

M20 PERSONAL COMPUTER

PCOS (Professional Computer Operating System)

User Guide



olivetti

PREFACE

This book describes the Professional Computer Operating System (PCOS) and may be used with any M20 model.

It is directed at the user who has some experience of computer programming and is familiar with computing terminology.

The book is made-up of two parts.

Part I comprises Chapters 1 to 12 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 M20, both hardware and software.

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

Chapters 5 to 12 provide operational details of how to use PCOS.

Part II comprises Chapter 13 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.

The following are trademarks of Ing. C. Olivetti & C., S.p.A.:
OLICOM, GTL, OLITERM, OLIVORD, OLINUM, OLISTAT, OLITUTOR,
OLIENTRY, OLISORT, OLIMASTER.

MULTIPLAN is a registered trademark of MICROSOFT Inc.

MS-DOS is a trademark of MICROSOFT Inc.

CP/M and CP/M-86 are registered trademarks of Digital Research Inc.

CBASIC-86 is a trademark of Digital Research Inc.

Copyright © by Olivetti, 1983,
all rights reserved.

REFERENCE: BASIC Language
Reference Manual
Code 3982430 P

BASIC PCOS Pocket
Reference
Code 3985300 B

Installation and
Operations Guide
Code 3933340 V

I/O with External
Peripherals
Code 3982300 N

DISTRIBUTION: General (G)

EDITION: June 1983

RELEASE: 3.0

PUBLICATION ISSUED BY:

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

CONTENTS

PART I

| | | | |
|--|------|---|------|
| 1. INTRODUCTION | | <u>PERIPHERAL INTERFACES</u> | 2-12 |
| <u>INTRODUCTION</u> | 1-1 | EIA RS-232-C SERIAL INTER- FACE | 2-12 |
| 2. HARDWARE COMPONENTS | | CENTRONICS-LIKE PARALLEL | 2-12 |
| <u>MAIN UNIT</u> | 2-1 | IEEE-488 PARALLEL INTER- FACE | 2-12 |
| <u>MEMORY</u> | 2-2 | <u>PRINTERS</u> | 2-13 |
| <u>THE VIDEO DISPLAY UNIT</u> | 2-3 | | |
| <u>THE KEYBOARD</u> | 2-4 | 3. SOFTWARE COMPONENTS | |
| LINE TERMINATOR KEYS | 2-5 | <u>INTRODUCTION</u> | 3-1 |
| THE CTRL KEY | 2-5 | <u>PCOS</u> | 3-3 |
| LOCKING THE SHIFT KEY | 2-6 | MEMORY OPTIMISATION | 3-3 |
| LOGICAL RESET | 2-6 | RESIDENT AND TRANSIENT COMMANDS | 3-4 |
| ASSIGNING VALUES TO KEYS | 2-6 | PROGRAMMABLE KEYS | 3-4 |
| KEYBOARD BUFFER WARNING BUZZER | 2-7 | PROTECTION MECHANISMS | 3-4 |
| <u>PHYSICAL RESET</u> | 2-7 | LINE EDITOR FUNCTIONS | 3-5 |
| <u>HARD DISK AND DISKETTE DRIVES</u> | 2-8 | REAL-TIME CLOCK | 3-5 |
| <u>DISKETTE HANDLING</u> | 2-9 | ROUTING INPUT/OUTPUT | 3-5 |
| <u>LABELLING DISKETTES</u> | 2-9 | USER-DEFINED FONTS | 3-6 |
| <u>WRITE-PROTECTION</u> | 2-10 | CONTROL CHARACTER DISPLAY | 3-6 |
| INSERTING AND REMOVING DISKETTES | 2-11 | INITIALISATION FILES | 3-6 |
| | | USER-CONFIGURABLE OPERA- TING SYSTEM | 3-6 |

| | | | |
|-----------------------------|------|-----------------------------------|------|
| <u>BASIC INTERPRETER</u> | 3-7 | <u>COMMAND SYNTAX</u> | 4-2 |
| <u>VIDEO FILE EDITOR</u> | 3-7 | <u>COMMAND NAMES AND KEYWORDS</u> | 4-3 |
| <u>ASSEMBLER</u> | 3-8 | <u>PARAMETERS</u> | 4-3 |
| <u>PASCAL</u> | 3-9 | DEFAULT VALUES | 4-5 |
| <u>LINKER</u> | 3-9 | <u>RESIDENT AND TRANSIENT</u> | 4-5 |
| <u>PROGRAM DEBUGGER</u> | 3-9 | <u>COMMANDS</u> | |
| <u>STANDARD INTERFACE</u> | 3-9 | <u>COMMAND SEARCH PROCEDURE</u> | 4-6 |
| <u>HANDLERS</u> | | <u>FILE AND VOLUME IDENTIFI-</u> | 4-6 |
| <u>PCOS COMMAND LIBRARY</u> | 3-10 | <u>FIERS</u> | |
| CHANGING ENVIRONMENT | 3-10 | <u>WILD CARDS</u> | 4-10 |
| COMMANDS | | <u>NO INTERACTION FLAG</u> | 4-10 |
| PCOS CONFIGURING COMMANDS | 3-10 | | |
| SET SYSTEM GLOBAL COMMANDS | 3-11 | 5. INITIALISATION AND | |
| KEYBOARD-RELATED COMMANDS | 3-11 | CHANGING ENVIRONMENTS | |
| VOLUME HANDLING COMMANDS | 3-12 | <u>M20 ENVIRONMENTS</u> | 5-1 |
| FILE HANDLING COMMANDS | 3-12 | PCOS | 5-1 |
| STANDARD INTERFACE | 3-14 | BASIC | 5-1 |
| HANDLING COMMANDS | | VIDEO FILE EDITOR | 5-1 |
| PCOS GRAPHIC FACILITY | 3-14 | CHANGING ENVIRONMENTS | 5-2 |
| COMMANDS | | <u>MODES OF PCOS</u> | 5-5 |
| USER AIDS | 3-15 | COMMAND MODE | 5-5 |
| 4. ENTERING A COMMAND | | EXECUTION MODE | 5-5 |
| <u>NOTATION CONVENTION</u> | 4-1 | CHANGING MODES | 5-5 |

| | | | |
|---|------|--|------|
| <u>STANDARD INITIALISATION</u> | 5-6 | <u>SETTING THE ENVIRONMENT FOR AN RS-232-C COMMUNICATIONS PORT</u> | 6-15 |
| <u>INITIALISATION FILES</u> | 5-7 | | |
| <u>NON-STANDARD INITIALISATION</u> | 5-8 | <u>SETTING THE PRINTING ENVIRONMENT</u> | 6-15 |
| <u>INITIALISATION FOLLOWING A PSAVE COMMAND</u> | 5-9 | <u>RECONFIGURING THE KEYBOARD LANGUAGE</u> | 6-19 |
| <u>INITIALISATION FOLLOWING A PRUN COMMAND</u> | 5-9 | <u>SAVING A USER-CONFIGURED PCOS</u> | 6-21 |
| <u>THE INITIALISATION FLOW-CHART</u> | 5-9 | 7. <u>DEVICE RE-ROUTING</u> | |
| <u>BEGINNING AND ENDING A WORKING SESSION</u> | 5-12 | <u>INTRODUCTION</u> | 7-1 |
| 6. <u>CONFIGURING PCOS</u> | | <u>LOCAL DEVICE RE-ROUTING</u> | 7-2 |
| <u>INTRODUCTION</u> | 6-1 | <u>GLOBAL DEVICE RE-ROUTING</u> | 7-5 |
| <u>MAKING TRANSIENT COMMANDS RESIDENT</u> | 6-1 | <u>DEVICE RE-ROUTING FROM A BASIC PROGRAM</u> | 7-7 |
| <u>ASSIGNING STRINGS TO KEYS</u> | 6-5 | 8. <u>PROTECTION TOOLS</u> | |
| <u>CHANGING SYSTEM GLOBAL PARAMETERS</u> | 6-7 | <u>INTRODUCTION</u> | 8-1 |
| <u>SETTING THE SYSTEM GLOBAL ENVIRONMENT</u> | 6-7 | <u>VOLUME PASSWORDS</u> | 8-1 |
| <u>DISPLAYING AND MODIFYING DEVICE NAMES</u> | 6-11 | <u>FILE PASSWORDS</u> | 8-2 |
| <u>MODIFYING THE BASIC ENVIRONMENT</u> | 6-14 | <u>WRITE-PROTECTION</u> | 8-4 |
| | | <u>COPY-PROTECTION</u> | 8-4 |
| | | <u>BASIC PROGRAM SECURITY</u> | 8-5 |

9. VOLUME HANDLING

| | |
|--|------|
| <u>FORMATTING AND INITIALISING NEW VOLUMES</u> | 9-1 |
| FORMATTING A DISKETTE OR HARD DISK | 9-1 |
| INITIALISING A VOLUME | 9-3 |
| <u>LISTING A VOLUME</u> | 9-4 |
| LISTING A VOLUME USING THE VLIST COMMAND | 9-4 |
| LISTING A VOLUME USING THE VQUICK COMMAND | 9-5 |
| <u>COPYING VOLUMES</u> | 9-6 |
| COPYING VOLUMES ON A DUAL-DRIVE SYSTEM (VOLUMES OF EQUAL SIZE) | 9-7 |
| COPYING DISKETTES USING ONLY ONE DRIVE (VOLUMES OF EQUAL SIZE) | 9-8 |
| COPYING VOLUMES (OF DIFFERENT SIZES) | 9-9 |
| <u>NAMING AND PROTECTING A VOLUME</u> | 9-9 |
| WRITE-PROTECTION | 9-9 |
| PASSWORD PROTECTION | 9-10 |
| NAMING A VOLUME | 9-10 |

ALPHABETISING A VOLUME 9-11

10. FILE HANDLING

| | |
|--------------------------------------|-------|
| <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 | 10-2 |
| COPYING FILES USING ONE DRIVE | 10-5 |
| <u>LISTING FILES</u> | 10-6 |
| <u>PROTECTING FILES</u> | 10-8 |
| PASSWORD PROTECTION | 10-8 |
| WRITE-PROTECTION | 10-9 |
| FREEDING UNUSED FILE BLOCKS | 10-10 |
| <u>DELETING AND RECOVERING FILES</u> | 10-14 |
| DELETING FILES | 10-14 |
| RECOVERING DELETED FILES | 10-15 |
| <u>RENAMING FILES</u> | 10-15 |

11. PCOS GRAPHIC AND CON-
SOLE-RELATED FACILITIES

| | |
|---|-------|
| <u>INTRODUCTION</u> | 11-1 |
| <u>DISPLAYING LABELS</u> | 11-1 |
| <u>PRINTING THE SCREEN IMAGE</u> | 11-3 |
| <u>USING THE SPRINT AND LSCREEN COMMANDS</u> | 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 |
| USING THE PKEY COMMAND | 11-8 |
| <u>CREATING USER-DEFINED FONTS</u> | 11-10 |
| <u>CREATING A FONT MATRIX USING THE RFONT COMMAND</u> | 11-11 |
| <u>THE FONT MATRIX FILE</u> | 11-11 |
| <u>USING THE WFONT COMMAND</u> | 11-14 |
| <u>DISPLAYING CONTROL CHAR- ACTERS</u> | 11-15 |
| <u>THE LINE TERMINATION KEYS</u> | 11-19 |

12. VIDEO FILE EDITOR

| | |
|--|-------|
| <u>INTRODUCTION</u> | 12-1 |
| THE DISPLAY | 12-1 |
| THE KEYBOARD | 12-2 |
| <u>HOW TO INVOKE THE VIDEO FILE EDITOR</u> | 12-4 |
| EDIT.COMD | 12-4 |
| <u>GENERAL EDITING FUNCTION KEYS</u> | 12-6 |
| <u>WINDOW MOVING FUNCTION KEYS</u> | 12-12 |
| <u>EXITING AND SAVING FUNCTION KEYS</u> | 12-14 |
| <u>COMMANDS AND SEARCHING</u> | 12-15 |
| STRING SEARCHES | 12-15 |
| COMMANDS | 12-16 |

PART II

| | | | |
|-------------------|-------|---------------------------|-------|
| 13. PCOS COMMANDS | | FUNPROT.COMD | 13-40 |
| BASIC.COMD | 13-1 | FWPROT.COMD | 13-42 |
| BKEYBOARD.BAS | 13-2 | HELP.BAS | 13-44 |
| BVOLUME.SAV | 13-3 | IEEE.SAV | 13-44 |
| CI.SAV | 13-8 | LABEL.COMD | 13-45 |
| CKEY.COMD | 13-9 | LSCREEN.COMD | 13-50 |
| COMMANDS.BAS | 13-14 | LTERM (ALWAYS RESIDENT) | 13-52 |
| DCONFIG.COMD | 13-15 | PKEY.COMD | 13-53 |
| EDIT.COMD | 13-18 | PLOAD (ALWAYS RESIDENT) | 13-57 |
| EPRINT.SAV | 13-18 | PRUN.COMD | 13-59 |
| ERROR.BAS | 13-20 | PSAVE.COMD | 13-61 |
| FCOPY.COMD | 13-20 | PUNLOAD (ALWAYS RESIDENT) | 13-63 |
| FDEPASS.COMD | 13-26 | RFONT.COMD | 13-65 |
| FFREE.COMD | 13-28 | RKILL.COMD | 13-67 |
| FKILL.COMD | 13-30 | RS232.SAV | 13-69 |
| FLIST.COMD | 13-31 | SBASIC.COMD | 13-69 |
| FMOVE.COMD | 13-33 | SCOMM.COMD | 13-72 |
| FNEW.COMD | 13-35 | SDEVICE.COMD | 13-73 |
| FPASS.COMD | 13.37 | SFORM.COMD | 13.75 |
| FRENAME.COMD | 13-39 | | |

| | |
|-------------|--------|
| SLANG.CMD | 13-78 |
| SPRINT.CMD | 13-80 |
| SSYS.CMD | 13-82 |
| VALPHA.CMD | 13-86 |
| VCOPY.CMD | 13-88 |
| VDEPASS.CMD | 13-90 |
| VFORMAT.CMD | 13-91 |
| VLIST.CMD | 13-94 |
| VMOVE.SAV | 13-96 |
| VNEW.CMD | 13-98 |
| VPASS.CMD | 13-100 |
| VQUICK.CMD | 13-102 |
| VRENAME.CMD | 13-104 |
| VVERIFY.CMD | 13-106 |
| WFONT.CMD | 13-109 |

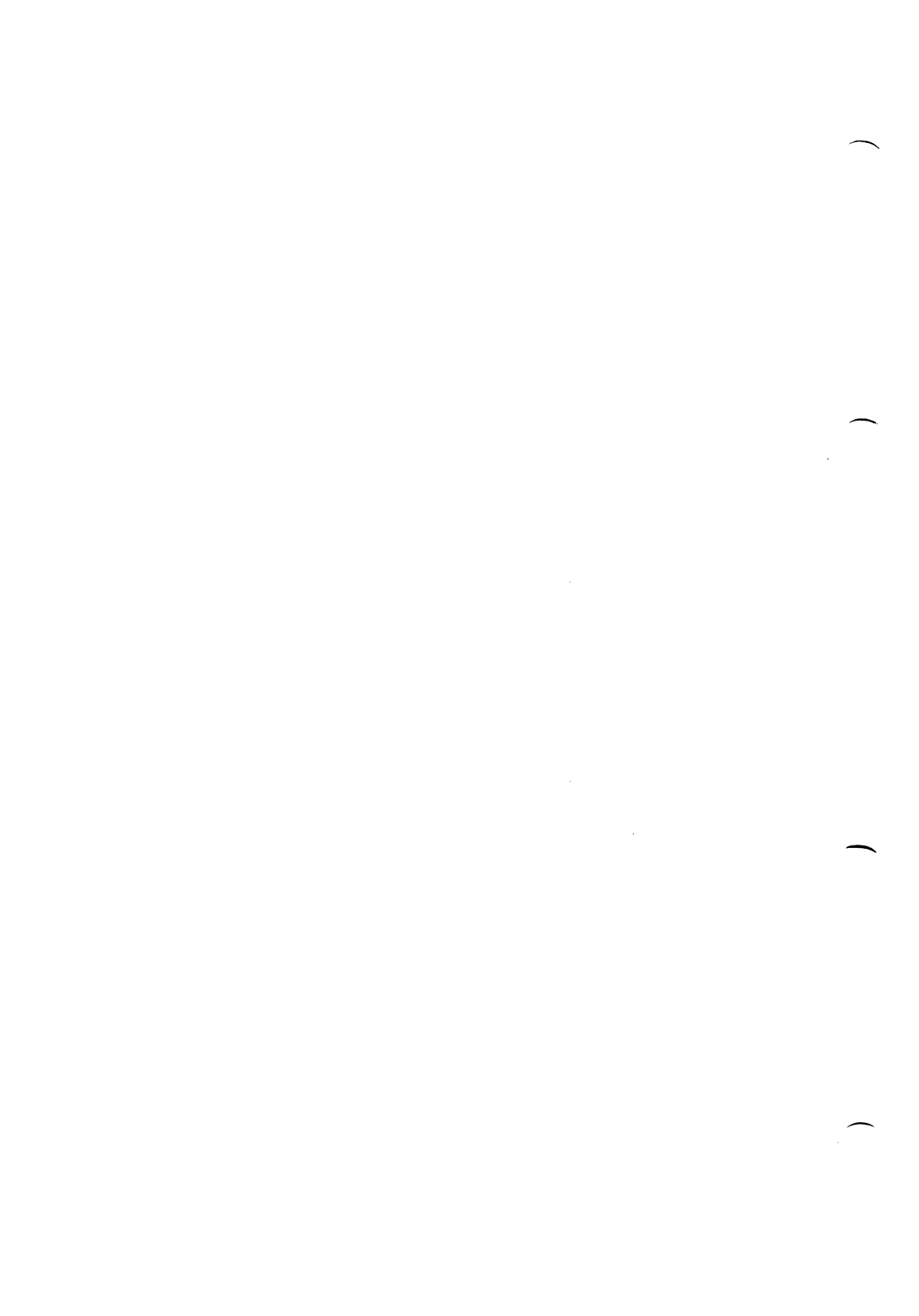
| | | | |
|--|------|--|------|
| A. ASCII CODE | | USA ASCII KEYBOARD | B-42 |
| <u>ASCII CODE</u> | A-1 | USA ASCII + BASIC KEYBOARD | B-45 |
| B. NATIONAL KEYBOARDS | | YUGOSLAVIA KEYBOARD | B-48 |
| <u>ASCII CHARACTER EQUIVALENCES</u> | B-1 | C. HARD DISK AND DISKETTE CHARACTERISTICS | |
| <u>NATIONAL KEYBOARD LAYOUTS AND CODES</u> | B-1 | <u>THE HARD DISK UNIT</u> | C-1 |
| DENMARK KEYBOARD | B-3 | CHARACTERISTICS | C-2 |
| FRANCE KEYBOARD | B-6 | <u>DISKETTES</u> | C-3 |
| GERMANY (ORIGINAL) KEYBOARD | B-9 | CHARACTERISTICS | C-4 |
| GERMANY (WEST) KEYBOARD | B-12 | D. DIAGNOSTIC/BOOTSTRAP ERROR MESSAGES | |
| GREAT BRITAIN KEYBOARD | B-15 | <u>DIAGNOSTIC ERROR MESSAGES</u> | D-1 |
| GREECE KEYBOARD | B-18 | <u>BOOTSTRAP ERROR MESSAGES</u> | D-3 |
| ITALY KEYBOARD | B-21 | E. PCOS AND BASIC ERROR MESSAGES | |
| NORWAY KEYBOARD | B-24 | <u>INTRODUCTION</u> | E-1 |
| PORTUGAL KEYBOARD | B-27 | <u>PCOS AND BASIC ERRORS</u> | E-1 |
| SPAIN KEYBOARD | B-30 | F. GET/PUT CONVERSION - PCOS 1-3 TO PCOS 2-0/3-0 | |
| SWEDEN/FINLAND KEYBOARD | B-33 | <u>GET/PUT CONVERSION</u> | F-1 |
| SWITZERLAND FRENCH KEYBOARD | B-36 | GETCONV.BAS | F-1 |
| SWITZERLAND GERMAN KEYBOARD | B-39 | | |

GLOSSARY OF TERMS

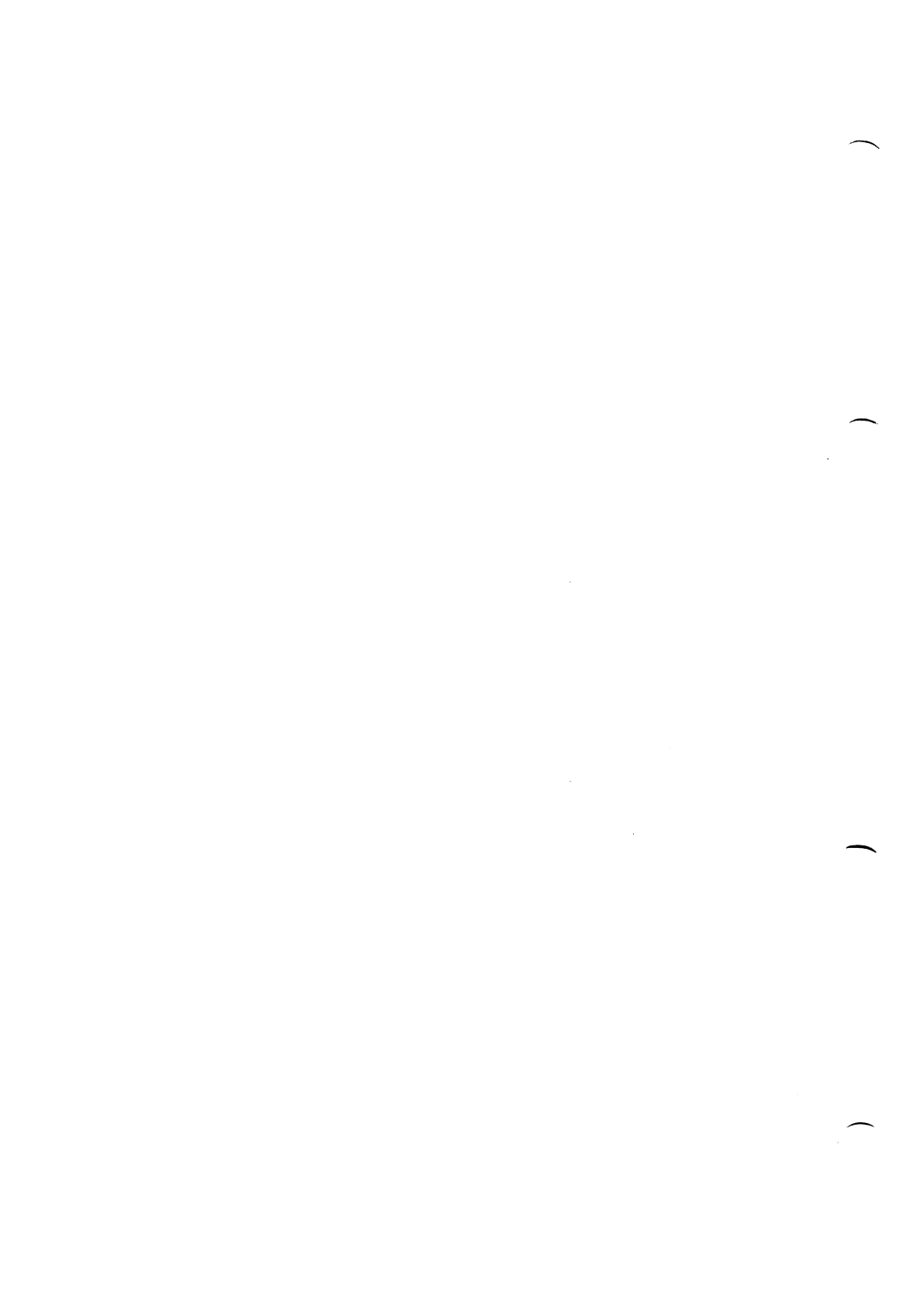
G. GLOSSARY OF TERMS G-1

H. COMMAND INDEX

COMMAND KEYWORD INDEX H-1



PART I



1. INTRODUCTION

ABOUT THIS CHAPTER

This chapter provides a general introduction to the M20.

CONTENTS

INTRODUCTION

1-1

INTRODUCTION

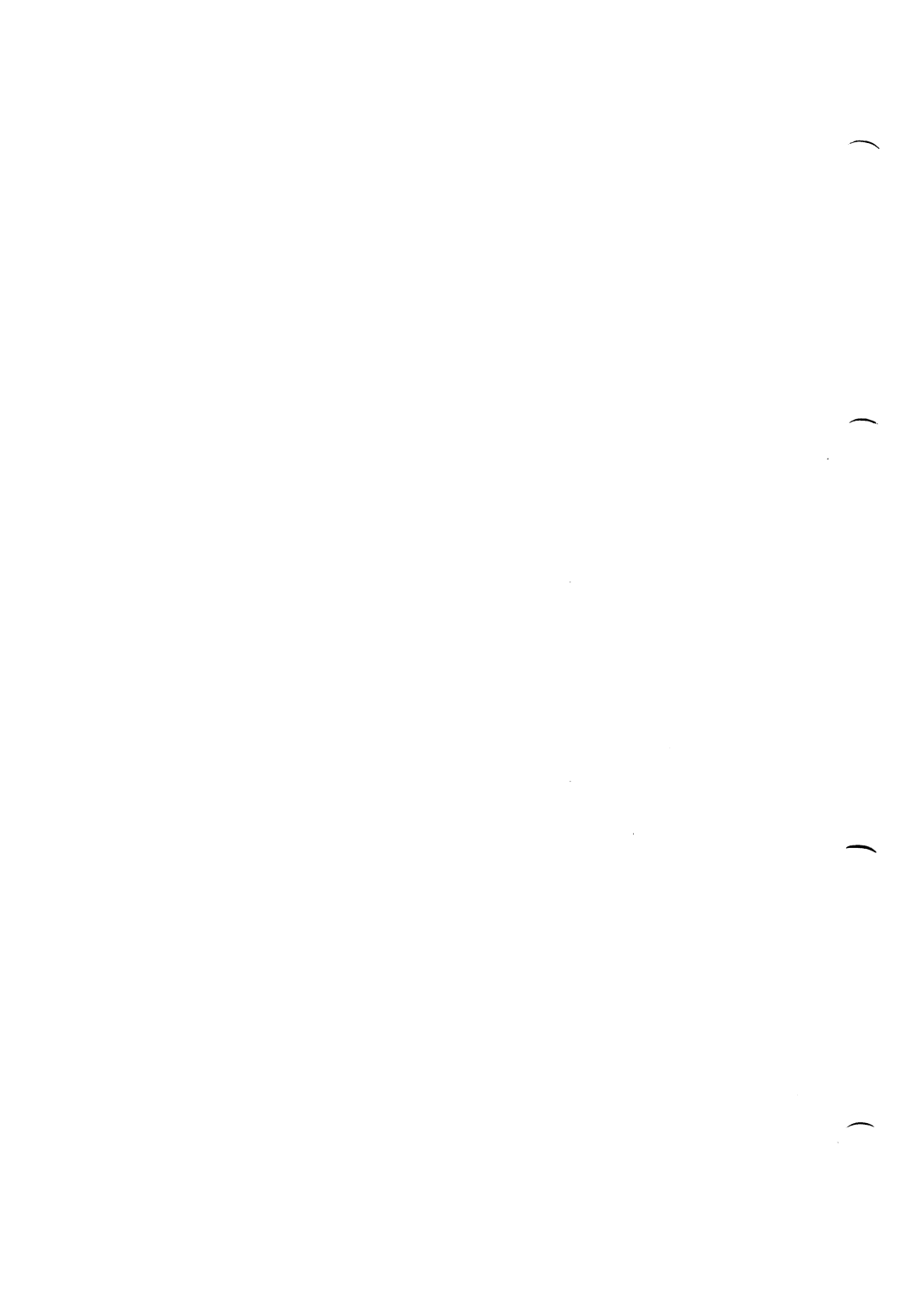
The Olivetti Model 20 (M20) is a stand-alone system designed for professional use as a problem-solving tool. It has the versatility to help the businessman, the scientist, the student and the technician to process information quickly and accurately.

Processing power is provided by a 16-bit Zilog Z8001 microprocessor in conjunction with read only and random access memory. Bulk storage is provided by 5 1/4 in. floppy disks and (optionally) a 5 1/4 in. hard disk. A keyboard and alphanumeric/graphic video (in either its monochromatic or colour version) serve for the user interface, enabling commands to be entered, prompts to be displayed, etc. Access to a range of printers, other peripherals and other computers is made possible via serial and parallel interfaces.

The whole is managed by the Professional Computer Operating System (PCOS) thus providing an environment for a set of programming tools that enable you to develop and run application programs. Programming facilities include: an extensive BASIC 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, Assembler or PASCAL to be created and modified.

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.



2. HARDWARE COMPONENTS

ABOUT THIS CHAPTER

This chapter describes the various hardware components that make up the M20. They are discussed from the viewpoint of the function performed, the options available and the physical controls.

CONTENTS

| | | | |
|--------------------------------------|-----|------------------------------------|------|
| <u>MAIN UNIT</u> | 2-1 | <u>DISKETTE HANDLING</u> | 2-9 |
| <u>MEMORY</u> | 2-2 | <u>LABELLING DISKETTES</u> | 2-9 |
| <u>THE VIDEO DISPLAY UNIT (VDU)</u> | 2-3 | <u>WRITE-PROTECTION</u> | 2-10 |
| <u>THE KEYBOARD</u> | 2-4 | INSERTING AND REMOVING DISKETTES | 2-11 |
| LINE TERMINATOR KEYS | 2-5 | <u>PERIPHERAL INTERFACES</u> | 2-12 |
| THE CTRL KEY | 2-5 | EIA RS-232-C SERIAL INTERFACE | 2-12 |
| LOCKING THE SHIFT KEY | 2-6 | CENTRONICS-LIKE PARALLEL INTERFACE | 2-12 |
| LOGICAL RESET | 2-6 | IEEE-488 PARALLEL INTERFACE | 2-12 |
| ASSIGNING VALUES TO KEYS | 2-6 | <u>PRINTERS</u> | 2-13 |
| KEYBOARD BUFFER WARNING BUZZER | 2-7 | | |
| <u>PHYSICAL RESET</u> | 2-7 | | |
| <u>HARD DISK AND DISKETTE DRIVES</u> | 2-8 | | |

MAIN UNIT

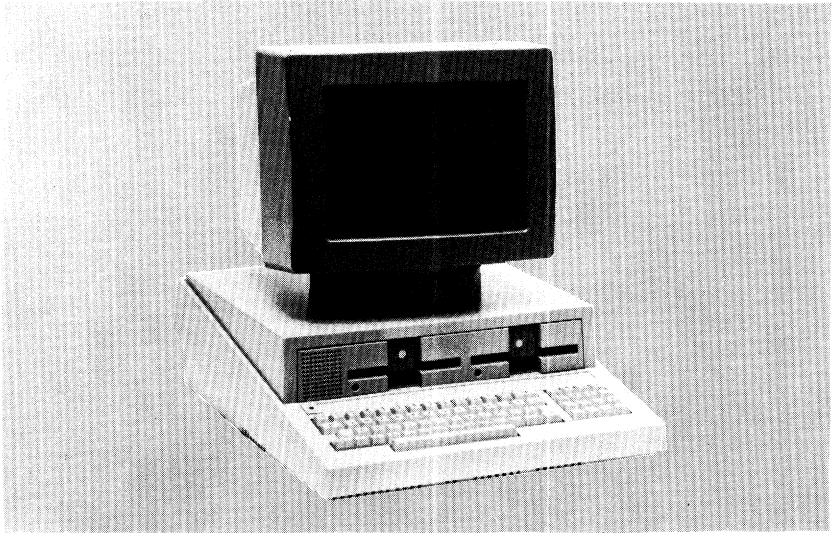


Fig. 2-1 The M20

Figure 2-1 shows the location of the major physical components of the M20. It comprises a video display unit (VDU) and a main unit.

The main unit houses the following components:

- the keyboard, comprising a standard alphanumeric keypad and an additional 16-key numeric keypad for the rapid entry of numerical data and arithmetic functions
- standard 5 1/4 in. floppy disk (diskette) drive and/or a hard disk unit
- the central processing unit (CPU); a Z8001 microprocessor
- read-only memory (ROM)
- random access memory (RAM)

The back panel of the main unit is shown in Figure 2-2.

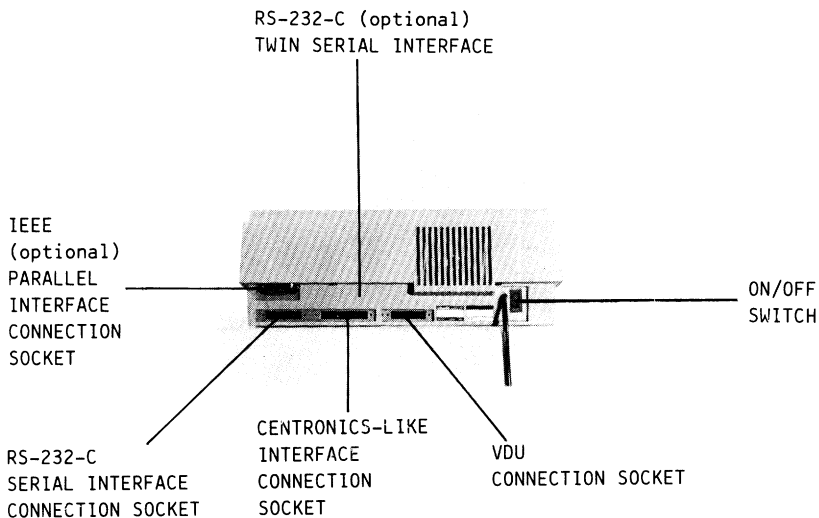


Fig. 2-2 The Back Panel

Figure 2-2 shows the location of the ON/OFF switch and the connection sockets. The latter provide connection points for the video and for peripherals utilising the standard interfaces.

MEMORY

The amount of memory that your M20 has depends on the number and type of memory expansion boards installed. The minimum configuration is one main memory board (the mother board) of 128K. Up to three 32K memory expansion boards or up to three 128K memory expansion boards may be added. 32K and 128K boards cannot be present in the same system. Four-colour systems require at least one memory expansion board (either 32K or 128K); eight-colour systems require at least two.

The mother board can contain up to four 16K blocks of read-only memory (ROM). The implemented version has 8K of ROM which contains self-testing diagnostics that are run on power-up, and the bootstrap loader. One block of RAM (16K) is used for the video bit map. The remaining seven blocks of RAM are available for the operating system, application packages, user programs, etc.

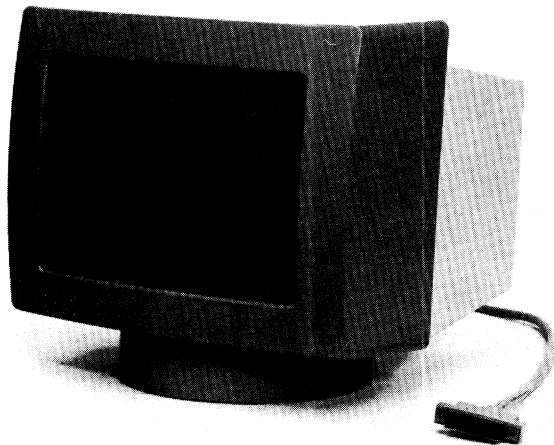
Each 32K memory expansion board adds two 16K blocks of RAM. For a four-colour system one block of the first expansion board is used for the video bit map, while the rest are added to the system memory. The video bit map for an eight-colour system requires one 16K block from each of

the first two expansion boards.

Each 128K memory expansion board adds eight blocks of RAM. Just as with 32K expansion boards, a four-colour system requires one 16K block from the first expansion board for the video bit map, and an eight-colour system requires one 16K block from each of the first two expansion boards. The remaining blocks are added to the system memory.

The memory is logically divided into segments. Each segment can contain up to 4 blocks. This is due to the architectural design of the CPU to enable different software components to occupy distinct areas of memory.

THE VIDEO DISPLAY UNIT (VDU)



BRIGHTNESS
CONTROL
(behind)

Fig. 2-3 The VDU

There are two types of VDU available with the M20:

- black and white; providing regular and reverse video
- eight colour; supporting the display of up to eight colours: black, red, green, yellow, blue, magenta, cyan and white. There are, however, two implementations of this VDU: one in which a maximum of four of these eight colours can be selected to be present at any one time; or one in which all eight can be displayed at once. This depends on the hardware present in the main unit.

Either VDU enables you to select either a 512 x 256 or a 480 x 256 pixel display; that is, 256 scanlines of either 512 or 480 pixels, where the term "pixel" is a contraction of "picture element" and "scanline" is a row of pixels.

The VDU can show up to 16 lines of 64 characters each, or 25 lines of 80 characters each. It can also display graphic images.

The VDU has a circular base on which it can be rotated and tilted to a viewing position which is comfortable for you. You can also alter the brightness of the images on the screen. This is done by adjusting the brightness control thumbwheel situated behind the top of the screen and to the right, as indicated in Figure 2-3.

THE KEYBOARD

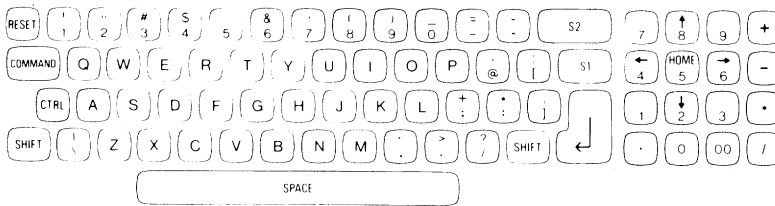


Fig. 2-4 The Keyboard

The keyboard shown in Figure 2-4 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.

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.

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 re-enter 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 right-most 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.

LOGICAL RESET

The key combination `/CTRL/ /RESET/` causes a logical reset of the system. This re-initialises the system as described in chapter 5.

ASSIGNING VALUES TO KEYS

All of the keys of the M20 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.

A "template" comes with the M20 to help you remember what values you have assigned to the keys. The template fits into the slot just above the keys.

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 re-initialised, including a set of diagnostic tests. (Note that logical reset, as explained above, does not reset all system parameters, neither does it cause diagnostic tests to be performed.) See Chapter 5 for details.

The physical reset switch is located in the small hole on the right hand side of the main unit. It is operated by inserting a ball-point pen or pencil into the hole (see Figure 2-5). A 'bleep' will be heard when contact is made.

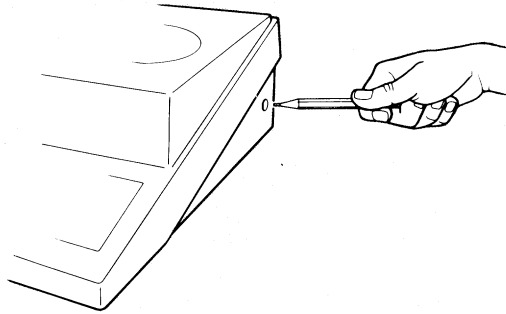


Fig. 2-5 Physical Reset

HARD DISK AND DISKETTE DRIVES

Hard disk and diskettes provide the bulk storage medium for information on the M20.

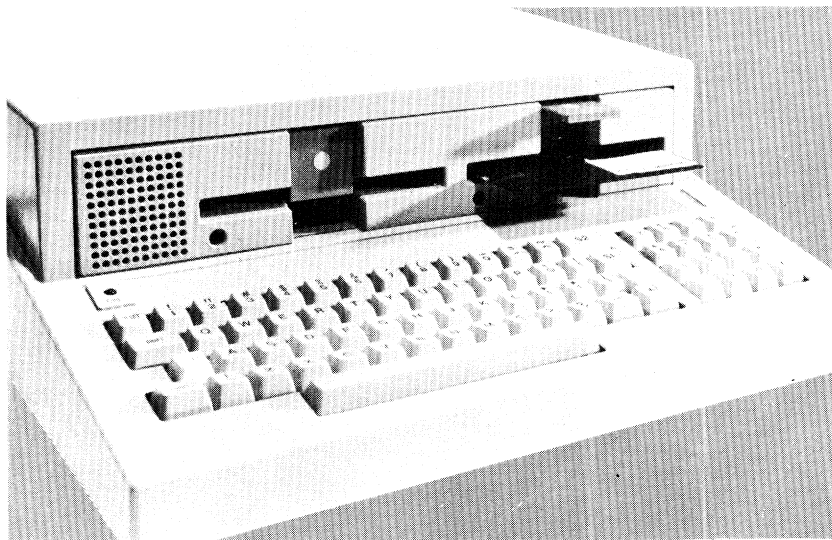


Fig. 2-6 The Diskette Drives

Figure 2-6 shows an M20 having two diskette drives - drive 0 on the right, drive 1 on the left. It also shows the position of the two indicator lights which tell you which drive is being accessed.

Your M20 will not necessarily have the configuration shown in Figure 2-6. In fact several combinations of disk and/or diskette drive are possible. Yours will be one of the following:

- one 160 Kbyte diskette drive
- one 320 Kbyte diskette drive
- one 640 Kbyte diskette drive
- two 160 Kbyte diskette drives
- two 320 Kbyte diskette drives
- two 640 Kbyte diskette drives

- one (fixed) hard disk drive (drive 10 on the right) and a 640 Kbyte diskette drive (drive 0 on the left)

Three types of diskette can be used:

- 160 Kbyte diskette (single-sided double-density)
- 320 Kbyte diskette (double-sided double-density)
- 640 Kbyte diskette (double-sided quadruple-density)

However note that while both 160 and 320 Kbyte diskettes can be used on a 320 Kbyte drive, only 160 Kbyte diskettes can be used on a 160 Kbyte drive. Moreover, while any diskette can be read via a 640 Kbyte drive, only 640 Kbyte diskettes can be written to on a 640 Kbyte drive.

DISKETTE HANDLING

Although diskettes are generally durable, damage to diskettes will be minimised 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
- keep dust out of the diskette drives by keeping the drive covers closed when not in use

LABELLING 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 aluminised write-protect labels is provided with every carton of diskettes. To apply write-protection simply fix an aluminised label over the write-protect notch as indicated in figure 2-7.

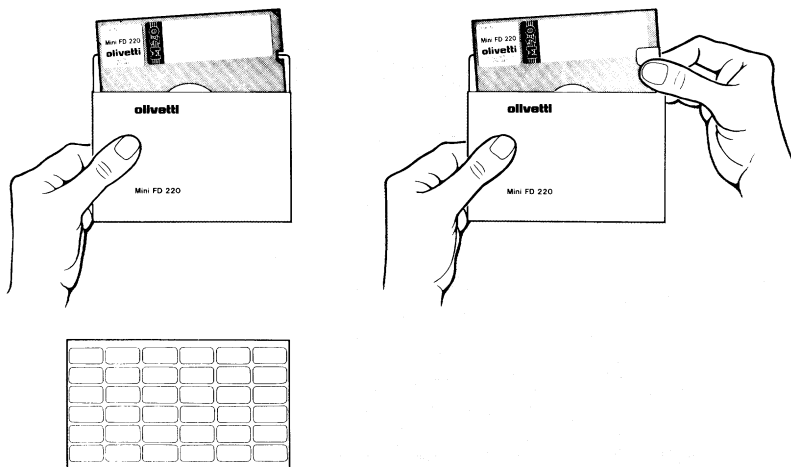


Fig. 2-7 Diskette Write-Protection

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

INSERTING AND REMOVING DISKETTES

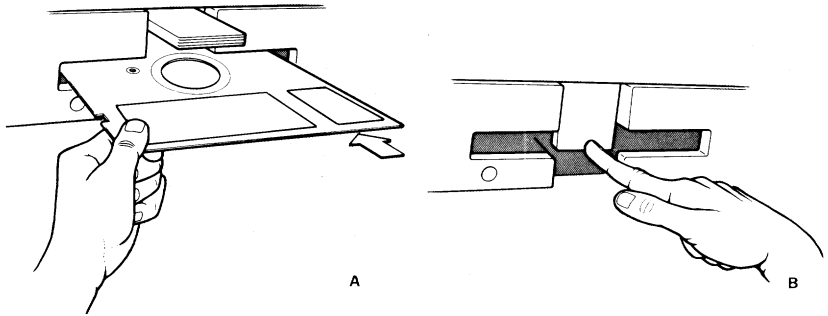


Fig. 2-8 Inserting a Diskette

To insert a diskette you do the following:

- Open the drive cover by pulling it outwards. It will spring open
- Insert the diskette into the slot with its label facing upward and nearest you (Figure 2-8A)
- 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 re-insert it
- Once the diskette has clicked into position, close the drive cover (Figure 2-8B)

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 M20 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.

PERIPHERAL INTERFACES

A standard M20 provides the following interfaces:

- EIA RS-232-C Serial Interface
- Centronics-like Parallel Interface

The following are optional:

- IEEE-488 Parallel Interface
- Twin EIA RS-232-C/20mA Current Loop Interface

EIA RS-232-C SERIAL INTERFACE

This standard I/O interface is offered on all M20 models to connect compatible devices (plotters, paper tape readers and punches, modems, etc...). It is programmable to correspond to baud rate (50 - 9600 baud), character length (in bits), presence or absence of parity, number of STOP bits in the data to be transmitted to or from the M20.

For further details refer to "I/O with External Peripherals User Guide".

CENTRONICS-LIKE PARALLEL INTERFACE

The M20 is fitted with this interface so that any compatible parallel printer may be connected to it.

To interact correctly with individual printers, the SFORM command allows the user to choose the desired print format.

IEEE-488 PARALLEL INTERFACE

This is an optional interface available on the M20.

Once the IEEE-488 driver is invoked by the IEEE command the M20 offers most of the IEEE-488 standard features to:

- read/write data from and to other devices
- assign 'talker'/'listener' status to other devices so that one talker may transmit data to several listeners

- receive service requests, conduct serial polls to identify the requesting device and respond with a user-programmed action
- transfer devices from 'remote' to 'local' control and vice versa
- act as controller and send commands to any device

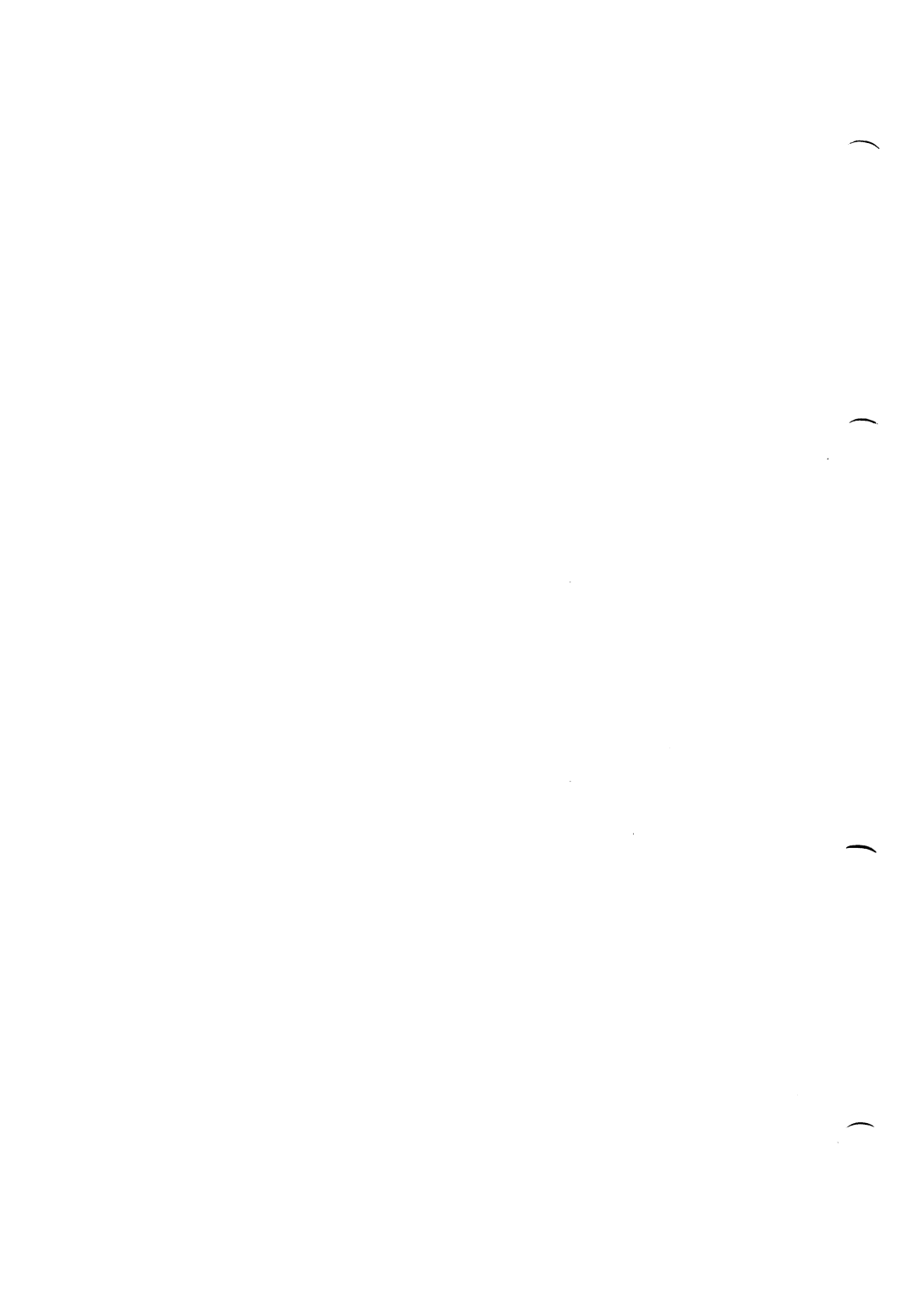
All these features are invoked by use of BASIC IEEE-488 extension statements.

For further details refer to the "I/O with External Peripherals User Guide".

PRINTERS

The M20 is compatible with a wide range of printers. These can be connected either via the Centronics-like parallel interface, or the RS-232-C interface. For details of how to install and operate your particular printer(s) consult the appropriate manual:

- PR 1450 Operator Guide
- PR 1471 Operator Guide
- PR 1481 Operator Guide
- PR 2300 Operator Guide
- PR 2400 Operator Guide
- PR 2835 Operator Guide
- PR 320 Operator Guide
- PR 430 Operator Guide
- ET 121 Operator Guide
- ET 231 Operator Guide



3. SOFTWARE COMPONENTS

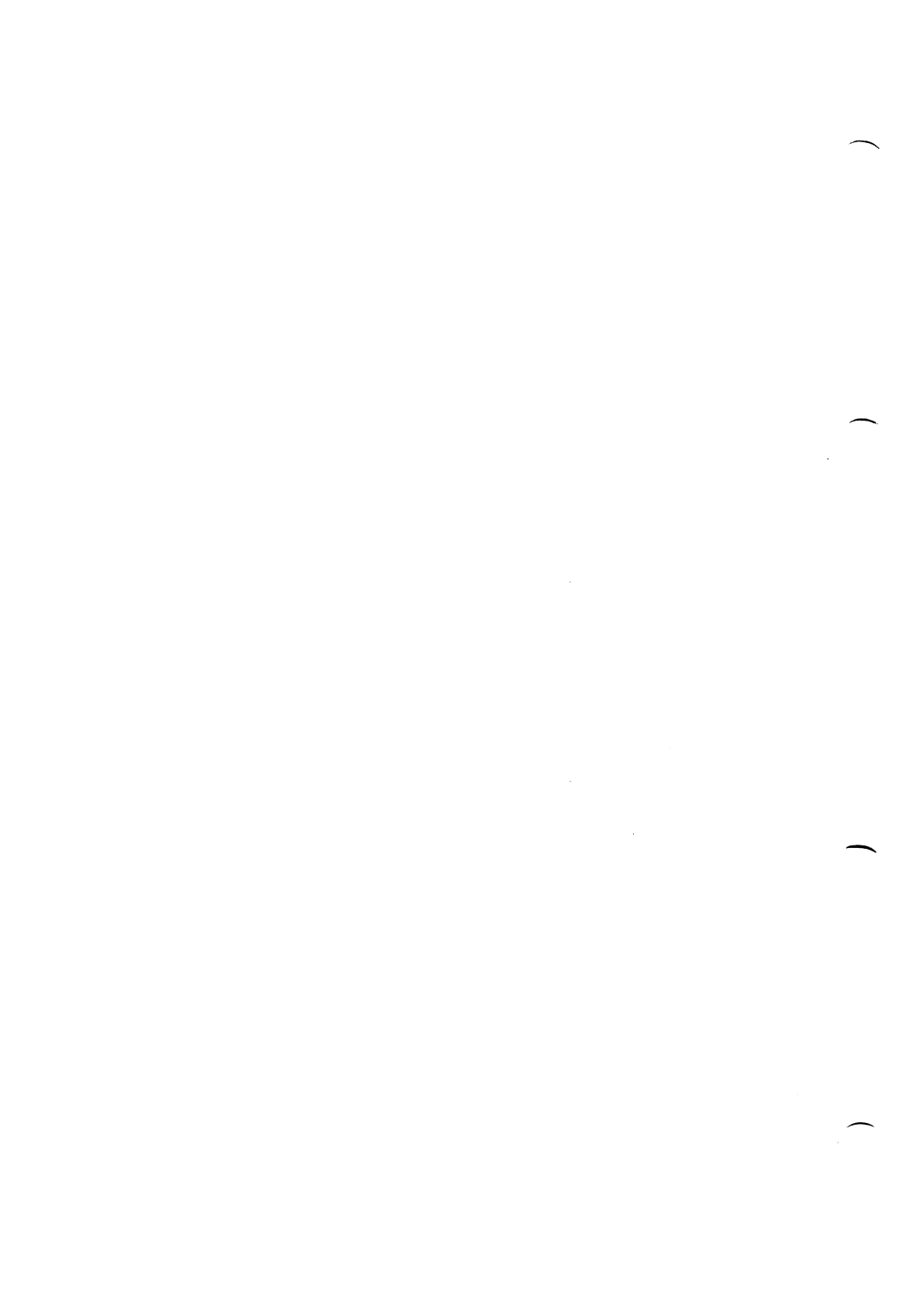
ABOUT THIS CHAPTER

This chapter describes the PCOS-related software components of the M20.

CONTENTS

| | | | |
|---------------------------------------|-----|------------------------------------|------|
| <u>INTRODUCTION</u> | 3-1 | <u>BASIC INTERPRETER</u> | 3-7 |
| <u>PCOS</u> | 3-3 | <u>VIDEO FILE EDITOR</u> | 3-7 |
| MEMORY OPTIMISATION | 3-3 | <u>ASSEMBLER</u> | 3-8 |
| RESIDENT AND TRANSIENT COMMANDS | 3-4 | <u>PASCAL</u> | 3-9 |
| PROGRAMMABLE KEYS | 3-4 | <u>LINKER</u> | 3-9 |
| PROTECTION MECHANISMS | 3-4 | <u>PROGRAM DEBUGGER</u> | 3-9 |
| LINE EDITOR FUNCTIONS | 3-5 | <u>STANDARD INTERFACE HANDLERS</u> | 3-9 |
| REAL-TIME CLOCK | 3-5 | <u>PCOS COMMAND LIBRARY</u> | 3-10 |
| ROUTING INPUT/OUTPUT | 3-5 | CHANGING ENVIRONMENT COMMANDS | 3-10 |
| USER-DEFINED FONTS | 3-6 | PCOS CONFIGURING COMMANDS | 3-10 |
| CONTROL CHARACTER DISPLAY | 3-6 | SET SYSTEM GLOBAL COMMANDS | 3-11 |
| INITIALISATION FILES | 3-6 | KEYBOARD-RELATED COMMANDS | 3-11 |
| USER-CONFIGURABLE OPERATING SYSTEM | 3-6 | VOLUME HANDLING COMMANDS | 3-12 |

| | |
|---|------|
| FILE HANDLING COMMANDS | 3-14 |
| STANDARD INTERFACE HANDLING COMMANDS | 3-14 |
| USER AIDS | 3-15 |



INTRODUCTION

The M20 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).

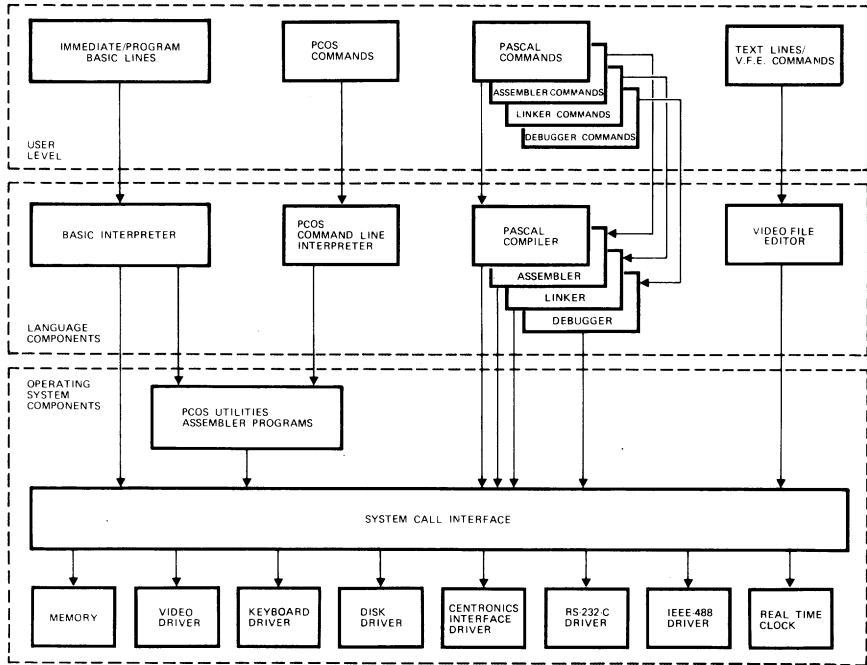


Fig. 3-1 M20 Software Components

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)
- 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 components are:

- PCOS utilities (such as BASIC.COMD, BVOLUME.SAV, ...WFONT.COMD) 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 M20 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

The M20 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 twin serial interface (as an option) for access to RS-232-C and/or 20mA Current Loop peripherals and/or computers
 - . 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 Comand Line Interpreter enables you to communicate with PCOS using a library of over 50 commands.

MEMORY OPTIMISATION

The M20 has a system of memory optimisation which is handled dynamically by allocating memory according to need and usage.

Memory optimisation 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 13.

RESIDENT AND TRANSIENT COMMANDS

In an attempt to maximise user memory space, only three commands are always loaded into memory when the system is initialised. 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).

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 M20 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 1st December 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

The M20 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 the M20 by specifying "device re-routing 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 re-initialised

For more details about device re-routing 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 re-initialised

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

INITIALISATION FILES

PCOS enables you to create an initialisation file that will be executed automatically every time the system is initialised. Any sequence of commands, programs, BASIC statements, etc., can be executed in this way to initialise 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 enable 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 re-initialising the system from the PSAVEd file, you will then restore the system to the state it was at the time it was saved.

For details on how to configure PCOS, refer to Chapter 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 "M20 BASIC Language Reference 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. These include facilities to :
 - . move the cursor around the screen
 - . insert text either as a new line between adjacent lines or within an existing line

- . delete text either one line at a time or one character at a time
- . recall a deleted line
- . delete a block of text and restore it elsewhere
- . split and join lines of text
- window moving functions. These enable you to move the window up or down the file
- exit and save functions. These enable you to:
 - . save the edited text and exit the editor
 - . exit the editor without saving the edited text
 - . save the edited text without exiting the editor
- search functions to search the file for a specified string
- a subset of "high-level" commands. These commands 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 12.

ASSEMBLER

The (optional) M20 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 "Assembler Language User Guide".

PASCAL

The (optional) M20 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. The M20 PASCAL is an extended version of the Microsoft PASCAL in that it can call run-time routines including graphics features.

For details refer to the "PASCAL Language User Guide".

LINKER

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

For details refer to either the "Assembler Language User Guide" or the "PASCAL 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 "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) twin RS-232-C interface, 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

For details refer to the "I/O with External Peripherals User Guide".

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 recognise as a keyword. This is explained in chapter 4.

For details of a particular command see chapter 13.

CHANGING ENVIRONMENT COMMANDS

| KEYWORD | | COMMAND TITLE |
|------------|---------------|-----------------------------|
| SHORT FORM | FULL MNEMONIC | |
| 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 | | COMMAND FUNCTION |
|------------|-----------------------|-----------------------------|
| SHORT FORM | FULL MNEMONIC | |
| p1 | 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

SET SYSTEM GLOBAL COMMANDS

| KEYWORD | | COMMAND FUNCTION |
|------------|---------------|---|
| SHORT FORM | FULL MNEMONIC | |
| 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 | | COMMAND FUNCTION |
|------------|---------------------|--|
| SHORT FORM | FULL MNEMONIC | |
| bk | BKEYBOARD.BAS | Activates the BASIC verbs facility |
| 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

VOLUME HANDLING COMMANDS

| KEYWORD | | COMMAND FUNCTION |
|------------|---------------|---|
| SHORT FORM | FULL MNEMONIC | |
| bv | BVOLUME.CMD | Searches the volume directory, returns free disk space, or returns the name of the current volume (from BASIC only) |
| va | VALPHA.CMD | Alphabetises a directory |
| vc | VCOPY.CMD | Copies a volume (drive to drive) |
| vd | VDEPASS.CMD | Removes a password from a volume |
| vf | VFORMAT.CMD | Formats a volume |
| vl | VLIST.CMD | Lists a volume directory (full form) |
| vm | VMOVE.SAV | Copies a volume (using one drive) |
| vn | VNEW.CMD | Initialises 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

FILE HANDLING COMMANDS

| KEYWORD | | COMMAND FUNCTION |
|------------|---------------|---|
| SHORT FORM | FULL MNEMONIC | |
| 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

STANDARD INTERFACE HANDLING COMMANDS

| KEYWORD | | COMMAND FUNCTION |
|------------|---------------|---|
| SHORT FORM | FULL MNEMONIC | |
| 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 |

Table 3-7 Standard Interface Handling Commands

PCOS GRAPHIC FACILITY COMMANDS

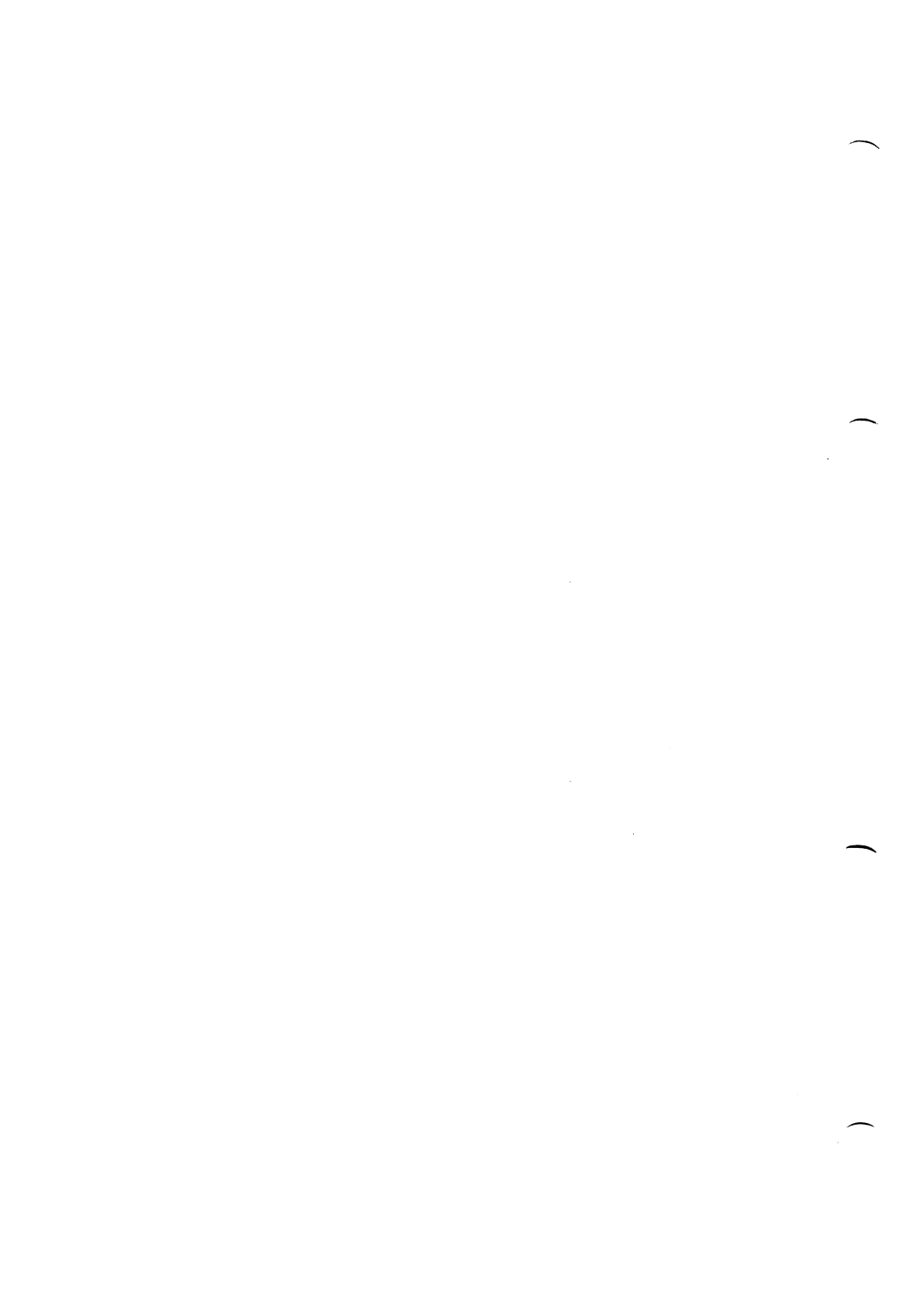
| KEYWORD | | COMMAND FUNCTION |
|------------|---------------|--|
| SHORT FORM | FULL MNEMONIC | |
| la | LABEL.CMD | Displays a label string |
| ls | LSCREEN.CMD | Prints the displayed text |
| rf | RFONT.CMD | Creates an ASCII font matrix file from the currently active font |
| sp | SPRINT.CMD | Prints 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

USER AIDS

| KEYWORD | | COMMAND FUNCTION |
|------------|---------------|--|
| SHORT FORM | FULL MNEMONIC | |
| co | COMMANDS.BAS | Displays details about PCOS commands |
| dc | DCONFIG.CMD | Displays the hardware and/or memory configuration |
| ep | EPRINT.SAV | Displays error messages |
| er | ERROR.BAS | Displays details about PCOS/BASIC error messages |
| he | HELP.BAS | Provides a series of display frames as a guide to using PCOS |

Table 3-9 User Aids



4. ENTERING A COMMAND

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 descriptions of the commands mentioned in this chapter refer to Chapter 13.

CONTENTS

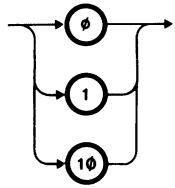
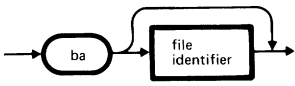
| | |
|--|------|
| <u>NOTATION CONVENTION</u> | 4-1 |
| <u>COMMAND SYNTAX</u> | 4-2 |
| <u>COMMAND NAMES AND KEYWORDS</u> | 4-3 |
| <u>PARAMETERS</u> | 4-3 |
| <u>DEFAULT VALUES</u> | 4-5 |
| <u>RESIDENT AND TRANSIENT COMMANDS</u> | 4-5 |
| <u>COMMAND SEARCH PROCEDURE</u> | 4-6 |
| <u>FILE AND VOLUME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

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 mentioned elements indicate any options which can be taken. Furthermore, a loop indicates any elements that may be repeated

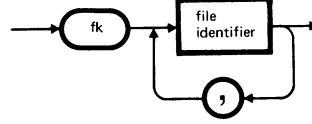
The following table shows some examples of the use of syntax diagrams. Note in particular the use of the flow lines.

| 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 by-pass 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:

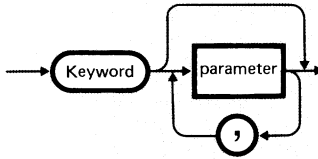


Fig. 4-1 General Format of a PCOS Command

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, user-written Assembly Language and PASCAL programs are all executed in this way. For details on executing Assembly Language programs refer to the "Assembler Language User Guide". For details of executing PASCAL programs refer to the "PASCAL User Guide".

PARAMETERS

Parameters are user-selected strings of alphabetic characters and of integers which can be optional. They are recognised 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="289 619 862 686"> <tr> <td data-bbox="289 619 540 639">+ (plus)</td> <td data-bbox="551 619 862 639">" (double quotation mark)</td> </tr> <tr> <td data-bbox="289 639 540 660">& (ampersand)</td> <td data-bbox="551 639 862 660">' (single quotation mark)</td> </tr> <tr> <td data-bbox="289 660 540 686">SPACE</td> <td data-bbox="551 660 862 686">, (comma)</td> </tr> </table> <p>In which 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="text-align: center;">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="text-align: center;">la 'title' /CR/</p> <p>has four nil parameters after 'title'. All of these will assume their default values</p> | | | | | | |

| | |
|--|--|
| <p>device re-routing parameter</p> | <p>a parameter that re-routes input/output from/to specified devices or files. These are recognised by a "+" or a "-" sign as the first character.</p> <p>For example +dprt:</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 re-initialised. 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 M20 will load the BASIC Interpreter, enter into BASIC execution mode, and run the file with the BAS extension
4. 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:vl 0: /CR/
```

will search only drive 1 for the VLIST command.

FILE AND VOLUME IDENTIFIERS

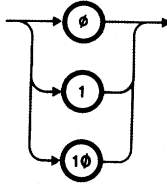
A disk may contain one or more program and/or data files. A single file, however, may not extend beyond one disk.

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.

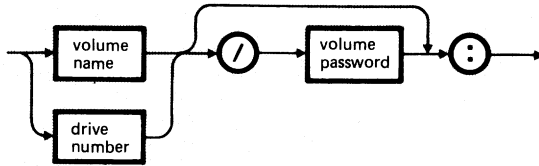
You may assign an identifier to a file either by an OPEN statement (data files), or by a SAVE command (program files), or by an FNEW, FCOPY, FMOVE, FRENAME or EDIT command.

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

drive number



volume identifier



file identifier

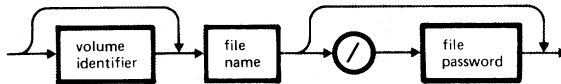


Fig. 4-2 File and Volume Identifier

| SYNTAX ELEMENT | MEANING |
|--------------------|--|
| <p>volume name</p> | <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> <p style="text-align: center;">fp VOL1/vpass:myfile,newpass /CR/</p> |

| | |
|--------------|---|
| | <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>Note: 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;">f1 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.

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

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

The M20 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 M20 displays the current values of the SBASIC command parameters. That is, no further interaction takes place. But in other cases

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 M20 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.

(

(

(

(

5. INITIALISATION AND CHANGING ENVIRONMENTS

ABOUT THIS CHAPTER

This chapter describes the operational modes of the M20 and the initialisation process.

CONTENTS

| | | | |
|---|-----|---|------|
| <u>M20 ENVIRONMENTS</u> | 5-1 | INITIALISATION FOLLOWING A PRUN COMMAND | 5-9 |
| PCOS | 5-1 | <u>THE INITIALISATION FLOWCHART</u> | 5-9 |
| BASIC | 5-1 | <u>BEGINNING AND ENDING A WORKING SESSION</u> | 5-12 |
| 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 | 5-5 | | |
| <u>STANDARD INITIALISATION</u> | 5-6 | | |
| <u>INITIALISATION FILES</u> | 5-7 | | |
| <u>NON-STANDARD INITIALISATION</u> | 5-8 | | |
| INITIALISATION FOLLOWING A PSAVE COMMAND | 5-9 | | |

M20 ENVIRONMENTS

The M20 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 the M20 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 Language Reference Guide". The modes of operation within the Video File Editor are described in Chapter 12.

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 either the Video File Editor from BASIC, neither can you enter BASIC from the Video File Editor.

The following figure illustrates the possible changes of environment.

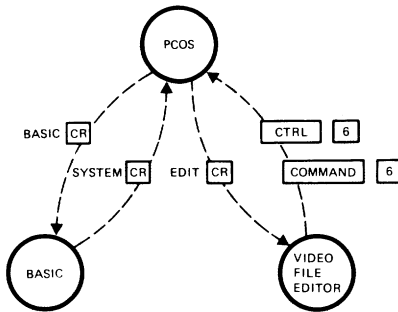


Fig. 5-1 M20 Environments

From PCOS to BASIC and Vice Versa

| IF the M20 is in... | AND you enter... | THEN... |
|---------------------|---|---|
| PCOS | ba /CR/ | the M20 enters the BASIC environment |
| PCOS | a file name with the extension BAS OR the BASIC command with a file identifier as a parameter | the M20 enters the BASIC environment and executes the specified program |
| PCOS or BASIC | /CTRL/ /RESET/ simultaneously (logical reset) | the system is re-initialised but without running diagnostics |
| PCOS or BASIC | physical reset | the system is re-initialised and diagnostic tests are performed |
| BASIC | SYSTEM /CR/ OR the SYSTEM command is encountered during execution of a BASIC program | the system returns to the PCOS environment |

From PCOS to Video File Editor and Vice Versa

| IF M20 is in... | AND you enter... | THEN... |
|-------------------|--|--|
| PCOS | ed file identifier /CR/ | the M20 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 M20 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 M20 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 M20 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 re-enter the command.

On completion of command execution PCOS returns to command mode.

CHANGING MODES

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

| IF the M20 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 INITIALISATION

Initialisation 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. On completion you will hear two 'beeps'. If, during diagnostics you press any of the keys /L/, /D/, /F/, /S/, or /B/ then a non-standard initialisation takes place (see next section). Otherwise, the initialisation is standard and proceeds as described below.

A logical reset (/CTRL/ /RESET/) also causes initialisation, 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: Initialisation 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.

INITIALISATION FILES

Once the keyboard is initialised, PCOS starts a search on both drives for an initialisation 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 initialisation file is found then the system enters PCOS command mode. The effect of an initialisation 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 initialisation. 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 initialisation, 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 initialisation. 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.CMD nor INIT.SAV exist, then the BASIC Interpreter and INIT.BAS are loaded into RAM by PCOS at initialisation. On encountering the extension BAS, the M20 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.CMD and BASIC.ABS

NON-STANDARD INITIALISATION

The initialisation process can be modified by pressing one of the keys /L/, /D/, /F/, /B/ or /S/ while the power-up diagnostics are still running. Each of these options is described in the following table:

| IF, during power-up diagnostics you press | THEN... |
|---|--|
| /L/ | <p>the system loops on diagnostics indefinitely. Its effect can be cancelled by</p> <ul style="list-style-type: none"> - pressing one of the keys /D/, /F/, /B/ or /S/, in which case the corresponding non-standard initialisation process is performed - hitting any key other than /L/, /D/, /F/, /B/ or /S/, in which case initialisation continues as if /L/ had not been pressed in the first instance |
| /D/ | <p>the system loops indefinitely on disk drive diagnostics, repeatedly reading track zero, side zero of the first ready drive</p> |
| /F/ | <p>the bootstrap loader examines the diskette drive before the hard disk drive when looking for a bootable file</p> |

| | |
|-----|--|
| /B/ | the system passes directly into BASIC command mode without attempting to execute any initialisation file |
| /S/ | the system passes directly into PCOS command mode without executing any initialisation file |

INITIALISATION FOLLOWING A PSAVE COMMAND

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

INITIALISATION FOLLOWING A PRUN COMMAND

Following a PRUN command the bootstrap loader searches the drives in the same order as for a standard initialisation, but looks for a file (containing the operating system) specified by the parameter to the PRUN command. This file may be anywhere on diskette or hard disk, that is, it need not be the last PSAVE'd file on the volume.

The message

Invalid File Error (xx) on drive (x)

is produced after each drive has been searched and either the specified file has not been found, or the specified file has been found but is not bootable.

No diagnostic tests are performed.

THE INITIALISATION FLOWCHART

The following figure summarises the initialisation process - both standard and non-standard - by means of a flowchart.

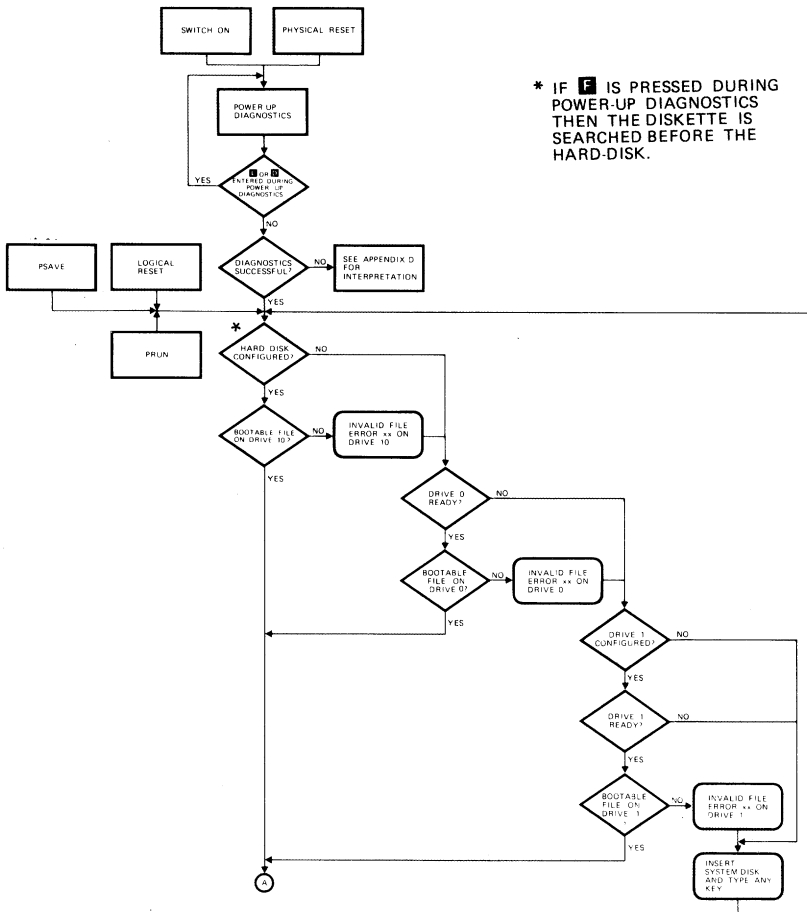


Fig. 5-2 The Initialisation Flowchart

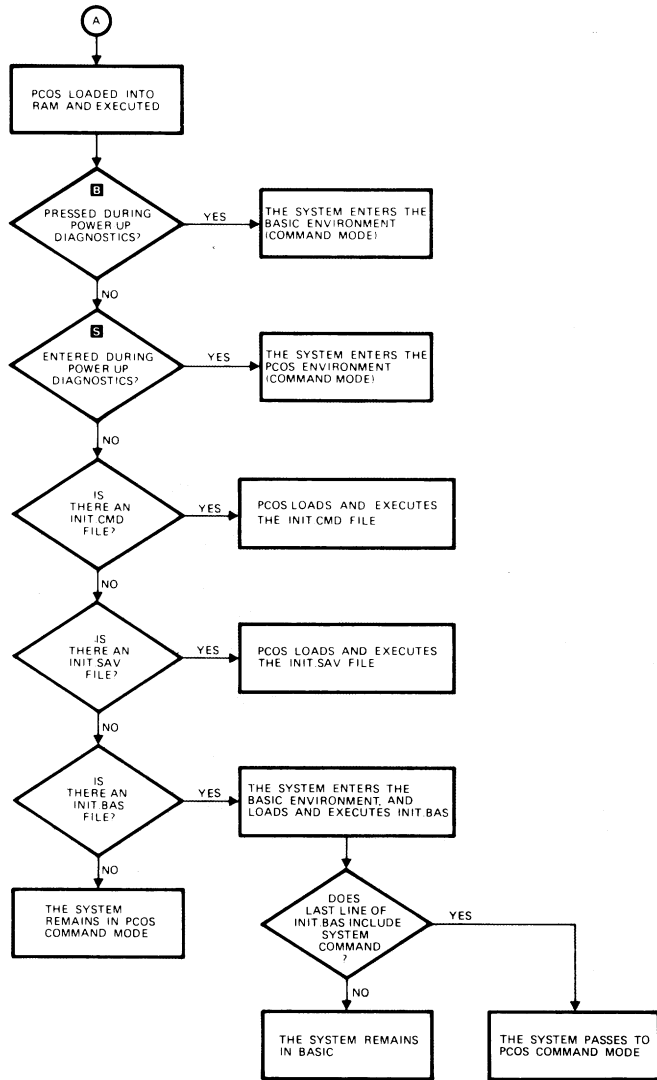


Fig. 5-2 The Initialisation Flowchart

BEGINNING AND ENDING A WORKING SESSION

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

| IF the M20 is... | AND you... | THEN... |
|---|--|---|
| switched off | switch on the M20 pressing the ON/OFF switch on the back panel | power-up diagnostics and initialisation follow, after which the bootstrap loader attempts to load a bootable file. If successful a new working session is commenced |
| switched on (in any environment) | perform a physical reset by pressing the physical reset button located down the right hand side of the main unit | the current working session is terminated. Power-up diagnostics and initialisation 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/ /RESET/ | the current working session is terminated, after which a standard initialisation (but without diagnostics) is performed. If a bootable file can be loaded the M20 passes into the PCOS environment and a new working session is commenced |

6. CONFIGURING PCOS

ABOUT THIS CHAPTER

This chapter describes how to modify PCOS to meet your own needs. This 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 13.

CONTENTS

| | | | |
|--|------|---|------|
| <u>INTRODUCTION</u> | 6-1 | <u>SETTING THE PRINTING ENVIRONMENT</u> | 6-15 |
| <u>MAKING TRANSIENT COMMANDS RESIDENT</u> | 6-1 | <u>RECONFIGURING THE KEY-BOARD LANGUAGE</u> | 6-19 |
| <u>ASSIGNING STRINGS TO KEYS</u> | 6-5 | <u>SAVING A USER-CONFIGURED PCOS</u> | 6-21 |
| <u>CHANGING SYSTEM GLOBAL PARAMETERS</u> | 6-7 | | |
| <u>SETTING THE SYSTEM GLOBAL ENVIRONMENT</u> | 6-7 | | |
| <u>DISPLAYING AND MODIFYING DEVICE NAMES</u> | 6-11 | | |
| <u>MODIFYING THE BASIC ENVIRONMENT</u> | 6-14 | | |
| <u>SETTING THE ENVIRONMENT FOR AN RS-232-C COMMUNICATIONS PORT</u> | 6-15 | | |

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 "Assembler Language 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 at a single key stroke
- 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 initialisation is by means of an initialisation file. If such a file is present on an enabled volume, it will be loaded and executed at initialisation to automatically perform functions that you always require at initialisation (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 CI, RS232, IEEE, EPRINT,

VMOVE, PDEBUG (not included on the system diskette), PLOAD, PUNLOAD and LTERM

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

The M20'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.

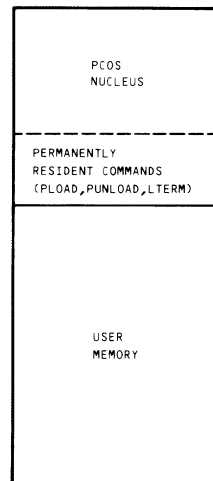


Fig. 6-1 M20 Memory after Initialisation of a Standard PCOS

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 M20'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 requires 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
- IEEE - for loading the (optional) IEEE-488 package
- 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
- KANA - for enabling the Japanese keyboard (special project)

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.CMD 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. 3.x
Operation Mode = Segmented / System
Main entry    = <0A>%B88C; Init entry = --None--
Memory allocated:
  Block No. %00; Starting address = <0A>%B88C; Size = %04FA
```

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.

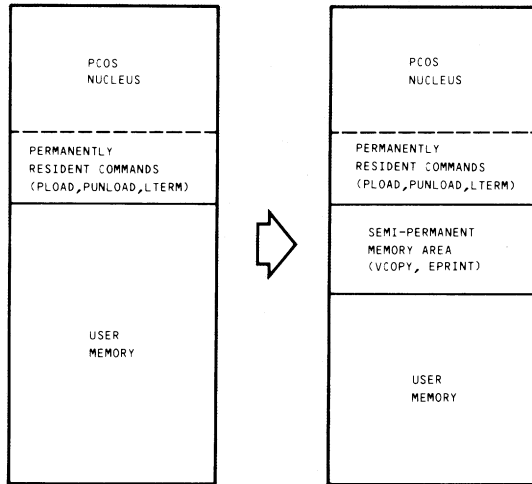


Fig. 6-2 M20 Memory after Making two Transient Commands Semi-Permanent

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/, /RESET/, /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 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'

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/.

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 bytes

and the effect on M20 memory will be as shown in Figure 6-3.

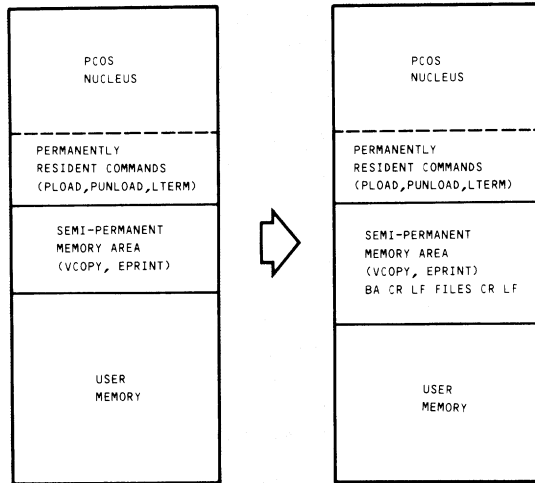


Fig. 6-3 The Effect of a PKEY command on M20 Memory

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".

The template that fits into the slot above the top row of the keyboard can be used as a memory aid to the keys that you have programmed. Alternatively, you can display a list of programmed keys, along with their string assignments by entering

```
pk /CR/
```

A key assignment may be cancelled 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 cancelled by entering

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

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

SETTING THE SYSTEM GLOBAL ENVIRONMENT

If, after initialisation from a standard PCOS, you enter the SSYS command, the M20 will display the default values of the system global environment parameters. That is, if you enter

```
ss /CR/
```

then the M20 will respond with a display as follows

Date = 01/01/1982 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 legal name-printable character

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

The default value is 01/01/1982.

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

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 legal name-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 either you 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> | <pre>statement 600 displays the time.</pre> |
| <pre>700 TIME\$="07:40:15" . . .</pre> | <pre>statement 700 sets the time</pre> |

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 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 minimised 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 12/12/82,09:30:00,,,1 /CR/
```

then the system date is changed to December 12 1982, 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. M20 responds by displaying the new parameter values as follows

```
Date = 12/12/1982   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 (except Disk time) 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 initialisation 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 |
| com2: | 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 only displayed 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 first 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.

com2:

This value is as for com1: except that it refers to the second expansion RS-232-C communications port.

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 re-routing output from the console to the printer you would type

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

See Chapter 7 for further details on device re-routing.

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 initialising the system from a standard PCOS, you subsequently enter the SBASIC command, M20 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 "I/O with External Peripherals User Guide".

SETTING THE PRINTING ENVIRONMENT

If, upon initialising 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 ... pr1450, 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 as any other Set System Global command; but note that OFF is the default value, and in this respect the SFORM command behaves differently to the other Set System Global Commands.

On entering the SFORM command immediately after initialisation 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. Its value must be one of:

- pr1450
- pr1471
- pr1481
- pr2300
- pr2400

- pr2835
- pr-320
- pr-430
- et-121
- et-231
- transp

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.

Note: The et-121 and et-231 are typewriters but can be connected as printers.

The default value is pr1450.

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) interface

pa - the Centronics-like (parallel) interface

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 ,pr1471,30,2,,, 'PCOS USER GUIDE' /CR/
```

then the printing environment parameters will have the following values:

auto - OFF - by default

ptype - pr1471

```

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 PR1471 printer connected to the M20 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
pstype -       pr1471 - 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

```
sl /CR/
```

then M20 displays a menu as follows:

Available Country Configurations:

| | | | |
|----------------------|---|------------------------|----|
| Italy | 0 | Yugoslavia | 10 |
| West Germany | 1 | Norway | 11 |
| France | 2 | Greece | 12 |
| Great Britain | 3 | Switzerland/France ... | 13 |
| United States | 4 | Switzerland/Germany .. | 14 |
| Spain | 5 | Germany (Original) ... | 15 |
| Portugal | 6 | Datev | 16 |
| Sweden/Finland | 7 | Delta | 17 |
| Denmark | 8 | | |

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 M20 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".

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 re-routing 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 and SLANG) to provide the system parameter values you require |
| 5. | Insert a newly formatted 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 from the new operating system which is then ready for use</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 aluminised 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 to the one you have just PSAVEd, the system will still re-boot from the newly PSAVEd file. Moreover, any future attempt to re-initialise 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".

Remark

If your system has a hard disk, then the user-configured PCOS can be saved 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.

7. DEVICE RE - ROUTING

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 commands mentioned in this chapter refer to Chapter 13.

CONTENTS

| | |
|---|-----|
| <u>INTRODUCTION</u> | 7-1 |
| <u>LOCAL DEVICE RE-ROUTING</u> | 7-2 |
| <u>GLOBAL DEVICE RE-ROUTING</u> | 7-5 |
| <u>DEVICE RE-ROUTING FROM A BASIC PROGRAM</u> | 7-7 |