l length of x-axis
d distance between tick-marks
t1 length of first tick-mark
t2 length of second tick-mark

yaxis,l,d, Draws y-axis.
t1,t2
l length of y-axis
d distance between tick-marks
t1 length of first tick-mark
t2 length of second tick-mark

line,x1,x2, Draws line from (x1,y1) to (x2,y2)
y1,y2

digint,xvar, Activates plotter for interactive
yvar digitizing. The green "digitize"
LED will light. The 8-position pad
may then be used to move the pen or
digitizing reticle to the desired
position. (Do not attempt to move
the pen carriage manually without
consulting the manual.) When the
digitize/select button is pressed,
the current coordinates will be
returned as ASCII strings to the
M24. PLOT will convert the values
to binary form and place them in
xvar and yvar. For example,

10 A% = 0

20 B% = 0

.

.

80 CALL "plot" ("digint",@A%,@B%)

90 PRINT A%,B%

will: 1) initialize the numeric variables A%, B%; 2) permit the user to

	return to the M24 the x- and y-coordinates of a selected point, to be stored in A% and B%; and 3) display the values returned. In the event of lockup due to issuance of digitizing commands from within infinite loops, plotter malfunction, etc, the user may return control to the BASIC level by entering /CTRL/ /C/ at the keyboard
mover,x,y	Moves pen to relative coordinates x,y, with pen raised. If x, y are not specified, the pen will be moved to the most recently transmitted coordinates
drawr,x,y	Moves pen to relative coordinates x,y, with pen lowered. If x, y are not specified, the pen will be moved to the most recently transmitted coordinates
chartadv,n	The pen is lifted if it was down, and the paper is shifted n centimeters to the left
int	Integers or integer variables (when appropriate) with operation-specific definitions
text	A text string (when appropriate)

Characteristics

The PLOT utility supports the serially-interfaced GOERZ Servogor 281 plotter. The serial parameters (parity, baud rate, etc.) have variable, user-determined settings at the plotter. Thus, it is the user's responsibility to set the serial parameters on the M24 to matching values by loading and initializing the serial driver with the commands

```
rs
sc com:;....
```

before loading BASIC.

Examples

IF you enter...	THEN...
plot penup	The pen is raised

plot pendn	The pen is lowered
plot circle,100,50,10	A circle is drawn, center at coordinates 100 and 50, radius 1 mm
plot chars,,,,,"This is a circle"	Prints "This is a circle" at the current pen position
plot lf	Issues a line feed. Character plotting continues on next line.
plot cr	Issues a carriage return. (Disables character mode)
plot chars,,,,,"A perfect one"	Prints "A perfect one" on the next line
plot cr	Disables character mode

Remarks

There is an alternative method for controlling the plotter through the M24. After loading "rs" and setting the serial parameters with "sc" as above, the user enters the command SFORM as follows:

```
sf ,transp,,,,se
```

to configure the printer driver to transparent mode, serial port operation. Then the BASIC statement LPRINT may be used. For example,

```
LPRINT CHR$(1) + "P"; (Enable command string execution)
```

```
LPRINT " HF7 100/100KX3000 50 20 20H"; (Pen up, fetch pen 7, move pen to (100,100), draw x-axis, etc.)
```


Loads and initializes alternative PCOS operating systems.

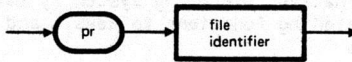


Fig. 14-31 PRUN

Where

SYNTAX ELEMENT	MEANING
file identifier	Specifies the file containing the operating system to be loaded

Example

IF you enter...	THEN...
pr 0:alt.1 /CR/	The operating system previously saved in file "alt.1" on the diskette inserted in drive 0 is loaded and initialized

Characteristics

The command is used in conjunction with the PSAVE command. That is, an operating system must have been previously configured and then saved using the PSAVE command before it can be reloaded and initialized by the PRUN command.

This command enables the user to configure different operating systems for different functions, then to alternate between the different operating systems. The following table shows an example of this feature:

STEP	OPERATION
1	Define an alternative operating system by setting global parameters, assigning functions to keys, and PLOADing selected commands
2	Save the new operating system in, for instance, file "alt.1" on the diskette inserted in, for instance, drive 1 by entering ps 1:alt.1 /CR/
3	Define another alternative operating system by setting global parameters, assigning functions to keys, and PLOADing selected commands
4	Save it, for instance, on the same diskette by entering ps 1:alt.2 /CR/
5	Enter pr 1:alt.1 /CR/ to reload and initialize the operating system saved in "alt.1"
6	Enter pr 1:alt.2 /CR/ to reload and initialize the operating system saved in "alt.2"

PSAVE.CMD

Saves the current PCOS on an unprotected or enabled volume.

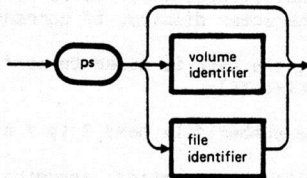


Fig. 14-32 PSAVE

Where

SYNTAX ELEMENT	MEANING
volume identifier	<p>A volume name or a drive number specifying the volume on which the current PCOS configuration is to be saved.</p> <p>If this parameter is omitted the PSAVE command will select drive 0 (even if a hard disk unit is configured)</p>
file identifier	<p>The name of the file which is to contain the current PCOS configuration, possibly preceded by a volume identifier.</p> <p>If the file identifier is omitted, the PSAVE command will assign the name PCOS.SAV</p>

Characteristics

The PSAVEd file will include any utilities which have been PLOADED, any programmed key definitions in effect, and the state of the system as set by the global commands (see Chapter 6). Moreover, any currently active user-defined font, and any globally specified device rerouting will also be included, as will control character display, if currently specified.

After entering a PSAVE command, the user is asked to confirm the intention to save PCOS on the specified file.

Save system on file drive number:file name ? (y / any other key)

A "y /CR/" response will complete the operation, assuming the corresponding drive is ready, and the specified volume is either unprotected or enabled. Entering any other key will abort the operation without error.

A PSAVEd file will follow other files that may already exist on the volume in question, but it is normal practice to PSAVE onto an empty diskette.

If a file with the same name as the PSAVEd file already exists on the specified volume, it will be overwritten. In any case, once the file is PSAVEd, the system is immediately rebooted via a standard boot file search (see Chapter 5). Furthermore, any future attempt to boot the system from this volume will cause the last PSAVEd file to be booted (unless the PRUN command is used).

Both write-protection and password protection can be applied to a PSAVEd file to protect it from being overwritten or modified by unauthorized users.

The no-interaction flag (%n) has no effect with the PSAVE command.

Examples

IF you enter...	THEN...
ps /CR/	The current PCOS configuration is saved on the diskette inserted in drive 0 in a file called PCOS.SAV as no file is specified

ps 1:alt1 /CR/

The current PCOS configuration is saved on the diskette inserted in drive 1 in a file called "alt1"

Remarks

You may find PSAVE useful to set the value of global parameters; for example, the default value for memory size, which may be set too low for some activities in BASIC.

PUNLOAD (ALWAYS RESIDENT)

Removes PLOADED commands from memory.

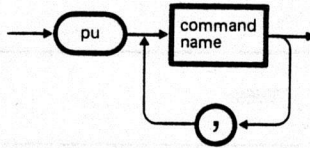


Fig. 14-33 PUNLOAD

Where

SYNTAX ELEMENT	MEANING
command name	The name of a PLOADED command to be made transient. Wild cards are not permitted

Characteristics

This command can be used to remove from memory commands that have either been PLOADED during the current working session or PLOADED and PSAVED during a previous working session. However, some commands change PCOS tables and therefore cannot be removed from memory. That is, the PUNLOAD command will not work on the following commands:

- CI
- RS232
- EPRINT
- VMOVE
- PDEBUG (not included on the system diskette)

Furthermore, the permanently resident commands PLOAD, PUNLOAD, and LTERM cannot be removed from memory.

Example

IF you enter...	THEN...
pl vc,ps /CR/ .	The transient commands VCOPY and PSAVE are loaded into memory
pu vc /CR/	The system responds PUnloading vcopy.cmd and the VCOPY command is removed from memory and becomes transient once again

RFont.COMD

Converts the system font data into a text file that describes each font character as a series of upper-case 'X's on a 10 by 8 matrix. This is the first step in creating a user-defined font.

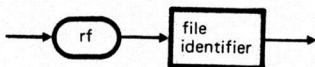


Fig. 14-34 RFont

Where

SYNTAX ELEMENT	MEANING
file identifier	The name of the file in which the font matrices are to be created, plus any necessary password and volume identifier. If the file does not already exist, it is created. If the file does already exist, a prompt appears asking the user to confirm the intention. A "y" response overwrites the file

Characteristics

The file generated by the RFont command comprises four lines of header information :

- The country identifier (USA, ITALY, etc.)
- The country number (see SLANG command)
- The font height (in scan lines). This must always be 10
- The number (decimal) of characters defined in the file (95 to 190)

This is followed by the font matrices. Each character is described in 11 lines, the first of which is the decimal code of the character; the

remaining 10 describe the character. For example:

```
50
-----
----XXX-
---X---X
-----X
-----X-
-----X--
----X---
---XXXXX
-----
-----
```

where "50" is the decimal code of the number "2".

To change the character, invoke the Video File Editor on the file and place X's so that they show the intended appearance of the character. It is advised that the first two or three columns appear blank so that characters do not run together. Once the changes are saved and the editor is exited, the WFONT command can then be entered to direct the system to display characters as they appear in the matrices.

The range of modified characters can be extended to twice the standard 95 character printable range. By adding 10 line font matrices (each preceded by the appropriate decimal code) to the end of the text font file, and by updating the "number of characters" that appears on the third line of the file, the desired number of characters can be defined. Confusing results may occur if matrix entries are not numbered consecutively from the decimal code 32, associated with "space" - the first printable ASCII character. Furthermore, the matrices must always be ordered in ascending ASCII sequence; otherwise the key/character correspondence will be destroyed. If the number of matrices exceeds the number of characters specified in the third line of the file, the extra fonts will not be defined.

Example

IF you enter...	THEN...
rf 1:italicfont /CR/	The system font data is converted into an ASCII font matrix file and saved on the disk inserted in drive 1 in the file named "italicfont"

Recovers (whenever possible) a recently deleted file.

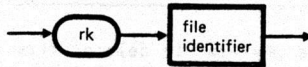


Fig. 14-35 RKILL

Where

SYNTAX ELEMENT	MEANING
file identifier	The name of the file to be recovered. It is not necessary to specify the file password

Characteristics

RKILL will recover only files that have not yet been overwritten. Once partly overwritten, a recently deleted file (either FKILLED from PCOS or KILLED from BASIC) cannot be recovered. Moreover, a file cannot be recovered on a volume that has been alphabetized since the file was deleted.

Examples

IF you enter...	THEN...
fn 1:myfile,20 /CR/	The file "myfile" is created on the diskette inserted in drive 1

vl 1: /CR/	A list of files resident on the diskette in drive 1 is displayed. "myfile" will appear at the end of this list
fk 1:myfile /CR/	The file "myfile" is deleted from the diskette in drive 1. (This can be checked using the VLIST command)
rk 1:myfile /CR/	The file "myfile" is recovered, and it will again appear in a volume list

Remarks

RKILL will not be executed if the volume is write-protected.

RS232.SAV

Loads the RS-232-C extension package.

When called, this utility allows the user to communicate, via the RS-232-C interface, with devices that are compatible with this interface. The transmission environment is set using the SCOMM command. However, it is necessary to execute the RS232 command before the SCOMM command.

For more details about the RS-232-C package refer to the "Serial Interface for I/O Peripherals User Guide".

SBASIC.CMD

Sets the BASIC environment for programming.

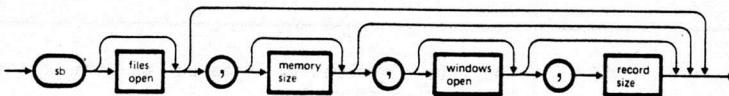


Fig. 14-36 SBASIC

Where

SYNTAX ELEMENT	MEANING	RANGE	
		MIN	MAX
files	The maximum number of files that can be opened concurrently under BASIC	0	15
memory	Maximum number of bytes available for user memory in BASIC	0	Actual value can be determined using the DCONFIG command (it can never be greater than 58K)
windows	The number of windows for which memory space is preallocated. Additional windows can always be opened if there is enough space	1	16
record size	Maximum record size (in bytes) for random files	1	4096

Default Values

At initialization from a standard PCOS these parameters have the default values shown in the table below. Alternatively, the user can establish other values either by using an initialization program (for example, INIT.BAS) or by loading a user-configured PCOS (obtained using PSAVE).

In each case the value of these parameters can be changed at any moment using the SBASIC command. If this is done, the values become effective only when the system goes into BASIC. If the SBASIC command is called from a BASIC program using an EXEC or a CALL statement, the newly set values will not be taken into account in the current program (otherwise the current program could be destroyed). These will become operative in subsequent programs.

A parameter, once changed, maintains its last-assigned value until it is again modified or until the system is reinitialized (by switching the system ON, or by a reset). Thus nil parameters in an SBASIC command keep their last-specified values in preceding commands (or default values).

The following table gives the standard default values.

PARAMETER	DEFAULT VALUE	MEANING
files	3	A maximum of 3 files can be opened concurrently
memory	36000 bytes	PCOS has allocated 36000 bytes for BASIC programs, data which is being processed by BASIC programs, open file tables, etc.
windows	1	The first window allocated requires 38 bytes. Any additional windows for which space is allocated will require a further 108 bytes of user memory each
record size	256 bytes	The maximum record size for random files is 256 bytes

Table 14-1 BASIC Environment Default Parameters

Examples

IF you enter...	THEN...
sb /CR/	<p>The current value of the parameters will be displayed (as no parameters were specified in the command).</p> <p>For example</p> <p>files = 3 memory = 36000 windows = 1 record size = 128</p>
sb 5,38000,,80 /CR/	<p>The command will establish that 5 files can be opened concurrently, that user memory is 38000 bytes and that the maximum record size for random files is 80 bytes. The omitted parameter (windows) will revert to the value specified in the preceding SBASIC command since the last initialization or in the last SBASIC command followed by a PSAVE (if a user-configured PCOS is being used) or to the default value (precedence in the order stated)</p>

BASIC Memory

When the SBASIC command is used, and you enter BASIC, user memory is reduced by the following amount in bytes:

$$- 829 + (578 + R)F$$

where F is the number of files that can be opened, and R is the record size

- 108 bytes for each window (except the first) that space is allocated for

It must be kept in mind that utilities, auxiliary commands, and functions assigned to keys reduce user memory and therefore reduce the maximum value of the "memory" parameter.

SCOMM.CMD

Sets the transmission environment of an RS-232-C communications port.

The RS-232-C communications driver must be loaded prior to the execution of SCOMM.

For more details, refer to the "Serial Interface for I/O Peripherals User Guide".

SDEVICE.CMD

Displays the device name table and optionally renames a device.

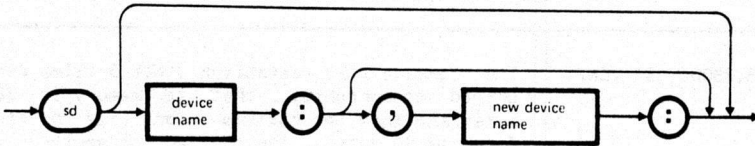


Fig. 14-37 SDEVICE

Where

SYNTAX ELEMENT	MEANING
device name	The current name of any device that exists on the device table consisting of, at most, 13 printable characters (see illegal characters in Chapter 4). This can be either the name assigned in the last SDEVICE command (in the current working session) or, in the absence of a preceding SDEVICE command, the default device name. Alternatively, if a user-configured PCOS is being used, the PSAVED device name
new device name	The name to be assigned to the device in question. The name must be, at most, 13 characters long (for illegal characters, see Chapter 4)

Characteristics

The colon (:) must follow immediately after the device name.

Default Device Names

prt: - PCOS Printer Driver

cons: - PCOS Console Driver (video and keyboard)

com: - Standard RS-232-C communication port on Z8001 card

com1: - Expansion RS-232-C communication port on motherboard

Examples

IF you enter...	THEN...
sd /CR/	The device table will be displayed on the VDU. The information displayed gives the device name and its type: W for a write-only type of device, R for a read-only type, and R/W for a device that can be read from and written to
sd prt: /CR/	The system will display the device name "prt:" and "W" meaning that the printer is a write-only device
sd cons:, kboard.video: /CR/	"cons:", the name of the console, is changed to "kboard.video:". The system will display the old name and device type first, followed by the message "changed to" and the new name and device type

SFORM.CMD

This command specifies the type of printer which is being used and the printing format.

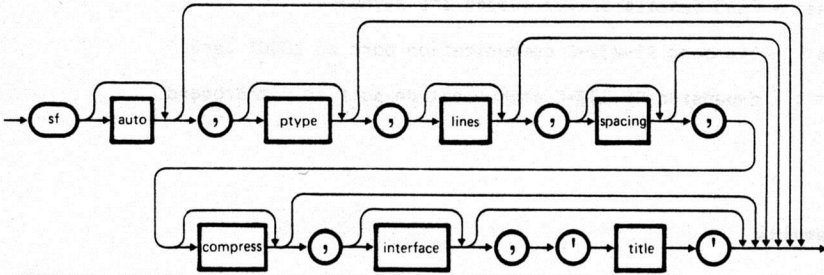


Fig. 14-38 SFORM

Where

SYNTAX ELEMENT	MEANING	DEFAULT VALUE
auto	<p>The status of the nil parameters</p> <p>'off' indicates they are to have their default values</p> <p>'on' indicates they are to have the values set by the most recent SFORM command since switch-on or, failing that, a previous SFORM command that was followed by a PSAVE command</p>	off

PCOS COMMANDS

<p>p<code>type</code></p>	<p>The type of printer connected. It is entered as a valid printer number, or as <code>transp</code>.</p> <p>"<code>transp</code>" sets transparent mode. That is, files are printed in free format, irrespective of the type of printer attached</p>	<p>PR15B</p>
<p><code>lines</code></p>	<p>The number of printed lines per page; 0 implies that no form feed will be issued</p>	<p>60</p>
<p><code>spacing</code></p>	<p>The number of interline spaces between printed lines. Single spacing = 1, double-spacing = 2, etc.</p>	<p>1</p>
<p><code>compress</code></p>	<p>A two-letter parameter where the first letter indicates if the characters are bold (<code>w</code>) or normal (<code>n</code>), and the second letter the type of character. This can be:</p> <ul style="list-style-type: none"> - <code>c</code> (compressed) = 16.6 chars/in. - <code>e</code> (elite) = 12 chars/in. - <code>p</code> (pica) = 10 chars/in. 	<p><code>ne</code> (normal, elite)</p>

interface	<p>Whether the interface through which the system controls the printer is serial or parallel:</p> <p>se built-in RS-232-C 8251 port (serial)</p> <p>pa Centronics type (parallel)</p> <p>s1 8250 port on mother board (serial)</p>	pa
title	<p>The title to be printed at the top of each page. It must not exceed 24 characters</p>	no title

Characteristics

To print a title at the top of the first page of a listing, the user must issue a form feed from BASIC; for example, "lprint chr\$(12)".

For the title format, a string of ")" deletes the previous title, without replacement.

If there is no printer attached to the corresponding port, the following error message will be issued

ERROR - RS232 CANNOT BE OPENED

Before specifying a printer at any serial port the user must:

- execute the RS232 command
- set the serial port parameters using the SCOMM command.

Example

IF you enter...	THEN...
sf on,,50 /CR/	The command specifies that the nil parameters are to take the values specified either in the last SFORM command followed by a PSAVE command, or in the last SFORM command since switch-on. The third parameter specifies the page length as 50 lines

SLANG.CMD

Reconfigures the keyboard to that of another country.

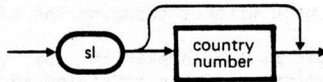


Fig. 14-39 SLANG

Where

SYNTAX ELEMENT	MEANING
country number	A number corresponding to the keyboard configuration of a particular country. See menu below

Characteristics

For the SLANG command to be executed, the files "font.all" and "kb.all" must both be on an enabled volume.

The SLANG command can be executed in one of two ways:

- By entering the command without specifying the country number, in which case the following menu is displayed:

Available Country Configurations :

Italy	0	Denmark	8
West Germany	1	Yugoslavia.....	10
France	2	Norway	11
Great Britain	3	Greece	12
United States	4	Switzerland/France...	13
Spain	5	Switzerland/Germany...	14
Portugal	6	Germany (original)....	15
Sweden/Finland	7	Datav	16

Enter Your Selection by Number (or q to quit) -->

The user must then select the country by entering the corresponding number followed by /CR/.

If the user enters a number that does not correspond to a supported language, then the prompt reappears.

Entering "q /CR/" quits the menu without changing the country number.

- By entering the country number as a parameter in the command line. In this case an invalid country number is indicated by an error message;

ERROR 76 --- error in parameter

The new keyboard configuration remains operative until either it is changed by another SLANG command, or the current working session is terminated.

The new keyboard configuration can be made permanent by means of the PSAVE command.

The keyboard layouts and the codes generated by the keys are shown for all supported national standards in Appendix B.

Examples

IF you enter...	THEN...
sl /CR/	The SLANG command displays the country number menu.
1 /CR/	The West Germany keyboard is simulated
sl 0 /CR/	The Italy keyboard is simulated

SPRINT.CMD

Loads and initializes the SPRINT utility to print the image of either the screen or a specified window.

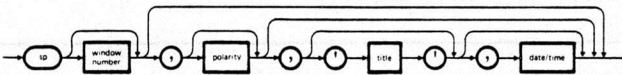


Fig. 14-40 SPRINT

Where

SYNTAX ELEMENT	MEANING	DEFAULT
window number	An integer ≥ 0 and ≤ 16 representing the window to be printed. 0 indicates the entire screen, whether or not the screen is divided into windows	0 (whole screen)
polarity	n, p, or c, describing the paper image. Negative (n) gives black on paper for white on screen, while positive (p) gives black on paper for black on screen. For color systems, all colors that are not the background color are considered to be foreground. Color (c) specifies color output to the PR 1481 printer	n (negative)
title	An alphanumeric title string, beginning with a nonnumeric character, and no longer than 24 characters: to be printed above the graphic output	no title
date/time	dt or no determining whether (dt) or not (no) the current date and time are to be printed above the graphics output	no (not required)

Example

IF you enter...	THEN...
EXEC "sp 1,,'BAR CHART',dt" /CR/	The contents of window 1 are printed beneath the heading "BAR CHART" and the date and time

Remarks

The SPRINT command can be used to draw graphic elements only on printers equipped for graphics.

After you have PLOADED SPRINT, the single keystroke /SCR PRT/ produces the same result as the SPRINT command.

Windows cannot be opened in PCOS.

SSYS.CMD

Sets system global parameters.

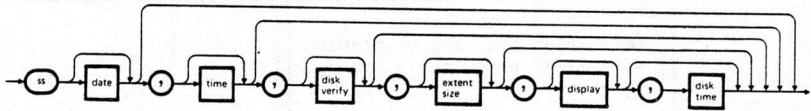


Fig. 14-41 SSYS

Where

SYNTAX ELEMENT	MEANING	RANGE	
		MIN	MAX
Date	Any legal date after 1899. Date consists of month, day and year (for USA ASCII and USA ASCII + BASIC keyboards), distinguished one from the other by a date separator. The order is day, month, year on other keyboard versions		

	month: any number, representing January as 01,...December as 12	01	12
	day: any legal day according to the month of the year	01	31
	year: any valid value in the range 1900 to 1999. (Note that only the last two digits need to be specified)	1900	1999
	date separator: any printable nonnumeric character (:, /, *, etc.)		
Time	Time consists of hour, minute and second, distinguished one from the other by a time separator		
	hour: any valid value	00	23
	minute: any valid value	00	59
	second: any valid value	00	59
	time separator: any printable nonnumeric character (/, :, etc.)		

Disk Verify	Whether or not verification is performed after each hard disk or diskette input/output operation. When verification is on, data that is written to the hard disk or diskette is subsequently read and checked. If an error is detected, "ERROR 57 --- disk i/o error" is displayed	0 (off)	1 (on)
Extent size	The number of sectors to be allocated to a file when an output operation requires additional space	1	1087
Display	Whether the 64 x 16, or 80 x 25 display mode is in effect	0 (64x16)	1 (80x25)
Disk time	The number of seconds that the diskette drive motor will remain on after accessing that drive	1	30

Default Values

At initialization from a standard PCOS, parameters attain default values shown in the following table. However, the user can assign these parameters different values at initialization by using a user-configured PCOS, or by using an initialization program (for example, INIT.BAS), which sets the parameters by calling the SSYS command.

In each case the parameter values can be changed by the user at any subsequent moment by using the SSYS command. Once modified, a parameter maintains the last value assigned to it until it is again modified by another SSYS command, or until the system is reinitialized. In an SSYS command, if a nil parameter is entered, it will revert to the value current before the command was entered.

The following table gives the standard default values:

PARAMETER	DEFAULT VALUE	MEANING
Date	01/01/84	January 1 1984, and the date separator is a slash (/)
Time	00:00:01	System time starts from 1 second past midnight and the time separator is a colon (:)
Disk Verify	0 (off)	No verification after any disk or diskette read/write operation
Extent size	8	Eight logically contiguous sectors will be allocated to a file when an output operation requires additional space
Display	0	16 rows of 64 characters each
Disk time	2	The diskette drive motor remains on for two seconds after the last access

Table 14-2 Set System Global Parameters Default Values

Example

IF you enter...	THEN...
ss 01/31/84, 00:00:01,0 /CR/	The command sets the date to January 31 1984, the time to one second after midnight and prescribes no verification after a read/write operation

The remaining unspecified parameters (extent size, display, and disk time) will take either their default values or the values specified in the last SSYS command that was either followed by a PSAVE command (if a user-configured PCOS is being used) or entered during the current working session

TLANG.SAV

Enables translation of SLANG character codes between PCOS and 8-bit devices and systems.

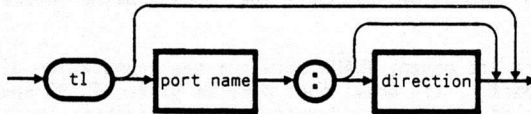


Fig. 14-42 TLANG

Where

SYNTAX ELEMENT	MEANING
port name	device name as defaulted by the system or as set by the SDEVICE utility. May be 'PRT:', 'COM:', or 'COM1:'
direction	direction of translation; one of the following: N: do not translate W: translate to 8-bit codes during output R: translate from 8-bit codes during input R/W: simultaneously do translations W and R

Characteristics

PCOS uses 7-bit ASCII codes for its various national-equivalent character codes as set by SLANG. To enhance translation between PCOS and those systems and devices using 8-bit character sets, TLANG is selectively applied to individual I/O channels of PCOS.

The TLANG utility is automatically PLOADed when first executed. A translation direction cannot be set until the corresponding device driver is loaded.

If both parameters are given, TLANG sets the new translation direction of the specified port. If only port name is given, TLANG specifies the current direction of the specified port. If no parameter is given, TLANG displays all the current non-null translation directions with corresponding port names.

Examples

IF you enter...	THEN...
t1 prt: w /CR/	PCOS 7-bit codes will be translated to 8-bit codes during output to the system printer
t1 com: r/w /CR/	<p>Translation is enabled in both directions to and from the first serial port.</p> <p>Before executing this operation, you must use the RS232 command to load the RS-232 device driver, or an error message will be displayed. ERROR 111 -invalid device name</p>

VALPHA.CMD

Sorts a volume directory into alphabetic order and frees any unused directory blocks, thus improving access time whenever the directory is scanned

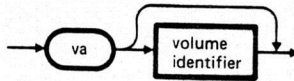


Fig. 14-43 VALPHA

Where

SYNTAX ELEMENT	MEANING
volume identifier	The volume name, or the drive number in which the volume is inserted and, if the volume is not enabled, its password. If the volume identifier is not specified the volume in the last selected drive is assumed

Example

IF you enter...	THEN...
va 1: /CR/	The directory of the diskette inserted in drive 1 will be reordered, and all the files will be listed in alphabetical order

Remarks

Any volume write-protect label present must be removed before the VALPHA command can function.

The VALPHA command orders the directory such that files specified in upper case appear before those specified in lower case. On a standard system diskette the PCOS.SAV file will therefore always be placed first on the volume directory. But note that any file that starts with an upper case letter alphabetically higher than "P" will appear before the PCOS.SAV file after using the VALPHA command.

Copies a diskette from drive to drive

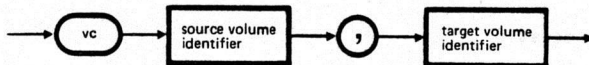


Fig. 14-44 VCOPY

Where

SYNTAX ELEMENT	MEANING
source volume identifier	The volume name or the drive number in which the diskette to be copied is inserted. It is not necessary to specify the password
target volume identifier	The volume name or the drive number in which the target volume is inserted. It is unnecessary to specify the password. The target volume must not be write-protected

Characteristics

Before copying a diskette, it is advisable to write-protect the source diskette to avoid accidentally overwriting it.

You must ensure that the copy process is being carried out in the correct direction - VCOPY will overwrite your intended source volume with the target volume, if you instruct it to do so.

This command copies the first volume onto the second volume specified, as long as the two volume identifiers do not select the same drive, and the source volume is of the same capacity as the target volume. That is, it is possible to use the VCOPY command only when copying a 640 Kbyte diskette onto another 640 Kbyte diskette, or a 320 Kbyte diskette onto another 320 Kbyte diskette.

When the command is called, the screen will display the following message

Warning - VCopy deletes all files. Copy disk? (y/n)

Entering

y /CR/

starts the process.

After VCOPY, the two diskettes will be identical (except for the diskette's creation data, which never changes throughout the life of a diskette).

It is not possible to copy a copy-protected diskette.

Examples

IF you enter...	THEN...
vc 1:,0: /CR/	The volume on drive 1 is copied onto the volume on drive 0
vc source/pass:, target: /CR/	The volume named "source" which has the password "pass" is copied to the volume named "target". Subsequently, they will both have the same name "source" and the same password "pass". It is not necessary, however, to specify the password "pass". If the volumes are to have different names, the VRENAME command must be used. If the password is not required, then VDEPASS must be used first

VDEPASS.COMD

Deletes the password from a non write-protected volume. In order to do this, the password must be known to the user.

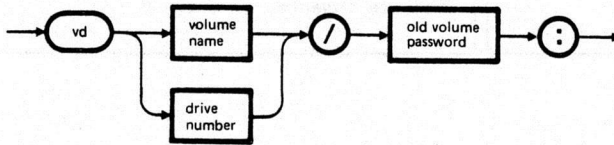


Fig. 14-45 VDEPASS

Where

SYNTAX ELEMENT	MEANING
volume name	The name of the volume whose password is to be deleted
drive number	0, 1, or 10 depending on the drive in which the diskette or hard disk resides
old volume password	The password that is to be deleted. The password must be known if it is to be deleted

Example

IF you enter...	THEN...
vd 0/mypassword: /CR/	The password "mypassword" is removed from the diskette inserted in drive 0

Remarks

VDEPASS will not operate if a volume write-protect label is present. This should be removed and replaced again after the operation.

If the password is not specified correctly, error code 72 "volume not enabled" will be displayed.

Once a volume password has been removed, accessing the volume will be the same as for a volume that never had a password.

VLIST.CMD

Lists the directory (or a specified part thereof) of a specified volume.

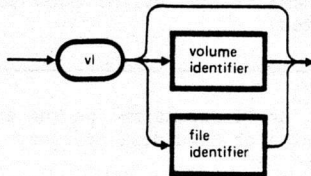


Fig. 14-46 VLIST

Where

SYNTAX ELEMENT	MEANING
volume identifier	The name of the volume to be listed or the drive number. It is not necessary to specify the volume password. If no volume identifier is specified the last selected drive is assumed
file identifier	The name of a file, including any necessary volume identifier. Groups of files can be specified using the wild card facility, or a list of files may be specified. It is not necessary to specify the file password

Characteristics

In a volume list, the system displays the following information:

- The drive in which the volume is inserted.
- The name of the volume (if it exists).
- The requested file list, specifying for each file:
 - . Its name.
 - . Its length (in bytes).
 - . Sectors used.
 - . Sectors allocated.
 - . The number of extents.
 - . Whether write-protected or not (shown as WP).
 - . Whether a password exists or not (shown as /PW).
- The total number of files, the total number of blocks allocated and used, and the total number of extents used by the files listed.
- The total free space (in blocks) available on the volume.

If the number of files is too great to fit them all onto one screen of

data, the list is displayed one screen at a time. A message is then displayed to prompt you to hit any key to examine the next screenful.

If the no-interaction (%n) flag is used, then the list is scrolled until the last screen of data is displayed.

If the file identifier is not specified, then all the files resident on the volume in question are listed.

For a typical example of a volume list, refer to Chapter 9.

Examples

IF you enter...	THEN...
vl /CR/	The system displays the directory of the volume inserted in the last drive selected
vl 1:b* /CR/	The system displays a volume list of all the files with names that start with "b", which are resident on the diskette inserted in drive 1
vl 10: /CR/	The system displays a volume list of all files on the hard disk

VMOVE.SAV

Copies a diskette using only one drive.

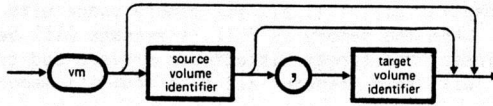


Fig. 14-47 VMOVE

Where

SYNTAX ELEMENT	MEANING
source volume identifier	The volume name or drive number (must be drive 0) of the diskette to be copied, plus any necessary volume password
target volume identifier	The volume name or drive number (must be drive 0) of the diskette to which the source volume is to be copied, plus any necessary volume password

Characteristics

Before copying a diskette, it is advisable to write-protect the source diskette to prevent accidentally overwriting it.

As VMOVE can do all that VCOPY does, using only one drive, VMOVE is useful on single drive systems, or for copying diskettes on hard disk systems. It cannot be used to copy the hard disk.

If neither the source nor the target volume is password-protected, then the command name alone is sufficient. If the source volume is password-protected, then the full source volume identifier - including the password - must be specified in the command line. If the target volume is password-protected, then both the source and target volume identifiers must be specified.

When VMOVE is called, the system displays a message warning the user that VMOVE deletes all files and PCOS, and asks the user to confirm the

intention. Following a "y /CR/" (yes) reply, the system displays another message asking the user to insert the source diskette in drive 0, and to hit any key.

At this point the computer will fill all its memory space with data from the source diskette. When the memory is full, a message will be displayed asking the user to insert the target diskette in drive 0 and to hit any key. The computer will then transfer all data from its memory onto the target diskette.

This process will be repeated (typically) two more times. At the end of the copy operation, you can either continue to make more copies, or reboot PCOS.

Note: After VMOVE, PCOS has to be rebooted because VMOVE uses all memory space as a buffer, thus overwriting PCOS.

The target diskette must not have a write-protection label applied.

Any name and/or password assigned to the source volume will be copied to the target volume.

VNEW.COMD

Initializes an unprotected or enabled volume with the option to name the volume.

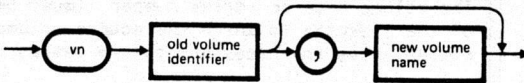


Fig. 14-48 VNEW

Where

SYNTAX ELEMENT	MEANING
old volume identifier	The volume name or the drive number and, if the volume is not enabled, its password
new volume name	The new name which is to be assigned to the volume. If omitted, the new volume will have no name

Characteristics

This command can be used to initialize any diskette or the hard disk. That is, VNEW performs a logical format on the specified volume.

If a volume write-protect label is present on the diskette, it must be removed before VNEW can function.

When this command is called, the screen will display the following message:

```
Warning -vnew deletes all files. Initialize disk? y/n
```

Entering

```
y /CR/
```

starts the file system initialization immediately. Once started, the VNEW process cannot be aborted.

Once initialized, any old volume name will no longer apply.

Examples

IF you enter...	THEN...
vn mydisk:,newdisk /CR/	The diskette called "mydisk" is initialized and renamed "newdisk"
vn 10: /CR/	The hard disk is initialized
vn 0: /CR/	The diskette inserted in drive 0 is initialized

Remarks

A volume can be initialized (logically formatted) using VNEW only if it has previously been physically formatted. Diskettes supplied by Olivetti for use with PCOS are preformatted but not initialised. The hard disk must have a PCOS partition (see Chapter 12.) If the user has failed to create a PCOS partition, as the HDU, Error 57...disk I/O error is likely to occur.

If the volume is not already formatted, or if there is some error at initialization, the system will display an error message.

CAUTION: If the volume password has been forgotten, VNEW cannot be used. VNEW will not allow you to reinitialize a disk that is password-protected.

However, if you still have an M20, you can reformat the diskette on that machine using VFORMAT. This should be done if VNEW returns any of the following errors: 57, 59, 63, 72, 73, or 74, all of which indicate that there is something wrong with the state of the diskette that is best solved by fully reinitializing it.

After performing a VVERIFY destructive test (%d option), VNEW is used to reinitialize the PCOS portion of the hard disk.

VPASS.COMD

Assigns a password to an unprotected volume or changes the password of a volume that already has one.

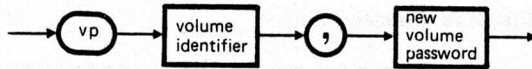


Fig. 14-49 VPASS

Where

SYNTAX ELEMENT	MEANING
volume identifier	A volume name or the number of the drive in which the volume resides and, if the volume has one, its password
new volume password	The password to be assigned to the volume. If the volume has a password, this will be replaced by the new one

Characteristics

If a volume write-protection label is present on the diskette, it must be removed before VPASS can operate.

When a volume has a password assigned, the user will not be able to use the volume without specifying the password as part of the volume identifier. Once this has been done a first time, it need not be specified again until the volume is removed and another one is referenced on the drive unit on which the volume was inserted. This point is illustrated in the following sequence:

STEP	OPERATION
1	Insert a diskette having the password "mypass" into drive 0 and rename it by entering <pre>vr 0/mypass:,newname /CR/</pre>
2	Rename it again by entering <pre>vr 0:,testdisk /CR/</pre> <p>Note that the password need not be specified as the volume is already enabled</p>
3	Remove "testdisk" from drive 0 and insert another diskette that has a password
4	Attempt to rename it by entering <pre>vr 0:,newdisk /CR/</pre> <p>The message</p> <pre>ERROR 72 volume not enabled</pre> <p>is displayed because the new diskette has not had its password specified</p>
5	Remove "newdisk" and reinsert "testdisk"

6	<p>Attempt to rename "testdisk" by entering</p> <pre style="text-align: center;">vr testdisk:,mydisk /CR/</pre> <p>and ERROR 72 is again displayed because another diskette (see step 4) has been referred to in drive zero. The original diskette must have its password restated to reenable it</p>
---	---

Example

IF you enter...	THEN...
<pre>vp myvol:,psswd /CR/</pre>	<p>The volume "myvol" is given the password "psswd"</p>

VQUICK.COMD

Lists all the file names, or a desired subset of the file names resident on a volume, and displays the number of free blocks on the volume.

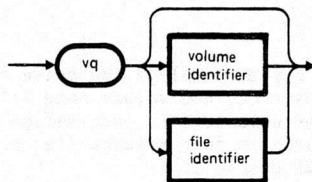


Fig. 14-50 VQUICK

Where

SYNTAX ELEMENT	MEANING
volume identifier	<p>The volume name or the drive number. If omitted, the last selected drive is assumed.</p> <p>It is not necessary to specify the volume password</p>
file identifier	<p>The name of the file plus any necessary volume identifier.</p> <p>The file name can be expressed using wild card characters to specify a group of files.</p> <p>If the file identifier is not specified, then all the file names resident on the volume in question are displayed.</p> <p>It is not necessary to specify the file password</p>

Characteristics

In a volume quick list the system displays the drive number in which the volume in question is inserted, the volume name (if it exists) and the number of free blocks on the volume. This information is followed by a list of requested file names in four columns (for a 64 x 16 display) or five columns (for an 80 x 25 display).

The no-interaction flag (%n), if used, is ignored.

A typical volume quick list is given in Chapter 9.

Examples

IF you enter...	THEN...
vq /CR/	The system displays a volume quick list. The last selected drive is assumed
vq 0:*.cmd /CR/	Only the files with the file name extension CMD and resident on the diskette inserted in drive 0 will be displayed in the list
vq 1:?s* /CR/	The only files listed are those resident on the diskette inserted in drive 1 and which have "s" as the second letter of their file name

VRENAME.COMD

Names or renames an unprotected or enabled volume.

If the new volume name is not specified, the old name will be deleted.

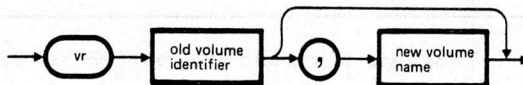


Fig. 14-51 VRENAME

Where

SYNTAX ELEMENT	MEANING
old volume identifier	The old volume name, or the drive number and, if the volume is not enabled, its password
new volume name	The name the volume is to be given. Leaving this parameter out results in the volume being given no name. It can then be referenced only by drive number (and its password if it has one)

Examples

IF you enter...	THEN...
vr 0: /CR/	The name is removed from the diskette in drive 0
vr 0:,mydisk /CR/	The name "mydisk" is assigned to the diskette inserted in drive 0
vr mydisk:,d1 /CR/	The name is changed from "mydisk" to "d1"

Remarks

Any volume write-protect label present must be removed before VRENAME can function.

A volume can be named (again, or for the first time) either by referencing the drive number, or by specifying a new name in the second parameter.

VRENAME changes only the volume name. The password is unaffected.

If a new volume name is not specified, the old one will be deleted and in future it will be possible to refer to the volume only by the drive number (followed by its password if it exists).

VVERIFY.CMD

Checks the hard disk for faulty blocks.

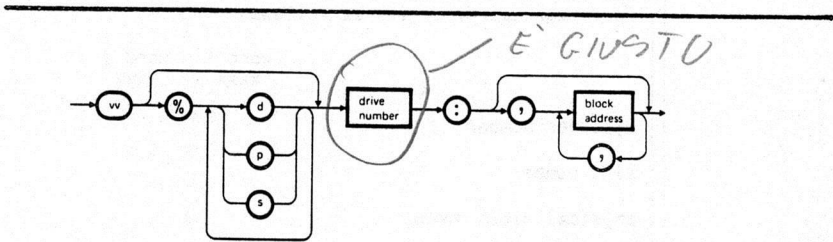


Fig. 14-52 VVERIFY

Where

SYNTAX ELEMENT	MEANING
d	Destructive test. The hard disk is checked for faulty blocks. Because the destructive test destroys all the information held on the hard disk, the user is prompted to confirm the intention. Throughout the test the user is kept informed of the test's progress by a series of messages that tell the user which side of which cylinder is currently being tested. On completion of the test, a further message displays the number of bad blocks that there are on the hard disk, but not their addresses

	<p>Following a destructive test, the PCOS partition of the hard disk cannot be reused until it has been reinitialized using the VNEW command.</p>
p	<p>Displays on the VDU, the physical addresses of the faulty blocks. No test is performed. Each block address is displayed as two words, each comprising four hexadecimal digits as follows:</p> <pre> word 1 word 2 XXXX XXXX cylinder number side number physical block number </pre>
s	<p>Block(s) specified by the block address parameter(s) are defined as being faulty. This is particularly useful if the user suspects some bad block(s) as it allows the block(s) to be added to the bad block list without performing the test.</p>
drive number	<p>The drive number of the hard disk is always 10.</p>
block address	<p>The physical block address on the disk. It must be entered as an eight-digit hexadecimal number preceded by the letter "h". The format is as follows:</p> <pre> hXXXXXXXX cylinder number side number block number </pre>

Characteristics

If VVERIFY is entered without parameters (other than the drive number parameter), then a nondestructive test is performed. Faulty blocks are listed on cylinder 0 side 0 block 1, but the information held on the hard disk remains intact. The nondestructive test, however, takes longer than the destructive test and is less exhaustive. Throughout the test, the cylinder and side number currently being tested are displayed on the VDU. The final message displays the total number of bad blocks found, and the beginning and ending cylinder numbers of the PCOS partition.

If the contents of the hard disk are no longer required, the %d option may be specified, thereby reducing execution time. Furthermore, if the user already suspects which blocks are faulty, these blocks may be added to the bad block list using the %s option, thereby not performing any test at all.

The following are the possible program flag combinations:

- %dp - performs the destructive test, then displays a list of the updated bad block list
- %pd - displays the bad block list, then performs the destructive test
- %sp - adds the specified block addresses to the bad block list, then displays the updated list
- %ps - displays the bad block list, then adds the specified block addresses to it

The no-interaction flag (%n) can be used to suppress interactive messages and the display of the command title. It does not suppress the display of the bad block list with the %p option, nor will it suppress the display of error messages.

The VVERIFY command can be used only on a formatted drive. Any attempt to do otherwise will result in the message

ERROR 111 --- invalid device

Examples

IF you enter...	THEN ...
vv 10: /CR/	The hard disk is checked for faulty blocks. The bad block list on the PCOS partition is updated
vv %p 10: /CR/	The bad block list is displayed
vv %np 10: /CR/	The bad block list is displayed, but display of the command name is suppressed
vv %nd 10: /CR/	The hard disk is checked for faulty blocks. The entire contents of the hard disk are destroyed. Only error messages are displayed
vv %s h00010101 /CR/	Cylinder 1, side 1, block 1 is added to the bad block list. VVERIFY will check this address, and return an error message if it is not within the PCOS partition

WFONT.CMD

Makes an active font from an ASCII file which was first created by the RFONT command, then subsequently modified using the Video File Editor.

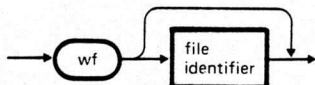


Fig. 14-53 WFONT

Where

SYNTAX ELEMENT	MEANING
file identifier	The file containing the character font matrices. This must be complete with any necessary volume identifier and file password. The file must be structured like the output from an RFONT command. If the volume is not enabled, the volume password must also be specified. If the file identifier is omitted, the system will return to the font that was active at initialization

Characteristics

At the beginning of the file is a four-line header. All but the character count line can be modified without affecting the program.

- line 1: Any text that describes the file. (The content is ignored by the WFONT command. It is there for reference only.)
- line 2: The country number that was active when the file was created. (The content is ignored by the WFONT command. It is for reference only.)
- line 3: The height (in lines) of a valid font matrix - 10 lines. (The content is ignored by the WFONT command.)
- line 4: The character count followed by at least one other word (like "characters"). The count must match the number of font matrices that follow. The minimum is 95 characters, and the maximum is 190 characters.

Example

```

USA
 4
10
95 ch

```

Each matrix is defined as follows:

- line 1: The character number. This is not read by the WFONT command. Changing this number will have no effect, as long as the line is not deleted altogether.
- line 2 - 11: Any combination of X's (upper case only) and -'s is acceptable.

Example of a Matrix

```
50
-----
----XXX-
---X---X
-----X
-----X-
-----X-
----X---
---XXXXX
-----
-----
```

When the command is entered, assuming the specified font matrix file is on an active volume and that enough memory is available, the character code will be read, converted to binary, and written into memory.

Once execution is completed, the new fonts will replace those known to the system from initialization. The system will return to the font that was active at initialization when reinitialization occurs, or when the WFONT command is invoked with no parameter.

User-defined fonts can be made permanent by means of the PSAVE command.

Examples

IF you enter...	THEN...
wf italicfont /CR/	The font matrix file named "italicfont" is converted to binary and made active
wf /CR/	The original system font is reactivated

Remarks

1. In 64 x 16 display mode, the leftmost 3 columns are generally reserved for spacing between characters and should not be used except in special cases where regular spacing between characters is not desired.
2. In 80 x 25 display mode, the leftmost two columns are not displayed at all. The third column should be left empty for spacing, unless connected characters are desired.
3. For fonts that will work well in either display mode, the user is urged to confine X's to the rightmost 5 columns.
4. If a numeric character is placed after the 8th character but not by means of the editor's "insert mode," it is ignored and no error will be issued. The same is not true however, for nonnumeric characters.
5. More font matrices can be added to a font file by adding a new font matrix at the end of the file (with a character number) and by updating the character count (4th header line). If a font matrix is inserted between two existing matrices, this serves only to increment the character numbers of all matrices beyond the one inserted after Write Font is invoked with this file. Any character numbers that appear before each matrix are not interpreted and are only for reference. The added characters are read by position. That is, the character following the one referenced to ASCII code 127, will correspond to code 128, whatever the character number is.
6. Within the range of character codes 127 through 222, many character codes do not have keys that correspond to them. If it is desired to have a key or a key combination that corresponds to one of the character codes in this range, use the CKEY command.

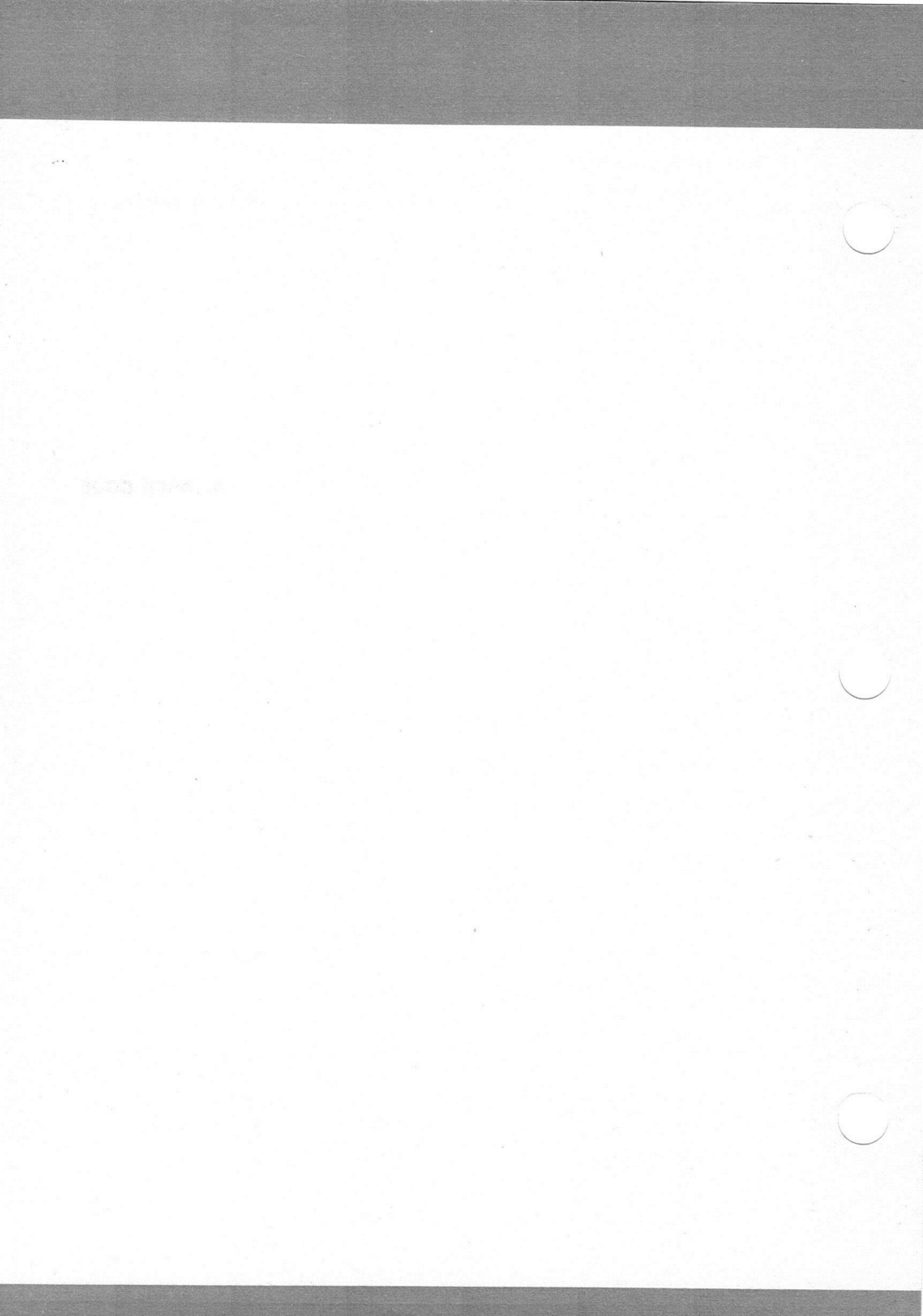
Memory Usage

The WFONT command allocates user memory each time it is invoked with a valid font file. This memory space is released either by reinitializing PCOS, or by invoking the WFONT command with no parameter. In order to save memory, it is advisable to release space allocated by the WFONT command before activating another user-defined font set.

Example

IF you enter...	THEN...
wf 1:script.font /CR/ . .	Script is made active
wf /CR/ . .	Memory for script is freed
wf 1:bold.font /CR/ . .	Bold print is made active
wf 1:script.font /CR/	Script is made active, but data for bold print cannot be released from memory without reinitializing PCOS

A. ASCII CODE



ASCII CODE

ASCII CODE

This table shows decimal (a), hexadecimal (b), and binary (c) representation of the ASCII code. (Boxed items are different on national keyboards.)

a	b	c	d	a	b	c	d	a	b	c	a	b	c
0	00	0000 0000	NUL	64	40	0100 0000	Ⓢ	128	80	1000 0000	192	C0	1100 0000
1	01	0000 0001	SOH	65	41	0100 0001	A	129	81	1000 0001	193	C1	1100 0001
2	02	0000 0010	STX	66	42	0100 0010	B	130	82	1000 0010	194	C2	1100 0010
3	03	0000 0011	ETX	67	43	0100 0011	C	131	83	1000 0011	195	C3	1100 0011
4	04	0000 0100	EQT	68	44	0100 0100	D	132	84	1000 0100	196	C4	1100 0100
5	05	0000 0101	ENQ	69	45	0100 0101	E	133	85	1000 0101	197	C5	1100 0101
6	06	0000 0110	ACK	70	46	0100 0110	F	134	86	1000 0110	198	C6	1100 0110
7	07	0000 0111	BEL	71	47	0100 0111	G	135	87	1000 0111	199	C7	1100 0111
8	08	0000 1000	BS	72	48	0100 1000	H	136	88	1000 1000	200	C8	1100 1000
9	09	0000 1001	HT	73	49	0100 1001	I	137	89	1000 1001	201	C9	1100 1001
10	0A	0000 1010	LF	74	4A	0100 1010	J	138	8A	1000 1010	202	CA	1100 1010
11	0B	0000 1011	VT	75	4B	0100 1011	K	139	8B	1000 1011	203	CB	1100 1011
12	0C	0000 1100	FF	76	4C	0100 1100	L	140	8C	1000 1100	204	CC	1100 1100
13	0D	0000 1101	CR	77	4D	0100 1101	M	141	8D	1000 1101	205	CD	1100 1101
14	0E	0000 1110	SO	78	4E	0100 1110	N	142	8E	1000 1110	206	CE	1100 1110
15	0F	0000 1111	SI	79	4F	0100 1111	O	143	8F	1000 1111	207	CF	1100 1111
16	10	0001 0000	DLE	80	50	0101 0000	P	144	90	1001 0000	208	D0	1101 0000
17	11	0001 0001	DC	81	51	0101 0001	Q	145	91	1001 0001	209	D1	1101 0001
18	12	0001 0010	DC	82	52	0101 0010	R	146	92	1001 0010	210	D2	1101 0010
19	13	0001 0011	DC	83	53	0101 0011	S	147	93	1001 0011	211	D3	1101 0011
20	14	0001 0100	DC	84	54	0101 0100	T	148	94	1001 0100	212	D4	1101 0100
21	15	0001 0101	NAK	85	55	0101 0101	U	149	95	1001 0101	213	D5	1101 0101
22	16	0001 0110	SYN	86	56	0101 0110	V	150	96	1001 0110	214	D6	1101 0110
23	17	0001 0111	ETB	87	57	0101 0111	W	151	97	1001 0111	215	D7	1101 0111
24	18	0001 1000	CAN	88	58	0101 1000	X	152	98	1001 1000	216	D8	1101 1000
25	19	0001 1001	EM	89	59	0101 1001	Y	153	99	1001 1001	217	D9	1101 1001
26	1A	0001 1010	SUB	90	5A	0101 1010	Z	154	9A	1001 1010	218	DA	1101 1010
27	1B	0001 1011	ESC	91	5B	0101 1011	[155	9B	1001 1011	219	DB	1101 1011
28	1C	0001 1100	FS	92	5C	0101 1100	\	156	9C	1001 1100	220	DC	1101 1100
29	1D	0001 1101	GS	93	5D	0101 1101]	157	9D	1001 1101	221	DD	1101 1101
30	1E	0001 1110	RS	94	5E	0101 1110	^	158	9E	1001 1110	222	DE	1101 1110
31	1F	0001 1111	US	95	5F	0101 1111	_	159	9F	1001 1111	223	DF	1101 1111
32	20	0010 0000	SPACE	96	60	0110 0000	␣	160	A0	1010 0000	224	E0	1110 0000
33	21	0010 0001	!	97	61	0110 0001	a	161	A1	1010 0001	225	E1	1110 0001
34	22	0010 0010	"	98	62	0110 0010	b	162	A2	1010 0010	226	E2	1110 0010
35	23	0010 0011	#	99	63	0110 0011	c	163	A3	1010 0011	227	E3	1110 0011
36	24	0010 0100	\$	100	64	0110 0100	d	164	A4	1010 0100	228	E4	1110 0100
37	25	0010 0101	%	101	65	0110 0101	e	165	A5	1010 0101	229	E5	1110 0101
38	26	0010 0110	&	102	66	0110 0110	f	166	A6	1010 0110	230	E6	1110 0110
39	27	0010 0111	'	103	67	0110 0111	g	167	A7	1010 0111	231	E7	1110 0111
40	28	0010 1000	(104	68	0110 1000	h	168	A8	1010 1000	232	E8	1110 1000
41	29	0010 1001)	105	69	0110 1001	i	169	A9	1010 1001	233	E9	1110 1001
42	2A	0010 1010	*	106	6A	0110 1010	j	170	AA	1010 1010	234	EA	1110 1010
43	2B	0010 1011	+	107	6B	0110 1011	k	171	AB	1010 1011	235	EB	1110 1011
44	2C	0010 1100	,	108	6C	0110 1100	l	172	AC	1010 1100	236	EC	1110 1100
45	2D	0010 1101	-	109	6D	0110 1101	m	173	AD	1010 1101	237	ED	1110 1101
46	2E	0010 1110	.	110	6E	0110 1110	n	174	AE	1010 1110	238	EE	1110 1110
47	2F	0010 1111	/	111	6F	0110 1111	o	175	AF	1010 1111	239	EF	1110 1111
48	30	0011 0000	0	112	70	0111 0000	p	176	B0	1011 0000	240	F0	1111 0000
49	31	0011 0001	1	113	71	0111 0001	q	177	B1	1011 0001	241	F1	1111 0001
50	32	0011 0010	2	114	72	0111 0010	r	178	B2	1011 0010	242	F2	1111 0010
51	33	0011 0011	3	115	73	0111 0011	s	179	B3	1011 0011	243	F3	1111 0011
52	34	0011 0100	4	116	74	0111 0100	t	180	B4	1011 0100	244	F4	1111 0100
53	35	0011 0101	5	117	75	0111 0101	u	181	B5	1011 0101	245	F5	1111 0101
54	36	0011 0110	6	118	76	0111 0110	v	182	B6	1011 0110	246	F6	1111 0110
55	37	0011 0111	7	119	77	0111 0111	w	183	B7	1011 0111	247	F7	1111 0111
56	38	0011 1000	8	120	78	0111 1000	x	184	B8	1011 1000	248	F8	1111 1000
57	39	0011 1001	9	121	79	0111 1001	y	185	B9	1011 1001	249	F9	1111 1001
58	3A	0011 1010	:	122	7A	0111 1010	z	186	BA	1011 1010	250	FA	1111 1010
59	3B	0011 1011	;	123	7B	0111 1011	[187	BB	1011 1011	251	FB	1111 1011
60	3C	0011 1100	<	124	7C	0111 1100	\	188	BC	1011 1100	252	FC	1111 1100
61	3D	0011 1101	=	125	7D	0111 1101]	189	BD	1011 1101	253	FD	1111 1101
62	3E	0011 1110	>	126	7E	0111 1110	^	190	BE	1011 1110	254	FE	1111 1110
63	3F	0011 1111	?	127	7F	0111 1111	_	191	BF	1011 1111	255	FF	1111 1111

CONFIDENTIAL - SECURITY INFORMATION
This document contains information that is exempt from public release under the Freedom of Information Act, 5 U.S.C. 552, and the Privacy Act, 5 U.S.C. 552a.

Serial Number	Date	Time	Location	Remarks
1	10/15/68	1400	10101	Normal
2	10/15/68	1405	10101	Normal
3	10/15/68	1410	10101	Normal
4	10/15/68	1415	10101	Normal
5	10/15/68	1420	10101	Normal
6	10/15/68	1425	10101	Normal
7	10/15/68	1430	10101	Normal
8	10/15/68	1435	10101	Normal
9	10/15/68	1440	10101	Normal
10	10/15/68	1445	10101	Normal
11	10/15/68	1450	10101	Normal
12	10/15/68	1455	10101	Normal
13	10/15/68	1500	10101	Normal
14	10/15/68	1505	10101	Normal
15	10/15/68	1510	10101	Normal
16	10/15/68	1515	10101	Normal
17	10/15/68	1520	10101	Normal
18	10/15/68	1525	10101	Normal
19	10/15/68	1530	10101	Normal
20	10/15/68	1535	10101	Normal
21	10/15/68	1540	10101	Normal
22	10/15/68	1545	10101	Normal
23	10/15/68	1550	10101	Normal
24	10/15/68	1555	10101	Normal
25	10/15/68	1600	10101	Normal
26	10/15/68	1605	10101	Normal
27	10/15/68	1610	10101	Normal
28	10/15/68	1615	10101	Normal
29	10/15/68	1620	10101	Normal
30	10/15/68	1625	10101	Normal
31	10/15/68	1630	10101	Normal
32	10/15/68	1635	10101	Normal
33	10/15/68	1640	10101	Normal
34	10/15/68	1645	10101	Normal
35	10/15/68	1650	10101	Normal
36	10/15/68	1655	10101	Normal
37	10/15/68	1700	10101	Normal
38	10/15/68	1705	10101	Normal
39	10/15/68	1710	10101	Normal
40	10/15/68	1715	10101	Normal
41	10/15/68	1720	10101	Normal
42	10/15/68	1725	10101	Normal
43	10/15/68	1730	10101	Normal
44	10/15/68	1735	10101	Normal
45	10/15/68	1740	10101	Normal
46	10/15/68	1745	10101	Normal
47	10/15/68	1750	10101	Normal
48	10/15/68	1755	10101	Normal
49	10/15/68	1800	10101	Normal
50	10/15/68	1805	10101	Normal
51	10/15/68	1810	10101	Normal
52	10/15/68	1815	10101	Normal
53	10/15/68	1820	10101	Normal
54	10/15/68	1825	10101	Normal
55	10/15/68	1830	10101	Normal
56	10/15/68	1835	10101	Normal
57	10/15/68	1840	10101	Normal
58	10/15/68	1845	10101	Normal
59	10/15/68	1850	10101	Normal
60	10/15/68	1855	10101	Normal
61	10/15/68	1900	10101	Normal
62	10/15/68	1905	10101	Normal
63	10/15/68	1910	10101	Normal
64	10/15/68	1915	10101	Normal
65	10/15/68	1920	10101	Normal
66	10/15/68	1925	10101	Normal
67	10/15/68	1930	10101	Normal
68	10/15/68	1935	10101	Normal
69	10/15/68	1940	10101	Normal
70	10/15/68	1945	10101	Normal
71	10/15/68	1950	10101	Normal
72	10/15/68	1955	10101	Normal
73	10/15/68	2000	10101	Normal
74	10/15/68	2005	10101	Normal
75	10/15/68	2010	10101	Normal
76	10/15/68	2015	10101	Normal
77	10/15/68	2020	10101	Normal
78	10/15/68	2025	10101	Normal
79	10/15/68	2030	10101	Normal
80	10/15/68	2035	10101	Normal
81	10/15/68	2040	10101	Normal
82	10/15/68	2045	10101	Normal
83	10/15/68	2050	10101	Normal
84	10/15/68	2055	10101	Normal
85	10/15/68	2100	10101	Normal
86	10/15/68	2105	10101	Normal
87	10/15/68	2110	10101	Normal
88	10/15/68	2115	10101	Normal
89	10/15/68	2120	10101	Normal
90	10/15/68	2125	10101	Normal
91	10/15/68	2130	10101	Normal
92	10/15/68	2135	10101	Normal
93	10/15/68	2140	10101	Normal
94	10/15/68	2145	10101	Normal
95	10/15/68	2150	10101	Normal
96	10/15/68	2155	10101	Normal
97	10/15/68	2200	10101	Normal
98	10/15/68	2205	10101	Normal
99	10/15/68	2210	10101	Normal
100	10/15/68	2215	10101	Normal

**D. DIAGNOSTIC/BOOTSTRAP ERROR
MESSAGES**

ABOUT THIS APPENDIX

This appendix provides an explanation of the error messages that might occur during the diagnostic/bootstrap process.

CONTENTS

DIAGNOSTIC ERROR MESSAGES D-1

BOOTSTRAP ERROR MESSAGES D-4

DIAGNOSTIC/BOOTSTRAP ERROR MESSAGES

At power-on or when the hardware reset button is pressed, the handshake between the 8086 processor (the CPU on the motherboard) and the Z8001 takes place.

Upon primary boot, the 8086 executes its diagnostic tests and looks for the presence of an alternate processor. If, before timeout, there is a response from an alternate processor (in this case, the Z8001 CPU), the 8086 performs various functions to initialize the Z8001.

The Z8001 then goes through the cold start procedure again and starts the diagnostic tests.

DIAGNOSTIC ERROR MESSAGES

Diagnostic error numbers and messages appear in the extreme top left corner of the screen.

DISPLAYED CHARACTER	MESSAGE TYPE	DESCRIPTION
0	Parallel printer discrete logic	The parallel printer port discrete logic (which is tested and initialized to enable detailed error reporting) has failed
1	CPU diagnostic	A malfunction has been detected during the CPU diagnostic test. Correct operation of all registers, addressing modes, and instruction classes is verified
2	ROM diagnostic	<p>A failure has been detected during the ROM diagnostic test. An error message is sent to the parallel printer port with one of the following formats:</p> <p>E RH</p> <p>or</p> <p>E RL</p> <p>indicating either "ROM high" or "ROM low"</p>
3	RAM diagnostic	<p>A malfunction has been detected during the RAM diagnostic test. An error message is sent to the parallel printer port with the following format:</p> <p>E SS 0000 ssss wwww</p> <p>where</p> <p>SS = segment number 0000 = offset from segment ssss = what data should be wwww = the actual data</p>

DIAGNOSTIC/BOOTSTRAP ERROR MESSAGES

DISPLAYED CHARACTER	MESSAGE TYPE	DESCRIPTION
4	Trap and Interrupt diagnostic	An illegal trap or interrupt has been detected.
5	LSI diagnostic	<p>An error has been detected in one of the LSI chips. An error message is sent to the parallel printer port with the following format:</p> <p>E Cx</p> <p>where x is a number indicating the nature of the error:</p> <p>1 = 8253-5 PIT 2 = 8259 PIC 3 = 8251 USARTS 4 = 8237 DMAC 5 = FUC</p>
6	Interrupt test	<p>Test of nonvectored interrupt (real-time clock) only. An error message is sent to the parallel printer port with the format</p> <p>E NI</p>
7	Floppy disk initialization	Disk initialization only; no test, no error message. Execution of the diagnostics continues
8	Keyboard test	<p>A keyboard failure has been detected. An error message is sent to the parallel printer port with the format</p> <p>E Ky</p> <p>The pattern "8" remains on the screen, and execution of the diagnostics continues</p>

At the end of the keyboard test two beep tones are heard which signal the user to input the appropriate character for the special-purpose looping. The user has approximately 5 seconds to respond before timeout. If no key is pressed, the test is considered done and the program flow jumps directly to the bootstrap.

If the 'd' key is pressed, the process loops back to the diskette initialization step and then continues to the keyboard test. There is no second pair of beep tones.

If the 'l' key is pressed, the system enters a loop which continually runs the diagnostic software and prints the passcount and any error information to the RS-232-C serial port. This loop may be interrupted by pressing any of the remaining special keys 'f', 'm', 'b', or 's'.

BOOTSTRAP ERROR MESSAGES

2bagliato

The messages listed in the following table can occur during the bootstrap procedure.

MESSAGE	MEANING
Insert system disk and type any key	Neither of the diskette drives is READY. Insert the system diskette and strike any key
Invalid File Error (xx) on drive (n)	where (xx) is 00 - invalid extent count for the file descriptor block 01 - invalid file type 02 - invalid block count 03 - end of file error 04 - parameter out of range for the diskette drive

DIAGNOSTIC/BOOTSTRAP ERROR MESSAGES

MESSAGE	MEANING
Disk Error: (xx)	<p>A diskette or hard disk error has occurred where xx is a two-digit hexadecimal number indicating the diskette or hard disk driver return code. This value must be decoded into an 8-bit number, 6 bits of which represent an error condition, as follows:</p> <ul style="list-style-type: none"> bit 7 - drive not ready <li style="padding-left: 2em;">- (most significant bit) bit 6ror.

1947

1947

1. The first part of the report is a general introduction to the subject of the study. It discusses the importance of the study and the objectives of the research. It also mentions the scope of the study and the limitations of the study.

2. The second part of the report is a detailed description of the methodology used in the study. It discusses the data sources, the data collection methods, and the data analysis methods.

3. The third part of the report is a discussion of the results of the study. It discusses the findings of the study and the implications of the findings.

E. PCOS AND BASIC ERROR MESSAGES

1964-1971 MONTHLY STATE OF OHIO

INTRODUCTION

Errors returned from the BASIC Interpreter are not displayed with their error number (only the description is displayed). Errors returned from PCOS are always displayed with the error number; the descriptive label will only be displayed if the EPRINT command is present in memory (via execution, PLOAD or PSAVE), for example:

```
v1 %n novol /CR/
```

```
ERROR 69 --- volume name not found
```

The PCOS and BASIC error codes are listed in the table below. For each code the displayed message is given along with a descriptive comment. The table also indicates which codes apply to BASIC and which to PCOS.

PCOS AND BASIC ERRORS

ER- ROR CODE	MESSAGE	PCOS OR BASIC	COMMENT
1	NEXT without FOR	BASIC	A NEXT statement has been encountered without a matching FOR
2	syntax error	BASIC	A line has been encountered which includes an incorrect sequence of characters (misspelled keyword, incorrect punctuation etc.)
3	RETURN without GOSUB	BASIC	A RETURN has been encountered for which there is no previous unmatched GOSUB statement
4	out of data	BASIC	A READ statement has been executed when there are no DATA statements with unread data remaining in the program

5 illegal function call	BASIC	<p>A parameter that is out of range has been passed to a numeric or a string function.</p> <p>Such an error may occur when:</p> <ol style="list-style-type: none"> a. An array subscript is either negative or too big b. A log function is assigned a negative or a null argument c. The SQR function is assigned a negative value d. A negative number has an exponent which is not an integer e. An incorrect argument has been made in one of the following functions: MID\$, LEFT\$, RIGHT\$, TAB, SPC, STRING\$, SPACE\$, INSTR, or ON...GOTO
6 overflow	BASIC	<p>The result of a calculation is too large to be represented in BASIC's number format.</p> <p>Note: With underflow, the result is taken as zero, and execution continues without indication of an error</p>
7 out of memory	PCOS OR BASIC	<p>A program is too big; or has too many loops, GOSUBS, variables; or has expressions too complicated to evaluate, or a command or Assembler routine has been called that cannot be accommodated in the current memory available</p>
8 undefined line number	BASIC	<p>A line reference is to a non-existent line from a GOTO, GOSUB, IF..THEN..ELSE or DELETE</p>

9	out of range	BASIC	An array element has been referred to either with a subscript that is outside the dimensions of the array or with the wrong number of subscripts
10	duplicate definition	BASIC	Two DIM statements have been given for the same array, or a DIM statement has been applied to an array after the default dimension of 10 was previously established for that array
11	division by zero	BASIC	A division by zero has been encountered or the value zero has been raised to a negative power. In the former case the result is machine infinity (with the appropriate sign) and in the latter case the result is positive machine infinity
12	illegal direct	BASIC	A statement which is invalid in immediate mode has been entered as an immediate command
13	type mismatch	PCOS OR BASIC	A string value has been entered when a numeric value is required or vice versa
14	out of string space	BASIC	String variables have caused BASIC to exceed the amount of free user memory remaining. (BASIC will allocate space dynamically until it runs out of memory)
15	string too long	BASIC	An attempt has been made to create a string of more than 255 characters
16	string formula too complex	BASIC	A string expression is too long or too complex to be processed. It should be broken into smaller expressions

17	can't continue	BASIC	An attempt has been made to continue a program that is non-continuable: one that was halted due to an error, was modified during a break in execution, or does not exist in user memory
18	undefined function	BASIC	A function has been called that has not been previously defined
19	no RESUME	BASIC	An error-trapping routine has been entered that contains no RESUME statement
20	RESUME without error	BASIC	A RESUME statement has been encountered before an error-trapping routine is entered
21	unprintable error	BASIC	An error message is not printable. That is, it corresponds to an error with an undefined error code
22	missing operand	BASIC	An expression contains an operator but no following operand
23	buffer overflow	BASIC	An attempt has been made to enter a line with more than 255 characters
26	FOR without NEXT	BASIC	A FOR has been encountered without a matching NEXT
29	WHILE without WEND	BASIC	A WHILE has been encountered without a matching WEND
30	WEND without WHILE	BASIC	A WEND has been encountered without a matching WHILE
31	IEEE: invalid talker/ listener address	BASIC	An invalid talker/listener address has been used
32	IEEE: talker = listener address	BASIC	An attempt has been made to talk to a talker, or listen to a listener
33	IEEE: unprintable error	BASIC	An IEEE error message is not printable. That is, it corresponds to an error with an undefined error code

34	IEEE: board not present	BASIC	An attempt has been made to use IEEE on a machine which does not have the optional IEEE interface
35	window not open	PCOS OR BASIC	An attempt has been made to use a window which is not at present open
36	unable to create window	PCOS OR BASIC	The window to be created is too big or too small for its mode (graphics or text). (This error can be returned while executing an Assembly Language program)
37	invalid action verb	BASIC	An action verb has been incorrectly spelled or used
38	parameter out of range	BASIC	One or more parameters have exceeded the limits set for their range
39	too many dimensions	BASIC	An attempt has been made to use an array of more than one dimension, in graphics mode
50	field overflow	BASIC	A FIELD statement has attempted to allocate more bytes than were specified for the record length of a random file
51	internal error	BASIC	An internal malfunction has occurred. Report the conditions under which the error occurred to your support organisation
52	bad file number	BASIC	A statement or command refers to: - a file that does not have a file number within the range specified at initialisation - a file that is not open
53	file not found	PCOS OR BASIC	A LOAD, KILL or OPEN statement or a PCOS command refers to a file that does not exist on an enabled volume

54	bad file mode	PCOS OR BASIC	An attempt has been made to use random file operations (GET or PUT) with a sequential file; or to use the sequential operation LOAD with a random file; or to use an illegal file mode with OPEN, that is, not A, I, O, or R
55	file already open	PCOS OR BASIC	A sequential OPEN, O has been issued for a file that is already open, or a KILL has been applied to a file that is open
57	disk I/O error	PCOS OR BASIC	An input/output error has occurred during a disk I/O operation. It is a termination error from which PCOS/BASIC cannot recover - apply a RESET
58	file already exists	PCOS OR BASIC	The file name specified in a NAME statement, or the file name you are attempting to assign in a PCOS command is identical to a file name already in use on the volume
59	disk type mismatch	PCOS	A volume has been specified that is an invalid size for the operation
60	disk not initialized	PCOS	An attempt has been made to access a diskette or hard disk that has not been initialised
61	disk filled	PCOS OR BASIC	All disk storage space available is in use
62	end of file	PCOS OR BASIC	An INPUT statement has been executed: after all the data has been assigned, or for an empty (null) file. Note: the EOF function can be used to detect end of file
63	invalid record number	PCOS OR BASIC	The record number used with a GET or PUT statement exceeds range. That is, it is 0 or greater than 32767

PCOS AND BASIC ERROR MESSAGES

64	invalid file name	PCOS OR BASIC	An invalid form of filename has been used with KILL, LOAD, OPEN, SAVE or a PCOS command. For example - too long - includes illegal characters such as space or hyphen
66	direct statement in file	BASIC	An immediate statement has been encountered when loading an ASCII format file. The LOAD operation is terminated
67	too many files	BASIC	An attempt has been made to create a new file (using SAVE or OPEN) when the present directory is already full
68	internal error	PCOS OR BASIC	An internal malfunction has occurred. Report the conditions to your support organisation
69	volume name not found	PCOS OR BASIC	The volume name referred to does not match any diskette or hard disk currently present
70	rename error	PCOS OR BASIC	An attempt has been made to rename a volume with an invalid name
71	invalid volume number	PCOS OR BASIC	The specified volume number is invalid
72	volume not enabled	PCOS OR BASIC	The specified volume has not been enabled
73	invalid password	PCOS OR BASIC	The specified password does not match that of the file
74	illegal disk change	PCOS OR BASIC	The diskette has been changed since the last file was last used
75	write protected file	PCOS OR BASIC	An attempt has been made to write to a write-protected file

76	error in parameter	PCOS OR BASIC	One or more of the quoted parameters contains an unacceptable value
77	invalid number of parameters	PCOS OR BASIC	More than the required number of parameters have been specified
78	file not open	PCOS OR BASIC	An attempt has been made to access a file that is not open
79	printer error	PCOS OR BASIC	A printer error has been returned indicating that some operator response is required, such as out of ribbon
80	copy protected file	PCOS	An attempt has been made to copy a file that is copy-protected
61	paper empty	PCOS OR BASIC	The printer has run out of paper
82	printer fault	PCOS OR BASIC	The printer has a hardware fault
92	command not found	PCOS	An invalid keyword has been entered
99	bad load file	PCOS	The program file specified is not compatible with the PCOS version being used
101	error in time or date	PCOS	An invalid time or date has been entered
108	call user error	PCOS	An error has been encountered in a call to an Assembly Language routine or a PCOS command
110	time out	PCOS	A time-out error has occurred
111	invalid device	PCOS	The specified device name does not exist. Specifying a negative argument to a command may also cause this error, since PCOS will interpret the minus sign as an attempt to cancel device re-routing

F. PCOS 4-0 ENHANCEMENTS

ABOUT THIS APPENDIX

This appendix summarises the differences between PCOS 4-0 and PCOS 3-0.

CONTENTS

<u>PCOS ENHANCEMENTS -- RELEASE</u> <u>3-0 TO RELEASE 4-0</u>	F-1
NEW COMMANDS	F-1
ENHANCEMENTS TO EXISTING COMMANDS	F-2

PCOS ENHANCEMENTS -- RELEASE 3-0 TO RELEASE 4-0

Seven new commands have been added to the PCOS command library for release 4-0. In addition there are modifications to two existing commands.

NEW COMMANDS

CCOPY.CMD

This command enables you to copy a file from a CP/M-86 diskette to a PCOS volume, or from a PCOS volume to a CP/M-86 diskette.

CDIR.CMD

This command enables you to examine the directory of a CP/M-86 diskette from PCOS.

HBACKUP.CMD

This command enables the hard disk to be backed-up onto a series of diskettes without the need to enter a command line each time a new diskette is inserted. Moreover, it provides a means for hard disk-based files that are too large to fit onto one diskette to be backed-up.

HDISK.CMD

This command enables you to partition the hard disk for use by operating systems MS-DOS, CP/M-86 and/or UCSD p-System as well as PCOS.

HRESTORE.CMD

This command enables a back-up made using the FBACKUP command to be restored to the hard disk.

MDIR.CMD

This command enables you to examine the directory of an MS-DOS diskette from PCOS.

MCOPY.CMD

This command enables you to copy a file from an eight blocks per track MS-DOS diskette to a PCOS volume, or from a PCOS volume to an eight blocks per track MS-DOS diskette.

ENHANCEMENTS TO EXISTING COMMANDS

SFORM.CMD

This command has been enhanced to allow serial printers to be connected to the twin RS-232-C expansion ports.

SPRINT.CMD

This command has been enhanced to permit colour printing.

THE UNITED STATES OF AMERICA
DEPARTMENT OF STATE
OFFICE OF THE SECRETARY

G. GLOSSARY OF TERMS

ABOUT THIS APPENDIX

This appendix defines the terms used in this manual that vary from general EDP terminology.

CONTENTS

GLOSSARY OF TERMS

G-1

GLOSSARY OF TERMS

The following table defines the non-standard terminology in this manual and also provides a page reference.

TERM	MEANING	REFERENCE
bootable file	a file of a specific format that the bootstrap loader can load into memory to initialize the system	
device re-routing	a facility that enables input to be accepted from devices or files other than the keyboard, and output to be directed to devices and files other than the video	
diskette	a single or double-sided 5 1/4in. floppy disk	
	It may be 10 - hard disk 0 - first diskette drive 1 - second diskette drive	
environment	an operational environment in which the M24 responds to keyboard input in a specific way. Three distinct environments are mentioned in this manual PCOS BASIC Video File Editor	

extent	the number of sectors to be allocated to a file	
global command	a PCOS command that allows the user to change global parameters	
global parameter	a parameter that defines a feature of the system environment	
hard disk	a 5 1/4in. Winchester disk unit	
initialization file	a file written in either Assembly Language or BASIC that is automatically loaded and executed on system initialization. It may have one of the following names: <ul style="list-style-type: none"> - INIT.COMD - INIT.SAV - INIT.BAS 	
logical reset	a reset of all global parameters (except those controlled by the real-time clock) and reinitialization of the system (without performing diagnostic tests). It is caused by pressing /CTRL/ /ESC/, simultaneously	
nil parameter	a parameter to a command where the parameter in question is not specified in the command line. The parameter therefore assumes either a default value, or the last specified value (in the case of global commands)	
non-standard initialization	a system initialization where /L/, /D/, /F/, /B/, or /S/ is pressed during power-up diagnostics, or following a PRUN command	

GLOSSARY OF TERMS

<p>PCOS nucleus</p>	<p>that part of the operating system that is loaded into memory on initializing the system, and remains there until the working session is terminated</p>	
<p>permanent memory area</p>	<p>that part of memory occupied by the PCOS nucleus, and those commands, assembler programs, programmed key definitions and user defined fonts made permanent by a PSAVE command</p>	
<p>physical reset</p>	<p>a system reinitialization caused by pressing the physical reset button. The subsequent initialization includes diagnostic tests and a reset of all global parameters (including those controlled by the real-time clock</p>	
<p>programmed key</p>	<p>a key that has either had its associated ASCII code changed by means of a CKEY command, or had a string assigned to it by means of the PKEY command</p>	
<p>raw key code</p>	<p>the immediate code generated by a key (or key combination) corresponding to the physical position of the key on the keyboard, independent of system tables</p>	
<p>sector</p>	<p>256 bytes of available disk space</p>	
<p>semi-permanent memory area</p>	<p>that part of memory occupied by loaded commands and assembler programs, PKEYed strings and user-defined fonts that will be released on termination of the current working session</p>	

standard initialization	initialization following switch-on, physical reset, or logical reset; not having /L/, /D/, /F/, /B/ or /S/ pressed during power-up diagnostics	
standard PCOS	the PCOS configuration supplied by Olivetti on the system diskette	
text file	an ASCII file whose records are separated either by CR/LF, or by record separator (RS) characters	
transient command	a command that is not loaded into memory at initialization. This includes commands that are loaded and purged (those with CMD extension), and those that are loaded, but remain in memory after execution (those with SAV extension)	
volume	the entire contents of a diskette or hard disk	
wild card character	a special symbol used to represent any single character (?), or any string of characters (*)	
working session	the time between booting PCOS and the next boot of PCOS or switch-off	

ABOUT THIS APPENDIX

This appendix gives the keyboard changes.

CONTENTS

INTRODUCTION H-1

FUNCTION KEYS H-1

NEW KEYS H-2

NEW ACTIONS PERFORMED H-3

NEW MAPPINGS H-3

INTRODUCTION

The following is a description of the new features of the M24 PCOS Keyboard for the Olivetti Keyboard 2.

There are three main aspects to the modifications on the keyboard:

- The function keys (F1 through F18) have a mapping that is based on the M20 keyboard layout.
- All the new keys, relative to the M20, generate a PKeyable code.
- Some of the new keys, though they generate a PKeyable code, perform alternate actions.

FUNCTION KEYS

The function keys are a remapping of certain M20 keys. The way that these keys have been mapped is as follows:

M24 Key Shift	M20 Key Shift
base shift	control shift
shift shift	command shift
control shift	shift shift
command shift	base shift

This means that if you use the M24 function key without any shift, what will be returned is the equivalent of the old M20 control shift of the key just below that function key. For example, Function 1 will return what used to be /CTRL/ /1/ on the M20. This scheme works perfectly until the 14th Function key. At this point the mapping refers to the keys as follows:

Function Key	M24 Layout key
F14	7'
F15	8'
F16	9'
F17	+'
F18	-'

Note: the "'"' refers to those keys on the keypad to the right of the main layout.

NEW KEYS

There are a variety of new keys on the M24 Olivetti Keyboard 2 that did not previously exist on the M20. These keys are also given a value that corresponds to an M20 key combination. They are as follows:

M24 KEYP TOP	M20 KEYP TOP
ESC	/CTRL/ /5/
CLEAR	/CTRL/ /00/
BREAK	/CTRL/ /[/
HELP	/CTRL/ /] /
FUNCT.LOCK	/CTRL/ /,/
SCR/PRT	/CTRL/ /\ /
SCROLL ON	/CTRL/ /S /
arrow keys	/SHIFT/ /2/ /4/ /6/ /8/
CAPS LOCK	/COMMAND/ /? /

An example would be as follows: The /BREAK/ key on the M24 will generate the same code as the /CTRL/ /[/, which means that if the /CTRL/ key and the /[/ key are used at the same time, they will generate a code that is generated by the /BREAK/ key. Some of these keys had special meaning on the M20, such as /COMMAND/ /? / which would shift all the alpha keys to capitals. Moreover, some new keys perform the same as some M20 keystroke combination. For example, /SCROLL ON/ will perform the same function as the /CTRL/ /S / M20 combination.

The keys just described also have a PKEYable code and a CKEYable code that can be used for programming. Table 1 gives a description of the suggested values to give to some of these keys as an alternate. What must be understood is that if one key is CKEYed, another key may be affected by the change too.

NEW ACTIONS PERFORMED

There are new keys that perform an added activity when used. These keys are the /BREAK/, /CLEAR/, and /FUNCT.LOCK/ keys. The /BREAK/ and /CLEAR/ keys both will return the equivalent of a /CTRL/ /C/ to the user buffer. Thus, the buffer will be "cleared," or the process that is running will get a "break" character and stop.

Unlike these keys, the /FUNCT.LOCK/ key performs a more complicated action. This key has been programmed so as to act as an attractive feature in the BASIC environment. The /FUNCT.LOCK/ key will allow the keys on the keypad and the arrows to perform as editing keys in the BASIC Editor. Table H-2 describes not only how the /FUNCT.LOCK/ key acts but also how the /HELP/ key and the /SHIFT/ /HELP/ key combinations might act when programmed as suggested.

Thus, in BASIC you can build an environment that suits quite well to editing. In a session, for instance, you can begin by PKEYing a value to the /FUNCT.LOCK/ key to "edit ". Whenever you use the key from then on, it is equivalent to the "line edit," and you can enter the line number you want to edit. Typing a carriage return will enable the arrow keys until another carriage return or a break (/CTRL/ /C/) is used. An added feature is that the /ESC/ key functions like the /CTRL/ /5/ to exit out of ins(ert) editing. All of these features can add up to make line editing faster in BASIC.

NEW MAPPINGS

The basic philosophy behind the reprogramming of the M24 PCOS Keyboard driver was to get as much use out of the new keys as possible while retaining as much of the old driver as possible. Some key combinations that existed on the M20 have been modified on the M24 to make sure that this philosophy was sustained as much as possible. Table 3 shows those keys that have been modified on the M24 Keyboard 2 and their value as passed by the M24 keyboard chip, the unshifted key on the M20 to which they are mapped and the references to the M20 table, and finally, the new value returned to the PCOS environment. The last value in the table is the value to which the key should be CKEYed to get back the old M20 mapping. Striking this key is the equivalent of performing the SCREEN PRINT command with the default parameters.

Key Top	Alternative Programming
SCR PRT	CKEY &61,&7F PKEY &7F,"SP options"
/CTRL/ /S1/	CKEY &EA,&A8 to get previous M20 results; SCROL-ON will follow.
/CTRL/ /[/	CKEY &89,&FB to get previous M20 results; BREAK will follow.
/CTRL/ /00/	CKEY &EE,&B2 to get previous M20 results; CLEAR will follow.
/CTRL/ /,/	CKEY &8C, new code PKEY &new code,&FE will turn /FUNCT.LOCK/ off.
/CTRL/ /\	CKEY &61,&7F will get previous M20 results, and will turn internal screen print off.

Tab. H-1 Alternative Key Programming

Key Top	Recommended Programming
<p data-bbox="183 307 252 330">/HELP/</p> <p data-bbox="183 354 342 377">/SHIFT/ /HELP/</p> <p data-bbox="183 471 321 495">/FUNCT.LOCK/</p>	<p data-bbox="454 307 670 330">PKEY &1D, 'HELP',&D</p> <p data-bbox="454 354 705 377">PKEY &9F, 'COMMAND',&D</p> <p data-bbox="454 401 888 448">This key top should correspond to the "HELP" diskette supplied by Olivetti</p> <p data-bbox="454 471 620 495">PKEY &FE, 'EDIT'</p> <p data-bbox="454 519 874 581">This key top facilitates BASIC editing by the following rules: When in the BASIC immediate mode,</p> <ol style="list-style-type: none"> <li data-bbox="454 605 863 652">1) Depress /FUNCT. LOCK/ (Red LED is illuminated) <li data-bbox="454 675 814 699">2) Enter Line Number, then /CR/. <li data-bbox="454 722 874 793">3) Some Special Keys are now automatically translated to get BASIC editing functions: <ol style="list-style-type: none"> <li data-bbox="487 816 863 840">a) Left arrow maps into backspace <li data-bbox="487 863 831 911">b) Right arrow maps into space forward. <li data-bbox="487 934 885 981">c) /HOME/ maps into "l" to list the line and go to the beginning <li data-bbox="487 1005 885 1052">d) /END/ maps into "x" to go to the end of line and insert <li data-bbox="487 1075 852 1099">e) /INS/ maps into "i" to insert <li data-bbox="487 1122 852 1146">f) /DEL/ maps into "d" to delete <li data-bbox="487 1169 896 1216">g) (Note that /ESC/ will always terminate the insert mode)

Tab. H-2 Recommended Key Programming

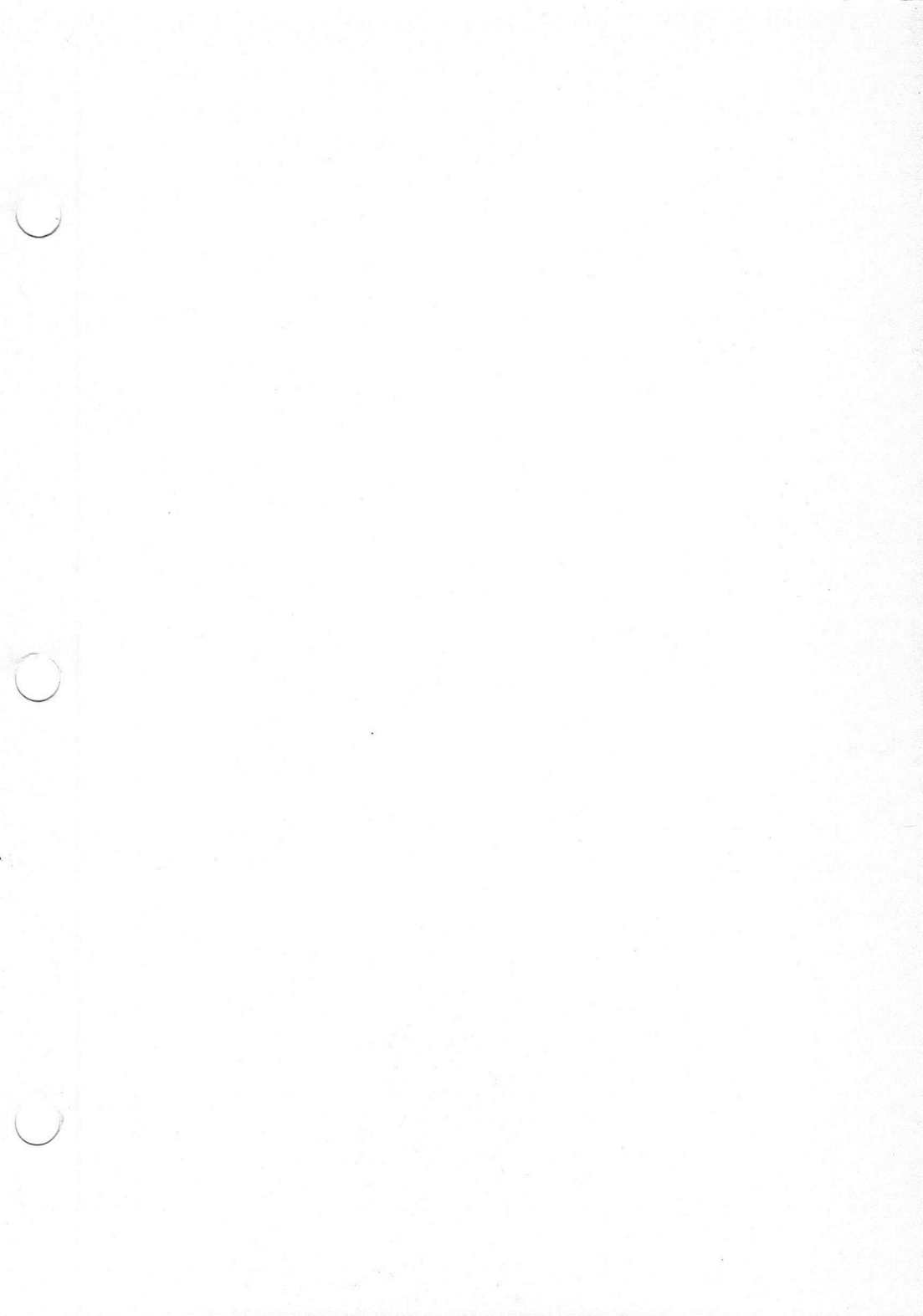
/FUNCT. LOCK/ (cont.)	4) During BASIC line editing the following keys have significance:
	a) /CR/: terminates the editing of the line and extinguishes the /FUNCT. LOCK/ LED
	b) /CTRL / /C/, /CLEAR/, or /BREAK/: Have no effect on the BASIC editor, but the /FUNCT. LOCK/ LED is extinguished and the special translations are terminated
	c) /FUNCT. LOCK/: same as (b); has no effect on BASIC, and terminates special translations and extinguishes the LED

Tab. H-2 Recommended Key Programming (cont.)

Key Code from M24	M24 to M20 PCOS Conversion	New M24 PCOS Code	Old M20 PCOS Code
/SCR PRT/ (55)	/CTRL/ /\/ (61, 91, 31, 01)	AD	7F
/FUNCT. LOCK/ (5E)	/CTRL/ /./ (8D, 8D, 5D, 2D)	FE	same
/HELP/ (29)	/CTRL/ /]/ (8C, BC, 5C, 2C)	1D	same
/CLEAR/ (5C)	/CTRL/ /00/ (EE, C6, DA, C6)	A2	B2
/BREAK/ (5D)	/CTRL/ /[/ (89, B9, 59, 29)	A2	FB
/SCROLL ON/ (56)	/CTRL/ /S1/ (EA, C2, D6, C2)	A3	A8

Tab. H-1 Key Code Equivalencies

Date	Description	Amount	Balance
1954
1954
1954
1954
1954
1954



Code 4001540 D (0)
Printed in Italy



olivetti