DIGITAL
RESEARCH(r)

CP/M Plus TM
(CP/M Version 3)

Operating System
User's Guide
Programmer's Guide
System Guide

This manual is reprinted by Commodore Business Machines Inc. with the permission of Di ital Research Inc.

P.O. Box 579
160 Central Avenue
Pacific Grove, CA 93950

COPYRIGHT

DISCLAIMER

NOTICE TO USER

From time to time changes are made in the filenames and in the files actually included on the distribution disk. This manual should not be construed as a representation or warranty that such files or materials and facilities exist on the distribution disk or as part of the materials and programs distributed. Most distribution disks include a "README.DOC" file. This file explains variations from the manual which do constitute modification of the manual and the items included therewith. Be sure to read this file before using the software.

TRADEMARKS

The CP/M Plus (CP/M Version 3) OperatingSystem User's Guide was printed in the United States of America.

First Edition: January 1983
Second Edition: March 1983

Foreword

Welcome to the world of microcomputers opened to you by your eight-bit microprocessor. Welcome also to the world of application software accessible with your Digital Research CP/M Plus" operating system, also called CP/Ma 3. Digital Research designed CP/M(r) 3 especially for the 8080, 8085, Z80(r) or equivalent microprocessor that is the heart of your computer.

What CP/M 3 Does For You

CP/M 3 manages and supervises your computer's resources, including memory and disk storage, the console (screen and keyboard), printer, and communications devices. It also manages information stored magnetically on disks by grouping this information into files of programs or data. CP/M 3 can copy files from a disk to your computer's memory, or to a peripheral device such as a printer, To do this, CP/M 3 places various programs in memory and executes them in response to commands you enter at your console.

Once in memory, a program executes through a set of steps that instruct your computer to perform a certain task. You can use CP/M 3 to create your own programs, or you can choose from the wide variety of CP/M 3 application programs that entertain you, educate you, and help you solve commercial and scientific problems.

What You Need to Run CP/M 3 on Your Computer

Digital Research provides two kinds of CP/M 3 systems: banked and nonbanked. Your computer dealer can tell you if you have a banked or nonbanked system. The banked system requires more memory, but in turn provides more memory space for application programs. The banked version also has additional enhancements that are noted in the text.

The minimum hardware requirement for both versions of CP/M 3 is a computer based on an 8080, 8085, or equivalent microprocessor, a console device (generally a keyboard and display device such as a CRT screen), and at least one floppy disk drive. To use all the capabilities of CP/M 3, you should have two disk drives. At least one should be a single density floppy drive, because CP/M 3 and most CP/M applications are distributed on single density floppy disks.

The nonbanked system requires at least 32K (kilobytes) of Random Access Memory (RAM). The larger banked system requires at least 96K of RAM. If you want to expand beyond these requirements you will appreciate that the banked system can include up to sixteen banks of memory.

CP/M 3 and its utility programs are distributed on two floppy disks. The system disk contains the operating system and the most commonly used utility programs. A second disk contains additional utilities.

How To Use CP/M 3 Documentation

The CP/M 3 documentation set includes three manuals:

o CP/M Plus (CP/M Version 3) Operating System User's Guide
o CP/M Plus (CP/M Version 3) Operating System Programmer's Guide
o Programmer's Utilities Guide for the CP/M Family of Operating Systems

The CP/M Plus (CP/M Version 3) Operating System User's Guide introduces you to the CP/M 3 operating system and tells you how to use it. The User's Guide assumes that the version of CP/M 3 on your distribution disk is ready to run on your computer. To use this manual, you must be familiar with the parts of your computer, know how to set it up and turn it on, and how to handle, insert, and store disks. However, you do not need a great deal of experience with computers.

The CP/M Plus (CP/M Version 3) Operating System Programmer's Guide presents information for application programmers who are creating or adapting programs to run under CP/M 3. The Programmer's Utilities Guide for the CP/M Family of Operating Systems includes information on the CP/M assemblers and debuggers that experienced programmers use to create new CP/M 3 programs.

How This Guide is Organized

This guide begins with simple examples, proceeds with basic concepts, then presents a detailed reference section on commands. The first four sections describe CP/M 3 operation for the first-time user. Section 1 introduces CP/M 3 and tells you how to start the operating system, enter commands, edit the command line, and create back-up copies of your distribution disks. Section 2 discusses files, disks, and drives. Section 3 describes how you can use CP/M 3 to manage your printer and console. Section 4 develops the concepts you need to use CP/M 3 commands. If you are new to CP/M read the first four sections carefully to get a general understanding of how to use CP/M 3 before you proceed to the specific command descriptions.

Section 5 rovides detailed information on each CP/M 3 utility program, arranged alphabetically for easy reference. Many of these are programming utilities that you will not use until you start writing your own CP/M 3 programs. Section 6 tells you how to use ED, the CP/M 3 file editor. With ED, you can create and edit program source codes, text, and data files.

Appendix A lists the messages CP/M 3 displays when it encounters special condi- tions, and describes corrective action where necessary. Appendix B provides an ASCII to hexadecimal conversion table. Appendix C lists the filetypes associated with CP/M 3. Appendix D lists and defines the CP/M 3 control characters. Appendix E provides a glossary of commonly used computer terms.

If you are new to computers, you might find some of the topics, such as the programming utilities, difficult to understand at first. Learning to use your computer is a challenge, and we hope you will find it fun. This book proceeds step-by-step so that you can quickly proceed from opening your new system disk package to mastering CP/M 3's powerful facilities.

Table of Contents

Table of Contents (continued)

Table of Contents continued

Table of Contents (continued)

6 ED, The CP/M 3 Context Editor

Table of Contents (continued)

Appendixes

List of Tables

Table of Contents (continued)

Figure

Section 1

Introduction to CP/M 3

This section tells you how to start CP/M 3, how to enter and edit the command line, and how to make back-up copies of your CP/M 3 distribution disks.

1.1 How to Start CP/M 3

Starting or loading CP/M 3 means reading a copy of the operating system from your CP/M 3 system disk (1 of 2 of your distribution disks) into your computer's memory,

First, check that your computer's power is on. Next, insert the CP/M 3 system disk into your initial drive. In this section, assume that the initial drive is A and the disk is removable. Close the drive door. Then, press the RESET or RESTART button. This automatically loads CP/M 3 into memory. This process is called booting, cold starting, or loading the system.

After CP/M 3 is loaded into memory, a message similar to the following is displayed on your screen:

```
CP/M 3          Version V.V
```

The version number, represented above by V.V, tells you the version of CP/M 3 that you own. After this display, the following two-character message appears on your screen:

```
A >
```

This is the CP/M 3 system prompt. The system prompt tells you that CP/M 3 is ready to read a command from your keyboard. In this example, the prompt also tells you that drive A is your default drive. This means that until you tell CP/M 3 to do otherwise, it looks for program and data files on the disk in drive A. It also tells you that you are logged in as user 0, by the absence of a user number other than 0.

1.2 The Command Line

CP/M 3 performs tasks according to specific commands that you type at your keyboard. A CP/M 3 command line is composed of a command keyword, an optional command tail, and a carriage return keystroke. The command keyword identifies a command (program) to be executed. The command tail can contain extra information for the command, such as a filename or parameters. To end the command line, you must press the carriage return or ENTER key. The following example shows a command line.

A>DIR MYFILE

The characters that the user types are slanted to distinguish them from characters that the system displays. In this example, DIR is the command keyword and MYFILE is the command tail. The carriage return keystroke does not appear on the screen or in the example. You must remember to press the carriage return key to send a command line to CP/M 3 for processing. Note that the carriage return key can be marked ENTER, RETURN,, CR, or something similar on your keyboard. In this guide, RETURN signifies the carriage return key.

As you type characters at the keyboard, they appear on your screen. The single-character position indicator, called the cursor, moves to the right as you type characters. If you make a typing error, press either the BACKSPACE key (if your keyboard has one) or CTRL-H to move the cursor to the left and correct the error. CTRL is the abbreviation for the Control key. To type a control character, hold down the Control key and press the required letter key. For example, to move the cursor to the left, hold down CTRL and press the H key.

You can type the keyword and command tail in any combination of upper-case and lower-case letters. CP/M 3 treats all letters in the command line as upper-case.

Generally, you type a command line directly after the system prompt. However, CP/M 3 does allow spaces between the prompt and the command keyword.

CP/M 3 recognizes two different types of commands: built-in commands and transient utitity commands. Built-in commands execute programs that reside in memory as a part of the CP/M 3 operating system. Built-in commands can be executed immediately. Transient utility commands are stored on disk as program files. They must be loaded from disk to perform their task. You can recognize transient utility program files when a directory is displayed on the screen because their filenames are followed by COM. Section 4 presents lists of the CP/M 3 built-in and transient utility commands.

For transient utilities, CP/M 3 checks only the command keyword. If you include a command tail, CP/M 3 passes it to the utility without checking it because many utilities require unique command tails. A command tail cannot contain more than 128 characters. Of course, CP/M 3 cannot read either the command keyword or the command tail until you press the RETURN key.

Let's use one command to demonstrate how CP/M 3 reads command lines. The DIR command, which is -n abbreviation for directory, tells CP/M 3 to display a directory of disk files on your screeii. Type the DIR keyword after the system prompt, omit the command tail, and press RETURN.

```
A>DIR
```

CP/M 3 responds to this command by writing the names of all the files that are stored on the disk in drive A. For example, if you have your CP/M 3 system disk in drive A, these filenames, among others, appear on your screen:

```
COPYSYS com
PIP     com
SET     com
```

CP/M 3 recognizes only correctly spelled command keywords. If you make a typing error and press RETURN before correcting your mistake, CP/M 3 echoes the command line followed with a question mark. If you mistype the DIR command, as in the following example, CP/M 3 responds

A>DJR
DJR?

to tell you that it cannot find the command keyword. To correct simple typing errors, use the BACKSPACE key, or hold down the CTRL key and press H to move the cursor to the left. CP/M 3 supports other control characters that help you efficiently edit command lines. Section 3 tells how to use control characters to edit command lines and other information you enter at your console.

DIR accepts a filename as a command tail. You can use DIR with a filename to if a specific file is on the disk. For example' to check that the transient utility program COPYSYS.COM is on your system disk, type

A>DIR COPYSYS*CDM

CP/M 3 performs this task by displaying either the name of the file you specified, or the message

No F i le.

Be sure you type at least one space after DIR to separate the command keyword from the command tail. If you do not, CP/M 3 responds as follows.

A >DIRCOPYSYS, COM
DIRCOPYSYS.COM?

1.3 Why You Should Back Up Your Files

Humans have faults, and so do computers. Human or computer errors sometimes destroy valuable programs or data files. By mistyping a command, for example, you could accidentally erase a program that you just created or a data file that has been months in the making. A similar disaster could result from an electronic component failure.

Data processing professionals avoid losing programs and data by making copies of valuable files. Always make a working copy of any new program that you purchase and save the original. If the program is accidentally erased from the working copy, you can easily restore it from the original.

It is also wise to make frequent copies of new programs or data files as you develop them. The frequency of making copies varies with each programmer. However, as a general rule, make a copy at the point where it takes ten to twenty times longer to reenter the information than it takes to make the copy.

So far, we have not discussed any commands that change information recorded on your CP/M 3 system disk. Before we do, make a few working copies of the your distribution disks.

1.4 How to Make Copies of Your CP/M 3 Disks

To back up your CP/M 3 disks, you need two or more floppy disks for the back-ups. The back-up disks can be new or used. You might want to format new, or reformat used disks with the disk formatting program that accompanies your particular computer. If the disks are used, be sure that there are no other files on the disks.

If your computer's manufacturer has provided a special program to copy disks, you might use it to make back-ups of your distribution disks. Otherwise, use the COPYSYS and PIP utility programs found on your CP/M 3 distribution disks. PIP can copy all program and data files, but only COPYSYS can copy the operating system. Note that the COPYSYS utility distributed by Digital Resarch only functions with eight-inch, single-density drives. However, your computer's manufacturer might have modified COPYSYS to work with your equipment.

This section shows how to make distribution disk back-ups on a system that has two drives: drive A and drive B. Your drives might be named with other letters from the range A through P. To make a copy of your CP/M 3 distribution system disk, labeled 1 of 2, first use the COPYSYS utility to copy the operating system loader. Make sure that your distribution system disk is in drive A, the default drive, and the blank disk is in drive B. Then enter the following command at the system prompt:

```
A >COPYSYS
```

CP/M 3 loads COPYSYS into memory and runs it. COPYSYS displays the following output on your screen. When the program prompts you, press RETURN only when you have verified that the correct disk is in the correct drive.

Copysys Ver 3,0

Source drive (or return for default) ?A

Source on A then type return

Function complete

Destination drive name (or return to reboot) ?B

Destination on B then type return

Function complete

Do You wish to copy CPM3.SYS? Yes

(CP/M 3 repeats the above prompts to copy CPM3.SYS.)

A>

You now have a copy of the operating system only. To copy the remaining files from disk I of 2, enter the following PIP command.

A>PIP B:=A:*.*

This PIP command copies all the files in your disk directory to drive B from drive A. PIP displavs the message COPYING followed by each filename as the copy operation proceeds. When PIP finishes copying, CP/M 3 displays the system prompt.

Now you have an exact copy of the distribution disk 1 of 2 in drive B. Remove the original from drive A and store it in a safe place. If your original remains safe and unchanged, you can easily restore your CP/M 3 program files if something happens to your working copy.

Remove the copy from drive B and insert it in drive A. Use this copy as your CP/M 3 system disk to make more back-ups, to try the examples shown throughout this manual, and to start CP/M 3 the next time you turn on your computer. Cold start the computer to check copy operations.

You still need to make a back-up copy of distribution disk 2 of 2. This disk contains programmer's utility programs and source files. Place another new or reformatted disk in drive B. This time, type only the command keyword:

A>PIP

PIP responds with an asterisk prompt, *. You can now remove disk I of 2 from drive A and insert the disk you want to copy, disk 2 of 2. Type the following PIP command after the asterisk prompt, for example,

*B: =A: *. *

Again, PIP displays the message COPYING, followed by each filename. When PIP completes the copy and displays the asterisk prompt, press RETURN. CP/M 3 then displays the familiar A> system prompt. You now have a copy of disk 2 of 2 in drive B. Remove both 2 of 2 disks and store them in a safe place. You can now reinsert your working system disk and continue to use the system.

End of Section 1

Section 2

Files, Disks, and Drives

CP/M 3's most important task is to access and maintain files on your disks. With CP/M 3 you can create, read, write, copy, and erase disk files. This section tells you what a file is, how to create, name, and access a file, and how files are stored on your disks. It also tells how to change disks and change the default drive.

2.1 What is a File?

A CP/M 3 file is a collection of related information stored on a disk. Every file must have a unique name because CP/M 3 uses that name to access that file. A directory is also stored on each disk. The directory contains a list of the filenames stored on that disk and the locations of each file on the disk.

In general, there are two kinds of files: program (command) files and data files. A program file contains an executable program, a series of instructions that the computer follows step-by-step. A data file is usually a collection of information: a list of names and addresses, the inventory of a store, the accounting records of a business, the text of a document, or similar related information. For example, your computer cannot execute names and addresses, but it can execute a program that prints names and addresses on mailing labels.

A data file can also contain the source code for a program. Generally, a program source file must be processed by an assembler or compiler before it becomes a program file. In most cases, an executing program processes a data file. However, there are times when an executing program processes a program file. For example, the copy program PIP can copy one or more program files.

2.2 How Are Files Created?

There are many ways to create a file. One way is to use a text editor. The CP/M 3 text editor ED (described in Section 6) can create a file and assign it the name you specify. You can also create a file by copying an existing file to a new location, perhaps renaming it in the process. Under CP/M 3, you can use the PIP command to copy and rename files. Finally, some programs such as MAC" create output files as they process input files.

2.3 How Are Files Named?

CP/M 3 identifies every file by its unique file specification. A file specification can be simply a one- to eight-character filename, such as:

MYFILE

A file specification can have four parts: a drive specifier, a filename, a filetype, and a password.

The drive specifier is a single letter (A-P) followed by a colon. Each drive in your system is assigned a letter. When you include a drive specifier as part of the file specification, you are telling CP/M 3 that the file is stored on the disk currently in that drive. For example, if you enter

B:MYFILE

CP/M 3 looks in drive B for the file MYFILE.

The filename can be from one to eight characters. When you make up a filename, try to let the name tell you something about what the file contains. For example, if you have a list of customer names for your business, you could name the file:

CUSTOMER

so that the name gives you some idea of what is in the file.

As you begin to use your computer with CP/M 3, you will find that files fall naturally into categories. To help you identify files belonging to the same category, CP/M 3 allows you to add an optional one- to three-character extension, called a filetype, to the filename. When you add a filetype to the filename, separate the filetype from the filename with a period. Try to use three letters that tell something about the file's category. For example, you could add the following filetype to the file that contains a list of customer names:

CUSTOMER.NAM

When CP/M 3 displays file specifications in response to a DIR command, it adds blanks to short filenames so that you can compare filetypes quickly. The program files that CP/M 3 loads into memory from a disk have different filenames, but all have the filetype COM.

In banked CP/M 3, you can add a password as an optional part of the file specification. The password can be from one to eight characters. If you include a password, separate it from the filetype (or filename, if no filetype is included) with a semicolon, as follows:

CUSTOMER.NAM;ACCOUNT

If a file has been protected with a password, you must ENTER the password as part of the file specification to access the file. Section 2.7.3 describes passwords in more detail.

We recommend that you create filenames, filetypes, and passwords from letters and numbers. You must not use the following characters in filenames, filetypes, or passwords because they have special meanings for CP/M 3:

< > = , ! I * ? & / $ [ ] ( ) . : ; \ + -

A complete file specification containing all possible elements consists of a drive specification, a primary filename, a filetype, and a password, all separated by their appropriate delimiters, as in the following example:

A:DOCUMENT.LAW;SUSAN

2.4 Do You Have the Correct Drive?

When you type a file specification in a command tall without a drive specifier, the program looks for the file in the drive named by the system prompt, called the default drive. For example, if you type the command

A>DIR COPYSYS.COM

DIR looks in the directory of the disk in drive A for COPYSYS.COM. If you have another drive, B for example, you need a way to tell CP/M 3 to access the disk in drive B instead. For this reason, CP/M 3 lets you precede a filename with a drive specifier. For example, in response to the command

A>DIR B:MYFILE.LIB

CP/M 3 looks for the file MYFILE.LIB in the directory of the disk in drive B. When you give a command to CP/M 3, note which disk is in the default drive. Many application programs require that the data files they access be stored in the default drive.

You can also precede a program filename with a drive specifier, even if you use the program filename as a command keyword. For example, if you type the following command

```
A>B : P I P
```

CP/M 3 looks in the directory of the disk in drive B for the file PIP.COM. If CP/M 3 finds PIP on drive B, it loads PIP into memory and executes it.

If you need to access many files on the same drive, you might find it convenient to change the default drive so that you do not need to repeatedly enter a drive specifier. To change the default drive, enter the drive specifier next to the system prompt and press RETURN. In response, CP/M 3 changes the system prompt to display the new default drive:

```
A>B:
B>
```

Unlike the filename and filetype which are stored in the disk directory, the drive specifier for a file changes as you move the disk from one drive to another. Therefore, a file has a different file specification when you move a disk from one drive to another. Section 4 presents more information on how CP/M 3 locates program and data files.

2.5 Do You Have the Correct User Number?

CP/M 3 further identifies all files by assigning each one a user number which ranges from 0 to 15. CP/M 3 assigns the user number to a file when the file is created. User numbers allow you to separate your files into sixteen file groups. User numbers are particularly useful for organizing files on a hard disk.

When you use a CP/M 3 utility to create a file, the file is assigned to the current user number, unless you use PIP to copy the file to another user number. You can determine the current user number by looking at the system prompt.

```
4A>     User number 4, drive A
A >     User number 0, drive A
2B>     User number 2,  drive B
```

The user number always precedes the drive identifier. User 0, however, is the default user number and is not displayed in the prompt.

You can use the built-in command USER to change the current user number.

```
A >USER 3
3A>
```

You can change both the user number and the drive by entering the new user number and drive specifier together at the system prompt:

```
A >3B
3 B >
```

Most commands can access only those files that have the current user number. For example, if the current user number is 7, a DIR command with no options displays only the files that were created under user number 7. However, if a file resides in user 0 and is marked with a special file attribute, the file can be accessed from any user number. (Section 2.7.1 discusses file attributes.)

2.6 Accessing More Than One File

Certain CP/M 3 built-in and transient utilities can select and process several files when special wildcard characters are included in the filename or filetype. A file specification containing wildcards is called an ambiguous filespec and can refer to more than one file because it gives CP/M 3 a pattern to match. CP/M 3 searches the disk directory and selects any file whose filename or filetype matches the pattern.

The two wildcard characters are ?, which matches any single letter in the same position, and *, which matches any character at that position and any other characters remaining in the filename or filetype. The following list presents the rules for using wildcards.

- o A ? matches any character in a name, including a space character.
- o An * must be the last, or only, character in the filename or filetype. CP/M 3 internally replaces an * with ? characters to the end of the filename or filetype.
- o When the filename to match is shorter than eight characters, CP/M 3 treats the name as if it ends with spaces.
- o When the filetype to match is shorter than three characters, CP/M 3 treats the filetype as if it ends with spaces.

Suppose, for example, you have a disk that contains the following six files:

A. COM AA. COM AAA*COM B. COM A. ASM and B.ASM

The following wildcard specifications match all, or a portion of, these files:

.	is treated as ????????.???
????????.???	matches all six names
*.COM	is treated as ????????.COM
????????.COM	matches the first four names
?.COM	matches A.COM and B.COM
? . *	is treated as ?. ? ? ?
?*???	matches A.COM, B.COM, A.ASM, and B.ASM
A?.COM	matches A.COM and AA.COM
A*.COM	is treated as A???????.COM
A???????*COM	matches A.COM, AA.COM, and AAA.COM

Remember that CP/M 3 uses wildcard patterns only while searching a disk direc-tor-y, and therefore wildcards are valid only in filenames and filetypes. You cannot use a wildcard character in a drive specifier. You also cannot use a wildcard character as part of a filename or filetype when you create a file.

2.7 How to Protect Your Files

Under CP/M 3 you can organize your files into groups to protect them from accidental change and from unauthorized access. You can specify how your files are displayed in response to a DIR command, and monitor when your files were last accessed or modified. CP/M 3 supports these features by assigning the following to files:

- user numbers
- attributes
- time and date stamps
- passwords (banked CP/M 3 only)

All of this information for each file is recorded in the disk directory.

2.7.1 File Attributes

File attributes control how a file can be accessed. When you create a file, CP/M 3 gives it two attributes. You can change the attributes with a SET command.

The first attribute can be set to either DIR (Directory) or SYS (System). This attribute controls whether CP/M 3 displays the file's name in response to a DIR command or DIRSYS command. When you create a file, CP/M 3 automatically sets this attribute to DIR. You can display the name of a file marked with the DIR attribute with a DIR command. If you give a file the SYS attribute, you must use a DIRSYS command to display the filename. Simple DIR and DIRSYS commands display only the filenames created under the current user number.

A file with the SYS attribute has a special advantage when it is created under user 0. When you give a file with user number 0 the SYS attribute, you can read and execute that file from any user number. This feature gives you a convenient way to make your commonly used programs available under any user number. Note, however, that a user 0 SYS file does not appear in response to a DIRSYS command unless 0 is the current user number.

The second file attribute can be set to either R/W (Read-Write) or R/O (Read-Only). If a file is marked R/0, any attempt to write data to that file produces a Read-Only error message. Therefore, you can use the R/O attribute to protect important files. A file with the R/W attribute can be read or written to, or erased at any time, unless the disk is physically write-protected.

2.7.2 Date and Time Stamping

If you use date and time stamps, you can quickly locate the most recent copy of a file, and check when it was last updated or changed. You can choose to have the system tell you either when you created the file, or when you last read from or wrote to the file. You use the SET command to enable date and time stamping, and the DIR command with the DATE option to display a file's time and date stamp.

A SET command enables the option you want to monitor. You can use the following commands to enable time and date stamping on a disk, but you must choose between ACCESS and CREATE. If you choose ACCESS, the stamp records the last time the file was accessed. If you choose CREATE, the stamp records when the file was created.

```
A>SET [ACCESS=ON]
A>SET [CREATE=ON]
A>SET [UPDATE=ON]
```

Files created on or copied to a disk that has time and date stamping are automatically stamped. The DATE command allows you to display and reset the time and date that CP/M 3 is using. For a complete discussion of time and date stamping, see the descriptions of the SET and INITDIR commands in Section 5.

2.7.3 Passwords (Banked CP/M 3 Only)

Passwords allow you to protect your files from access by other users. You can use passwords to limit access to certain files for security purposes.

The SET utility allows you to enable password protection on a drive, assign a password to SET itself (so that unauthorized users cannot disable password protection on a drive), and assign passwords to specific files that have already been created. You can assign passwords to all program and data files. This means that a command line could require the entry of two passwords in order to execute: one password to access the command program, and a second password to access the file specified in the command tail.

Some CP/M 3 commands and most word processing, accounting, and other application programs running under CP/M 3 do not accept passwords in the command tail. If you want to protect your file and still use those programs, you can set a default password before executing the application program. See the description of the SET command in Section 5 for an explanation of this process.

2.8 How Are Files Stored on a Disk?

CP/M 3 records the filename, filetype, password, user number, and attributes of each file in a special area of the disk called the directory. In the directory, CP/M 3 also records which parts of the disk belong to which file.

CP/M 3 allocates directory and storage space for a file as records are added to the file. When you erase a file, CP/M 3 reclaims storage in two ways: it makes the file's directory space available to catalog a different file, and frees the file's storage space for later use. It is this dynamic allocation feature that makes CP/M 3 powerful. You do not have to tell CP/M 3 how big your file will become, because it automatically allocates more storage for a file as needed, and releases the storage for reallocation when the file is erased. Use the SHOW command to determine how much space remains on the disk.

2.9 Changing Floppy Disks

CP/M 3 cannot, of course, do anything to a file unless the disk that holds the file is inserted into a drive and the drive is ready. When a disk is in a drive, it is online and CP/M 3 can access its directory and files.

At some time, you will need to take a disk out of a drive and insert another that contains different files. You can replace an online disk whenever you see the system prompt at your console. This is a clear indication that no program is reading or writing to the drive.

You can also remove a disk and insert a new one when an application program prompts you to do so. This can occur, for example, when the data that the program uses does not fit on one floppy disk.

Note: you must never remove a disk if a program is reading or writing to it.

You can change disks on the drive without sending any special signals to CP/M 3. This allows you to insert another disk at a program's request and read files from or create files on the new disk.

2.10 Protecting a Drive

Under CP/M 3, drives can be marked R/O just as files can be given the R/O attribute. The default state of a drive is R/W. You can give a drive the R/O attribute by using the SET command described in Section 5. To return the drive to R/W, use the SET command or press a CTRL-C at the system prompt.

End of Section 2

Section 3

Console and Printer

This section describes how CP/M 3 communicates with your console and printer. It tells how to start and stop console and printer output, edit commands you enter at your console, and redirect console and printer input and output. It also explains the concept of logical devices under CP/M 3.

3.1 Controlling Console Output

Sometimes CP/M 3 displays information on your screen too quickly for you to read it. Sometimes an especially long display scrolls off the top of your screen before you have a chance to study it. To ask the system to wait while you read the display, hold down the CONTROL (CTRL) key and press S. A CTRL-S keystroke causes the display to pause. When you are ready, press CTRL-Q to resume the display. If you press any key besides CTRL-Q during a display pause, CP/M 3 sounds the console bell or beeper.

DIR, TYPE, and other CP/M 3 utilities support automatic paging at the console. This means that if the program's output is longer than what the screen can display at one time, the display automatically halts when the screen is filled. When this occurs, CP/M 3 prompts you to press RETURN to continue.

3.2 Controlling Printer Output

You can also use a control command to echo console output to the printer. To start printer echo, press a CTRL-P. To stop, press CTRL-P again. While printer echo is in effect, any characters that appear on your screen are listed at your printer.

You can use printer echo with a DIR command to make a list of files stored on a floppy disk. You can also use CTRL-P with CTRL-S and CTRL-Q to make a hard copy of part of a file. Use a TYPE command to start a display of the file at the console. When the display reaches the part you need to print, press CTRL-S to stop the display, CTRL-P to enable printer echo, and then CTRL-Q to resume the display and start pr inting. You can use another CTRL-S, CTRL-P, CTRL-Q sequence to terminate printer echo.

3.3 Console Line Editing

You can correct simple typing errors with the BACKSPACE key. CP/M 3 also supports additional line-editing functions for banked and nonbanked systems that you perform with control characters. You can use the control characters to edit command lines or input lines to most programs.

3.3.1 Line Editing in Nonbanked CP/M 3

Nonbanked CP/M 3 allows you to edit your command line using the set of characters listed'in Table 3-1. To edit a command line in nonbanked CP/M 3, use control characters to delete characters left of the cursor, then replace them with new characters.

In the following example command line, the command keyword PIP is mistyped. The underbar represents the cursor.

A>POP A:=B:*.*-

To move the cursor to the letter 0, hold down the CTRL key and press the letter H eleven times. CTRL-H deletes characters as it moves the cursor left, leaving the following command line:

A>P_

Now, type the correct letters, press RETURN, and send the command to CP/M 3.

A>PIP A:=B:*.*_

Table 3-1 lists all line-editing control characters for nonbanked CP/M 3.

Table 3-1. Nonbanked CP/M 3 Line-editing Control Characters

Character	Meaning
CTRL-E	Forces a physical carriage return but does not send the command line to CP/M 3. Moves the cursor to the beginning of the next line without erasing your previous input.
CTRL-H	Deletes a character and moves the cursor left one character position.
CTRL-1	Moves the cursor to the next tab stop. Tab stops are automatically set at each eighth column. Has the same effect as pressing the TAB key.
CTRL-J	Sends the command line to CP/M 3 and returns the cursor to the left of the current line. Has the same effect as a RETURN or a CTRL-M.
CTRL-M	Sends the command line to CP/M 3 and returns the cursor to the left of the current line. Has the same effect as a RETURN or a CTRL-J.
CTRL-R	Places a # at the current cursor location, moves the cursor to the next line, and displays any partial command you typed so far.
CTRL-U	Discards all the characters in the command line, places a # at the current cursor position, and moves the cursor to the next command line.
CTRL-X	Discards all the characters in the command line, and moves the cursor to the beginning of the current line.

You probably noticed that some control characters have the same meaning. For example, the CTRL-J and CTRL-M keystrokes have the same effect as pressing the RETURN key; all three send the command line to CP/M 3 for processing. Also, CTRL-H has the same effect as pressing the BACKSPACE key.

3.3.2 Line Editing in Banked CP/M 3

Banked CP/M 3 allows you to edit your command line without deleting all characters. Using the line-editing control characters listed in Table 3-2, you can move the cursor left and right to insert and delete characters in the middle of a command line. You do not have to retype everything to the right of your correction. In banked CP/M 3, you can press RETURN when the cursor is in any position in the command line; CP/M 3 reads the entire command line. You can also recall a command for reediting and reexecution.

In the following sample session, the user mistyped PIP, and CP/M 3 returned an error message. The user recalls the erroneous command line by pressing CTRL-W and corrects the error (the underbar represents the cursor):

```
A>POP A:=B:*.*_          (PIP mistyped)
POP?

A>POP A:=B:*#*_          (CTRL-W recalls the line)

A>POP A:=B:*.*           (CTRL-B to beginning of line)

A>POP A:=B:*.*           (CTRL-F to move cursor right)

A>PP A:=B:*.*            (CTRL-G to delete error)

A>PIP A:=B:*.*           (type I to correct the command name)
```

To execute the corrected command line, the user can press return even though the cursor is in the middle of the line. A return keystroke, or one of its equivalent control characters, not only executes the command, but also stores the command in a buffer so that you can recall it for editing or reexecution by pressing CTRL-W.

When you insert a character in the middle of a line, characters to the right of the cursor move to the right. If the line becomes longer than your screen is wide, char acters disappear off the right side of the screen. These characters are not lost. They reappear if you delete characters from the line or if you press CTRL-E when the cursor is in the middle of the line. CTRL-E moves all characters to the right of the cursor to the next line on the screen.

Table 3-2 gives a complete list of line-editing control characters for a banked CP/M 3 system.

Table 3-2. Banked CP/M 3 Line-editing Control Characters

Character	Meaning
CTRL-A	Moves the cursor one character to the left.
CTRL-B	Moves the cursor to the beginning of the command line without having any effect on the contents of the line. If the cursor is at the beginning, CTRL-B moves it to the end of the line.
CTRL-E	Forces a physical carriage return but does not send the command line to CP/M 3. Moves the cursor to the beginning of the next line without erasing the previous input.
CTRL-F	Moves the cursor one character to the right.
CTRL-G	Deletes the character indicated by the cursor. The cursor does not move.
CTRL-H	Deletes a character and moves the cursor left one character position.
CTRL-1	Moves the cursor to the next tab stop. Tab stops are automatically set at each eighth column. Has the same effect as pressing the TAB key.
CTRL-J	Sends the command line to CP/M 3 and returns the cursor to the beginning of a new line. Has the same effect as a RETURN or a CTRL-M keystroke.
CTRL-K	Deletes to the end of the line from the cursor.
CTRL-M	Sends the command line to CP/M 3 and returns the cursor to the beginning of a new line. Has the same effect as a RETURN or a CTRL-J keystroke.
CTRL-R	Retypes the command line. Places a # at the current cursor location, moves the cursor to the next line, and retypes any partial command you typed so far.

Table 3-2. (continued)

Character	Meaning
CTRL-U	Discards all the characters in the command line, places a # at the current cursor position, and moves the cursor to the next line. However, you can use a CTRL-W to recall any characters that were to the left of the cursor when you pressed CTRL-U.
CTRL-W	Recalls and displays previously entered command line both at the operating system level and within executing programs, if the CTRL-W is the first character entered after the prompt. CTRL-J, CTRL-M, CTRL-U, and RETURN define the command line you can recall. If the command line contains characters, CTRL-W moves the cursor to the end of the command line. If you press RETURN, CP/M 3 executes the recalled command.
CTRL-X	Discards all the characters left of the cursor and moves the cursor to the beginning of the current line. CTRL-X saves any characters right of the cursor.

You probably noticed that some control characters have the same meaning. For example, the CTRL-J and CTRL-M keystrokes have the same effect as pressing the RETURN key; all three send the command line to CP/M 3 for processing. Also, CTRL-H has the same effect as pressing the BACKSPACE key. Notice that when a control character is displayed on your screen, it is preceded by an up-arrow, T. For example, a CTRL-C keystroke appears as TC on your screen.

3.4 Redirecting Input and Output

CP/M 3's PUT command allows you to direct console or printer output to a disk file. You can use a GET command to make CP/M 3 or a utility program take console input from a disk file. The following examples illustrate some of the conveniences GET and PUT offer.

You can use a PUT command to direct console output to a disk file as well as the console. With PUT, you can create a disk file containing a directory of all files on that disk, as follows-.

A>PUT CONSOLE OUTPUT TO FILE DIR,PRN
Putting console output to file: DIR.PRN

```
A>DIR
A: FILENAME TEX : FRONT   TEX : FRONT     BAK : ONE    BAK : THREE   TEX
A: FOUR     TEX : ONE     TEX : LINEDIT   TEX : EXAMPI TXT : TWO     BAK
A: TWO      TEX : THREE   BAK : EXAMP2    TXT

A>TYPE DIR.PRN
A: FILENAME TEX : FRONT   TEX : FRONT     BAK : ONE    BAK : THREE   TEX
A: FOUR     TEX : ONE     TEX : LINEDIT   TEX : EXAMPI TXT : TWO     BAK
A: TWO      TEX : THREE   BAK : EXAMP2    TXT
```

You can use a similar PUT command to direct printer output to a disk file as well as the printer.

A GET command can direct CP/M 3 or a program to read a disk file for console input instead of the keyboard. If the file is to be read by CP/M 3, it must contain standard CP/M 3 command lines. If the file is to be read by a utility program, it must contain input appropriate for that program. A file can contain both CP/M 3 command lines and program input if it also includes a command to start a program.

You add or omit the SYSTEM option in the GET command line to specify whether CP/M 3 or a utility program is to start reading the file, as shown in the following sample session. If you omit the SYSTEM option, the system prompt returns so that you can initiate the program that is to take input from the specified file. If you include the SYSTEM option, CP/M 3 immediately takes input from the specified file.

```
3A>type Pip.dat
b:=front.tex
b:=one.tex
b:=two.tex

3A>set console input from file Pip.dat

Getting console input from file: PIP.DAT
3A>Pip
CP/M 3 PIP VERSION 3.0
*b:=front.tex

*b:=one.tex

*b:=two.tex

* ^< c r>

3A>type ccp.da't
d i r
show
dirsys

3A>get console input from file ccp.dat [system]
Getting console input from file; CCP.DAT
3A>dir

A: FILENAME TEX : FRONT    TEX : FRONT    BAK : ONE      BAK : THREE    TEX
A: FOUR     TEX : ONE      TEX : LINEDIT  TEX : EXAMPI   TXT : TWO      BAK
A: TWO      TEX : EXAMP3       : EXAMP2   TXT : PIP      DAT : EXAMP4
A: THREE    BAK : EXAMP5       : CCP      DAT

3A>show
A: RW, Space:       3,392K
B: RW, Space:         452K

3A>dirsys
NON-SYSTEM FILE(S) EXIST
```

See the descriptions of GET and PUT in Section 5 for more ways to use redirected input and output.

3.5 Assigning Logical Devices

Most CP/M 3 computer systems have a traditional console with a keyboard and screen display. Many also have letter-quality printers. If you use your computer for unusual tasks, you might want to add a different kind of character device to your system: a line printer, a teletype terminal, a modem, or even a joystick for playing games. To keep track of these physically different input and output devices, CP/M 3 associates different physical devices with logical devices. Table 3-3 gives the names of CP/M 3 logical devices. It also shows the physical devices assigned to these logical devices in the distributed CP/M 3 system.

Table 3-3. CP/M 3 Logical Devices

Logical Device Name	Device Type	Physical Device Assignment
CONIN:	Console input	Keyboard
CONOUT:	Console output	Screen
AUXIN:	Auxiliary input	Null
AUXOUT:	Auxiliary output	Null
LST:	List output	Printer

In some implementations of CP/M 3, you can change these assignments with a DEVICE command. If your system supports the DEVICE command, you can, for example, assign AUXIN and AUXOLTT to a modem so that your computer can communicate with others over the telephone.

End of Section 3

Section 4

CP/M 3 Command Concepts

As we discussed in Section 1, a CP/M 3 command line consists of a command keyword, an optional command tall, and a carriage return keystroke. This section describes the two kinds of programs the command keyword can identify, and tells how CP/M 3 searches for a program file on a disk. This section also tells how to execute multiple CP/M 3 commands, and how to reset the disk system.

4.1 Two Kinds of Commands

A command keyword identifies a program that resides either in memory as part of CP/M 3, or on a disk as a program file. Commands that identify programs in memory are called built-in commands. Commands that identify program files on a disk are called transient utility commands.

CP/M 3 has six built-in commands and over twenty transient utility commands. You can add utilities to your system by purchasing various CP/M 3-compatible application programs. If you are an experienced programmer, you can also write your own utilities that operate with CP/M 3.

4.2 Built-in Commands

Built-in commands are part of CP/M 3 and are always available for your use regardless of which disk you have in which drive. Built-in commands reside in memory as a part of CP/M 3 and therefore execute more quickly than the transient utilities.

Some built-in commands have options that require support from a related transient utility. The related transient has the same name as the built-in and has a filetype of COM. This type of transient utility is loaded only when a built-in command line contains options that cannot be performed by the built-in command.

If you include certain options in the command tall for a built-in command, CP/M 3 might return a .C0M Required message, This means that the command tail options require support from a related transient utility and CP/M 3 could not find that program file. The following files must be accessible to support all the functions these built-ins offer: ERASE.COM, RENAME.COM, TYPE.COM, and DIR.COM.

Section 5 explains in detail the built-in commands listed in Table 4-1.

Table 4-1. Built-in Commands

Command	Function
DIR	Displays filenames of all files in the directory except those marked with the SYS attribute.
DIRSYS	Displays filenames of files marked with the SYS (system) attribute in the directory.
ERASE	Erases a filename from the disk directory and releases the storage space occupied by the file.
RENAME	Renames a disk file.
TYPE	Displays contents of an ASCII (TEXT) file at your screen.
USER	Changes to a different user number.

CP/M 3 allows you to abbreviate the built-in commands as follows:

DIRSYS	DIRS
ERASE	ERA
RENAME	REN
TYPE	TYP
USER	USE

4.3 Transient Utility Commands

When you enter a command keyword that identifies a transient utility, CP/M 3 loads the program file from the disk and passes it any filenames, data, or parameters you entered in the command tail. Section 5 provides the operating details for the CP/M 3 transient utilities listed in Table 4-2.

Table 4-2. Transient Utility Commands

Name	Function
COPYSYS	Creates a new boot disk.
DATE	Sets or displays the date and time.
DEVICE	Assigns logical CP/M devices to one or more physical devices, changes device driver protocol and baud rates, or sets console screen size.
DUMP	Displays a file in ASCII and hexadecimal format.
ED	Creates and alters character files.
GET	Temporarily gets console input from a disk file rather than the keyboard.
HELP	Displays information on how to use CP/M 3 commands.
HEXCOM	Uses the output from MAC to produce a program file.
INITDIR	Initializes a disk directory to allow time and date stamping.
LINK	Links REL (relocatable) program modules produced by RMAC" (relocatable macro assembler) and produces program files.
MAC	Translates assembly language programs into machine code form.
PIP	Copies files and combines files,

Table 4-2. (continued)

Name	Function
PUT	Temporarily directs printer or console output to a disk file.
RMAC	Translates assembly language programs into relocatable program modules.
SET	Sets file options including disk labels, file attributes, type of time and date stamping, and password protection.
SETDEF	Sets system options including the drive search chain.
SHOW	Displays disk and drive statistics.
SID	Helps you check your programs and interactively correct programming errors.
SUBMIT	Automatically executes multiple commands.
XREF	Produces a cross-reference list of variables used in an assembler program.

4.4 How CP/M 3 Searches for Program and Data Files

This section describes how CP/M 3 searches for program and data files on disk. If it appears that CP/M 3 cannot find a program file you specified in a command line, the problem might be that CP/M 3 is not looking on the drive where the file is stored. Therefore, you need to understand the steps CP/M 3 follows in searching for program and data files.

4.4.1 Finding Data Files

As you recall, when you enter a command line, CP/M 3 passes the command tail to the program identified by the command keyword. If the command tail contains a file specification, the program calls CP/M 3 to search for the data file. If CP/M 3 cannot find the data file, the program displays an error message at the console. Typically, this message is File not found or No File, but the message depends on the program identified by the command keyword.

If you do not include a drive specifier with the filename in a command tail, CP/M 3 searches the directory of the current user number on the default drive. If the file is not there, CP/M 3 looks for the file with the SYS attribute in the directory of user 0 on the default drive. If CP/M 3 finds the file under user 0, it allows the program Read-Only access to the file. For example, if you enter the following command line,

3A>TYPE MYFILE.TXT

CP/M 3 first searches the directory for user 3 on drive A. If it does not find MYFILE.TXT there, it searches the directory of user 0 on drive A for MYFILE.TXT marked with the SYS attribute. If the file is not in either directory, CP/M 3 returns control to TYPE, which then displays No F i 1 e.

Some CP/M 3 utilities, such as PIP and DIR, restrict their file search to the current user number. Because CP/M 3 does not allow Read-Write access to SYS files,-ERASE and RENAME also restrict their search to the current user number.

The search procedure is basically the same if you do include a drive specifier with the filename. CP/M 3 first looks in the directory of the current user number on the specified drive. Then, if it does not find the file, it looks in the directory for user 0 on the specified drive for the file with the SYS attribute. If CP/M 3 does not find the data file after these two searches, it displays an error message.

4.4.2 Finding Program Files

The search procedure for a program file can be very different from a data file search. This is because you can use the SETDEF command described in Section 5 to define the search procedure you want CP/M 3 to follow when it is looking for a program file. With SETDEF you can ask CP/M 3 to make as many as sixteen searches when you do not include a drive specifier before the command keyword, but that is a rare case! We will begin by describing how CP/M 3 searches for program files when you have not yet entered a SETDEF command.

If a command keyword identifies a transient utility, CP/M 3 looks for that program file on the default or specified drive. It looks under the current user number, and then under user 0 for the same file marked with the SYS attribute. At any point in the search process, CP/M 3 stops the search if it finds the program file. CP/M 3 then loads the program into memory and executes it. When the program terminates, CP/M 3 displays the system prompt and waits for your next command. However, if CP/M 3 does not find the command file, it repeats the command line followed by a question mark, and waits for your next command.

If you include a drive specifier before the command keyword, you are telling CP/M 3 precisely where to look for the program file. Therefore, CP/M 3 searches only two locations: the directory for the current user on the specified drive, and then for user 0 on the specified drive, before it repeats the command line with a question mark. For example, if you enter

4C>A:SH0W [SPACE]

CP/M 3 looks on drive Al, user 4 and then user 0 for the file SHOW.COM.

If you do not include a drive specifier before the command keyword,CP/M 3 searches directories in a sequence called a drive chain. When you first receive CP/M 3, there is only one drive in your chain, the default drive. Unless you change the chain with a SETDEF command, CP/M 3 looks in two places for the program file. For example, if you enter

7E>SHOW [SPACE]

CP/M 3 searches the following locations for the file SHOW.COM:

1. drive E, user 7
2. drive E, user 0

Remember that a SHOW.COM file under user 0 must be marked with the SYS attribute or else CP/M 3 cannot find it. Use a SET command to give program files under user 0 to the SYS attribute because they can then be accessed automatically from all other user areas. You do not have to duplicate frequently used program files in all user areas on all drives.

When you use a SETDEF command to define your own drive chain, include the default drive, and the drive that contains your most frequently used utilities. For an example, assume you defined your drive chain as * (the default drive) and drive A. When you enter the following command:

2D>SH0W [SPACE]

CP/M 3 looks for SHOW.COM in the following sequence:

1. drive D, user 2
2. drive D, user 0
3. drive A, user 2
4. drive A, user 0

You can include your default drive in your drive chain with an option in a SETDEF command. Any drive chain you specify with SETDEF remains in effect until you restart or reset the system.

You can also use a SETDEF command to enable automatic submit in your drive chain. See Section 4.5 for a description of automatic submit.

4.5 Executing Multiple Commands

In the examples so far, CP/M 3 has executed only one command at a time. CP/M 3 can also execute a sequence of commands. You can enter a sequence of commands at the system prompt, or you can put a frequently needed sequence of commands in a disk file. Once you have stored the sequence in a disk file, you can execute the sequence whenever you need to with a SUBMIT command.

To enter multiple commands at the system prompt, separate each command keyword and associated command tail from the next keyword with an exclamation point, !. When you complete the sequence, press RETURN. CP/M 3 executes your commands in order:

```
3A>dirsys!dir examp*.*!show [space]
NON-SYSTEM FILE(S) EXIST
3A>dir examp*.*
A: EXAMP7      : EXAMPI   TXT : EXAMP3      EXAMP2   TXT   EXAMP4
A: EXAMP5      : EXAMPG
3A>show [space]

A: RW, Space:    3,344K
```

If you find you need to execute the same command sequence frequently, store the sequence in a disk file. To create this file, use ED or another character file editor. The file must have a filetype of SUB. Each command in the file must start on a new line. For example, an UPDATE.SUB file might look like this:

```
DIR A:*.COM
ERA B:*.COM
PIP B:=A:*.COM
```

To execute this list,, enter the following command:

```
A>SUBMT UPDATE
```

The SUBMIT utility passes each command to CP/M 3 for sequential execution. While SUBMIT executes, the commands are usually echoed at the console, as well as any program's screen display, such as the directory or PIP's "COPYING..." message. When one command completes, the system prompt reappears either with the next command in the SUB file, or, when the SUB file is exhausted, by itself to wait for your next command from the keyboard.

If PROFILE exists, PROFILE.SUB is a special submit file that CP/M 3 automatically executes at each cold start. This feature is especially convenient if you regularly execute a standard set of commands, such as SETDEF and DATE SET, before beginning a work session. A PROFILE.SUB might already exist on your distribution disk. If not, you can create one using ED or another editor.

The description of the SUBMIT utility in Section 5 gives more details on how to create a SUB file and use SUBMIT parameters to pass options to the programs to be executed.

You can also use CTRL-C to reset the disk system. This is sometimes called a warm start. When you press CTRL-C and the cursor Is at the system prompt, CP/M 3 logs out all the active drives, then logs in the default drive. ne active drives are any drives you have accessed since the last cold or warm start. A SHOW [SPACE] command displays the remaining space on all active drives. In the following example, SHOW [SPACE] indicates that three drives are active. However, if you press CTRL-C immediately after this display and then enter another SHOW [SPACE] command, only the space for the default drive, A, is displayed.

4.6 Terminating Programs

You can use the two keystroke command CTRL-C to terminate program execution or reset the disk system. To enter a CTRL-C command, hold down the CTRL key and press C.

Not all application programs that run under CP/M can be terminated by a CTRL-C, However, most of the transient utilities supplied with CP/M 3 can be terminated immediately by a CTRL-C keystroke. If you try to terminate a program while it is sending a display to the screen, you might need to press a CTRL-S to halt the display before entering CTRL-C.

You can also use CTRL-C to reset the disk system. This is sometimes called a warm start. When you press CTRL-C and the cursor is at the system prompt, CP/M 3 logs out all the active drives, then logs in the default drive. The active drives are any drives you have accessed since the last cold or warm start. A SHOW [SPACE] command displays the remaining space on all active drives. In the following example, SHOW [SPACE] indicates that three drives are active. However, if you press CTRL-C immediately after this display and then enter another SHOW [SPACE] command, only the space for the default drive, A, is displayed.

```
A>SH0W [SPACE]
A: RW, Space:        9,4B8k
B: RO, Space:        2,454k
C: RO, Space:        1,665K
A> ^C
A>SHOW [SPACE]
A: RW# Space:         9,488K
```

4.7 Getting Help

CP/M 3 includes a transient utility command called HELP that can display a summary of what you need to know to use each command described in this manual. To get help, simply enter the command:

```
A>HELP
```

In response, the HELP utility displays a list of topics for which summaries are available. After HELP lists the topics available, it displays its own prompt:

HELP>

To this prompt, you can enter one of the topics presented in the list, for example,

HELP>SHOW

After displaying a summary of the SHOW command, HELP lists subtopics that detail different aspects of the SHOW command. To display the information on a subtopic when you have just finished reading the main topic, enter the name of the subtopic preceded by a period,

HELP >. OPTIONS

In the preceding example, HELP then displays the options available for the SHOW command. As you become familiar with HELP, you might want to call a HELP subtopic directly from the system prompt as follows:

A>HELP SHOW OPTIONS

HELP lets you learn the basic CP/M 3 commands quickly. You might find that you reference the command summary in Section 5 only when you need details not
ded in the HELP summaries. When you add new utilities, you can modify HELP
provi I
to add or subtract topics, or even modify the summaries HELP presents. See the description of HELP in Section 5 for complete details.

End of Section 4

Section 5

Command Summary

This section describes the commands and programs supplied with your CP/M 3 operating system. The commands are in alphabetical order. Each command is followed by a short explanation of its operation and examples. More complicated commands are described later in detail. For example, ED is described in Section 6. Other commands, such as SID and MAC, are described fully in other CP/M manuals.

CP/M 3 has replaced some commands from previous CP/M versions. @C replaces ASM; SHOW and DIR include the previous STAT functions; and SID replaces DDT.

5.1 Let's Get Past the Formalities

This section describes the parts of a file specification in a command line. There are four parts in a file specification; to avoid confusion, each part has a formal name:

- drive specifier-the optional disk drive A, B, C, ...P that contains the file or group of files to which you are referring. If a drive specifier is included in your command line, a colon must follow it.
- filename-the one- to eight-character first name of a file or group of files.
- filetype-the optional one- to three-character category name of a file or group of files. If the filetype is present, a period must separate it from the filename.
- password-the optional one- to eight-character password which allows you to protect your files. It follows the filetype, or the filename if no filetype is assigned, and is preceded by a semicolon.

If you do not include a drive specifier, CP/M 3 automatically uses the default drive. If you omit the period and the filetype, CP/M 3 automatically includes a filetype of three blanks.

This general form is called a file specification. A file specification names a particular file or group of files in the directory of the on-line disk given by the drive specifier. For example,

B:MYFILE.DAT

is a file specification that indicates drive B:, filename MYFILE, and filetype DAT. File specificat'on is abbreviated to

filespec

in the command syntax statements.

Some CP/M 3 commands accept wildcards in the filename and filetype parts of the command tall. For example,

B:MY*.A??

is a file specification with drive specifier B:, filename MY*, and filetype A??. This ambiguous file specification might match several files in the directory.

Put together, the parts of a file specification are represented like this:

d:filename.typ;password

In the preceding form, d: represents the optional drive specifier, filename represents the one- to eight-character filename, and typ represents the optional one- to three-character filetype. The syntax descriptions in this section use the term filespec to indicate any valid combination of the elements included in the file specification. The following list shows valid combinations of the elements of a CP/M 3 file specification.

- o filename
- o filename.typ
- o filename;password
- o filename.typ;password
- o d: filename
- o d:filename.typ
- o d:filename;password
- o d: filename. typ;password

The characters in Table S-1 have special meaning in CP/M 3, so do not use these characters in file specifications except as indicated.

Table 5-1. Reserved Characters

Character	Meaning
<= , ! | > [ ] tab space carriage return	file specification delimiters
:	drive delimiter in file specification
.	filetype delimiter in file specification
;	password delimiter in file specification
* ?	widcard characters in an ambiguous file specification
< > & ! | \ + -	option list delimiters
[ ]	option list delimiters for global and local options
( )	delimiters for multiple modifiers inside square brackets for options that have modifiers
/ $	option delimiters in a command line
;	comment delimiter at the beginning of a command line

CP/M 3 has already established several file groups. Table 5-2 lists some of their filetypes with a short description of each family. Appendix C provides the complete lst.

Table 5-2. CP/M 3 Filetypes

Filetype	Meaning
ASM	Assembler source file
BAS	CBASIC8 source program
COM	8080, 8085, or equivalent machine language program
HLP	HELP message file
SUB	List of commands to be executed by SUBMIT
$$$	Temporary file

In some commands, descriptive qualifiers are used with filespecs to further qualify the type of filespec accepted by the commands. For example, wildcard-filespec denotes wiIdcard spec'ficat'ons, dest-filespec denotes a destination filespec, and src-filespec denotes a source filespec.

You now understand command keywords, command tails, control characters, default drive, and wildcards. You also see how to use the formal names filespec, drive specifier, filename, and filetype. These concepts give you the background necessary to compose complete command lines.

5.2 How Commands Are Described

CP/M 3 commands appear in alphabetical order. Each command description is given in a specific form. This section also describes the notation that indicates the optional parts of a command line and other syntax notation.

- o The description begins with the command keyword in upper-case.
- o The syntax section gives you one or more general forms to follow when you compose the command line.

- o The explanation section defines the general use of the command keyword, and points out exceptions and special cases. The explanation sometimes includes tables or lists of options that you can use in the command line.

- o The examples section lists a number of valid command lines that use the command keyword. To clarify examples of interactions between you and the operating system, the characters that you enter are slanted. The responses that CP/M 3 shows on your screen are in vertical type.

The notation in the syntax lines describes the general command form using these rules:

- o Words in capital letters must be spelled as shown, but you can use any combination of upper- or lower-case letters.

- o The symbolic notation d:, filename, type, ;password, and filespec have the general meanings described in Section 5.1.

- o You must include one or more space characters where a space is shown, unless otherwise specified. For example, the PIP options do not need to be separated by spaces.

The following table defines the special symbols and abbreviations used in syntax lines.

Table 5-3. Syntax Notation

Symbol	Meaning
DIR	Directory attribute.
n	You can substitute a number for n.
0	Indicates an option or an option list.
RO	Read-Only.
RW	Read-Write.
s	You can substitute a string, which consists of a group of characters, for s.

Table 5-3. (continued)

<table>
<tr><th>Symbol</th><th>Meaning</th></tr>
<tr><td>SYS</td><td>System attribute.</td></tr>
<tr><td>{ }</td><td>Items within braces are optional. You can enter a command without the optional items. The optional items add effects to your command line.</td></tr>
<tr><td>[ ]</td><td>Items in square brackets are options or an option list. If you use an option specified within the brackets, you must type the brackets to enclose the option. If the right bracket is the last character on the command line, it can be omitted.</td></tr>
<tr><td>( )</td><td>Items in parentheses indicate a range of options. If you use a range from an option list, you must enclose the range in parentheses.</td></tr>
<tr><td>...</td><td>Ellipses tell you that the previous item can be repeated any number of times.</td></tr>
<tr><td>|</td><td>The or bar separates alternative items in a command line. You can select any or all of the alternatives specified. Mutually exclusive options are indicated in additional syntax lines or are specifically noted in the text.</td></tr>
<tr><td>^ or CTRL</td><td>Represent the CTRL key on your keyboard. (CTRL characters show as ' on your screen.)</td></tr>
<tr><td><cr></td><td>Indicates a carriage return keystroke.</td></tr>
<tr><td>*</td><td>Wildcard character-replaces all or part of a filename and/or filetype.</td></tr>
<tr><td>?</td><td>Wildcard character-replaces any single character in the same position of a filename or filetype.</td></tr>
</table>

Let's look at some examples of syntax notation. The CP/M 3 DIR (DIRectory) command displays the names of files cataloged in the disk directory and, optionally, displays other information about the files.

The syntax of the DIR command with options shows how to use the command line syntax notation:

```
Syntax:   DIR{fd:{|}filespec}{[options]}
                 |        |          |
              optional  optional  optional
```

This tells you that the command tail following the command keyword DIR is optional. DIR alone is a valid command, but you can include a file specification, or a drive specification, or 'ust the options in the command line. Therefore,

DIR
DIR filespec
DIR d:
DIR [RO]

are valid commands. Furthermore, the drive or file specification can be followed by another optional value selected from one of the following list of DIR options:

RO
RW
DIR
SYS

Therefore,

DIR d:filespec [RO]

is a valid command.

Recall that in Section 2 you learned about wildcards in filenames and filetypes. The DIR command accepts wildcards in the file specification, Using this syntax, you can construct several valid command lines:

```
DIR
DIR X.PAS
DIR X.PAS [RO]
DIR X.PAS [SYS]
DIR *.PAS
DIR *.*  [RW]
DIR X.* [DIR]
```

The CP/M 3 command PIP (Peripheral Interchange Program) is the file copy program. PIP can copy information from the disk to the screen or printer. PIP can combine two or more files into one longer file. PIP can also rename files after copying them. Look at one of the formats of the PIP command line for another example of how to use command line notation. PIP also copies files from one disk to another disk.

Syntax: PIP dest-filespec = src-filespecf{,filespec ... }

In the preceding example, dest-filespec is further defined as a destination file specification or peripheral device (printer, for example) that receives data. Similarly, src-filespec is a source file specification or peripheral device (keyboard, for example) that transmits data. PIP accepts wildcards in the filename and filetype. (See the PIP command for details regarding other capabilities of PIP.) There are, of course, many valid command lines that come from this syntax. Some examples follow.

```
PIP   NEWFILE.DAT=OLDFILE.DAT
PIP   B:=A:THISFILE.DAT
PIP   B:X.BAS=Y.BAS, Z.BAS
PIP   X.BAS=A.BAS, B.BAS, C.BAS
PIP   B:=A:*.BAK
PIP   B:=A:*.*
```

The remainder of this section contains a complete description of each CP/M 3 util ity. The descriptions are arranged alphabetically for easy reference.

The COPYSYS Command

Syntax: COPYSYS

Explanation: The COPYSYS command copies the CP/M 3 system from a CP/M 3 system disk to another disk. The disk must have the same format as the original system disk. For example, if the system disk is a single-density disk, the disk you use to copy onto must also be in single-density format.

The COPYSYS utility copies only the system tracks onto the new disk. To use the new disk as a CP/M 3 system disk, you must also copy the system file CPM3.SYS to the new disk. COPYSYS gives you the option to copy CPM3.SYS to the new disk. To copy other files onto the new disk,, use the PIP command.

Example: A>COPY'SYS

```
Copysys Ver 3.0
Source drive name ( or return for default )C
Source on C then type return

Place the disk to be copied in drive C, then enter <cr>.

Function Complete
Destination drive name (or return to reboot)C
Destination on C then type return

Replace the system disk in C with the new disk, then enter <cr>.

Function complete
Do You wish to copy CPM3.SYS?)'
Source drive name (or return for default)<cr,>
Source on default then type return
Function complete
Destination drive name (or return to reboot)C
Destination on C then type return
```

Place the disk to be copied in drive C then enter <cr>.

Function complete

The preceding example copies the CP/M 3 system using only one disk drive C. In the preceding messages, the word source refers to the disk that contains the CP/M 3 system, and the word destination refers to the disk to which the CP/M 3 system is to be copied.

The system file CPM3.SYS is copied from the default drive A to the -iew d'sk in drive C. CP/M 3 requires the file CPM3.SYS to be on the systetii disk.

The DATE Command

Syntax: DATE {CONTINUOUS}
DATE {time-specification}
DATE SET

Explanation: The DATE command is a transient utility that lets you display and set the date and time of day. When you start CP/M 3, the date and time are set to the creation date of your CP/M 3 system. Use DATE to change this initial value to the current date and time.

Display Current Date and Time

Syntax: DATE {CONTINUOUS}

Explanation: The preceding form of the DATE command displays the current date and time. The CONTINUOUS option allows continuous display of the date and time. The CONTINUOUS option can be abbreviated to C. You can stop the continuous display by pressing any key.

Examples:

```
A>DATE
A>DATE C
```

The first example displays the current date and time. A sample display might be:

```
Fri 08/13/82 09:15:37
```

The second example displays the date and time continuously until you press any key to stop the display.

Set the Date and Time

Syntax: DATE {time-specification}
DATE SET

Explanation: The first form allows the user to enter both date and time in the command. The time-specification format is

MM/DD/YY HH:MM:SS

where:

MM is a month value in the range 1 to 12.
DD is a day value in the range 1 to 31.
YY is the two-digit year value relative to 1900.
HH is the hour value in the range of 0 to 23.
MM is the minute value in the range of 0 to 59.
SS is the second value in the range of 0 to 59.

The system checks the validity of the date and time entry and determines the day for the date entered.

The second form prompts you to enter the date and the time. To keep the current system date or time, press the carriage return.

Examples: A>DATE 08114IB2 10:30:00

The system responds with

Press any key to set time

When the time occurs, press any key. DATE initializes the time at that instant, and displays the date and time:

Sat 08/14/82 10:30:00

A>DATE SET

The system prompts with

Enter today 's date (MM/DD/YY)

Press the carriage return to skip or enter the date. Then the system prompts with

Enter the time (HH:MM:SS)

Press the carriage return to skip or enter the time and the system prompts with

Press any key to set time

to allow you to set the time exactly.

The DEVICE Command

Syntax: DEVICE {NAMES | VALUES | physical-dev I logical-dev}
DEVICE logical-dev = physical-dev {option}
{,physical-dev {option},...}
DEVICE logical-dev = NULL
DEVICE physical-dev {option}
DEVICE CONSOLE [PAGE | COLUMNS = columns | LINES = lines}

Explanation: The DEVICE command is a transient utility that displays current assignments of CP/M 3 logical devices and the names of physical devices. DEVICE allows you to assign logical CP/M 3 devices to peripheral devices attached to the computer. The DEVICE command also sets the communications protocol and speed of a peripheral device, and displays or sets the current console screen size.

CP/M 3 supports the following five logical devices:

CONIN:
CONOUT:
AUXIN:
AUXOUT:
LST:

These logical devices are also known by the following names:

CON: (for CONIN: and CONOUT:)
CONSOLE: (for CONIN: and CONOUT:)
KEYBOARD (for CONIN:)
AUX: (for AUXIN: and AUXOUT:)
AUXILIARY: (for AUXIN: and AUXOUT:)
PRINTER (for LST:)

The physical device names on a computer vary from system to system. You can use the DEVICE command to display the names and attributes of the physical devices that your system accepts.

Display Device Characteristics and Assignments

Syntax: DEVICE { NAMES | VALUES | physical-dev | logical-dev}

Explanation: The preceding form of the DEVICE command displays the names and attributes of the physical devices and the current assignments of the logical devices in the system.

Examples: A>DEVICE

The preceding command displays the physical devices and current assignments of the logical devices in the system. The following is a sample response:

```
Physical Devices:
I=Input ,D=Output ,S=Serial ,X=Xon-Xoff
CRT      9600   IOS   LPT      9600   IOSX   CRTI     9600  IOSs
CRT2     9600   IOS   CRT3     4800   IOS    LPTI      134  IOSX
CEN      NONE   0     MODEM1   19200  IOS    MODEM2    300  S
CTRLRI   150    0     GRACRT   19200  IOS    DIABLO    110  0
CTRLR2   300    0     SCRTY    7200

Current Assignments:
CONIN:   = CRT
CONOUT:  = CRT
AUXIN:   = Null Device
AUXOUT:  = Null Device
LST:     = LPT

Enter new assignment or hit RETURN:
```

The system prompts for a new device assignment. You can enter any valid device assignment (as described in the next section). If you do not want to change any device assignments, press the RETURN key.

A> DEVICE NAMES

The preceding command lists the physical devices with a summary of the device characteristics.

A>DEVICE VALUES

The preceding command displays the current logical device assignments.

A >DEVICE CRT

The preceding command displays the attributes of the physical device CRT.

A >DEVICE CON

The preceding command displays the assignment of the logical device CON:

Assign a Logical Device

Syntax: DEVICE logical-dev = physical-dev {option}
{,physical-dev {option},...}
DEVICE logical-dev = NULL

Explanation: The first form assigns a logical device to one or more physical devices. The second form disconnects the logical device from any physical device.

Table 5-4. DEVICE Options

Option	Meaning
XON	refers to the XON/XOFF communications protocol. This protocol uses two special characters in the ASCII character set called XON and XDFF. XON signals transmission on, and XOFF signals transmission off. Before each character is output from the computer to the peripheral device, the computer checks to see if there is any incoming data from the peripheral. If the incoming character is XO,FF, the computer suspends all further output until it receives an XON from the device, indicating that the device is again ready to receive more data.
NOXON	indicates no protocol and the computer sends data to the device whether or not the device is ready to receive it.
baud-rate	is the speed of the device. The system accepts the following baud rates: 50 75 110 134 150 300 600 1200 1800 2400 3600 4800 7200 9600 19200

Examples: A >DEVICE CONOUT: =L PT ,CRT
A >DEVICE AUXIN: =CRT2 [XON , 9600]
A >DEVICE LST: =NULL

The first example assigns the system console output, CONOUT:, to the printer, LPT, and the screen, CRT. The second example assigns the auxiliary logical input device, AUXIN:, to the physical device CRT using protocol XON/XOFF and sets the transmission rate for the device at 9600. The third example disconnects the list output logical device, LST:.

Set Attributes of a Physical Device

Syntax: DEVICE physical-dev {option)

Explanation: The preceding form of the DEVICE command sets the attributes of the physical device specified in the command.

Example: A >DEVICE LPT [XON ,96OO]

The preceding command sets the XON/XOFF protocol for the physical device LPF and sets the transmission speed at 9600.

Display or Set the Current Console Screen Size

Syntax: DEVICE CONSOLE [PAGE | COLUMNS = columns | LINES = lines]

Explanation: The preceding form of the DEVICE command displays or sets the current console size.

Examples: A >DEVICE CONSOLE [PAGE]
A > DEVI CE CONSOL [COLUMNS=40, LINES=16]

The first example displays the current console page width in columns and length in lines. The second example sets the screen size to 40 columns and 16 lines.

The DIR Command

Syntax: DIR {d:}
DIR {filespec}

DIRSYS {d:}
DIRSYS {filespec}

DIR {d:} [options]
DIR {filespec} {filespec} ... [options]

Explanation: The DIR command displays the names of files and the attributes asso- crated with the files. DIR and DIRSYS are built-in utilities-, DIR with options is a transient uti ity.

Display Directory

Syntax: DIR {d:}
DIR {filespec}

DIRSYS {d:}
DIRSYS {filespec}

Explanation: The DIR and DIRSYS commands display the names of files cataloged in the directory of an on-line disk. The DIR command lists the names of files in the current user number that have the Directory (DIR) attri- bute. DIR accepts wildcards in the file specification. You can abbreviate the DIRSYS command to DIRS.

The DIRSYS command displays the names of files in the current user number that have the System (SYS) attribute. Although you can read System (SYS) files that are stored in user 0 from any other user number on the same drive, DIRSYS only displays user 0 files if the current user number is 0. DIRSYS accepts wildcards in the file specification.

If you omit the drive and file specifications, the DIR command displays the names of all files with the DIR attribute on the default drive for the current user number. Similarly, DIRSYS displays all the SYS files.

If the drive specifier is included, but the filename and filetype are omitted, the DIR command displays the names of all DIR files in the current user on the disk in the specified drive. DIRSYS displays the SYS files.

If the file specification contains wildcard characters, all filenames that satisfy the match are displayed on the screen.

If no filenames match the file specification, or if no files are cataloged in the directory of the disk in the named drive, the DIR or DIRSYS command displays the message:

No Fi le

If system (SYS) files match the file specification, DIR displays the message:

SYSTEM FI LE(S) EXI ST

If nonsystem (DIR) files match the file specification, DIRSYS displays the message:

NON-SYSTEM FILES(S) EXIST

The DIR command pauses after filling the screen. Press any key to continue the display.

Note: You can use the DEVICE command to change the number of columns displayed by DIR or DIRSYS.

Examples: A>DIR

Displays all DIR files cataloged in user 0 on the default drive A.

A >DIR B:

Displays all DIR files for user 0 on drive B.

A >DIR B: X. BAS

Displays the name X.BAS if the file X.BAS is present on drive B.

4A>DIR *.BAS

Displays all DIR files with filetype BAS for user 4 on drive A.

B >DIP A: X*. C?D

Displays all DIR files for user 0 on drive A whose filename begins with the letter X, and whose three character filetype contains the first chat-acter C and last character D.

A>DIRSYS

Displays all files for user 0 on drive A that have the system (SYS)
'bute.
attri

3A >DIRS *. COM

This abbreviated form of the DIRSYS command displays all SYS files with filetype COM on the default drive A for user 3.

Display Directory with Options

Syntax: DIR {d:} [options]
DIR {filespec} {filespec} ... [options]

Explanation: The DIR command with options is an enhanced version of the DIR command. The DIR command displays CP/M 3 files In a variety of ways. DIR can search for files on any or all drives, for any or all user numbers.

DIR allows the option list to occur anywhere in the command tall. These options modify the entire command line. Only one option list is allowed.

Options must be enclosed in square brackets. The options can be used individually, or strung together separated by commas or spaces. Options can be abbreviated to only one or two letters if the abbreviation unam-biguously identifies the option.

If a directory listing exceeds the size of your screen, DIR automatically halts the display when it fills the screen. Press any key to continue the display.

Table 5-5. DIR Display Options

Option	Function
ATT	displays the user-definable file attributes Fl, F2, F3, and F4.
DATE	displays files with date and time stamps. If date and time stamping is not active, DIR displays the message: Date and Time Stamping Inactive
DIR	displays only files that have the DIR attribute.
DRIVE = ALL	displays files on all accessed drives. DISK is also acceptable in place of DRIVE in all the DRIVE options.
DRIVE = (A,B,C,...,P)	displays files on the drives specified.
DRIVE = d	displays files on the drive specified by d.

Table 5-5. (continued)

Option	Function
EXCLUDE	displays the files on the default drive and user area that do not match the files specified in the command line.
FF	sends an initial fo:-m-feed to the printer device if the printer has been activated by CTRL-P. If the LENGTH= n option is also specified, DIR issues a form-feed every n lines. Otherwise, the FF option deactivates the default paged output display.
FULL	shows the name of the file and the size of the file. The size is shown as the amount of space in kilobytes and the number of 128-byte records allocated to the file. FULL also shows the attributes the file. (See the SET command for description of file attributes). If there is a directory label on the drive, DIR shows the password protection mode and the time stamps. The display is alphabetically sorted. FULL is the default output format for display when using DIR with options.
LENGTH n	displays n lines of output before inserting a table heading. n must be in the range between 5 and 65536. The default length is one full screen of information.

Table 5-5. (continued)

Option	Function
MESSAGE	displays the names of the specified drives and user numbers it is currently searching. If there are no files in the specified locations, DIR displays the file not found message.
NOPAGE	continuously scrolls information by on the screen. Does not wait for you to press a key to restart the scrolling movement.
NOSORT	displays files in the order it finds them on the disk. If this option is not included, DIR displays the files alphabetically.
RO	displays only the files that have the Read-Only 'bute. attri
RW	displays only the files that are set to Read-Write.
SIZE	displays the filename and file size in kilobytes.
SYS	displays only the files that have the SYS attribute.

Table 5-5. (continued)

Option	Function
USER= ALL	displays all files under all the user numbers for the default drive.
USER = n	displays the files under the user number specified by n.
USER = (0, 15)	displays files under the user numbers specified.

Examples: A >DIR C: [FULL]
A >DIR C: [SIZE]

The following is sample output of the [FULL] option display format shown in the first example of the DIR command:

```
Directory for Drive C: User 0

  Name.      Bytes   Recs   Attributes Prot      Update          Access
------------ ----- ------- ---------- ------------ ------- -------------------- ------------------
DITS     BAK   1K       1 Dir  RW   Read  09/01/82 13:04  09/01/82 13:07
DITS     TES   1K       1 Dir  RO   None  09/01/82 13:07  09/01/82 13:09
DITS     Y     1K       1 Dir  RW   None  08/25/82 03:33  08/25/82 03:33
DITS     ZZ    1K       1 Dir  RW   None  08/25/82 03:36  00/25/82 03:36
SETDEF   COM   4K      29 Dir  RO   None                  08/25/82 03:38
SUBMIT   TX2   1K       1 Dir  RO   None
SUBMIT   TXl   5K      43 Dir  RO   None

Total Bytes      =   14k   Total Records =   77   Files Found   =   7
Total lk BlocKs  =   14   Used/Max Dir Entries for Drive C:   1 1 /  64
```

The following is sample output of the [SIZE] option display format shown in the second example of the DIR command:

```
Directory for Drive C.     User 0

C; DITS      BAK    1K :  DITS      TES     lK   DITS       Y        lK
C: DITS      ZZ     lK :  SETDEF COM    4K   SUBMIT     TX2      IK
C: SUBMIT  TXl      5K :

Total Bytes             14K Total   Records        77 Files Found         7
Total IK Blocks          14   Used/Max Dir Entries for Drive C:       II/ 64
```

Both the full format and the size format follow their display with two lines of totals. The first line displays the total number of kilobytes, the total number of records, and the total number of files for that drive and user area. The second line displays the total number of 1K blocks needed to store the listed files. The number of 1K blocks shows the amount of storage needed to store the files on a single density disk, or on any drive that has a block size of one kilobyte. The second line also shows the number of directory entries used per number of directory entries available on the drive.

A>DIR [DRIVE=C,FF]

DIR sends a form-feed to the printer before displaying the files on drive C.

A>DIR D: [RW,SYS]

The preceding example displays all the files on drive D with Read-Write and SYS attributes.

A>DIR C: [USER=ALL]

Displays all the files under each user number (0-15) on drive C.

A>DIR [USER=2]

Displays all the files under user 2 on the default drive.

A >DIR C: [USER= (3 ,4 , 10)]

This example displays all the files under user numbers 3, 4, and 10 on drive C.

A>DIR [DRIVE=ALL]

Displays all the files under user 0 on all the drives in the drive search chain. (See the SETDEF command.)

4A>DIR [DRIVE=C]

Displays all the files under user 4 on drive C.

A >DIR [DRIVE= (B ,D)]

Displays all the files under user 0 on drives B and D.

A>DIR]lexclude] *.COM

The preceding example above lists all the files on the default drive and user 0 that do not have a filetype of COM.

A>DIR [user=all ,drive=all ,sys] *.PLI *.COM *.ASM

The preceding command line instructs DIR to list all the system files of type PLI, COM, and ASM on the system in the currently active drives for all the user numbers on the drives.

A >DIR X. SUB [MESSAGE , USER=ALL ,DRIVE=ALL]

The preceding command searches all drives under each user number for X.SUB. During the search, DIR displays the drives and user numbers.

A>DIR [drive=all ,user=all] TESTFILE.BOB

The preceding example instructs DIR to display the filename T'ESTFILE.BOB if it is found on any logged-in drive for any user number.

A>DIR [size,rw]D:

The preceding example instructs DIR to list each Read-Write file that resides on drive D with its size in kilobytes. Note that D: is equivalent to D:

The DUMP Command

Syntax: DUMP filespec

Explanation: Dump displays the contents of a file in hexadecimal and ASCII format.

Example: A >DUMP ABC.TEX

Console output can look like the following:

```
DUMP - Yersion 3.0
0000: 41 42 43 OD OA 44 45 46 OD OA 47 48 49 OD OA IA ABC..DEF..GHI ....
0010: 1A IA IA 1A IA LA IA IA IA 1A 1A 1A 1A IA 1A IA
```

The ED Command

Syntax: ED linput-filespec Id: I output-filespecll

Explanation: The ED transient utility lets you create and edit a disk file.

The ED utility is a line-oriented context editor. This means that you create and change character files line-by-line, or by referencing individual characters within a line.

The ED utility lets you create or alter the file named in the file specification. Refer to Section 6 for a description of the ED utility.

The ED utility uses a portion of your user memory as the active text buffer where you add, delete, or alter the characters in the file. You use the A command to read all or a portion of the file into the buffer. You use the W or E command to write all or a portion of the characters from the buffer back to the file.

An imaginary character pointer, called CP, is at the beginning of the buffer, between two characters in the buffer, or at the end of the buffer.

You interact with the ED utility in either command or insert mode. ED displays the * prompt on the screen when ED is in command mode. When the * appears, you can enter the single letter command that reads text from the buffer,, moves the CP, or changes the ED mode of operation. When in command mode, you can use the line-editing characters CTRL-C, CTRL-E, CTRL-H, CTRL-U, CTRL-X, and RUBOUT to edit your input. In insert mode, however, you use only CTRL-H, CTRL-U, CTRL-X, and RUBOUT.

Table 5-6. ED Command Summary

Command	Action
nA	Append n lines from original file to memory buffer.
0A	Append file until buffer is one half full.
*A	Append file until buffer is full (or end of file).
B, -B	Move CP to beginning (B) or bottom (-B) of buffer.
nC , -nC	Move CP n characters forward (C) or back (-C) through buffer.
nD , -nD	Delete n characters before (-D) or from (D) the CP.
E	Save new file and return to CP/M 3.
Fstring{^Z}	Find character string.

Table 5-6. (continued)

Command	Action
H	Save the new file, then reedit, using the new file as the original file,
I	Enter insert mode; use T Z or ESCape to exit insert mode.
Istririg{^Z}	Insert string at CP.

Note: upper-case I forces all input to upper-case; while lower-case i allows upper- and lower-case.

Command	Action
Jsearch-str^Zins-str^Zdel_to_str{^Z}	juxtapose strings.
nK , -nK	Delete (kill) n lines from the CP.
nL , -nL , 0L	Move CP n lines.
nMcommands	Execute commands n times.
n , - n	Move CP n lines and display that line.

Table 5-6. (continued)

Command	Action
n:	Move to line n.
:ncommand	Execute command through line n.
Nstring{^Z}	Extended find string.
O	Return to original file.
nP , -nP	Move CP n lines forward and display n lines at console.
Q	Abandon new file, return to CP/M 3.
R{^'Z}	Read X$$$$$$$.LIB file into buffer.
Rfilespec{^Z}	Read filespec into buffer.
Sdelete string^Zinsert string{^ Z}	Substitute string.

Table 5-6. (continued)

Command	Action
nT , -nT , 0T	Type n lines.
U, -U	Upper-case translation.
V -V , 0V	Line numbering on/off, display free buffer space.
nW	Write n lines to updated file.
n X{^Z}	Write or append n lines to X$$$$$$$.LIB.
nXfilespec{^Z}	Write n lines to filespec or append if previous x command applied to the same file.
oX{^Z}	Delete file X$$$$$$$.LIB.
0Xfilespec{^Z}	Delete filespec.
nZ	Wait n seconds.

Section 6 gives a detailed description of the overall operation of the ED utility and the use of each command.

If you do not include a command tail in the ED command, it prompts you for the input filespec and the output filespec as follows:

Enter Input File :

After you enter the input filespec, ED prompts again:

Enter Output Fi le :

Enter a filename or drive if you want the output file or its location to be different from that of the input file. Press RETURN if you want the output file to replace the input file. In this case, the input file is renamed to type BAK.

If the second file specification contains only the drive specifier, the second filename and filetype become the same as the first filename and filetype.

If the file given by the first file specification is not present, ED creates the file and writes the message:

NEW FILE

If the file given by the first filespec is already present, you must issue the A command to read portions of the file to the buffer. If the size of the file does not exceed the size of the buffer, the command

a

reads the entire file to the buffer.

The 1 (Insert) command places ED in insert mode. In this mode, any characters you type are stored in sequence in the buffer starting at the current CP.

Any single letter commands typed in insert mode are not interpreted as commands, but are simply stored in the buffer. To return from insert mode to command mode, press CTRL-Z or the ESC kev. Note that you can always substitute the ESC key for CTRL-Z in ED.

The single letter commands are usually typed in lower-case. The commands that must be followed by a character sequence end with CTRL-Z if they are to be followed by another command letter.

Any single letter command typed in upper-case tells ED to internally translate to upper-case all characters up to the CTRL-Z that ends the command.

When enabled, line numbers that appear on the left of the screen take the form:

nnnnn:

where nnnnn is a number in the range 1 through 65535. Line numbers are displayed for your reference and are not contained in either the buffer or the character file. The screen line starts with

when the CP is at the beginning or end of the buffer.

Examples: A'>ED MYPROG,.PAS

If not already present, this command line creates the file MYPROG.PAS on drive A. The command prompt

appears on the screen. This tells you that the CP is at the beginning of the buffer. If the file is already present, issue the command

: *

to fill the buffer. Then type the command

:*#a

to fill the screen with the first n lines of the buffer , where n is the current default page size (See the DEVICE command to set the page size).

Type the command

```
:*e
```

to stop the ED utility when you are finished changing the character file. The ED utility leaves the original file unchanged as MYPROG.BAK and the altered file as MYPROG.PAS.

```
A >ED MYPROG.PAS B:NEWPROG.PAS
```

The original file is MYPROG.PAS on the default drive A. The original file remains unchanged when the ED utility finishes, with the altered file stored as NEWPROG.PAS on drive B.

```
A >B:ED MYPROG.PAS B :
```

The ED.COM file must be on drive B. The original file is MYPROG.PAS located on drive A. It remains unchanged, with the altered program stored on drive B as MYPROG.PAS.

The ERASE Command

Syntax: ERASE ifilespecl i[CONFIRM]l

Explanation: The ERASE command removes one or more files from a disk's directory in the current user number. Wildcard characters are accepted in the filespec. Directory and data space are automatically reclaimed for later use by another file. The ERASE command can be abbreviated to ERA.

Use the ERASE command with care because all files in the current user number that satisfy the file specification are removed from the disk directory.

Command lines that take the form

ERASE {d:}wlldcard-filespec

require your confirmation because they erase an entire group of files, not 'ust one file. The system prompts with the following message:

ERASE {d:}wildcard-filespec (Y/N)?

Respond with y if you want to remove all matching files, and n if you want to avoid erasing any files.

If no files match the file specification, you see the following message:

No Fi le

The CONFIRM option informs the system to prompt for verification before erasing each file that matches the filespec. You can abbreviate CONFIRM to C.

If you use the CONFIRM option with wildcard-filespec, then ERASE prompts for confirmation for each file. You can selectively erase the files you want by responding Y to the confirm message, or keep the files by responding N to the confirm message.

Examples: A>ERASE X.PAS

This command removes the file X.PAS from the disk in drive A.

A> ERA *.PRN

The system asks to confirm:

ERASE *.PRN (Y/N)?Y

All files with the filetype PRN are removed from the disk in drive A.

B>ERA A:MY*. * [CONFIRM]

Each file on drive A with a filename that begins with MY is displayed with a question mark for confirmation. Type Y to erase the file displayed, N to keep the file.

A >ERA B: *.*
ERASE B:*.* (Y/N) ? Y

All files on drive B are removed from the disk.

The GENCOM Command

Syntax: GENCOM ICOM-filespecl fRSX-filespecl...
F[LOADERINULLISCB = (offset,value)])

Explanation: The GENCOM command is a transient utility that creates a special COM file with attached RSX files. RSX files are used as Resident System Extensions and are discussed in detail in the CPIM Plus (CP/M Version 3) Operating System Programmer's Guide. GENCOM places a special header at the beginning of the output program file to indicate to the system that RSX loading is required. It can also set a flag to keep the program loader active.

The GENCOM command can also restore a file already processed by GENCOM to the original COM file without the header and RSXS. GENCOM has three options that help you attach RSX files:

0 The LOADER option sets a flag to keep the program loader active. (For complete details on the LOADER option read about CP/M function 59 in the CPIM Plus (CPIM Version 3) Operating System Programmer's Guide.) This option is used only if no RSX files are attached to the COM file.

0 The NULL option indicates that only RSX files are specified. GENCOM creates a dummy COM file for the RSX files. The output COM filename is taken from the filename of the first RSX-filespec.

0 The SCB = (offset,value) option sets the System Control Block from the program by using the hex values specified by (offset,value). For complete details on the SCB option read about CP/M function 49 in the CPIM Plus (CPIM Version 3) Operating System Programmer's Guide.

Attach RSX Files to a COM File

Syntax: GENCOM COM-filespec RSX-filespec...
{[LOADER|SCB = (offset,value)]}

Explanation: The preceding form of the GENCOM command creates a COM file with a header and attached RSXS. A maximum of 15 RSXs can be attached. GENCOM expects the first filespec to be a COM file and the following filespecs to be RSX files. Note that the original COM file is replaced by the newly-created COM file.

Example: A >GENCOM MYPROG PROG1 PROG2

The preceding command generates a new COM file MYPROG.COM with attached RSXs PROG1 and PROG2.

Generate a COM File Using only RSX Files

Syntax: GENCOM RSX-filespec {RSX-filespec}...
[NULL {SCB = (offset,value)}]

Explanation: The preceding form of the GENCOM command attaches the RSX files to a dummy COM file. GENCOM creates a COM file with the file-name of the first RSX-filespec in the command tail. This format allows the system to load RSXs directly.

Example: A>GENCOM PROGI PROG2 [NULL]

The preceding command creates a COM file PROG1.COM with Resident System Extensions PROG1.RSX and PROG2.RSX.

Restore a File with Attached RSXs to Original COM File

Syntax: GENCOM filename

Explanation: The preceding form of the GENCOM file takes a file that has already been processed by GENCOM and restores it to its original COM file format. This form of the command assumes a filetype of COM.

Example: A> GENCOM MYPROG

In the preceding command, GENCOM takes MYPROG.COM, strips off the header and deletes all attached RSXs to restore it to its original COM format.

Update (Add or Replace) RSX Files

Syntax: GENCOM COM-filespec RSX-filespec...
I[LOADER I SCB = (offset,value)]l

Explanation: The preceding form of the GENCOM command adds and/or replaces RSX files to a file already processed by GENCOM.

GENCOM inspects the list of RSX files. If they are new, they are added to the file already processed by GENCOM. If they already exist, then GENCOM replaces the existing RSXs with the new RSX files.

Example: A> GENCOM MYPROG PROGI PROG2

In the preceding example, GENCOM looks at MYPROG.COM, which is already processed by GENCOM, to see if PROG1.RSX and PROG2.RSX are already attached RSX files in the module. If either one is already attached, GENCOM replaces it with the new RSX module. Otherwise, GENCOM appends the specified RSX files to the COM file.

Attach a Header Record

Syntax: GENCOM filename [SCB = (offset,value),... | LOADER]

Explanation: The preceding syntax line attaches a GENCOM header record, with the SCB or loader flag set, to a file of type COM that contains no RSXS. This form of the command does not attach RSXs to a file.

Examples: A>GENCOM FILETWO [loader]

The preceding command attaches a 256-byte header record to the file FILETWO.COM and sets the loader flag in the header record.

A>GENCOM FILEFOUR [scb=(l ,l)]

The preceding command causes the program loader to set byte 1 of the System Control Block to 1 when it loads FILEFOUR.COM.

For more information, see functions 49, Set/Get System Control Block, and 59, Load Overlay or Resident System Extensions, in the CP/M Plus (CP/M Version 3) Operating System Programmer's Guide.

The GET Command

Syntax: GET {CONSOLE INPUT FROM} FILE filespec
{[{ECHO|NO ECHO}|SYSTEM]}
GET {CONSOLE INPUT FROM} CONSOLE

Explanation: The GET command is a transient utility that directs CP/M 3 to take console input from a file. The file can contain CP/M 3 system commands and/or input for a user program. If you use the SYSTEM option, GET immediately takes the next system command from the file.

Console input is taken from a file until the program terminates. If the file is exhausted before program input is terminated, the program looks for subsequent input from the console. If the program terminates before exhausting all its input, the system reverts back to the console for console input.

When the SYSTEM option is used, the system immediately goes to the file specified for console input. If you omit the SYSTEM option, you can enter one system command to initiate a user program whose console input is taken from the file specified in the GET command. The system reverts to the console for input when it reaches the end of the GET file input. The system also reverts to the console for console input if a GET CONSOLE INPUT FROM CONSOLE command is included in the input file.

Get Console Input from a File

Syntax: GET {CONSOLE INPUT FROM} FILE filespec {[options]}

Explanation: The preceding form of the GET command tells the system to get subsequent console input from a file. Table 5-7 lists the GET options that you use in the following format:

[{ECHO | NO ECHO} | SYSTEM]

Table 5-7. GET Options

Option	Meaning
ECHO	specifies that the input is echoed to the console. This is the default option.
NO ECHO	specifies that the file input is not to be echoed to the console. The program output and the system prompts are not affected by this option and are still echoed to the console.
SYSTEM	specifies that all system input is to be taken from the disk file specified in the command line. GET takes system and program input from the file until the file is exhausted or until GET reads a GET console command from the file.

Examples: A> GET FILE XINPUT
A>MYPROG

The preceding sequence of commands tells the system to activate the GET utility. However, because SYSTEM is not specified, the system reads the next input line from the console and executes MYPROG. If MYPROG program requires console input, it is taken from the file XINPUT. When MYPROG terminates, the system reverts to the console for console input.

A>GET FILE XIN2 [SYSTEM]

The preceding command immediately directs the system to get subsequent console input from file XIN2 because it includes the SYSTEM option. The system reverts to the console for console input when it reaches the end of file in XIN2. Or, XIN2 can redirect the system back to the console if it contains a GET CONSOLE command.

Terminate Console Input from a File

Syntax: GET {CONSOLE INPUT FROM} CONSOLE

Explanation: Ile preceding form of the GET command tells the system to get console input from the console.

Example: A >GET CONSOLE

The preceding GET command tells the system to get console input from the console. You can use this command in a file (previously specified in a GET FILE command) which is already being read by the system for console input. It is used to redirect the console input to the console before the end of the file is reached.

The HELP Command

Syntax: HELP {topic}{subtopicl subtopic2 ... subtopic8}{[NO PAGEILIST]}
HELP [EXTRACT]
HELP[CREATE]

Explanation: The HELP command is a transient utility that provides summarized information for all of the CP/M 3 commands described in this manual. In the distributed CP/M 3 system, HELP presents general information on a command as a topic and detailed information on a command as a subtopic. HELP with no command tall displays a list of all the available topics. HELP with a topic in the command tail displays information about that topic, followed by any available subtopics. HELP with a topic and a subtopic displays information about the specific subtopic.

After HELP displays the information for your specified topic, it displays the special prompt HELP> on your screen. Subtopics can be accessed by preceding the subtopic with a period. The period causes the subtopic search to begin at the last known level. You can continue to specify topics for additional information, or simply press the RETURN key to return to the CP/M 3 system prompt.

You can abbreviate the names of topics and subtopics. Usually one or two letters is enough to specifically identify the topics.

Display Information

Syntax: HELP topic {subtopicl ... subtopic8}{[NO PAGE!LIST]}
HELP>.Subtopic

Explanation: The preceding forms of the HELP command display the information for the specified topic and subtopics. Use the following two options with this form of the HELP command:

o The NOPAGE option disables the default paged display of every n lines, where n is the number of lines per page as set by the system or as set by the user. To stop the display, press CTRL-S. To resume the display, press CTRL-Q. You can abbreviate NOPAGE to N. (See the DEVICE command for more information about setting the number of lines per page.)

o The LIST option is the same as NOPAGE, except that it eliminates extra lines between headings. Use this option with CTRL-P to list the help information on the printer.

Examples: A>HELP

The preceding command displays a list of topics for which help is avaiable.

A>HELP DATE

This command displays general information about the DATE command. It also displays any available subtopics.

A>HELP DIR OPTIONS [N]

The preceding command includes the subtopic options. In response, HELP displays information about options associated with the DIR command. The display is not in paged mode.

A >HELP ED

The preceding command displays general information about the ED utility.

A:>HELP ED COMMANDS

This form of HELP displays information about commands internal to ED. The preceding example can also be entered as

A>HELP ED
HELP>.CDMMANDS

Add Your Own Descriptions to the HELP.HLP File

Syntax: HELP [EXTRACT,]
HELP [CREATE]

Explanation: CP/M 3 is distributed with two related HELP files: HELP.COM and HELP.HLP. The HELP.COM file is the command file that processes the text of the HELP.HLP file and displays it on the screen. The HELP.HLP file is a text file to which you can add customized information, but you cannot directly edit the HELP.HLP file. You must use the HELP.COM file to convert HELP.HLP to a file named HELP.DAT before you can edit or add your own text.

This form of the HELP command has the following options:

o The EXTRACT option accesses the file HELP.HLP on the default drive and creates a file called HELP.DAT on the default drive. You can now invoke a word processing program to edit or add your own text to the HELP.DAT file. EXTRACT can be abbreviated to E.
o The CREATE option accesses your edited HELP.DAT file on the default drive and builds a revised HELP.HLP file on the default drive. CREATE can be abbreviated to C.

You must add topics and subtopics to the HELP.DAT file in a specific format. A topic heading in the HELP.DAT file takes the form:

///nTopicname<cr>

The three backslashes are the topic delimiters and must begin in column one. In the preceding format statement, n is a number in the range from 1 through 9 that signifies the level of the topic. A main topic always has a level number of 1. The first subtopic has a level number of 2. The next level of subtopic has a level number of 3, and so forth, up to a maximum of nine levels. Topicname is the name of your topic, and allows a maximum of twelve characters. The entire line is terminated with a carriage return.

Use the following guidelines to edit and insert text into the HELP.DAT file.

o Topics should be placed in alphabetical order.
o Subtopics should be placed alphabetically within their respective supertopic.
o Levels must be indicated by a number 1-9.

Some examples of topic and subtopic lines in the HELP.HLP file follow

///lNEW UTILITY<cr>

///2COMMANDS<cr>

///3PARAMETERS<cr>

///2EXAMPLES<cr>

The first example illustrates the format of a main topic line. The second example shows how to number the first subtopic of that main topic. The third example shows how the next level subtopic under level 2 should be numbered. The fourth example shows how to return to the lower level subtopics Any topic name with a level number of 1 is a main topic. Any topic name with a level number of 2 is a subtopic within its main topic.

When you are executing the HELP.COM file, you need only enter enough letters of the topic to unambiguously identify the topic name. When referencing a subtopics you must type the topic name AND the subtopic, otherwise the HELP program cannot determine which main topic you are referencing. You can also enter a topic and subtopic following the program's internal prompt, HELP>, as follows

HELP>ED COMMANDS

This form of HELP displays information about commands internal to the editing program, ED.

The HEXCOM Command

Syntax: HEXCOM filename

Explanation: The HEXCOM command is a transient utility that generates a command file (filetype COM) from a HEX input file. It names the output file with the same filename as the input file but with filetype COM. HEXCOM always looks for a file with filetype HEX.

Example: A> HEXCOM B:PROGRAM

In the preceding command, HEXCOM generates a command file PROGRAM.COM from the input hex file PROGRAM.HEX.

The INITDIR Command

Syntax: INITDIR d:

Explanation: The INITDIR command can initialize a disk directory to allow date and time stamping of files on that disk or remove date and time stamps.

You must use INITDIR to initialize the directory for any disk on which you plan to record date and time stamps for your files. If the disk is blank, INITDIR initializes the directory to record date and time stamps. If files already exist on the disk, INITDIR checks the space available for date and time stamps in the directory. If there is not enough room for date and time stamps, INITDIR does not initialize the directory and returns an error message.

After you initialize the directory for date and time stamps, you must use the SET command to specify time stamp options on the disk.

Examples: A>INITDIR C:

The system prompts to confirm:

INITDIR WILL ACTIVATE TIME STAMPS FOR SPECIFIED DRIVE.
Do you really want to re-format the directory : C (Y/N) ?

If the directory has previously been initialized for date and time stamps, INITDIR displays the message:

Directory already re-formatted
Do you wish to recover date/time di rectory spac e
(Y / N) ?

Enter Y to reinitialize the directory to eliminate date and time stamps. If you enter N, date and time stamping remains active on your disk and INITDIR displays the following message:

Do you want the existing date/time stamps cleared (Y/N) ?

Enter Y to clear the existing stamps. Enter N to keep the existing date and time stamps.

The LIB Command

Syntax: LIB filespec{[I|M|P|D]}
LIB filespec{[I|M|P]}= filespec{modifier}
},filespec{modifier} ... }

Explanation:
A library file contains a collection of object modules. Use the LIB utility to create libraries, and to append, replace, select, or delete modules from an existing library. You can also use LIB to obtain information about the contents of library files.

LIB creates and maintains library files that contain object modules in MicroSoft@ REL format. These modules are produced by Digital Research's relocatable macro-assembler program, RMAC, or any other language translator that produces modules in MicroSoft REL format.

LINK-80" links the object modules contained in a library to other ob'ect files. LINK-80 automatically selects from the library only those
i
modules needed by the program being linked, and then forms an executable file with a filetype of COM.

The library file has the filetype REL or IRL depending on the option you choose. Modules in a REL library file must not contain backward references to modules that occur earlier in the library, because LINK-80 currently makes only one pass through a library.

Table 5-8. LIB Options

Option	Meaning
I	The INDEX option creates an indexed library file of type IRL. LINK-80 searches faster on indexed libraries than on nonindexed libraries.
M	The MODULE option displays module names.
P	The PUBLICS option displays module names and the public variables for the new library file.
D	The DUMP option displays the contents of object modules in ASCII form.

Use modifiers in the command line to instruct LIB to delete, replace, or select modules in a library file. Angle brackets enclose the modules to be deleted or replaced. Parentheses enclose the modules to be selected.

Unless otherwise specified, LIB assumes a filetype of REL for all source filenames. When you follow a filename by a group of module names enclosed in parentheses, these modules are included in the new library file. If modules are not specified, LIB includes all modules from the source file in the new library file.

Table 5-9. LIB Modifiers

Modifier	Meaning
Delete	<module = >
Replace	<module = filename.REL> If module name and filename are the same this shorthand can be used: <filename>
Select	(modFIRST-modLAST,modl,mod2,...,modN)

Examples:

A>LIB TEST4[p]
A>LIB TEST5[P]=FILE1 ,FILE2

The first example displays all modules and publics in TEST4.REL. The second example creates TEST5.REL from FILEL.REL and FILE2.REL, and displays all modules and publics in TEST5.REL.

A >L IB TEST= TESTI (MODI ,MOD4) , TEST2 (Cl -C4 ,C6)

In the preceding example LIB creates a library file TEST.REL from modules in two source files. TEST1.REL contributes MOD1 and MOD4. LIB extracts modules Cl, C4, all the modules located between them, and module C6 from TEST2.REL.

A>LIB FILE2=FILE3<MDDA=>

In this example, LIB creates FILE2.REL from FILE3.REL, omitting MODA which is a module in FILE3.REL.

A>LIB FILE6=FILE5<MODA=FILEB.REL>
A>LIB FILE6=FILE5<THISNAME>

In the first example, MODA is in the existing FILE5.REL. When LIB creates FILE6.REL from FILE5.REL, FILEB.REL replaces MODA.

In the second example, module THISNAME is in FILES.REL. When LIB creates FILE6.REI, from FILES.REL the file THISNAME.REL replaces the similarly named module THISNAME.

A>LIB FILE1[I]=B:FILE2(PLOTS,FIND,SEARCH-DISPLAY)

In this example LIB creates FILE1.IRL on drive A from the selected modules PLOTS, FIND, and modules SEARCH through the module DISPLAY, in FILE2.REL on drive B.

The LINK Command

Syntax: LINK d:{filespec,{[o]}=}filespeci{[o]}{,...}

Explanation:

The LINK command combines relocatable ob'ect modules such as those produced by RMAC and PL/1-80" into a COM file ready for execution. Relocatable files can contain external references and publics. Relocatable files can reference modules in library files. LINK searches the library files and includes the referenced modules in the output file. The LINK command is the LINK-80 utility and are synonymous in this discussion. See the Programmer's Utilities Guide for the CPIM Family of Operating Systems for a complete description of LINK-80.

You can use LINK option switches to control the execution parameters of LINK-80. LINK options follow the file specifications and are enclosed within square brackets. Multiple switches are separated by commas.

Table 5-10. LINK Options

Option	Meaning
A	Additional memory; reduces buffer space and writes temporary data to disk.
B	BIOS link in banked CP/M 3 system. Aligns data segment on page boundary; puts length of code segment in header; defaults to SPR filetype.
Dbhhh	Data origin; sets memory origin for common and data area.
Gn	Go; set start address to label n.
Lhhhh	Load; change default load address of module to hhhh. Default 0100H.
Mhhhh	Memory size; define free memory requirements for MP/M" modules.
NL	No listing of symbol table at console.

cp,m User's Guide The LINK Command

Table 5-10. (continued)

Option	Meaning
NR	No symbol table file.
OC	Output COM command file. Default.
OP	Output PRL page relocatable file for execution under MP/M in relocatable segment.
OR	Output RSP Resident System Process file for execution under MP/M.
OS	Output SPR System Page Relocatable file for execution under MP/M.
Phhhh	Program origin; changes default program origin address to hhhh. Default is 0100H.
Q	Lists symbols with leading question mark.
S	Search preceding file as a library.
$Cd	Destination of console messages, d, can be X for console, Y for printer, or Z for zero output. Default is X.
$Id	Source of intermediate files; d is disk drive A-P. Default is current drive.
$Ld	Source of library files; d is disk drive A-P. Default is current drive.
$Od	Destination of object file; d can be Z, or disk drive A-P. Default is to same drive as first file in the LINK-80 command.
$Sd	Destination of symbol file; d can be Y, Z, or disk drive A-P. Default is to same drive as first file in LINK-80 command.

Examples: A>LINK B:MYFILE[NR]

LINK-80 on drive A uses as input MYFILE.REL on drive B and produces the executable machine code file MYFILE.COM on drive B. The [NR] option specifies no symbol table file.

A>LINK ml ,m2, m3

LINK-80 combines the separately compiled files ml, m2, and m3, resolves their external references', and produces the executable machine code file ml.COM.

A>LINK m=m1, m2, m3

LINK-80 combines the separately compiled files ml, m2, and m3 and produces the executable machine code file M.COM.

A>LINK MYFILE,FILE5[s]

The [s] option tells LINK-80 to search FILE5 as a library. LINK-80 combines MYFILE.REL with the referenced subroutines contained in FILE5.REL on the default drive A and produces MYFILE.COM on drive A.

The MAC Command

Syntax: MAC filename {$options}

Explanation:

MAC, the CP/M Macro Assembler, is a transient utility that reads assembly language statements from a disk file of filetype ASM. MAC assembles the statements and produces three output files with the input filename and output filetypes of HEX, PRN, and SYM.

Filename.HEX contains IntelO hexadecimal format ob'ect code. You can debug the HEX file with a debugger, or use HEX COM to create a COM file and execute it.

Filename.PRN contains an annotated source listing that can be printed or examined at the console. The PRN file includes a 16-column wide listing at the left side of the page that shows the values of literals, machine code addresses, and generated machine code. An equal sign denotes literal addresses to eliminate confusion with machine code addresses.

Filename.SYM contains a sorted list of symbols defined in the program.

Before invoking MAC, you must prepare a source program file with the filetype ASM containing assembly language statements.

You can direct the input and output of MAC using the options listed in the following table. Use a letter with the option to indicate the source and destination drives, console, printer, or zero output. Valid drive names are A through 0. X directs output to the console. P directs output to the printer. Z specifies that output files will not be created.

Table 5-11. Input/Output Options

Option	Meaning
A	source drive for ASM file (A-0)
H	destination drive for HEX file (A-0, Z)
L	source drive for macro library LIB files called by the MACLIB statement.
P	destination drive for PRN file (A-0, X, P, Z)
S	destination drive for SYM file (A-0, X, P, Z)

Table 5-12. Output File Modifiers

Modifier	Meaning
+L	lists input lines read from macro library LIB files
- L	suppresses listing (default)
+M	lists all macro lines as they are processed during assembly
- M	suppresses all macro lines as they are read during assembly
*M	lists only hex generated by macro expansions
+Q	lists all LOCAL symbols in the symbol list
- Q	suppresses all LOCAL symbols in the symbol list (default)
+S	appends symbol file to print file
- S	suppresses creation of symbol file
+1	produces a pass 1 listing for macro debugging in PRN file
-1	suppresses listing on pass 1 (default)

Examples: A >MAC SAMPLE

In the preceding example MAC is invoked from drive A and operates on the file SAMPLE.ASM also on drive A.

A>MAC SAMPLE $PB AA HB SX

In this example, an assembly option parameter list follows the MAC command and the source filename. The parameters direct the PRN file to drive B. obtain the ASM file from drive A, direct the HEX file to drive B,, and send the SYM file to the console. You can use blanks between option parameters.

The PATCH Command

Syntax: PATCH filename {typ} {n}

Explanation: The PATCH command displays or installs patch number n to the CP/M 3 system or CP/M 3 command files.

Only CP/M 3 system files of filetype COM, PRL, or SPR can be patched with the PATCH command. If the typ option is not specified, the PATCH utility looks for a file with a filetype of COM.

The patch number n must be between 1 and 32 inclusive.

Examples: A > PATCH SHOW 2

The preceding command patches the system SHOW.COM file with patch number 2. The system displays the followin question:

Do you want to indicate that Patch #2
ha been installed for SHOW.COM? Y

If the patch is successful, the system displays the message:

Patch Installed

If the patch is not successful, the system displays the following message:

Patch not Installed

One of the following error messages might be displayed:

o ERROR: Patch requires CP/M 3.
o ERROR: Invalid filetype typ.
o ERROR: Serial Number mismatch.
o ERROR: Invalid patch number n

The PIP Command

Syntax: PIP dest-filespec|d:{[Gn]} = src-filespec{[o]}{,...} | d:{[o]}

Explanation: PIP is a transient utility that copies one or more files from one disk and/or user number to another. PIP can rename a file after copying it. PIP can combine two or more files into one file. PIP can also copy a character file from disk to the printer or other auxiliary logical output device. PIP can create a file on disk from input from the console or other logical input device. PIP can transfer data from a logical input device to a logical output device, thus the name Peripheral Interchange Program.

PIP copies file attributes with the file. This includes Read-Write or Read-Only and SYS or DIR file attributes and the user-definable attributes Fl through F4. If a file is password-protected, you must enter the password in the command line following the filename and/or filetype to which it belongs. If the password fails, the file is skipped and the failure noted.

When you specify a destination file with a password, PIP assigns that password to the destination file and automatically sets the password protection mode to READ. When you specify a destination file with no password, PIP does not assign a password to the destination file. When you speci'fy only a destination drive, PIP assigns the same password and password protection mode to the destination file as specified in the source file. When you specify a destination file with a password, PIP automatically sets the password protection mode to READ. This means that you need a password to read the file. (See the SET command.)

Single File Copy

Syntax:

PIP d:{[Gn]} = src-filespec{[options]}

PIP dest-filespec{[Gn]} = d:{[options]}

PIP dest-filespec{[Gn]} = src-filespec{[o]}

Explanation: The first form shows the simplest way to copy a file. PIP looks for the file named by src-filespec on the default or optionally specified drive. PIP copies the file to the drive specified by d: and gives it the name specified by src-filespec. If you want, you can use the [Gn] option to place your destination file (dest-filespec) in the user number specified by n. The only option recognized for the destination file is [Gn]. Several options can be combined together for the source file specification (src-filespec). See the Table 5-13, PIP options.

The second form is a variation of the first. PIP looks for the file named by dest-filespec on the drive specified by d:, copies it to the default or optionally specified drive, and gives it the name specified by dest-filespec.

The third form shows how to rename the file after you copy it. You can copy it to the same drive and user number, or to a different drive and/or user number. Rules for options are the same. PIP looks for the file specified by src-filespec, copies it to the location specified in dest-filespec, and gives it the name indicated by dest-filespec.

Remember that PIP always goes to and gets from the current default user number unless you specify otherwise with the [Gn] option.

Before you start PIP, be sure that you have enough free space in kilobytes on your destination disk to hold the entire file or files that you are copying. Even if you are replacing an old copy on the destination disk with a new copy, PIP still needs enough room for the new copy before it deletes the old copy. Use the DIR command to determine filesize and the SHOW command to determine disk space. If there is not enough space, you can delete the old copy first by using the ERASE command.

Data is first copied to a temporary file to ensure that the entire data file can be constructed in the space available on the disk. PIP gives the temporary file the filename specified for the destination, with the filetype $$$. If the copy operation is successful, PIP changes the temporary filetype $$$ to the filetype specified in the destination.

If the copy operation succeeds and a file with the same name as the destination file already exists, the old file with the same name is erased before renaming the temporary file.

File attributes (DIR, SYS, RO, RW) are transferred with the files.

I

If the existing destination file is set to Read-Only (RO), PIP asks you if you want to delete it. Answer Y or N. Use the [W] option to write over Read-Only files.

You can include PIP options following each source name. There is one valid option ([Gn]-go to user number n) for the destination file specification. Options are enclosed in square brackets. Several options can be included for the source files. They can be packed together or separated by spaces. Options can verify that a file was copied correctly, allow PIP to read a file with the system (SYS) attribute, cause PIP to write over Read-Only files, cause PIP to put a file into or copy it from a specified user number, transfer from lower- to upper-case, and much more.

Examples:

A >PIP B:=A:oldfile.dat
A >PIP B:oldfile.dat=A:

Both forms of this command cause PIP to read the file oldfile.dat from drive A and put an exact copy of it onto drive B. This is called the short form of PIP, because the source or destination names only a drive and does not include a filename. When using this form you cannot copy a file from one drive and user number to the same drive and user number. You must put the destination file on a different drive or in a different user number. (See the section on PIP Options, and the USER Command.) The second short form produces exactly the same result as the first one. PIP looks for the file oldfile.dat on drive A, the drive specified as the source.

A > PIP B:newfile.dat=A:oldfile.dat

This command copies the file oldfile.dat from drive A to drive B and renames it to newfile.dat. The file remains as oldfile.dat on drive A. This is the long form of the PIP command, because it names a file on both sides of the command line.

A>PIP newfile.dat=oldfile.dat

Using this long form of PIP, you can copy a file from one drive and user number (usually user 0 because CP/M 3 automatically starts out in user 0-the default user number) to the same drive and user number. This gives you two copies of the same file on one drive and user number, each with a different name.

A >PIP B: PROGRAM. BAK=A:PROGRAM.DAT [GI]

The preceding command copies the file PROGRAM.DAT from user 1 on drive A to the current selected user number on drive B and renames the filetype on drive B to BAK.

B > PIP program2.dat=A:programl. dat [E V G3]

In this command, PIP copies the file named programl.dat on drive A and echoes [E] the transfer to the console, verifies [V] that the two copies are exactly the same, and gets [G3] the file programl.dat from user 3 on drive A. Because there is no drive specified for the destination,, PIP automatically copies the file to the default user number and drive, in this case user 0 and drive B.

Multiple File Copy

Syntax: PIP d:{[Gn]} ={d:}wildcard-filespec{[options]}

Explanation: When you use a wildcard in the source specification, PIP copies matching files one-by-one to the destination drive, retaining the original name of each file. PIP displays the message COPYING followed by each filename as the copy operation proceeds. PIP issues an error message and aborts the copy operation if the destination drive and user number are the same as those specified in the source.

Examples: A > PIP B:=A:*.COM

This command causes PIP to copy all the files on drive A with the filetype COM to drive B.

A > PI P B: =A:

This command causes PIP to copy all the files on drive A to drive B. You can use this command to make a back-up copy of your distribution disk. Note, however, that this command does not copy the CP/M 3 system from the system tracks. COPYSYS copies the system for you.

A>PIP B:=A:PROG????.*

The preceding command copies all files whose filenames be in with PROG

91

from drive A to drive B.

A>PIP B:[GI]=A:*.BAS

This command causes PIP to copy all the files with a filetype of BAS on drive A in the default user number (user 0) to drive B in user number 1. Remember that the DIR, TYPE, ERASE, and other commands only access files in the same user number from which they were invoked. (See the USER Command.)

Combining Files

Syntax: PIP dest-filespec{[Gn]} = src-filespec{[o]}, src-filespec{[o]}{,...}

Explanation: This form of the PIP command lets you specify two or more files in the source. PIP copies the files specified in the source from left to right and combines them into one file with the name indicated by the destination file specification. This procedure is called file concatenation. You can use the [Gn] option after the destination file to place it in the user number specified by n. You can specify one or more options for each source file.

Some of the options force PIP to copy files character-by-character. In these cases, PIP looks for a CTRL-Z character to determine where the end of the file is. All of the PIP options force a character transfer except the following:

A, C, Gn, K, O, R, V, and W.

Copying data to or from logical devices also forces a character transfer.

You can terminate PIP operations by typing CTRL-C.

When concatenating files, PIP only searches the last record of a file for the CTRL-Z end-of-file character. However, if PIP is doing a character transfer, it stops when it encounters a CTRL-Z character.

Use the [O] option if you are concatenating machine code files. The [O] option causes PIP to ignore embedded CTRL-Z (end-of-file) characters, which indicate the end-of-file character in text files, but might be valid data in object code files.

Examples: A>PIP NEWFILE=FILEI,FILE2,FILE3

The three files named FILE1, FILE2, and FILE3 are 'oined from left to right and copied to NEWFILE.$$$. NEWFILE.$$$ is renamed to EW-FII,E upon successful completion of the copy operation. All source and destination files are on the disk in the default drive A.

A>PIP B:X.BAS = Y.BAS, B:Z.BAS

The file Y.BAS on drive A is joined with Z.BAS from drive B and placed in the temporary file X.$$$ on drive B. The file X.$$$ is renamed to X.BAS on drive B when PIP runs to successful completion.

Copy Files to and from Auxiliary Devices

Syntax: PlPdest-filespecl[Gn]) = src-filespecf[o])

AUX:	AUX: f [o]l
CON:	CON: l[ojl
PRN:	NUL:
LST:	EOF:

Explanation: This form is a special case of the PIP command line that lets you copy a file from a disk to a device, from a device to a disk or from one device to another. The files must contain printable characters. Each peripheral device is assigned to a logical device that identifies a source device that can transmit data or a destination device that can receive data. (See the DEVICE command.) A colon follows each logical device name so it cannot be confused with a filename. Enter CTRL-C to abort a copy operation that uses a logical device in the source or destination.

The logical device names are listed as follows:

- o CON: Console input or output device. When used as a source, usually the keyboard; when used as a destination, usually the screen.
- o AUX: Auxiliary Input or Output Device.
- o LST: The destination device assigned to the list output device, usually the printer.

The following three device names have special meaning:

- o NUL: A source device that produces 40 hexadecimal zeros.
- o EOF: A source device that produces a single CTRL-Z, the CP/M 3 end-of-file mark.
- o PRN: The printer device with tab expansion to every eighth column, line numbers, and page ejects every sixtieth line.

Examples: B>PIP PRN:=CON:MYDATA.DAT

Characters are first read from the console input device, generally the keyboard, and sent directly to your printer device. You type a CI'RL-Z character to tell PIP that keyboard input is complete. At that time, PIP continues by reading character data from the file MYDATA.DAT on drive B. Because PRN: is the destination device, tabs are expanded, line numbers are added, and page ejects occur every sixty lines.

Note that when the CON: device is the source you must enter both the carriage return (RETURN) and line-feed (LF) keys for a new line.

A>PIP B:FUNFILE.SUE = CON:

Whatever you type at the console is written to the file FUNFILE.SUE on drive B. End the keyboard input by typing a CTRL-Z.

A>PIP LST:=CON:

Whatever you type at the console keyboard is written to the list device, generally the printer. Terminate input with a CTRL-Z.

A>PIP LST:=B:DRAFT.TXT[TB]

The file DRAFT.TXT on drive B is written to the printer device. Any tab characters are expanded to the nearest column that is a multiple of 8.

A>PIP PRN:=B:DRAFT.TXT

The preceding command causes PIP to write the file DRAFT.TXT to the list device. It automatically expands the tabs, adds line numbers, and ejects pages after sixty lines.

Multiple Command Mode

Syntax: PIP

Explanation: This form of the PIP command starts the PIP utility and lets you type multiple command lines while PIP remains in user memory.

PIP writes an asterisk on your screen when ready to accept input command lines.

You can type any valid command line described under previous PIP formats following the asterisk prompt.

Terminate PIP by pressing only the RETURN key following the asterisk prompt. The empty command line tells PIP to discontinue operation and return to the CP/M 3 system prompt.

Examples:

```
A>PIP
CP/M 3 PIP VERSION 3*0
*NEWFILE=FILE1,FILE2,FILE3
* APROG.COM=BPROG.COM
* A:=B:X.BAS
* B : = *.*
*^M
A>
```

This command loads the PIP program. The PIP command input prompt, *, tells you that PIP is ready to accept commands. The effects of this sequence of commands are the same as in the previous examples, where the command line is included in the command tail. PIP is not loaded into memory for each command. To exit this PIP command mode, press RETURN or one of its equivalent control characters, CTRL-J or CTRL-M as shown.

Using Options With PIP

Explanation: With options you can process your source file in special ways. You can expand tab characters, translate from upper- to lower-case, extract portions of your text, verify that the copy is correct, and much more.

The PIP options are listed in Table 5-13 using n to represent a number and s to represent a sequence of characters terminated by a CTRL-Z. An option must immediately follow the file or device it affects. The option must be enclosed in square brackets [ ]. For those options that require a numeric value, no blanks can occur between the letter and the value.

You can include the [Gn] option after a destination file specification. You can include a list of options after a source file or source device. An option list is a sequence of single letters and numeric values that are optionally separated by blanks and enclosed in square brackets

Table 5-13. PIP Options

Option	Function
A	Copy only the files that have been modified since the last copy. To back up only the files that have been modified since the last back-up, use PIP with the archive option, [A].
C	Prompt for confirmation before performing each copy operation. Use the [C] option when you want to copy only some files of a particular filetype.
Dn	Delete any characters past column n. This parameter follows a source file that contains lines too long to be handled by the destination device, for example, an 80-character printer or narrow console. The number n should be the maximum column width of the destination device.
E	Echo transfer at console. When this parameter follows a source name, PIP displays the source data at the console as the copy is taking place. The source must contain character data.
F	Filter form-feeds. When this parameter follows a source name, PIP removes all form-feeds embedded in the source data. To change form-feeds set for one page length in the source file to another page length in the destination file, use the F command to delete the old form-feeds and a P command to simultaneously add new form-feeds to the destination file.
Gn	Get source from or go to user number n. When this parameter follows a source name, PIP searches the directory of user number n for the source file. When it follows the destination name, PIP places the destination file in the user number specified by n. The number must be in the range 0 to 15.

Table 5-13. (continued)

Option	Function
H	Hex data transfer. PIP checks all data for proper Intel hexadecimal file format. The console displays error messages when errors occur.
I	Ignore :00 records in the transfer of Intel hexadecimal format file. The I option automatically sets the H option.
L	Translate upper-case alphabetics in the source file to lower-case in the destination file. This parameter follows the source device or filename.
N	Add line numbers to the destination file. When this parameter follows the source filename, PIP adds a line number to each line copied, starting with 1 and incrementing by one. A colon follows the line number. If N2 is specified, PIP adds leading zeros to the line number and inserts a tab after the number. If the T parameter is also set, PIP expands the tab.
0	Object file transfer for machine code (noncharacter and therefore nonprintable) files. PIP ignores any CTRL-Z end-of-file during concatenation and transfer. Use this option if you are combining object code files.
Pn	Set page length. n specifies the number of lines per page. When this parameter modifies a source file, PIP includes a page eject at the beginning of the destination file and at every n lines. If n = 1 or is not specified, PIP inserts page e'ects every sixty lines. When you also specify the F option, PIP ignores form-feeds in the source data and inserts new form-feeds in the destination data at the page length specified by n.

Table 5-13. (continued)

Option	Function
Qs	Quit copying from the source device after the string s.When used with the S parameter, this parameter can extract a portion of a source file. The string argument must be terminated by CTRL-Z.
R	Read system (SYS) files. Usually, PIP ignores files marked with the system attribute in the disk directory. But when this parameter follows a source filename, PIP copies system files, including their attributes, to the destination.
Ss	Start copying from the source device at the string s. The string argument must be terminated by CTRL-Z. When used with the Q parameter, this parameter can extract a portion of a source file. Both start and quit strings are included in the destination file.
Tn	Expand tabs. When this parameter follows a source filename, PIP expands tab (CTRL-1) characters in the destination file. PIP replaces each CTRL-1 with enough spaces to position the next character in a column divisible by n.
U	Translate lower-case alphabetic characters in the source file to upper-case in the destination file. This parameter follows the source device or filename.
V	Verify that data has been copied correctly. PIP compares the destination to the source data to ensure that the data has been written correctly. The destination must be a disk file.

Table 5-13. (continued)

Option	Function
W	Write over files with RO (Read-Only) attribute. Usually, if a PIP command tail includes an existing RO file as a destination, PIP sends a query to the console to make sure you want to write over the existing file. When this parameter follows a source name, PIP overwrites the RO file without a console exchange. If the command tail contains multiple source files, this parameter need follow only the last file in the list.
Z	Zero the parity bit. When this parameter follows a source name, PIP sets the parity bit of each data byte in the destination file to zero. The source must contain character data.

Examples: A >PIP NEWPROG.BAS=CODE.BAS[L],DATA.BAS[U]

This command constructs the file NEWPROG.BAS on drive A by joining the two files CODE.BAS and DATA.BAS from drive A. During the copy operation, CODE.BAS is translated to lower-case, while DATA.BAS is translated to upper-case.

A> PIP CON: =WIDEFILE.BAS[D80]

This command writes the character file WIDEFILE.BAS from drive A to the console device, but deletes all characters following the 80th column position.

A > PIP B: =LETTER.TXT[E]

The file LETTER.TXT from drive A is copied to LETTER.TXT on drive B. The LETTER.TXT file is also written to the screen as the copy operation proceeds.

A > PIP LST: =B: LONGPAGE.TXT[FP65]

This command writes the file LONGPAGE.TXT from drive B to the printer device. As the file is written, form-feed characters are removed and reinserted at the beginning and every 65th line thereafter.

B > PIP LST: = PROGRAM.BAS[NTBU]

This command writes the file PROGRAM.BAS from drive B to the printer device. The N parameter tells PIP to number each line. The T8 parameter expands tabs to every eighth column. The U parameter translates lower-case letters to upper-case as the file is printed.

A > PIP PORTION.TXT=LETTER.TXT[SDear Sr^Z QSincerely^Z]

This command abstracts a portion of the LET-FER.TXT file from drive A by searching for the character sequence "Dear Sir" before starting the copy operation. When found, the characters are copied to PORTION.TXT on drive A until the sequence "Sincerely" is found in the source file.

B > PI P B:=A:*.COM[VMR]

This command copies all files with filetype COM from drive A to drive B. The V parameter tells PIP to read the destination files to ensure that data was correctly transferred. The W parameter lets PIP overwrite any destination files that are marked as RO (Read-Only). The R parameter tells PIP to read files from drive A that are marked with the SYS (System) attribute.

The PUT Command

Syntax:

PUT CONSOLE {OUTPUT TO} FILE filespec{[o]}
PUT PRINTER {OUTPUT TO} FILE filespec{[o]}
PUT CONSOLE {OUTPUT TO} CONSOLE
PUT PRINTER {OUTPUT TO} PRINTER

Explanation:

The PUT command is a transient utility that lets you direct console output or printer output to a file. PUT allows you to direct the system to put console output or printer output to a file for the next system command or user program entered at the console. Or, PUT directs all subsequent console or printer output to a file when you include the SYSTEM option.

Console output is directed to a file until the program terminates. Then, console output reverts to the console. Printer output is directed to a file until the program terminates. Then printer output is directed back to the printer.

When you use the SYSTEM option, all subsequent console/printer output is directed to the specified file. This option terminates when you enter the PUT CONSOLE or PLTT PRINTER command.

The syntax for the option list is

[{ECHO | NO ECHO} {FILTER | NO FILTER} | {SYSTEM}]

Table 5-14 defines the preceding option list.

Table 5-14. PUT Options

Option	Meaning
ECHO	specifies that the output is echoed to the console. ECHO is the default option when you direct console output to a file.
NO ECHO	specifies that the file output Is not to be echoed to the console.
FILTER	specifies that filtering of control characters is allowed, which means that control characters are translated to printable characters. For example, an escape character is translated to ^[.
NO FILTER	means that PUT does not translate control characters. This is the default option.
SYSTEM	specifies that system output and program output is written to the file specified by filespec. Output is written to the file until a subsequent PUT CONSOLE command redirects console output back to the console.

Direct Console Output to a File

Syntax: PUT CONSOLE {OUTPUTI TO FILE filespec f[o])

Explanation: The preceding form of the PUT command tells the system to direct subsequent console output to a file.

Example: A> PUT CONSOLE OUTPUT TO FILE XOUT [ECHO]

The preceding command directs console output to file XOUT with the output echoed to the console.

Put Printer Output to a File

Syntax: PUT PRINTER {OUTPUT TO} FILE filespec {[o]}

Explanation: The preceding form of the PUT command directs printer output to a file.

The options are the same as in the PUT CONSOLE command, except that option NO ECHO is the default for the PUT PRINTER command. Note that if ECHO is specified, printer output is echoed to the printer.

Examples: A> PUT PRINTER OUTPUT TO FILE XOUT
A>MYPROG

The preceding example directs the printer output of program MYPROG to file XOUT. The output is not echoed to the printer.

A> PUT PRINTER OUTPUT TO FILE XOUT2 [ECHO ,SYSTEM]

The preceding command directs all printer output to file XOUT2 and to the printer, and the PUT is in effect until you enter a PUT PRINTER OUTPUT TO PRINTER command.

The printer output can be directed to one or more files. The output to these files is terminated when you revert printer output to the printer using the following command:

PUT PRINTER OUTPUT TO PRINTER

Terminate Console Output to a File

Syntax: PUT CONSOLE FOUTPUT TOI CONSOLE

Explanation: The preceding form of the PUT command directs console output to the console.

Example: A > PUT CONSOL E OUTPUT TO CONSOLE

The preceding command directs console output to the console.

Terminate Printer Output to a File

Syntax: PUT PRINTER {OUTPUT TO} PRINTER

Explanation: The preceding form of the PUT command directs the printer output to the printer.

Example: A> PUT PRINTER OUTPUT TO PRINTER

The preceding example directs printer output to the printer.

The RENAME Command

Syntax: RENAME {new-filespec = old-filespec}

Explanation: The RENAME command lets you change the name of a file that is cataloged in the directory of a disk. It also lets you change several filenames if you use wildcards in the filespecs. You can abbreviate RENAME to REN.

The new-filespec must not be the name of any existing file on the disk. The old-filespec identifies an existing file or files on the disk.

The RENAME command changes the file named by old-filespec to the name given as new-filespec.

RENAME does not make a copy of the file. RENAME changes only the name of the file.

If you omit the drive specifier, RENAME assumes the file to rename is on the default drive. You can include a drive specifier as a part of the newname. If both file specifications name a drive, it must be the same drive.

If the file given by oldname does not exist, RENAME displays the following message on the screen:

No File

If the file given by newname is already present in the directory, RENAME displays the following message on the screen:

Not renamed filename.typ file already exists
delete (Y/N ?

If you want to delete the old file, type Y to delete. Otherwise, type N to keep the old file and not rename the new file.

If you use wildcards in the filespecs, the wildcards in the new filespec must correspond exactly to the wildcards in the old filespec. For example, in the following two commands, the wildcard filespecs correspond exactly:

```
A > REN *.TXI=*.TEX
A >REN A*.T*=S*.T*
```

In the following example, the wildcards do not match and CP/M 3 returns an error message.

```
A >REN A*.TEX=A.T*
```

Examples: A >RENAME NEWASM.BAS=OLDFILE.BAS

The file OLDFILE.BAS changes to NEWASM.BAS on drive A.

```
A>RENAME
```

The system prompts for the filespecs:

```
Enter New Name:X.PRN
Enter Old Name:Y.PRN
Y         PRN=X              PRN
A>
```

File Y.PRN is renamed X.PRN on drive A.

```
B>REN A:X.PAS=Y. PLI
```

The file Y.PLI changes to X.PAS on drive A.

```
A >RENAME S*.TEX=A*.TEX
```

The preceding command renames all the files matching the wildcard A*.TEX to files with filenames matching the wildcard S*.TEX, respectively.

A'> REN B: NEWLIST=B:OLDLIST

The file OLDLIST changes to NEWLIST on drive B. Because the second drive specifier, B: is implied by the first, it is unnecessary in this example. The preceding command line has the same effect as the following:

A>REN B:NEWLIST=OLDLIST

or

A >REN NEWLIST=B: OLDLIST

The RMAC Command

Syntax: RMAC filespec {$Rd | $Sd | $Pd}

Explanation: RMAC is a relocatable macro assembler that assembles files of type ASM into REL files that can be linked to create COM files.

The RMAC command options specify the destination of the output files. The additional specifier d defines the destination drive of the output files. A-0 specifies drives A through 0. X means output to the console, P means output to the printer, and Z means zero output. Table 5-15 lists the RMAC command options.

Table 5-15. RMAC Command Options

Option	d =output option
R drive for REL file	(A-O, Z)
S drive for SYM file	(A-O, X, P, Z)
P drive for PRN file	(A-O, X, P, Z)

In the MAC command, the assembly parameter of H controls the destination of the HEX file. In the RMAC command this parameter is replaced by R, which controls the destination of the REL file; however, you cannot direct the REL file to the console or printer, RX or RP, because the REL file is not an ASCII file.

Examples: A>RMAC TEST $PX SB RB

In the preceding example RMAC assembles the file TEST.ASM from drive A, sends the listing file (TEST.PRN) to the console, puts the symbol file (TEST.SYM) on drive B and puts the relocatable object file (TEST.REL) on drive B.

The SAVE Command

Syntax: SAVE

Explanation: The SAVE command copies the contents of memory to a file. To use the SAVE utility, first issue the SAVE command, then run your program which reads a file into memory. When your program exits, it exits to the SAVE utility. The SAVE utility prompts you for the filespec to which the memory is to be copied, and the beginning and ending address of the memory to be saved.

Example: A > SAVE

The preceding command activates the SAVE utility. Now enter the name of the program that loads a file into memory.

A >SID dumip.com

Next, execute the program.

g0

When the program exits, SAVE intercepts the return to the system and prompts you for the filespec and the bounds of memory to be saved.

```
SAYE Ye r 3 . 0
File (or RETURN to exit)?dump2.com
Delete dump2.com?Y
From?100
To?400

A>
```

The contents of memory from 100H, hexadecimal, to 400H is copied to file DUMP2.COM.

The SET Command

Syntax: SET [options]
SET d: [options]
SET filespec [options]

Explanation: The SET command initiates password protection and time stamping of files in the CP/M 3 system. It also sets file and drive attributes, such as the Read-Only, SYS, and user-definable attributes. It lets you label a disk and password protect the label.

The SET command include options that affect the disk directory, the drive, or a file or set of files. The discussion of the SET command explicitly states which of the three categories are affected.

To enable time stamping of files, you must first run INITDIR to format the disk directory.

Set File Attributes

Syntax: SET filespec[attribute-options]

Explanation: The preceding SET command sets the specified attributes of a file or a group of files.

Table 5-16. SET File Attributes

Option	Meaning
DIR	Sets the file from the SYS directory to the (DIR) attribute.
SYS	Gives the file the System SYS attribute.
RO	Sets the file attribute to allow Read-Only access.
RW	Sets the file attribute to allow Read-Write access.

Table 5-16. (continued)

Option	Meaning
ARCHIVE=OFF	Sets the archive attribute to off. This means that the file has not been backed up (archived). PIP with the [A] option can copy files with the archive attribute set to OFF. PIP with this option requires an ambiguous filespec and copies only files that have been created or changed since the last time they were backed up with the PIP[A] option. PIP then sets the archive attribute to ON for each file successfully copied.
ARCHIVE=ON	Sets the archive attribute to on. This means that the file has been backed up (archived). The archive attribute can be turned on explicitly by the SET command, or it can be turned on by PIP when copying a group of files with the PIP [A] option. The archive attribute is displayed by DIR.
Fl = ON|OFF	Turns on or off the user-definable file attribute Fl.
F2 = ON|OFF	Turns on or off the user-definable file attribute F2.
F3 = ON|OFF	Turns on or off the user-definable file attribute F3.
F4 = ON|OFF	Turns on or off the user-definable file attribute F4.

Example: A >SET MYFILE.TEX[RO SYS]

The preceding command sets MYFILE.TEX to Read-Only and System.

A>SET MYFILE.TEX [RW DIR]

The preceding command sets MYFILE.TEX to Read-Write with the Directory (DIR) attribute.

Set Drive Attribute

Syntax: SET {d:} [RO]
SET {d:} [RW]

Explanation: The preceding SET commands set the specified drive to Read-Only or Read-Write.

If a drive is set to Read-Only, PIP cannot copy a file to it, ERASE cannot delete a file from it, RENAME cannot rename a file on it. You cannot perform any operation that requires writing to the disk. When the specified drive is set to Read-Write, you can read or write to the disk in that drive. If you enter a CTRL-C at the system prompt, all drives are reset to Read-Write.

Example: A >SET B: [RO]

The preceding command sets drive B to Read-Only.

Assign a Label to the Disk

Syntax: SET {D:}[NAME = labelname.typ]

Explanation:

The preceding SET command assigns a label (name) to the disk in the specified or default drive.

CP/M 3 provides a facility for creating a directory label for each disk. The directory label can be assigned an eight-character name and a three-character type similar to a filename and filetype. Label names make it easier to catalog disks and keep track of different disk directories. The default label name is LABEL.

Example: A>SETINAME=DISK100J

The preceding example labels the disk on the default drive DISK100.

Assign Password to the Label

Syntax: SET [PASSWORD= password]
SET [PASSWORD= <cr>

Explanation: The first form of the preceding SET command assigns a password to the disk label. The second form of the command removes password protection from the label.

You can assign a password to the label. If the label has no password, any user who has access to the SET program can set other attributes to the disk which might make the disk inaccessible to you. However, if you assign a password to the label, then you must supply the password to set any of the functions controlled by the label. SET always prompts for the password if the label is password-protected.

Examples: A >SETEPASSWORD=SECRET]
A > SET [PASSWORD=<cr>

The first command assigns SECRET to the disk label. The second command nullifies the existing password.

Note: If you use password protection on your disk, be sure to record the password. If you forget the password, you lose access to your disk or files.

Enable/Disable Password Protection for Files on a Disk

Syntax: SET [PROTECT= ON]
SET [PROTECT= OFF]

Explanation: The first form of the SET command turns on password protection for all the files on the disk. The password protection must be turned on before you can assign passwords to individual files or commands.

The second SET command disables password protection for the files on your disk.

After a password is assigned to the label and the PROTECT option is turned on, you are ready to assign passwords to your files.

You can always determine if a disk is password-protected by using the SHOW command to display the label.

Assign Passwords to Files

Syntax: SET filespec[PASSWORD =password]

Explanation: The preceding SET command sets the password for filespec to the password indicated in the command tall. Passwords can be up to eight characters long. Lower-case letters are translated to upper-case.

You can use wildcards in the filespec. SET assigns the specified password to the files that match the wildcard-filespec.

Note: always record the passwords that you assign to your files. Without the password, you cannot access those files unless password protection is turned off for the whole disk. If you forget the password to the directory label, you cannot turn off the password protection for the disk.

Example: A>SET MYFILE.TEX[PASSWORD=MYFIL]

MYFIL is the password assigned to file MYFILE.TEX.

Set Password Protection Mode for Files with Passwords

Syntax:

```
SET filespec  [PROTECT= READ]
SET filespec [PROTECT= WRITE]
SET filespec [PROTECT= DELETE]
SET filespec [PROTECT= NONE]
```

Explanation: You can assign one of four modes of password protection to your file. The protection modes are READ, WRITE, DELETE, and NONE and are described in the following table.

Table 5-17. Password Protection Modes

Mode	Protection
READ	The password is required for reading, copying, writing, deleting, or renaming the file.
WRITE	The password is required for writing, deleting, or renaming the file. You do not need a password to read the file.
DELETE	The password is only required for deleting or re-naming the file. You do not need a password to read or modify the file.
NONE	No password exists for the file. If a password exists, this modifier can be used to delete the password.

Assign a Default Password

Syntax: SET [DEFAULT = password]

Explanation: The preceding set command assigns a default password for the system to use during your computer session. The system uses the default password to access password-protected files if you do not specify a password,, or if you enter an incorrect password. The system lets you access the file if the default password matches the password assigned to the file.

Example: B > SET *.TEX[PASSWORD = SECRET, PROTECT = WRITE]

The preceding command assigns the password SECRET to all the TEX files on drive B. Each TEX file is given a WRITE protect modc to prevent unauthorized editing.

Example: A>SET[DEFAULT=dd]

The preceding command instructs the system to use dd as a password if you do not enter a password for a password-protected file.

Set Time Stamp Options on Disk

Syntax: SET [CREATE= ON]
SET [ACCESS= ON]
SET [UPDATE= ON]

Explanation: The preceding SET commands allow you to keep a record of the time and date of file creation and update, or of the last access and update of your files.

[CREATE ON] turns on CREATE time stamps on the disk in the default drive. To record the creation time of a file, the CREATE option must have been turned on before the file is created.

[ACCESS = ON] turns on ACCESS time stamps on the disk in the default drive. ACCESS and CREATE options are mutually exclusive. This means that only one can be in effect at a time. If you turn on the ACCESS time stamp on a disk that has the CREATE time stamp, the CREATE time stamp is automatically turned off.

[UPDATE = ON] turns on UPDATE time stamps on the disk in the default drive. UPDATE time stamps record the time the file was last modified.

To enable time stamping', you must first run INITDIR to format the disk directory for time and date stamping.

Although there are three kinds of date/time stamps, only two date/time stamps can be associated with a given file at one time. You can choose to have either a CREATE date or an ACCESS date for files on a particular disk.

When you set both UPDATE and CREATE time stamps, notice that editing a file changes both the UPDATE and CREATE time stamps. This is because ED does not update the original file but creates a new version with the name of the original file.

Example: A>SET[ACCESS=ON]

The DIR with [FULL] option displays the following date and time stamps:

```
B>DIR   [FULL]

Directory for Drive  B:

   Name         Bytes  Recs  Attributes   Prot    Update       Access
------------------  ------ ----  ----------    ----  -------------- --------------
ONE      TEX      9K    71   Dir RW       None                08/03/81 10:56
THREE    TEX     12K    95   Dir RW       None                08/05/81 15:45
TWO      TEX     10K    76   Dir RW       None                08/10/81 09:13
```

The access time stamps displayed show the time the file was last displayed or edited. Note that displaying a filename in a directory listing does not constitute an access and is not recorded.

A>SET [CREATE=ON, UPDATE=ON]

The following DIR output below shows how files with create and update time stamps are displayed.

```
B>DIR [FULL]

Directory for Drive B

   Name            Bytes   Recs  Attributes  Prot     Update          Create
----------------------  -----    ----  ----------   ---- --------------     --------------
GENLED   .DAT   109K    873   Dir RW    None  08/05/81 14:01 08/01/81 09:36
RECEIPTS.DAT     59K    475   Dir RW    None  08/09/81 12:11 08/01/81 09:40
INVOICES.DAT     76K    608   Dir RW    None  08/08/81 08:46 08/01/81 10:15
```

Additional SET Examples

Examples: A>SET *.COM[SYS,RO,PASS=123,PROT=READ]

The preceding setting gives the most protection for all the COM files on drive A. With the password protection mode set to READ, you cannot even read one of the COM files without entering the password 123, unless the default password has been set to 123. Even if the correct password is entered, you still cannot write to the file because the file is Read-Only.

A>SET *.COM [RW, PROTECT=NONE,DIR]

The preceding command reverses the protection and access attributes of the COM files affected by the previous example. After executing the preceding command, there is no password protection, the files of type COM can be read from or written to, and are set to DIR files.

The SETDEF Command

Syntax: SETDEF {d:{,d:{,d:{,d:}}}} {TEMPORARY=d:] | [ORDER = (typ {,typ})]}
SETDEF [DISPLAY | NO DISPLAY]
SETDEF [PAGE | NO PAGE]

Explanation: The SETDEF command lets you display or define the disk search order, the temporary drive, and the filetype search order. The SETDEF definitions affect only the loading of programs and/or execution of SUBMIT (SUB) files. The SETDEF command also lets you turn on/off the DISPLAY and PAGE modes for the system. When DISPLAY mode is on, the system displays the location and name of programs loaded or SUB files executed. When PAGE mode is on, CP/M 3 utilities stop after displaying one full screen of information. Press any key to continue the display.

The system usually searches the specified drive or the default drive for files. The user can use the SETDEF command, to extend the search for program files and submit files, for execution purposes only.

Note: A CP/M 3 program file has a filetype of COM. A file containing commands to be executed by SUBMIT has a filetype of SUB.

Display the Program Loading Search Definitions

Syntax: SETDEF

Explanation: The preceding form of the SETDEF command displays the disk search order, the temporary drive, and the filetype search order.

Assign the Drive for Temporary Files

Syntax: SETDEF [TEMPORARY= D:]

Explanation: The preceding form of the SETDEF command defines the disk drive to be used for temporary files. The default drive used for temporary files is the system default drive.

Example: A>SETDEF [TEMPORARY=C:]

The preceding command sets disk drive C as the drive to be used for temporary files.

Define the Disk Drive Search Order

Syntax: SETDEF { d: {,d: {,d: {,d:}}}}

Explanation: The preceding form of the SETDEF command defines the disks to be searched by the system for programs and/or submit files to be executed. The CP/M 3 default is to search only the default drive.

Note: * can be substituted for d: to indicate that the default drive is to be included in the drive search order.

Example: A'>SETDEF C:,*

The preceding example tells the system to search for a program on drive C, then, if not found, search for it on the default drive.

Define the Filetype Search Order

Syntax: SETDEF [ORDER = (typ {,typ})]
where typ = COM or SUB

Explanation: The preceding form of the SETDEF command defines the filetype search order to be used by system for program loading. The filetype, indicated as typ in the syntax line, must be COM or SUB. The CP/M 3 default search is for COM files only.

Example: A>SETDEF [ORDER=(SUB,COM)]

The preceding command instructs the system to search for a SUB file to execute. If no SUB file is found, search for a COM file.

Turn On/Off System Display Mode

Syntax: SETDEF [DISPLAY | NO DISPLAY]

Explanation: The preceding command turns the system display mode on or off. The default system display mode is off. When the display mode is on, CP/M 3 displays the following information about a program file before loading it for execution: drive, filename, filetype (if any), and user number (if not the default user number).

Example: A>SETDEF [DISPLAY]

The preceding command turns on the system display mode. The system now displays the name and location of programs loaded or submit files executed. For example, if you enter the PIP command after turning on the system display mode, CP/M 3 displays the following:

```
A>PIP
A : P I P       COM
CP/M 3 PIP      VERSION 3.O
 *
```

indicating that the file PIP.COM was loaded from drive A under the current user number. If the current user number is not 0, and if PIP.COM does not exist under the current user number, then the system displays the location of PIP.COM as follows:

```
4A>PIP
A:PIP           COM (User 0)
CP/M 3 PIP VERSION 340
 *
```

indicating that PIP.COM was loaded from drive A under user number 0. This mode is in effect until you enter

SETDEF [NO DISPLAY]

to turn off the system DISPLAY mode,

Tum On/Off System Page Mode

Syntax: SETDEF [PAGE| NO PAGE]

Explanation: The preceding command turns on/off the system page mode. When the PAGE mode is set to on, CP/M 3 utilities stop after displaying one full screen of information, called a console page. The utilities resume after you press any key.

The default setting of the system page mode is ON.

Example: A>SETDEF [NO PAGE]

The preceding command turns off the system page mode. CP/M 3 utilities do not pause after displaying a full console page, but continue to scroll.

The SHOW Command

Syntax: SHOW {d:}{+4[SPACE |LABEL |USERS |DIR |DRIVE]}

Explanation: The SHOW command displays the following disk drive information:

- o access mode and amount of free disk space
- o disk label
- o current user number
- o number of files for each user number on the disk
- o number of free directory entries for the disk
- o drive characteristics

Display Access Mode and Disk Space Available

Syntax: SHOW {d:}{[SPACE]}

Explanation: The preceding form of the SHOW command displays the drive, the access mode for that drive, and the remaining space in kilobytes for the specified drive. SHOW by itself displays the information for all logged-in drives in the system.

Examples:

```
 A>SH0W B:
B: RW, Space:          9,488K
A>SHOW
A: RW, Space:              4K
B: RW, Space:          9,488K
```

The first example shows that drive B has Read-Write access and it has 9,488K bytes of space left. The second example shows that drive A also is Read-Write and has only 4K bytes left and drive B is Read-Write and has 9,488K bytes left.

Display Disk Label

Syntax: SHOW {d:}[LABEL]

Explanation: The preceding form of the SHOW command displays disk label information, Example: A > SH0W B: [LABEL]

The preceding command displays the following for drive B:

```
Label for drive B:

Directory       Passwds   Stamp Stamp   Label Created Label Updated
Label             Reqd    Create Update
------------------- -------   --------- ------- ----------------- --------------
TOMSDISK.         on        on        on    07/04/81 10:30 07/08/81 09:30
```

The first column, directory label, displays the name assigned to that drive directory. The second column, Passwds Reqd, shows that password protection has been turned on for that drive.

As described in the SET command, each file can have up to two time stamps. The first of these time stamps can be either the creation date and time for the file or the date and time of the last access to the file. Access is defined as reading from or writing to the file. The third column of the SHOW [LABEL] output displays both the type of stamp and whether or not it is on. In the preceding example, creation time stamps are given to new files as shown by the stamp create column heading.

The fourth column displays the status of the second time stamp field, the update time stamp. Update time stamps display the date and time of the last update to a file, that is, the last time someone wrote to the file. In the SHOW [LABEL] display, update time stamps are turned on.

Besides showing the password protection and the active time stamps on a drive, SHOW [LABEL] also displays the date and time that the label was created and last updated.

Display User Number Information

Syntax: SHOW Id:I[USERS]

Explanation: The preceding command displays the current user number and all the users on the drive and the corresponding number of files assigned to them.

Example: A>SH0W [USERS]

Active User : 1
Active Files : 0 2 3 4
A: # of files: 95 40 1 26
A: Number of free directory entries: 350
A>

Display Number of Free Directory Entries

Syntax: SHOW {d:}[DIR]

Explanation: The preceding command displays the number of free directory entries cil the specified drive.

Example: A >SHOW C: [DIR]

C: Number of free directory entries : 24

A>

The preceding command shows that there are only 24 free directory entries on drive C.

Display Drive Characteristics

Syntax: SHOW {d:}[DRIVE]

Explanation: The preceding form of the SHOW command displays th@ drive characteristics of the specified drive.

Example: A>SHOW [DRIVE]

The following is an example of the system display for the preceding command:

```
     A: Drive Characteristics
 3,600: 128 Byte Record Capacity
   450: Kilobyte Drive Capacity
    96: 32 Byte Directory Entries
    96: Checked Directory Entries
   128: Records / Directory Entry
    16: Records / Block
    48: Sectors / Track
   512: Bytes / Physical Record
```

The SID Command

Syntax: SID {pgm-filespec} {,sym-filespec}

Explanation: The SID (Symbolic Instruction Debugger) allows you to monitor and test programs developed for the 8080 microprocessor. SID supports real-time breakpoints, fully monitored execution, symbolic disassembly, assembly, and memory display and fill functions. Utility programs are supplied with CP/M 3 that can be dynamically loaded with SID to provide traceback and histogram facilities.

SID commands display memory and CPU registers and direct the breakpoint operations during the debugging session.

Without a file specification SID loads into memory without a test program. Use this form to examine memory or to write and test simple programs using the A command. You must not use the SID commands G, T, or U, described later, until you have first loaded a test program.

A SID command line with a pgm-filespec loads both SID and the test program into memory. If the filetype is omitted from the filespec, COM is assumed. SID optionally loads in a symbol table file specified by sym-filespec. The sym-filespec needs no filetype because SID looks for a file with filetype SYM. Use the C, G, T, or U command to begin execution of the test program under supervision of SID.

Use CTRL-S to halt the screen display. CTRL-Q restarts the display. Abort lengthy displays by typing any keyboard character. Use CTRL-C to exit from SID.

SID can address absolute memory locations through symbolic expressions. A symbolic expression evaluates to either an address or a data item.

A symbolic expression can be a name from a SYM file produced from your program by a CP/M Macro Assembler. When you precede the symbolic expression with a period, SID returns its address in hexadecimal. When you precede the symbolic expression with the at sign, @, SID returns the 16-bit value stored at that location and the next contiguous location. When you precede the symbolic expression with an equal sign, SID returns the 8-bit value stored at that location. For two-byte expressions, this is the low byte because the 8080 microprocessor stores the low value of a two-byte word first.

A symbolic expression can be a literal value in hex, decimal, or ASCII, as indicated in the following list:

- o SID uses literal hex values as given, but truncates any digits in excess of four on the left. The leftmost digit is the most significant digit. The rightmost digit is the least significant digit.
- o To indicate decimal values precede them with a pound sign, # Decimal values that evaluate to more than four hex digits are evaluated as the modulo of hex value FFFF. For example, #65534 = FFFEH, while #65536 = 0001H.
- o SID translates literal ASCII character strings between apostrophes to the hex value of the two rightmost ASCII characters.

You can combine symbolic expressions with the symbolic operators, + or -, to produce another symbolic expression. Symbolic expressions combined in this way can be used to calculate the offset of an indirectly addressed data item, for example a subscripted variable. A special up-arrow operator, , can reference the top-of-stack item. A string of n ' operators can reference the nth stack item without changing stack content or the stack pointer.

Table 5-18 lists the SID commands with their corresponding parameters and options. The actual command letter is printed in boldface. The parameters are in lower-case and follow the command letter. Optional items are in braces. Replace the arguments with the appropriate symbolic expressions as listed. Where two symbolic expressions are needed, SID can calculate the second one from the first using the symbolic operators described previously.

Table 5-18. SID Commands

Name	Syntax	Meaning
Assemble	As	Enter assembly language statements. s is the start address.
Call	Cs {b{,d}}	Call to memory location from SID. s is the called address, b is the value of the BC register pair, and d is the value of the DE register pair.
Display	D{W}{s}{,f}	Display memory in hex and ASCII. W specifies a 16-bit word format, s is the start address, and f is the finish address.
Load	Epgm-filespec {,sym-filespec}	Load program and symbol table for execution.
Load	E* sym-filespec	Load a symbol table file.
Fill	Fs,f,d	Fill memory with constant value. s is the start address, f is the finish address, and d is an 8-bit data item.
Go	G{p}{,a{,b}}	Begin execution. p is a start address, a is a temporary breakpoint, and b is a second temporary breakpoint. GO exits SID by performing a warm boot.
Hex	H Ha Ha,b	Displays all symbols with addresses in hex. The first syntax displays hex, decimal, and ASCII values of a. The second syntax performs number and character conversion, where a is a symbolic expression, and the third syntax computes hex sum and difference of a and b, where a and b are symbolic expressions.
Input	Icommand tail	Input CCP command line.
List	L {s}{,f}	List 8080 mnemonic instructions. s is the start address, and f is the finish address.

Table 5-18. (continued)

Name	Syntax	Meaning
Move	Ms,h,d	Move memory block. s is the start address, h is the high address of the block, and d is the destination start address.
Pass	P{p{,c}}	Pass point set, reset, and display. p is a permanent breakpoint address, and c is initial value of pass counter.
Read	Rfilespec{,d}	Read code/symbols. d is an offset to eac address.
Set	W{W}s	Set memory values. s is an address where value is sent, W is a 16-bit word.
Trace	Tfni,cll	Trace program execution. n is the number of program steps, and c is the utility entry address.
Trace	T{W}{n{,c}}	Trace without call. W instructs SID not to trace subroutines,, n is the number of program steps, and c is the utility entry address.
Untrace	U{W}{n{,c}}	Monitor execution without trace. n is the number of program steps, c is the utility entry address, W instructs SID not to trace subroutines.
Value	V	Display the value of the next available location in memory (NEXT), the next location after the largest file read in (MSZE), the current value of the program counter (PC), and the address of the end of available memory (END).
Write	Wfilespec{,s,}	Write the contents of a contiguous block of memory to filespec. s is the start address, f is the finish address.
Examine	X{f}{r}	Examine/alter CPU state. f is flag bit C, E, 1, M, or Z; r is register A, B, D, H, P or S.

Examples: A>SID

In the preceding example CP/M 3 loads SID from drive A into memory. SID displays the # prompt when it is ready to accept commands.

A>B:SID SAMPLE.HEX

In the preceding example, CP/M 3 loads SID and the program file SAMPLE.HEX into memory from drive B. SID displays:

```
NEXT MSZE    PC   END
nnnn   mmmm pppp eeee
```

In the preceding example, nnnn is a hexadecimal address of the next free location following the loaded program, and mmmm is the next location after the largest program. This is initially the same value as NEXT. pppp is the initial hexadecimal value of the the program counter. eeee is the hexadecimal address of the logical end of the TPA.

#DFEOO+#128+5

In the preceding example the first pound sign, #, is the SID prompt. This SID command, D, displays the values stored in memory starting at address FE80 (FEOO + #128) and ending at address FE85 (FE80 + 5).

SID Utilities

The SID utilities HIST.UTL and TRACE.UTL are special programs that operate with SID to provide additional debugging facilities. The mechanisms for system initialization, data collection, and data display are described in the CPIM SID Symbolic Instruction Debugger User's Guide. The following discussion illustrates how a utility is activated. You load the utility by naming it as a parameter when invoking SID:

SID filename.UTL

In the preceding example filename is the name of the utility. Following the initial sign-on, the utility can prompt you for additional debugging parameters.

The HIST utility creates a histogram (bar graph) showing the relative frequency of execution of code within selected program segments of the test program. The HIST utility allows you to monitor those sections of code that execute most frequently.

Upon start-up HIST prompts

TYPE HISTOGRAM BOUNDS

Enter the bounds in the following format-,

aaaa,bbbb

for a histogram between locations aaaa and bbbb inclusive. Collect data in U or T mode, then display results.

The TRACE utility obtains a traceback of the instructions that led to a particular breakpoint address in a program under test. You can collect the addresses of up to 256 instructions between pass points in U or T modes.

The SUBMIT Command

Syntax: SUBMIT {filespec} {argument} ... {argument}

Explanation: The SUBMIT command lets you execute a group or batch of commands from a SUB file, which is a file with filetype of SUB.

Usually, you enter commands one line at a time. If you must enter the same sequence of commands several times, you might find it easier to batch the commands together using the SUBMIT command. To do this, create a file and enter your commands in this file. The file is identified by the filename, and must have a filetype of SUB. When you issue the SUBMIT command, SUBMIT reads the file named by the filespec and prepares it for interpretation by CP/M 3. When the preparation is complete, SUBMIT sends the file to CP/M 3 line-by-line, as if you were typing each command.

The SUBMIT command executes the commands from a SUB file as if you are entering the commands from the keyboard.

You create the SUB file with the ED utility. It can contain CP/M 3 commands, nested SUBMIT commands, and input data for a CP/M 3 command or a program.

You can pass arguments to SUB files when you execute them. Each argument you enter is assigned to a parameter in the SUB file. The first argument replaces every occurrence of $1 in the file, the second argument replaces parameter $2, etc., up to parameter $9. For example, if your file START.SUB contains the following commands:

```
ERA $1.BAK
DIR $1
PIP $1 =A: $2.COM
```

and you enter the following SUBMIT command:

```
A> SUBMIT START SAM TEX
```

the argument SAM is substituted for every $1 in the START.SUB file, and TEX for every occurrence of $2 in the START.SUB file. SUBMIT then creates a file with the parameter substitutions and executes this file. This file now contains the following commands:

```
ERA SAM.BAK
DIR SAM
PIP SAM= A:TEX.COM
```

If you enter fewer arguments in the SUBMIT command than parameters in the SUB file,, the remaining parameters are not included in the commands.

If you enter more arguments in the SUBMIT command than parameters in the SUB file, the remaining arguments are ignored.

To include an actual dollar sign, $ in your SUB file, type two dollar signs, $$. SUBMIT replaces them with a single dollar sign when it substitutes an argument for a parameter in the SUB file. For example, if file AA.SUB contains line:

```
MAC $1 $$$2
```

and you enter the following SUBMIT command:

```
A>SUBMIT AA ZZ SZ
```

then the translated file contains the following:

```
MAC ZZ $SZ
```

Program Input Lines in a SUB File

A SUB file can contain program input lines. Any program input is preceded by a less than sign, <, as in the following example:

```
P I P
< B : =*. ASM
<CON:=DUMP#ASM
<
DIR
```

The three lines after PIP are input lines to the PIP command. The third line consists only of the < sign, indicating a carriage return. The carriage return causes PIP to return to the system to execute the final DIR command.

If the program terminates before using all of the input, SUBMIT ignores the excess input lines and displays the following warning message:

Warning : Program input ignored

If the program requires more input than is in the SUB file, it expects you to enter the remaining input from the keyboard.

You can enter control characters in a SUB file by using the usual convention of preceding the control character by an up-arrow character,, 1, followed by the letter to be converted to a control character. To enter an actual I character, use the combination 11. This combination translates to a single T in the same manner that $$ translates to a single

The SUB File

The SUB file can contain the following types of lines:

o Any valid CP/M 3 command
o Any valid CP/M 3 command with SUBMIT parameters
o Any data input line
o Any program input line with parameters ($0 to $9)

CP/M 3 command lines cannot exceed 128 characters.

Example: The following lines illustrate the variety of lines that can be entered in a SUB file:

```
D I R
D I R *.BAK
MAC   $1 $$$4
PIP LST: =$1.PRN[T$2 $3 $5]
DIR  *.ASM
PIP
< B:=*.A S M
<CON:=DUMP.ASM
 <
DIR B:
```

Executing the SUBMIT Command

Syntax: SUBMIT
SUBMIT filespec
SUBMIT filespec argument ... argument

If you enter only SUBMIT, the system prompts for the rest of the command. You enter the filespec and arguments.

Example: A>SUBMIT

The system displays the following prompt. Enter filespec and arguments here, such as:

Enter File to Submit: START B TEX

Another example could be

A > SUBMIT SUBA

Still another example using parameters is

A> SUBMIT AA ZZ SZ

where AA is the SUB file AA.SUB, ZZ is the argument to replace any occurrences of $1 in the AA.SUB file and SZ is the argument to replace all occurrences of $2 in the AA.SUB file.

The PROFILE.SUB Start-up File

Every time you turn on or reset your computer, CP/M 3 automatically looks for a special SUB file named PROFILE.SUB to execute. If it does not exist, then CP/M 3 resumes normal operation. If the PROFILE.SUB file exists, the system executes the commands in the file. This file is convenient to use if you regularly execute a set of commands before you do your regular session on the computer. For example, if you want to be sure that you always enter the current date and time on your computer before you enter any other commands, you can create the PROFILE.SUB file, with ED, and enter the DATE command as follows:

DATE SET

Then, whenever you bring up the system, the system executes the DATE command and prompts you to enter the date and time. By using this facility, you can be sure to execute a regular sequence of commands before starting your usual session.

The TYPE Command

Syntax: TYPE {filespec {[PAGE]|[NO PAGE]}}

Explanation: The TYPE command displays the contents of an ASCII character file on your screen. The PAGE option displays the console listing in paged mode, which means that the console listing stops automatically after listing n lines of text, where n is usually the system default of 24 lines per page. (See the DEVICE command to set n to a different value.) Press any character to continue listing another n lines of text. Press CTRL-C to exit back to the system. PAGE is the default mode.

The NO PAGE option displays the console listing continuously.

If you do not enter a file specification in the TYPE command the system prompts for a filename with the message:

Enter filename :

Respond with the filespec of the file you want listed.

Tab characters occurring in the file named by the file specification are expanded to every eighth column position of your screen.

At any time during the display, you can interrupt the listing by pressing CTRL-S. Press CTRL-Q to resume the listing.

Press CTRL-C to exit back to the system.

Make sure the file specification identifies a file containing character data.

If the file named by the file specification is not present on the specified drive, TYPE displays the following message on your screen:

No File

To list the file at the printer and on the screen, type a CTRL-P before entering the TYPE command line. To stop echoing console output at the printer, type a second CTRL-P. The type command displays the contents of the file until the screen is filled. It then pauses until you press any key to continue the display.

Examples: A > TYPE MYPROG. PLI

This command displays the contents of the file MYPROG.PLI on your screen, a page at a time.

A> TYPE B:THISFILE [NO PAGE]

This command continuously displays the contents of the file THISFILE from drive B on your screen.

The USER Command

Syntax: USER {number}

Explanation: The USER command sets the current user number. When you start CP/M 3, 0 is the current user number. You can use a USER command to change the current user number to another in the range 0-15.

CP/M 3 identifies every file with a user number. In general, you can access only files identified with the current user number. However, if you mark a file in user 0 with the SYS attribute, the file can be accessed from all other user numbers.

Examples: A > USER

The system command prompts for the user number, as follows:

Enter User# : 5
5A>

The current user number is now 5 on drive A.

A>USER 3
3A>

This command changes the current user number to 3.

The XREF Command

Syntax: XREF {d:} filename {$P}

Explanation: The XREF command provides a cross-reference summary of variable usage in a program. XREF requires the PRN and SYM files produced by MAC or RMAC for the program.

The SYM and PRN files must have the same filename as the filename in the XREF command tall. XREF outputs a file of type XRF.

Examples: A>XREF B:MYPROG

In this example, XREF is on drive A. XREF operates on the file MYPROG.SYM and MYPROG.PRN which are on drive B. XREF produces the file MYPROG.XRF on drive B.

A>XREF B:MYPROG $P

In the preceding example, the $P option directs output to the printer.

End of Section 5

Section 6

ED, The CP/M 3 Context Editor

6.1 Introduction to ED

To do almost anything with a computer you need some way to enter data, a way to give the computer the information you want it to process. The programs most commonly used for this task are called editors. They transfer your keystrokes at the keyboard to a disk file. CP/M 3's editor is named ED. Using ED, you can easily create and alter CP/M 3 text files.

The correct command format for invoking the CP/M 3 editor is given in Section 6.2, "Starting ED." After starting ED, you issue commands that transfer text from a disk file to memory for editing. Section 6.3, "ED Operation," details this operation and describes the basic text transfer commands that allow you to easily enter and exit the editor.

44

Section 6.4, Basic Editing Commands," details the commands that edit a file. Section 1,1, "Combining ED Commands," describes how to combine the basic commands to edit more efficiently. Although you can edit any file with the basic ED commands, ED provides several more commands that perform more complicated editing functions, as described in Section 6.6, "Advanced ED Commands."

During an editing session, ED can return two types of error messages. Section 6.7, "ED Error Messages," lists these messages and provides examples that indicate how to recover from common editing error conditions.

6.2 Starting ED

Syntax:

ED input-filespec {d: I output-filespecl

To start ED, enter its name after the CP/M 3 prompt. The command ED must be followed by a file specification, one that contains no wildcard characters, such as:

A > ED MYFILE.TEX

The file specification, MYFILE.TEX in the preceding example, specifies a file to be edited or created. The file specification can be preceded by a drive specification, but a drive specification is unnecessary if the file to be edited is on your default drive. Optionally, the file specification can be followed by a drive specification, as shown in the following example:

```
A>ED MYFILE.TEX B:
```

In response to this command, ED opens the file to be edited, MYFILE.TEX, on drive A, but sends all the edited material to a file on drive B.

Optionally, you can send the edited material to a file with a different filename, as in the following example:

```
A>ED MYFILE.TEX YOURFILE.TEX
```

The file with the different filename cannot already exist or ED prints the following message and terminates.

```
Output File Exists, Erase It
```

The ED prompt, *, appears at the screen when ED is ready to accept a command, as follows

```
A>ED MYFILE.TEX
    : *
```

If no previous version of the file exists on the current disk, ED automatically creates a new file and displays the following message:

```
NEW FILE
    : *
```

Note: before starting an editing session, use the SHOW command to check the amount of free space on your disk. Make sure that the unused portion of your disk is at least as large as the file you are editing, or larger if you plan to add characters to the file. When ED finds a disk or directory full, ED has only limited recovery mechanisms. These are explained in Section 6.7, "ED Error Messages."

6.3 ED Operation

With ED, you change portions of a file that pass through a memory buffer. When you start ED with one of the preceding commands, this memory buffer is empty. At your command, ED reads segments of the source file, for example MYFILE.TEX, into the memory buffer for you to edit. If the file is new, you must insert text into the file before you can edit. During the edit, ED writes the edited text onto a temporary work file, MYFILE.$$$.

When you end the edit, ED writes the memory buffer contents to the temporary file, followed by any remaining text in the source file. ED then changes the name of the source file from MYFILE.TEX to MYFILE.BAK, so you can reclaim this original material from the back-up file if necessary. ED then renames the temporary file, MYFILE.$$$, to MYFILE.TEX, the new edited file. The following figure illustrates the relationship between the source file, the temporary work file, and the new file.

Note: when you invoke ED with two filespecs, an input file and an output file, ED does not rename the input file to type BAK; therefore, the input file can be Read-Only or on a write-protected disk if the output file is written to another disk.

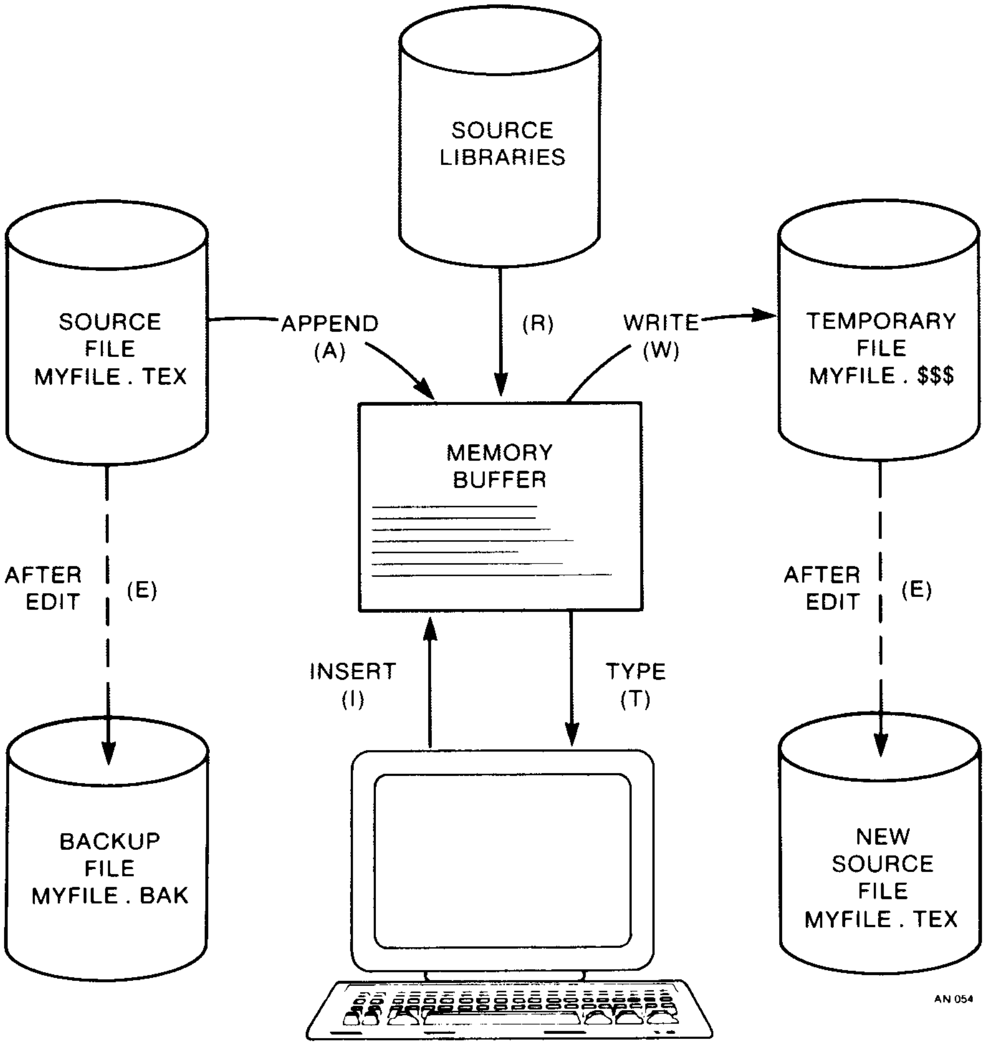

Figure 6-1. Overall ED Operation

In the preceding figure, the memory buffer is logically between the source file and the temporary work file. ED supports several commands that transfer lines of text between the source file, the memory buffer, and the temporary, and eventually final, file. The following table lists the three basic text transfer commands that allow you to easily enter the editor, write text to the temporary file, and exit the editor.

Table 6-1. Text Transfer Commands

Command	Result
nA	Append the next n unprocessed source lines from the source file to the end of the memory buffer.
nW	Write the first n lines of the memory buffer to the temporary file free space.
E	End the edit. Copy all buffered text to the temporary file, and copy all unprocessed source lines to the temporary file. Rename files.

6.3.1 Appending Text into the Buffer

When you start ED and the memory buffer is empty, you can use the A (append) command to add text to the memory buffer.

Note: ED can number lines of text to help you keep track of data in the memory buffer. The colon that appears when you start ED indicates that line numbering is turned on. Type -V after the ED prompt to turn the line number display off. Line numbers appear on the screen but never become a part of the output file.

The V (Verify Line Numbers) Command

The V command turns the line number display in front of each line of text on or off. The V command also displays the free bytes and total size of the memory buffer. The V command takes the following forms:

```
V, -V, 0V
```

Initially, the line number display is on. Use -V to turn it off. If the memory buffer is empty, or if the current line is at the end of the memory buffer, ED represents the line number as five blanks. The OV command prints the memory buffer statistics in the form:

free/total

where free is the number of free bytes in the memory buffer, and total is the size of the memory buffer. For example, if you have a total of 48,253 bytes in the memory buffer and 46,652 of them are free, the OV command displays this information as follows

46652/48253

If the buffer is full, the first field, which indicates free space, is blank.

The A (Append) Command

The A command appends, copies, lines from an existing source file into the memory buffer. The A command takes the following form:

nA

where n is the number of unprocessed source lines to append into the memory buffer. If a pound sign, #, is given in place of n, then the integer 65,535 is assumed. Because the memory buffer can contain most reasonably sized source files, it is often possible to issue the command #A at the beginning of the edit to read the entire source file into memory.

When n is 0, ED appends the unprocessed source lines into the memory buffer untill the buffer is approximately half full. If you do not specify n, ED appends one line from the source file into the memory buffer.

6.3.2 ED Exit

You can use the W (Write) command and the E (Exit) command to save your editing changes. The W command writes lines from the memory buffer to the new file without ending the ED session. An E command saves the contents of the buffer and any unprocessed material from the source file and exits ED.

The W (Write) Command

The W command writes lines from the buffer to the new file. The W command takes the form:

nW

where n is the number of lines to be written from the beginning of the buffer to the end of the new file. If n is greater than 0, ED writes n lines from the beginning of the buffer to the end of the new file. If n is 0, ED writes lines until the buffer is half empty. The OW command is a convenient way of making room in the memory buffer for more lines from the source file. If the buffer is full, you can use the OW command to write half the contents of the memory buffer to the new file. You can use the #W command to write the entire contents of the buffer to the new file. Then you can use the OA command to read in more lines from the source file.

Note: after a W command is executed, you must enter the H command to reedit the saved lines during the current editing session.

The E (Exit) Command

An E command performs a normal exit from ED. The E command takes the form:

E

followed by a carriage return.

When you enter an E command, ED first writes all data lines from the buffer and the original source file to the $$$ file. If a BAK file exists, ED deletes it, then renames the original file with the BAK filetype. Finally, ED renames the $$$ file from filename.$$$ to the original filetype and returns control to the operating system.

The operation of the E command makes it unwise to edit a back-up file. When vou edit a BAK file and exit with an E command, ED erases your original file because it has a BAK filetype. To avoid this, always rename a back-up file to some other filetype before editing it with ED.

Note: any command that terminates an ED session must be the only command on the line.

6.4 Basic Editing Commands

The text transfer commands discussed previously allow you to easily enter and exit the editor. This section discusses the basic commands that edit a file.

ED treats a file as a long chain of characters grouped together in lines. ED displays and edits characters and lines in relation to an imaginary device called the character pointer (CP). During an edit session, you must mentally picture the CP's location in the memory buffer and issue commands to move the CP and edit the file.

The following commands move the character pointer or display text in the vicinity of the CP. These ED commands consist of a numeric argument and a single command letter and must be followed by a carriage return. The numeric argument, n, determines the number of times ED executes a command; however, there are four special cases to consider in regard to the numeric argument:

- If the numeric argument is omitted, ED assumes an argument of 1.
- Use a negative number if the command is to be executed backwards through the memory buffer. The B command is an exception.
- If you enter a pound sign, #, in place of a number, ED uses the value 65,535 as the argument. A pound sign argument can be preceded by a minus sign to cause the command to execute backwards through the memory buffer, -#.
- ED accepts 0 as a numeric argument only in certain commands. In some cases, 0 causes the command to be executed approximately half the possible number of times, while in other cases it prevents the movement of the CP.

The following table alphabetically summarizes the basic editing commands and their valid arguments.

Table 6-2. Basic Editing Commands

Command	Action
B, -B	Move CP to the beginning (B) or end (-B) of the memory buffer.
nC, -nC	Move CP n characters forward (nC) or backward (-nC) through the memory buffer.
nD, -nD	Delete n characters before (-nD) or after (nD) the CP.
I	Enter insert mode.
Istring CTRL-Z	Insert a string of characters.
nK, -nK	Delete (kill) n lines before the CP (-nK) or after the CP (nK).
nL, -nL	Move the CP n lines forward (nL) or backward (-nL) through the memory buffer.
nT. -nT	Type n lines before the CP (-nT) or after the CP (nT).
n, -n	Move the CP n lines before the CP (-n) or after the CP (n) and display the destination line.

The following sections discuss ED's basic editing commands in more detail. The examples in these sections illustrate how the commands affect the position of the character pointer in the memory buffer. Later examples in Section 6.5, "Combining ED Commands," illustrate how the commands appear at the screen. For these sections, however, the symbol ' in command examples represents the character pointer, which you must imagine in the memory buffer.

6.4.1 Moving the Character Pointer

This section describes commands that move the character pointer in useful increments but do not display the destination line. Although ED is used primarily to create and edit program source files, the following sections present a simple text as an example to make ED easier to learn and understand.

The B (Beginning[Bottom) Command

The B command moves the CP to the beginning or bottom of the memory buffer. The B command takes the following forms:

B, -B

-B moves the CP to the end or bottom of the memory buffer; B moves the CP to the beginning of the buffer.

The C (Character) Command

The C command moves the CP forward or backward the specified number of characters. The C command takes the following forms:

nC, -nC

when n is the number of characters the CP is to be moved. A positive number moves the CP towards the end of the line and the bottom of the buffer. A negative number moves the CP towards the beginning of the line and the top of the buffer. You can enter an n large enough to move the CP to a different line. However, each line is separated from the next by two invisible characters: a carriage return and a line-feed, represented by <cr><If>. You must compensate for their presence. For example, if the CP is pointing to the beginning of the line, the command 30C moves the CP to the next line:

Emily Dickinson said,<cr><If>
"I fin^d ecstasy in living -<cr><If>

The L (Line) Command

-The L command moves the CP the specified number of lines. After an L command, the CP always points to the beginning of a line. The L command takes the following forms:

nL, -nL

where n is the number of lines the CP is to be moved. A positive number moves the CP towards the end of the buffer. A negative number moves the CP back toward the beginning of the buffer. The command 2L moves the CP two lines forward through the memory buffer and positions the character pointer at the beginning of the line.

"I find ecstasy in living -<cr><If>
the mere sense of living<cr><If>
is joy enough." <cr><If>

The command -L moves the CP to the beginning of the previous line, even if the CP originally points to a character in the middle of the line. Use the special character 0 to move the CP to the beginning of the current line.

The n (Number) Command

The n command moves the CP and displays the destination line. The n command takes the following forms:

n, -n

where n is the number of lines the CP is to be moved. In response to this command, ED moves the CP forward or backward the number of lines specified, then prints only the destination line. For example, the command -2 moves the CP back two lines.

Emily Dickinson said,<cr><If>
^"I find ecstasy In living -<cr><If>
the mere sense of living<cr><If>
is Joy enough."<cr><If>

A further abbreviation of this command is to enter no number at all. In response to a carriage return without a preceding command, ED assumes a n command of 1 and moves the CP down to the next line and prints it, as follows

Emily Dickinson said,<cr><If>
"I find ecstasy in living -<cr><If>
^the mere sense of living < cr> < If >

Also, a minus sign, -, without a number moves the CP back one line.

6.4.2 Displaying Memory Buffer Contents

ED does not display the contents of the memory buffer until you specify which part of the text you want to see. The T command displays text without moving the CP.

The T (Type) Command

The T command types a specified number of lines from the CP at the screen. The T command takes the forms

nT, -nT

where n specifies the number of lines to be displayed. If a negative number is entered, ED displays n lines before the CP. A positive number displays n lines after the CP. If no number is specified, ED types from the character pointer to the end of the line.
The CP remains in its original posit' ton no matter how many lines are typed. For example, if the character pointer is at the beginning of the memory buffer, and you instruct ED to type four lines (4T), four lines are displayed at the screen, but the CP stays at the beginning of line 1.

^Emily Dickinson said,<cr><If>
"I find ecstasy in living -<cr><If>
the mere sense of living<cr><If>
is joy enough."<cr><If>

If the CP is between two characters in the middle of the line, a T command with no number specified types only the characters between the CP and the end of the line, but the character pointer stays in the same position, as shown in the following memory buffer example:

"I find ec^stasy in living -

Whenever ED is displaying text with the T command, you can enter a CTRL-S to stop the display, then press any key when you are ready to continue scrolling. Enter a CTRL-C to abort long type-outs.

6.4.3 Deleting Characters

The D (Delete) Command

The D command deletes a specified number of characters and takes the forms:

nD, -nD

where n is the number of characters to be deleted. If no number is specified, ED deletes the character to the right of the CP. A positive number deletes multiple characters to the right of the CP, towards the bottom of the file. A negative number del etes charac ters to the left of the CP, towards the top of the file. If the character pointer is positioned in the memory buffer as follows

```
Emily Dickinson said,<cr><lf>
"I find ecstasy in living -<cr><lf>
the mere sense of living<cr><lf>
is joy ^enough."<cr><lf>
```

the command 6D deletes the six characters after the CP, and the resulting memory buffer looks like this:

```
Emily Dickinson said,<cr><lf>
"I find ecstasy in living -<cr><lf>
the mere sense of I'tving<cr><lf>
is joy^."<cr><lf>
```

You can also use a D command to delete the <cr><lf> between two lines to join them together. Remember that the <cr> and <lf> are two characters.

The K (-Kill) Command

The K command kills or deletes whole lines from the memory buffer and takes the forms:

nK, -nK

where n is the number of lines to be deleted. A positive number kills lines after the CP. A negative number kills lines before the CP. When no number is specified, ED kills the current line. If the character pointer is at the beginning of the second line,

```
Emily Dickinson said,<cr><If>
"I find ecstasy in living -<cr><If>
the mere sense of living<cr><If>
is joy enough." <cr><If >
```

then the command -K deletes the previous line and the memory buffer changes:

```
"I find ecstasy in living -<cr><If>
the mere sense of living<cr><If>
is joy enough."<cr><If>
```

If the CP is in the middle of a line, a K command kills only the characters from the CP to the end of the line and concatenates the characters before the CP with the next line. A -K command deletes all the characters between the beginning of the previous line and the CP. A OK command deletes the characters on the line up to the CP.

You can use the special # character to delete all the text from the CP to the beginning or end of the buffer. Be careful when using #K because you cannot reclaim lines after they are removed from the memory buffer.

6.4.4 Inserting Characters into the Memory Buffer

The I (Insert) Command

To insert characters into the memory buffer from the screen, use the I command.

If you enter the command in upper-case, ED automatically converts the string to upper-case. The I command takes the forms:

```
I
Istring^Z
```

When you type the first command, ED enters insert mode. In this mode, all keystrokes are added directly to the memory buffer. ED enters characters in lines and does not start a new line until you press the enter key.

```
A>ED B:QUOTE.TEX

 NEW FILE
    : *i
   1:  Emily Dickinson said,
   2:   "I find ecstasy, in living -
   3:   the mere sense of livig
   4:  is joy enough."
   5:  ^Z
   : *
```

Note: to exit from insert mode, you must press CTRL-Z or ESC. When the ED prompt, *, appears on the screen, ED is not in insert mode.

In command mode, you can use CP/M 3 command line-editing control characters. In insert mode, you can use the control characters listed in Table 6-3.

Table 6-3. CP/M 3 Line-editing Controls

Command	Result
CTRL-H	Delete the last character typed on the current line.
CTRL-U	Delete the entire line currently being typed.
CTRL-X	Delete the entire line currently being typed. Same as CTRL-U.
Backspace	Remove the last character.

When entering a combination of numbers and letters, you might find it inconvenient to press a caps-lock key if your terminal translates the upper-case of numbers to special characters. ED provides two ways to translate your alphabetic input to upper-case without affecting numbers. The first is to enter the insert command letter in uppercase: I. All alphabetics entered during the course of the capitalized command, either in insert mode or as a string, are translated to upper-case. If you enter the insert command letter in lower-case, all alphabetics are inserted as typed. The second method is to enter a U command before inserting text. Upper-case translation remains in effect until you enter a -U command.

The Istring^Z (Insert String) Command

The second form of the I command does not enter insert mode. It inserts the character string into the memory buffer and returns immediately to the ED prompt. You can use CP/M 3's line-editing control characters to edit the command string.

To insert a string, first use one of the commands that position the CP. You must move the CP to the place where vou want to insert a string. For example, if you want to insert a string at the beginning of the first line, use a B command to move the CP to the beginning of the buffer. With the CP positioned correctly, enter an insert string, as follows

iIn 1870, ^Z

This inserts the phrase "In 1870," at the beginning of the first line, and returns immediately to the ED prompt. In the memory buffer, the CP appears after the inserted string, as follows

In 1870,^ Emily Dickinson said, <cr><lf>

6.4.5 Replacing Characters

The S (Substitute) Command

The S command searches -the memory buffer for the specified string, but when it finds it, automatically substitutes a new string for the search string. Whenever you enter a command in upper-case, ED automatically converts the string to upper-case. The S command takes the form:

nssearch string^Znew string

where n is the number of substitutions to make. If no number is specified, ED searches for the next occurrence of the search string in the memorv buffer. For example, the command

sEmily Dickinson^ZThe Poet

searches for the first occurrence of "Emily Dickinson" and substitutes "The poet." In the memory buffer, the CP appears after the substituted phrase, as follows

The poet^ said, <cr>< If>

If upper-case translation is enabled by a capital S command letter, ED looks for a capitalized search string and inserts a capitalized insert string. Note that if you combine this command with other commands, you must terminate the new string with a CTRL-Z,

6.5 Combining ED Commands

It saves keystrokes and editing time to combine the editing and display commands. You can type any number of ED commands on the same line. ED executes the command string only after you press the carriage return key. Use CP/M 3's line-editing controls to manipulate ED command strings.

When you combine several commands on a line, ED executes them in the same order they are entered, from left to right on the command line. There are four restrictions to combining ED commands:

- o The combined-command line must not exceed CP/M 3's 128 character maximum.
- o If the combined-command line contains a character string, the line must not exceed 100 characters.
- o Commands to terminate an editing session must not appear in a combined-command line.
- o Commands, such as the 1, J, R, S, and X commands, that require character strings or filespecs must be either the last command on a line or must be terminated with a CTRL-Z or ESC character, even if no character string or filespec is given.

While the examples in the previous section show the memory buffer and the position of the character pointer, the examples in this section show how the screen looks during an editing session. Remember that the character pointer is imaginary, but you must picture its location because ED's commands display and edit text in relation to the character pointer.

6.5.1 Moving the Character Pointer

To move the CP to the end of a line without calculating the number of characters, combine an L command with a C command, L-2C. This command string accounts for the <cr><If> sequence at the end of the line.

Change the C command in this command string to move the CP more characters to the left. You can use this command string if you must make a change at the end of the line and you do not want to calculate the number of characters before the change, as in the following example:

```
     1-. *T
     1: Emily Dickinson said,
     1: *L-7CT
said,
     1: *
```

6.5.2 Displaying Text

A T command types from the CP to the end of the line. To see the entire line, you can combine an L command and a T command. Type Olt to move the CP from the middle to the beginning of the line and then display the entire line. In the following example, the CP is in the middle of the line. OL moves the CP to the beginning of the line. T types from the CP to the end of the line, allowing you to see the entire line.

```
     3: *T
sense of living
     3: *0LT
     3: the mere sense of living
     3: *
```

The command 0TT displays the entire line without moving the CP.

To verify that an ED command moves the CP correctly, combine the command with the T command to display the line. The following example combines a C command and a T command.

```
    2: *8CT
ecstasy in living -
    2:  *


    4:  *B#T
    1: Emily Dickinson said,
    2: "I find ecstasy in living -
    3: the mere sense of living
    4:    is joy enough."
    1: *
```

6.5.3 Editing

To edit text and verify corrections quickly, combine the edit commands with other ED commands that move the CP and display text. Command strings like the one that follows move the CP, delete specified characters, and verify changes quickly.

```
1. *15C5D0LT
1: Emily DicKinson,
1: *
```

Combine the edit command K with other ED commands to delete entire lines and verify the correction quickly, as follows

```
1: *2L2KB#T
1: Emily Dickinson said,
2: "I find ecstasy in living -
1 : *
```

The abbreviated form of the I (insert) command makes simple textual changes. To make and verify these changes, combine the I command string with the C command and the OLT command string as follows. Remember that the insert string must be terminated by a CTRL-Z.

```
1 : *20Ci to a f riend ^Z0-LT
1:   Emily Dickinson said to a friend,
1:  *
```

6.6 Advanced ED Commands

The basic editing commands discussed previously allow you to use ED for all your editing. The following ED commands, however, enhance ED's usefulness.

6.6.1 Moving the CP and Displaying Text

The P (Page) Command

Although you can display any amount of text at the screen with a T command, it is sometimes more convenient to page through the buffer, viewing whole screens of data and moving the CP to the top of each new screen at the same time. To do this, use ED's P command. The P command takes the following forms:

nP, -nP

where n is the number of pages to be displayed. If you do not specify n, ED types the 23 lines following the CP and then moves the CP forward 23 lines. This leaves the CP pointing to the first character on the screen.

To display the current page without moving the CP, enter OP. The special character 0 prevents the movement of the CP. If you specify a negative number for n, P pages backwards towards the top of the file.

The n: (Line Number) Command

When line numbers are being displayed, ED accepts a line number as a command to specify a destination for the CP. The line number command takes the following form:

n:

where n is the number of the destination line. This command places the CP at the beginning of the specified line. For example, the command 4: moves the CP to the beginning of the fourth line.

Remember that ED dynamically renumbers text lines in the buffer each time a line is added or deleted. Therefore, the number of the destination line you have in mind can change during editing.

The :n (Through Line Number) Command

The inverse of the line number command specifies that a command should be executed through a certain line number. You can use this command with only three ED commands: the K (kill) command, the L (line) command, and the T (type) command. The :n command takes the following form:

:ncommand

where n is the line number through which the command is to be executed. The :n part of the command does not move the CP, but the command that follows it might.

You can combine n: with :n to specify a range of lines through which a command should be executed. For example, the command 2::4T types the second, third, and fourth lines:

```
1 :  *2: :4T
2: "I find ecstasy in living -
3: the mere sense of living
4:    is joy enough."
2: *
```

6.6.2 Finding and Replacing Character Strings

ED supports a find command, F, that searches through the memory buffer and places the CP after the word or phrase you want. The N command allows ED to search through the entire source file instead of just the buffer. The J command searches for and then juxtaposes character strings.

The F (Find) Command

The F command performs the simplest find function; it takes the form:

nfstring

where n is the occurrence of the string to be found. Any number you enter must be positive because ED can only search from the CP to the bottom of the buffer. If you enter no number, ED finds the next occurrence of the string in the file. In the following example, the second occurrence of the word living is found.

```
1: *2fliving
3: *
```

The character pointer moves to the beginning of the third line where the second occurrence of the word "living" is located. To display the line, combine the find command with a type command. Note that if you follow an F command with another ED command on the same line, you must terminate the string with a CTRL-Z, as follows

```
1 : *2fliving^Z0lt
3: *the mere sense of living
```

It makes a difference whether you enter the F command in upper- or lower-case. If you enter F, ED internally translates the argument string to upper-case. If you specify fl ED looks for an exact match. For example, Fcp/m 3 searches for CP/M 3 but fcp/m 3 searches for cp/m 3, and cannot find CP/M 3.

If ED does not find a match for the string in the memory buffer, it issues the message,

```
BREAK "#" AT
```

where the symbol # indicates that the search failed during the execution of an F command.

The N Command

The N command extends the search function beyond the memory buffer to include the source file. If the search is successful, it leaves the CP pointing to the first character after the search string. The N command takes the form:

```
nNstring
```

where n is the occurrence of the string to be found. If no number is entered, ED looks for the next occurrence of the string in the file. The case of the N command has the same effect on an N command as it does on an F command. Note that if you follow an N command with another ED command, you must terminate the string with a CTRL-Z.

When an N command is executed, ED searches the memory buffer for the specified string, but if ED does not find the string, it does not issue an error message. Instead, ED automatically writes the searched data from the buffer into the new file. Then ED performs a OA command to fill the buffer with unsearched data from the source file. ED continues to search the buffer, write out data,, and append new data until it either finds the string or reaches the end of the source file. If ED reaches the end of the source file, ED issues the following message:

BREAK "*" AT

Because ED writes the searched data to the new file before looking for more data in the source file, ED usually writes the contents of the buffer to the new file before finding the end of the source file and issuing the error message.

No,e: you must use the H command to continue an edit session after the source file is exhausted and the memory buffer is emptied.

The j (Juxtapose) Command

The j command inserts a string after the search string, then deletes any characters between the end of the inserted string to the beginning of the a third delete-to string. This juxtaposes the string between the search and delete-to strings with the insert string. The J command takes the form:

nJsearch string^Zinsert string^Zdelete-to string

where n is the occurrence of the search string. If no number is specified, ED searches for the next occurrence of the search string in the memory buffer. In the following example, ED searches for the word "Dickinson", inserts the phrase "told a friend" after it, and then deletes everything up to the comma.

```
1:  *#T
1: Emily Dickinson said,
2: "I find ecstasy in living -
3:   the mere of living
4:   is joy enough."
1:  *jDickinson^Z told a friend^Z,
1: *01 t
1: Emily Dickinson told a friend#
1: *
```

If you combine this command with other commands, you must terminate the delete-to string with a CTRL-Z or ESC, as in the following example. If an upper-case J command letter is specified, ED looks for upper-case search and delete-to strings and inserts an upper-case insert string.

The J command is especially useful when revising comments in assembly language source code, as follows

```
236: SORT      LXI     H, SW      ;ADDRESS TOGGLE SWITCH
236: *j;^ZADDRESS SWlTCH TOGGLE^Z^'L^ZOLT
236: SORT      LXI     H, SW      ;ADDRESS SWITCH TOGGLE
236: *
```

In this example, ED searches for the first semicolon and inserts ADDRESS SWITCH TOGGLE after the mark and then deletes to the <cr><If> sequence, represented by CTRL-L. In any search string, you can use CTRL-L to represent a <cr><If> when the phrase that you want extends across a line break. You can also use a CTRL-1 in a search string to represent a tab.

Note: if long strings make your command longer than your screen line length, enter a CTRL-E to cause a physical carriage return at the screen. A CTRL-E returns the cursor to the left edge of the screen, but does not send the command line to ED. Remember that no ED command line containing strings can exceed 100 characters. When you finish your command, press the carriage return key to send the command to ED.

The M (Macro) Command

An ED macro command, M,, can increase the usefulness of a string of commands. The M command allows you to group ED commands together for repeated execution. The M command takes the following form:

nMcommand string

where n is the number of times the command string is to be executed. A negative number is not a valid argument for an M command. If no number is specified, the special character # is assumed, and ED executes the command string until it reaches the end of data in the buffer or the end of the source file, depending on the commands specified in the string. In the following example, ED executes the four commands repetitively until it reaches the end of the memory buffer:

```
 I : *mfliving^Z-6diLiving^Z0lt
2: "I find ecstasy in Living -
3: the mere sense of Living

BREAK "*" AT ^Z
```

The terminator for an M command is a carriage return; therefore, an M command must be the last command on the line. Also, all character strings that appear in a macro must be terminated by CTRL-Z or ESC. If a character string ends the corn-bined-command string, it must be terminated by CTRL-Z, then followed by a <cr> to end the M command.

The execution of a macro command always ends in a B R E A K "#" message, even when you have limited the number of times the macro is to be performed, and ED does not reach the end of the buffer or source file. Usually the command letter displayed in the message is one of the commands from the string and not M.

To abort a macro command, press a CTRL-C at the keyboard.

The Z (Sleep) Command

Use the Z command to make the editor pause between operations. The pauses give you a chance to review what you have done. The Z command takes the following form:

```
nZ
```

where n is the number of seconds to wait before proceeding to the next instruction.

Usually, the Z command has no real effect unless you use it with a macro command. The following example shows you how you can use the Z command to cause a brief pause each time ED finds the word TEXT in a file.

```
A >*mfliving^Z0tt10z
```

6.6.3 Moving Text Blocks

To move a group of lines from one area of your data to another, use an X command to write the text block into a temporary LIB file, then a K command to remove these lines from their original location, and finally an R command to read the block into its new location.

The X (Transfer) Command

The X command takes the forms:

```
nX
nXfilespec^Z
```

where n is the number of lines from the CP towards the bottom of the buffer that are to be transferred to a file. Therefore, n must always be a positive number. The nX command with no file specified creates a temporary file named X$$$$$$$.LIB. This file is erased when you terminate the edit session. The nX command with a file specified creates a file of the specified name. If no filetype is specified, LIB is assumed. This file is saved when you terminate the edit session. If the X command is not the last command on the line, the command must be terminated by a CTRL-Z or ESC. In the following example, just one line is transferred to the temporary file:

```
1    *X
1: *t
1:   *Emily Dickinson said,
1:  *kt
1:  "I find ecstacy in living -
 1: *
```

If no library file is specified, ED looks for a file named X$$$$$$$.LIB. If the file does not exist, ED creates it. If a previous X command already created the library file, ED appends the specified lines to the end of the existing file.

Use the special character 0 as the n argument in an X command to delete any file from within ED.

The R (Read) Command

The X command transfers the next n lines from the current line to a library file. The R command can retrieve the transferred lines. The R command takes the forms:

```
R
Rfilepsec
```

If no filename is specified, X$$$$$$$ is assumed. If no filetype is specified, LIB is assumed. R inserts the library file in front of the CP; therefore, after the file is added to the memory buffer, the CP points to the same character it did before the read, although the character is on a new line number. If you combine an R command with other commands, you must separate the filename from subsequent command letters with a CTRL-Z as in the following example where ED types the entire file to verify the read.

```
1: *41
 : *R^ZB#T
1: "I find ecstasy in living -
2: the mere sense of living
3:    is joy enough."
4:    Emily Dickinson said,
 1: *
```

6.6.4 Saving or Abandoning Changes: ED Exit

You can save or abandon editing changes with the following three commands.

The H (Head of File) Command

An H command saves the contents of the memory buffer without ending the ED session, but it returns to the head of the file. It saves the current changes and lets you reedit the file without exiting ED. The H command takes the following form:

```
H
```

followed by a carriage return.

To execute an H command, ED first finalizes the new file, transferring all lines remaining 'in the buffer and the source file to the new file. Then ED closes the new file, erases any BAK file that has the same file specification as the original source file, and renames the original source file filename.BAK. ED then renames the new file, which has had the filetype $$$, with the original file specification. Finally, ED opens the newly renamed file as the new source file for a new edit, and opens a new $$$ file. When ED returns the * prompt, the CP is at the beginning of an empty memory buffer.

If you want to send the edited material to a file other than the original file, use the following command:

```
A>ED filespec differentfilespeo
```

If you then restart the edit with the H command, ED renames the file differentfile-name.$$$ to differentfilename.BAK and creates a new file of differentfilespec when you finish editing.

The 0 (Original) Command

An 0 command abandons changes made since the beginning of the edit and allows you to return to the original source file and begin reediting without ending the ED session. The 0 command takes the form:

```
0
```

followed by a carriage return. When you enter an 0 command, ED confirms that you want to abandon your changes by asking

```
0 (Y/N)?
```

You must respond with either a Y or an N; if you press any other key, ED repeats the question. When you enter Y, ED erases the temporary file and the contents of the memory buffer. When the * prompt returns, the character pointer is pointing to the beginning of an empty memory buffer, just as it is when you start ED.

The Q (Quit) Command

A Q command abandons changes made since the beginning of the ED session and exits ED. The Q command takes the form:

Q

followed by a carriage return.

When you enter a Q command, ED verifies that you want to abandon the changes by asking

Q (Y/N)?

You must respond with either a Y or an N; if you press any other key, ED repeats the question. When you enter Y, ED erases the temporary file, closes the source file, and returns control to CP/M 3.

Note: you can enter a CTRL-Break or a CTRL-C to return control immediately to CP/M 3. This does not give ED a chance to close the source or new files, but it prevents ED from deleting any temporary files.

6.7 ED Error Messages

ED returns one of two types of error messages: an ED error message if ED cannot execute an edit command, or a CP/M 3 error message if ED cannot read or write to the specified file. An ED error message takes the form:

BREAK "x" AT c

where x is one ,f the symbols defined in the following table and c is the command letter where the error occurred.

Table 6-4. ED Error Symbols

Symbol	Meaning
#	Search failure. ED cannot find the string specified in a F, S, or N command.
?c	Unrecognized command letter c. ED does not recognize the indicated command letter, or an E, H@ 0, or Q command is not alone on its command line.
0	No. LIB file. ED did not find the LIB file specified in an R command.
>	Buffer full. ED cannot put anymore characters in the memory buffer, or string specified in an F, N, or S command is too long.
E	Command aborted. A keystroke at the keyboard aborted command execution.
F	File error. Followed by either disk FULL or DIRECTORY FULL.

The following examples show how to recover from common editing error conditions. For example

BREAK ">" AT A

means that ED filled the memory buffer before completing the execution of an A command. When this occurs, the character pointer is at the end of the buffer and no editing is possible. Use the OW command to write out half the buffer or use an 0 or H command and reedit the file.

BREAK "*" AT F

means that ED reached the end of the memory buffer without matching the string in an F command. At this point, the character pointer is at the end of the buffer. Move the CP with a B or n: line number command to resume editing.

BREAK "F" AT F
DISK FULL

Use the 0X command to erase an unnecessary file on the disk or a B#Xd:buffer.sav command to write the contents of the memory buffer onto another disk.

BREAK "F" AT n
DIRECTORY FULL

Use the same commands described in the previous message to recover from this file error.

The following table defines the disk file error messages ED returns when it cannot read or write a file.

Table 6-5. ED Diskette File Error Messages

Message	Meaning
CP/M Error on d: BDOS Function = NN	Read/OnlyFi le File =FILENAME,TYP Disk d: has Read-Only attribute. This occurs if a different disk has been inserted in the drive since the last cold or warm boot.
** FILE IS READ ONLY **	The file specified in the command to invoke ED has the R/O attribute. ED can read the file so that the user can examine it, but ED cannot change a Read-Only file.

End of Section 6

Appendix A

CP/M 3 Messages

Messages come from several different sources. CP/M 3 can display error messages when the Basic Disk Operating System (BDOS) returns an error code. CP/M 3 can also display messages when there are errors in command lines. Each utility supplied with CP/M 3 has its own set of messages. The following table lists CP/M 3 messages and utility messages. If you are running an application program, you might see messages other than those listed here. Check the application program's documentation for explanations of those messages.

The messages in Table A-1 might be preceded by ERROR:. Some of them might also be preceded or followed by the filespec of the file causing the error condition. Sometimes the input line is flagged with an up arrow T , to indicate the character that caused the error. In this case,, the message, E r r o r a t t h e ', precedes the appropriate error message. Some of the messages are followed by an additional line preceded by INPUT:, OPTION:, or DRIVE: followed by the applicable error message.

Table A-1. CP/M 3 Messages

Message — Meaning

Assign a Password to this file.

SET. A password mode has been selected for this file but no password has been assigned.

Auxiliary device redirection not implemented

GET and PUT. AUXIN and AUXOUT cannot be redirected to a file.

Bad character, re-enter

GENCPM. The character entered was not a number.

Table A-1. (continued)

Message	Meaning
Bad close.	SAVE. An error occurred during the attempt to close the file, probably because the file is write-protected.
Bad Logical Device Assiginment ;	DEVICE. Only the following logical devices are valid: CONIN:, CONOUT:, AUXIN:, AUXOUT:, LST:.
BAD PARAMETER	PIP. You entered an illegal parameter in a PIP command. Retype the entry correctly.
Bad Password , I	RENAME. The password supplied by the user is incorrect.
Bank one not allowed.	GENCPM. Bank 1 cannot be defined as available during system generation.
Baud rate cannot be set for this device	DEVICE. Only physical devices that have the SOFT-BAUD attribute can have their baud rates changed. To check the attributes of the physical device, type DEVICE physical-dev.

Table A-1. (continued)

Message	Meaning
Break "x" at c	ED. "x" is one of the symbols described below and c is the command letter being executed when the error occurred. # Search failure. ED cannot find the string specified in an F, N, or S command. ? Unrecognized command letter c. ED does not recogn ize the indicated command letter, or an E, Hi 0, or Q command is not alone on its command line. 0 The file specified in an R command cannot be found. > Buffer full. ED cannot put any more characters in the memory buffer, or the string specified in an F, N, or S command is too long. E Command aborted. A keystroke at the console aborted command execution. F Disk or directory full. This error is followed by either the disk or directory full message. Refer to the recovery procedures listed under these messages.
CANNOT CLOSE: Cannot close f i le CANNOT CLOSE FILE. CANNOT CLOSE DESTINATION FILE - filespec	GENCOM, HEXCOM, LIB-80tm. LINK-80, MAC, PIP, RMAC, SUBMIT. An output file cannot be closed. This can occur if the disk is removed before the program terminates.
Cannot delete file ,	GENCOM. CP/M cannot delete a file. Check to see if the COM file is Read-Only or password-protected.

Table A-1. (continued)

Message	Meaning
Cannot have both create and access time stamps	
	SET. CP/M 3 supports either create or access time stamps, but not both.
Cannot label a drive with a file referenced	
	SET. SET does not allow mixing of files and drives.
CANNOT OPEN SOURCE FILE	
	HEXCOM. The HEX file is not on the specified drive(s).
Cannot redirect from BIOS.	
	GET, PUT. This message is displayed as a warning only if the system has an invalid BIOS.
Cannot set both RO and RW.	
	SET. A file cannot be set to both Read-Only and Read-Write.
Cannot set both SYS and DIR.	
	SET. A file cannot be set to both SYS and DIR.
CAN'T DELETE TEMP FILE	
	PIP. A temporary $$$ file already exists which is Read-Only. Use the SET command to change the attribute to Read-Write, then erase it.

Table A-1. (continued)

Meaning

CHEKCSUM ERROR,
Checksum error

HEXCOM, PIP. A hex record checksum error was enco
tered. The hex record that produced the error must be c
rected, probably by recreating the hex file.

Close error.

XREF. This message is preceded by the filename.XRF. The disk might have been removed before the program terminated.

Close operation failed.

COPYSYS. There was a problem in closing the file at the end of the file copy operation.

Closing file HELP.DAT
Closinsg file HELP.HLP

HELP. HELP encountered error while processing the HELP.DAT or the HELP.HLP file.

COM file found and NULL option

GENCOM. The NULL option implies that no COM file is to be loaded, just the RSXS.

.COM file recquired

DIR, ERASE, RENAME, TYPE. Options in the built-in command line require support from a transient COM file that CP/M 3 cannot find on disk.

COMMON ERROR:

LINK-80 An undefined common block has been selected.

Table A-1. (continued)

Message	Meaning
CORRECT ERROR , TYPE RETURN OR CTRL-Z	PIP. A hex record checksum was encountered during the transfer of a hex file. The hex file with the checksum error should be corrected, probably by recreating the hex file.
CPMLDR error: failed to open CPM3. SYS	CPMLDR. The system file CPM3.SYS is missing.
CPMLDR error: failed to read CPM3.SYS	CPMLDR. An error occurred while reading CPM3.SYS.
CP/M Error on d: Disk I/O BDOS Function = xx File = filespec	CP/M displays the preceding message if the disk is defective or improperly formatted (wrong density).
CP/M Error on d: Invalid Drive BDOS Function = xx File = filespec	CP/M 3 displays the preceding message when there is no disk in the drive, the drive latch is open, or the power is off. It also displays the message when the specified drive is not in the system.
CP/M Error on d:Read/Only Disk BDOS Function = xx File = filespec	CP/M 3 does not allow you to erase, rename, update, or set attributes of a file residing in a Read-Only drive. Use the SET command to set the drive attribute to Read-Write.

Table A-1. (continued)

Message	Meaning
CP/M Error on d: Read/Only File BDOS Function = xx File = filespec	
	CP/M 3 does not allow you to erase, rename, update, or set attributes of a file that is Read-Only. Use the SET command to set the file attribute to Read-Write.
Date and Time Stamping Inactive,,	
	DIR. The DATE option was specified, but the disk directory has not been initialized with date/time stamping.
DESTINATION IS R/O , DELETE (Y/N)?	
	PIP. The destination file specified in a PIP command already exists and it is Read-Only. If you type Y, the destination file is deleted before the file copy is done. If you type N, PIP displays the message **NOT DELETED** and aborts the copy operation.
Device Reassignment Not Supporte d Enter new assignment or hit RETURN.	
	DEVICE. A device assignment is invalid.
Directory already re-format t e d .	
	INITDIR. The directory already has date/time stamping.

Table A-1. (continued)

Message	Meaning
Directory full DIRECTORY FULL	ED. There is not enough directory space for the file being written to the destination disk. You can use the OXfilespec command to erase any unnecessary files on the disk without leaving the editor. SUBMIT. There is not enough directory space on the temporary file drive to write the temporary file used for processing SUBMIT files. Use the SETDEF command to determine which drive is the temporary file drive. Use the ERASE command to erase unnecessary files or set the temporary file drive to a different drive and retry. LIB-80, LINK-80. There is no directory space for the output or intermediate files. Use the ERASE command to remove unnecessary files. GENCPM. There is no directory space for CPM3.SYS. HEXCOM. There is no directory space for the output COM file.
Directory needs to be reformatted for date/time stamps.	SET. A date/time option was specified, but the directory has not been initialized for date/time stamping. Use the INITDIR command to initialize the directory for date/time stamping.
DISK FULL	ED. There is not enough disk space for the output file. This error can occur on the E, H, W, or X commands. If it occurs with X command, you can repeat the command prefixing the filename with a different drive.

Table A-1. (continued)

Message Meaning

DISK READ
DISK REID ERROR:
Disk read error: filespec
D I SK READ ERROR - filespec

GENCPM, HEXCOM, LIB-80, LINK-80, PIP. The disk file specified cannot be read.

DISK WRITE.
Disk Write Error
DISK WRITE ERROR:
DISK WRITE ERROR - filespec

HEXCOM,- LIB-80, LINK-80, PIP, SUBMIT. A disk write operation cannot be successfully performed probably because the disk is full. Use the ERASE command to remove unnecessary files.

Do You want another file? (Y/N)

PUT. Enter Y to redirect output to an additional file. Otherwise, enter N.

Drive defined twice in search Path

SETDEF. A drive can be specified only once in the search path order.

Drive Read Only

ERASE, RENAME. The specified file is on a Read-Only drive and cannot be erased or renamed.

Drives specified has not been defined,

GENCPM. The drive specified has not been defined yet. Buffer(s) have not been allocated for the drive.

Table A-1. (continued)

Message	Meaning
Duplicate RSX in header. Replacing old by new This file was not used	GENCOM. The specified RSX is already attached to the COM file. The old one is discarded.
Duplicate input RSX .	GENCOM. Two or more RSXs of the same name are specified. GENCOM uses only one of the RSXS.
Equals (=) delimiter missing at line NN.	GENCPM. The equal sign is missing in the specified line.
END OF F ILE , ^Z , ?	PIP encountered an unexpected end-of-file during a HEX file transfer.
End of line expected	DEVICE, GET, PUT, SETDEF. The command typed does not have any further parameters. An end-of-linc was expected. Any further characters on the line were ignored.
Error at end of line :	DEVICE, GET, PUT, SETDEF. The error detected occurred at the end of the 'input line.
Error on line nnnnn :	SUBMIT. The SUBMIT program displays its messages in the preceding format, where nnnnn represents the line number of the SUBMIT file. Refer to the message following the line number for explanation of the error.

Table A-1. (continued)

Message	Meaning
FILE ERROR	ED. Disk or directory is full, and ED cannot write anything more on the disk. This is a fatal error, so make sure there is enough space on the disk to hold a second copy of the file before invoking ED.
File already exists Delete it? (Y/N) file already exists delete (Y/N)?	PUT. Enter Y to delete the file. Otherwise the program terminates. RENAME. The above message is preceded by filespec. You have asked CP/M 3 to create or rename a file using a file specificat'on that is already assigned to another file. Either delete the existing file or use another file specification.
File cannot fit into GENCPM buffer: filename. SPR	GENCPM. There is not enough memory to generate a system.
File exists , erase it	ED. The destination filename already exists when you are placing the destination file on a different disk than the source. It should be erased or another disk selected to receive the output file.
FILE IS READ/ONLY File is Read Only	ED. The file specified in the command to invoke ED has the Read-Only attribute. ED can read the file so that you can examine it, but ED cannot change a Read-Only file. PUT. The file specified to receive the output is a Read-Only file.

Table A-1. (continued)

Message	Meaning
F ILE NAME ERROR:	LIB-80. The form of a source filename is invalid.
File not found . FILE NOT FOUND - filespec	DUMP, ED, GENCOM, GET, PIP, SET. An input file that you have specified does not exist. Check that you have entered the correct drive specification or that you have the correct disk in the drive.
First submitted file must be a COM file	GENCOM. A COM file is expected as the first file in the command tail. The only time GENCOM does not expect to see a COM file in the first position of the command tall is when the NULL option is specified.
FIRST COMMON NOT LARGEST:	LINK-80. A subsequent COMMON declaration is larger than the first COMMON declaration for the indicated block. Check that the files being linked are in the proper order, or that the modules in a library are in the proper order.
HELP. DAT not on current d rive .	HELP. HELP cannot find HELP.DAT file to process.
Illegal command tail.	DIR. The command line has an invalid format or option.

Table A-1. (continued)

Message	Meaning
Illegal Format Value.	DIR. Only SIZE and FULL options can be used for display formats.
Illegal Global/Local Drive Spec Mixing.	DIR. Both a filespec with a drive specifier and the DRIVE option appears in the command.
Illegal filename.	SAVE. There is an error in the filespec on the command line.
Illegal Option or Modifier.	DIR. An invalid option or abbreviation was used.
Illegal date/time specification.	DATE. Date/time format is invalid.
Incorrect file sepcification.	RENAME. The format of the filespec is invalid.
INDEX ERROR:	LINK-80. The index of an IRL contains invalid information.
Insufficient Memory INSUFFICIENT MEMORY:	GET, LINK-80, PUT, SUBMIT. There is not enough memory to allocate buffers, or there are too many levels of SUBMIT nesting.

Table A-1. (continued)

Message	Meaning
Invalid ASCII character	SUBMIT. The SUBMIT file contains an invalid character (OFFH).
Invalid character at line NN.	GENCPM. The character must be a number.
Invalid command.	GET and PUT. The string or substring typed in the command line was not recognized as a valid command in the context used.
Invalid delimiter.	DEVICE, GET, PUT, SETDEF. The delimiter, or space, - was not valid at the location used. For example, a [was used where an = should have been used.
INVALID DESTINATION:	PIP. An invalid drive or device was specified.
INVALID DIGIT - filespec	PIP. An invalid hex digit has been encountered while reading a hex file. The hex file with the invalid hex digit should be corrected, probably by recreating the hex file.
Invalid drive.	SETDEF. The specified drive was not a valid drive. Drives recognized by SETDEF are * (default drive) and A to P. GENCPM, TYPE. Valid drives are A to P.

Table A-1. (continued)

Message	Meaning
Invalid drive ignored at line NN.	GENCPM. Valid drives are A to P.
Invalid drive name (Use A, B, C, D)	COPYSYS, GENCPM. Only drives A, B, C and D are valid destination drives for system generation.
Invalid File, INVALID FILENAME Invalid file name. Invalid Filename. Invalid file specification.	ED, ERASE, GENCOM, GET, PIP, PUT, SET, SUBMIT, TYPE. The filename typed does not conform to the normal CP/M 3 file naming conventions.
INVALID FORMAT	PIP. The format of your PIP command is illegal. See the description of the PIP command.
INVALID HEX DIGIT.	HEXCOM. An invalid hex digit has been encountered while reading a hex file. The hex file with the invalid hex digit should be corrected by recreating the hex file.
Invalid number .	DEVICE. A number was expected but not found, or the number was out of range; numbers must be from 0 to 255.

Table A-1. (continued)

Message	Meaning
Invalid option.	DEVICE and GET. An option was expected and the string found was not a device option or was not valid in the context used. SETDEF. The option typed in the command line is not a valid option. Valid options are DISPLAY, NO DISPLAY, NO PAGE, ORDER, PAGE, TEMPORARY.
Invalid option modifier.	DIR, GET, PUT. The option typed is not a valid option.
INVALID PARAMETER:	MAC, RMAC. An invalid assembly parameter was found in the input line. The assembly parameters are printed at the console up to the point of the error.
Invalid paprameter or modifier.	GENCPM. The parameter variable does not exist. Check spelling.
INVALID PASSWORD Invalid Password or Passwords not allowed.	ED, PIP. The specified password is incorrect, or a password was specified, but the file is not password-protected.
Invalid Physical device.	DEVICE. A physical device name was expected. The name founc in the command string does not correspond to any physical device name in the system.

Table A-1. (continued)

Message	Meaning
INVALID REL FILE:	LINK-80. The file indicated contains an invalid bit pattern. Make sure that a REL or IRL file has been specified.
Invalid RSX type,	GENCOM. Filetype must be RSX.
Invalid SCB offset ,	GENCOM. The specified SCB is out of range. The SCB offset range is OOH-64H.
INVALID SEPARATOR	PIP. You have placed an invalid character for a separator between two input filenames.
INVALID SOURCE	PIP. An invalid drive or device was specified. AUX and CON are the only valid devices.
Invalid type for ORDER option.	SETDEF. The type specified in the command line was not COM or SUB.
Invalid SYM file format	XREF. The filename.SYM file input to XREF is invalid.
I NVALID USER NUMBER	PIP. You have specified a user number greater than 15. User numbers are in the range 0 to 15.

Table A-1. (continued)

Message	Meaning
Invalid wildcard.	RENAME. The filespec contained an invalid wildcard specification.
Invalid wild card in the FCB name or type field	GENCOM. GENCOM does not allow wildcards in filespecs.
LOAD ADDRESS LESS THAN 100.	HEXCOM. The program origin is less than 100H.
MAIN MODULE ERROR:	LINK-80. A second main module was encountered.
Make error	XREF. There is not more directory space on the specified drive.
Memory conflict - cannot trim segment.	GENCPM. The defined memory segment overlaps another segment.
Memory conflict - segment trimmed	GENCPM. The defined memory segment overlaps with other segments.
MEMORY OVERFLOW:	LINK- 80. There is not enough memory to complete the link operation.

Table A-1. (continued)

Message	Meaning
Minimum number of buffers is 1.	GENCPM. The first drive must have at least one buffer defined.
Missing delimeter or Unrecognised Option..	ERASE. The ERASE command line format is invalid.
Missing parameter variable at line NN.	GENCPM. The line is missing a variable name.
Missing left parenthesis.	GENCOM. The SCB option must be enclosed by a left parenthesis.
Missing right Parenthesis	GENCOM. The SCB option is not enclosed with a right parenthesis.
Missing SCB value.	GENCOM. The SCB option requires a value.
More than four rives specified.	SETDEF. More than four drives were specified for the drive search chain.
MULTIPLE DEFINITION:	LINK-80. The specified symbol is defined in more than one of the modules being linked.

Table A-1. (continued)

Message	Meaning
n?	USER. You specified a number greater than fifteen for a user area number. For example, if you type USER 18, the screen displays 18?.
No directory label exists.	SHOW. The LABEL option was requested but the disk has no label.
No directory space NO DIRECTORY SPACE -filespec	COPYSYS, GENCOM, MAC, PIP, RMAC, AND SAVE. There is not enough directory space for the output file. Use the ERASE command to remove unnecessary files on the disk and try again.
No disk space .	SAVE. There is not enough space on the disk for the output file. Use the SHOW command to display the amount of disk space left and use the ERASE command to remove unnecessary files from the disk, or use another disk with more file space.
No file NO FILE: NO FILE - filespec	DIR, ERASE, LIB-80, LINK-80, PATCH, PIP, RENAME, TYPE. The specified file cannot be found in the specified drive(s).
No HELP.HLP file on the default drive .	HELP. The file HELP.HLP must be on the default drive.

Table A-1. (continued)

Message	Meaning
NO INPUT FILE PRESENT ON DISK	
	DUMP. The file you requested does not exist.
No memory	
	There is not enough memory available for loading the specified program.
No modifier for this option	
	GENCOM. A modifier was specified but none is required.
NO MODULE:	
	LIB-80. The indicated module cannot be found.
No more space in the header for RSXs or SCB initialization.	
	GENCOM. The header has room for only 15 entries, or the combination of RSXs and SCBs exceed the maximum.
No options specified ,	
	SET. Specify an option.
No PRN file ,	
	XREF. The file filename.PRN is not present on the specified drive.
No Records Exist	
	DUMP. Only a directory entry exists for the file.

A CP/M 3 Messages CP/M 3 User's Guide

Table A-1. (continued)

Message	Meaning
No source file on disk.	COPYSYS. The file CPM3.SYS is not on the disk specified.
NO SOURCE FILE PRESENT:	MAC, RMAC. The source file cannot be found on the specified drive.
NO SPACE	SAVE. There is no space in the directory for the file being written.
No 'SUB ' f i 1 e f ound	SUBMIT. The SUB file typed in the command line cannot be found in the drive search process.
No such file to rename .	RENAME. The file to be renamed does not exist on the specified drive(s).
No SYM file	XREF. The file filename SYM is not present on the specified drive.
NON-SYSTEM FILE(S) EXIST	DIRS. If nonsystem (DIR) files reside on the specified drive, DIRS displays this message.

CPIM 3 User's Guilc A CP/M 3 Messages

Table A-1. (continued)

Message	Meaning
Not enough available memory. Not Enough Memory. Not Enough Memory for Sort.	DIR,, INITDIR. There is not enough memory for data or sort buffers.
Not enough room in directory.	INITDIR. There is not enough remaining directory space to allow for the date and time extension.
NOT FOUND	PIP. PIP cannot find the specified file.
Not renamed +, filespec read only .	RENAME. The specified file cannot be renamed because it is Read-Only.
OPEN FILE NONRECOVERABLE	PIP. A disk has the wrong format or a bad sector.
Option only for drives.	SET. The specified option is not valid for files.
Option requires a file reference.	SET. The specified option requires a filespec.
Out of data space.	COPYSYS. The destination drive ran out of space during the transfer of the CPM3.SYS file.

Table A-1. (continued)

Message	Meaning
OPtions not grouped together.	DIR. Options can only be specified within one set of brackets.
Output file exists, Erase it.	The output file specified must not already exist.
OUTPUT FILE READ ERROR:	MAC, RMAC. An output file cannot be written properly, probably because the disk is full. Use the ERASE command to delete unnecessary files from the disk.
OVERLAPPING SEGMENTS:	LINK-80. LINK-80 attempted to write a segment into memory already used by another segment.
Page and nopage option selected. No Page in effect.	SET. The preceding options are mutually exclusive.
Parameter Error.	SUBMIT. Within the SUBMIT file of type SUB, valid parameters are $0 through $9.
Password Error	DUMP, ERASE, GENCOM, TYPE. The password is incorrect.
Physical Device Does Not Exist.	DEVICE. The specified physical device is not defined in the system.

Table A-1. (continued)

Message	Meaning
Possible incompartible disk format.	COPYSYS. The system disk and the output disk have different formats.
PROGRAM INPUT IGNORED.	SUBMIT. This message is preceded by "WARNING". The SUBMIT file contains a line with <, and the program does not require additional input.
PUT >	PLTF. This prompt occurs when a program requests input while running a PUT FILE [NO ECHO] command.
PUT ERROR: FILE ERASED.	PUT. The PUT output file was erased and could not be closed.
QUIT NOT FOUND	PIP. The string argument to a Q parameter was not found in your input file.
Random Read	SUBMIT. An error occurred when reading the temporary file used by the SUBMIT command.
Read only .	GENCOM, SET. The drive or file specified is write-protected.
Read error	TYPE. An error occurred when reading the file specified in the TYPE command. Check the disk and try again.

Table A-1. (continued)

Message	Meaning
Reading file: filespec	GENCPM. An error occurred while attempting to read the file specified by filespec.
Reading file HELP.HLP Reading HELP.HLP index	HELP. An error occurred while reading HELP.HLP. Copy the HELP.HLP file from the system disk.
RECORD TOO LONG	PIP. A HEX record exceeds 80 characters in a file being copied with the [HI option.
Requires CP/M 3.0 or higher. I	DATE, DEVICE, DIR, ERASE, GENCOM, HELP, INITDIR, PIP, SET, SETDEF, SHOW, RENAME, TYPE. This version of the utility must only be run under CP/M 3.0 or higher.
R/O DISK	PIP. The destination drive is set to Read-Only and PIP cannot write to it.
R/O FILE	PIP. The destination file is set to Read-Only and PIP cannot write to it.
Sort Stack Overflow	DIR. There is not enough memory available for the sort stack.

Table A-1. (continued)

Message	Meaning
Source file is incomplete	GENCPM. GENCPM cannot use your CP/M 3 system source file.
SOURCE FILE READ ERROR:	MAC 7 RMAC. The source file cannot be read properly by MAC.
SOURCE FILENAME ERROR:	MAC , RMAC. The form of the source filename is invalid.
START NOT FOUND	PIP. The string argument to an S parameter cannot be found in the source file.
Symbol Table overflow	XREF. No space is available for an attempted symbol allocation.
Symbol Table reference overflow	XREF. No space is available for an attempted symbol reference allocation.
SYNTAX ERROR:	LIB. The LIB-80 command is not properly formed.
Too many entries in Index Table. Not enough memory	HELP. There is not enough memory available to hold the topic table while creating HELP.HLP.

Table A-1. (continued)

Message Meaning

Topic:
xxxxxx
Not found

HELP. The topic requested does not exist in the HELP.HLP file. HELP displays the topics available.

Total file size exceeds 64K .

GENCOM. The output file exceeds the maximum allowed.

Try ' PAGE ' or 'NO PAGE '

TYPE. The only valid option is PAGE or NO PAGE.

Unable to allocate DATA deblocking buffer space.

GENCPM. There is not enough space left in generated system to allocate a data deblocking buffer.

Unable to allocate DIR deblocking buffer space.

GENCPM. There is not enough space left in generated system to allocate a directory deblocking buffer.

Unable to allocate space for hash table.

GENCPM. There is not enough contiguous memory to allocate space for the hash table in the generated system.

Unable to close HELP.DAT.
Unable to close HELP.HLP.

HELP. An error occurred while closing file HELP.HLP or HELP.DAT. There might not be enough disk or directory space on the drive.

CPIM I Us,,r's Guide A CP/M 3 Messages

Table A-1. (continued)

Message	Meaning
Unable to find file HELP.HLP	
	HELP. HELP requires HELP.HLP file to operate. Copy it to your default drive from your CP/M 3 system disk.
Unable to Make HELP.DAT. Unable to Make HELP.HLP.	
	HELP. There is not enough space on the disk for HELP.HLP or HELP.DAT, or the files are Read-Only.
Unable to open : filename.SPR	
	GENCPM. The file specified cannot be found on the default drive.
UNBALANCED MACRO LIBRARY.	
	MAC, RMAC. A MACRO definition was started within a macro library, but the end of the file was found in the library before the balancing ENDM was encountered.
UNDEFINED START SYMBOL:	
	LINK-80. The symbol specified with the G switch is not defined in any of the modules being linked.
UNDEFINED SYMBOLS:	
	LINK-80. The symbols following this message are referenced but not defined in any of the modules being linked.
UNEXPECTED END OF HEX FILE - f ilespee	
	PIP. An end-of-file was encountered before a termination hex record. The hex file without a termination record should be corrected, probably by recreating the hex file.

Table A-1. (continued)

Message	Meaning
Unrecognized drive.	SHOW. The specified drive is not valid, Valid drives are A to P.
UNRECOGNIZED ITEM:	LINK-80. An unfamiliar bit pattern has been scanned and ignored by LINK-80.
Unrecognized input ,	SHOW. The SHOW command line has an invalid format.
Unrecognized option,	GENCOM and SHOW. An option typed in the command line is not valid for the command.
USER ABORTED	PIP. You stopped a PIP operation by pressing CTRL-C.
VERIFY ERROR - filespec	PIP. When copying with the V option, PIP found a difference when rereading the data just written and comparing it to the data in its memory buffer.

Table A-1. (continued)

Message	Meaning
Write error	XREF. This message is preceded by filename.XRF and indicates that no disk space is available, or no directory space exists on the specified drive.
Write Protected?	COPYSYS. The drive or disk to which the system is to be written is Read-Only.
Writing file : filespec	GENCPM, HELP. An error occurred while attempting to write the file specified by filespec.
Wrong Password .	SET. The specified password is incorrect or invalid.
Zero length segment not allowed.	GENCPM. A memory segment cannot have zero length.
OFFFFH is an invalid value in the DPH directory BCB address field .	GENCPM. This value is allowed only in the DTABCB field.
?	SID. SID has encountered an error.

End of Appendix A

Appendix B

ASCII and Hexadecimal Conversions

ASCII stands for American Standard Code for Information Interchange. The code contains 96 printing and 32 non-printing characters used to store data on a disk. Table B-1 defines ASCII symbols, then Table B-2 lists the ASCII and hexadecimal conversions. The table includes binary, decimal, hexadecimal, and ASCII conversions.

Table B-1. ASCII Symbols

Symbol	Meaning	Symbol	Meaning
ACK	acknowledge	FS	file separator
BEL	bell	GS	group separator
BS	backspace	HT	horizontal tabulation
CAN	cancel	LF	line-feed
CR	carriage return	NAK	negative acknowledge
DC	device control	NUL	null
DEL	delete	RS	record separator
DLE	data link escape	SI	shift in
EM	end of medium	SO	shift out
ENQ	enquiry	SOH	start of heading
EOT	end of transmission	SP	space
ESC	escape	STX	start of text
ETB	end of transmission	SUB	substitute
ETX	end of text	SYN	synchronous idle
FF	form-feed	US	unit separator
		VT	vertical tabulation

Table B-2. ASCII Conversion Table

Binary	Decimal	Hexadecimal	ASCII
0000000	0	0	NUL
0000001	1	1	SOH (CTRL-A)
0000010	2	2	STX (CTRL-B)
0000011	3	3	ETX (CTRL-C)
0000100	4	4	EOT (CTRL-D)
0000101	5	5	ENQ (CTRL-E)
0000110	6	6	ACK (CTRL-F)
0000111	7	7	BEL (CTRL-G)
0001000	8	8	BS (CTRL-H)
0001001	9	9	HT (CTRL-1)
0001010	10	A	LF (CTRL-J)
0001011	11	B	VT (CTRL-K)
0001100	12	c	FF (CTRL-L)
0001101	13	D	CR (CTRL-M)
0001110	14	E	SO (CTRL-N)
0001111	15	F	SI (CTRL-0)
0010000	16	10	DLE (CTRL-P)
0010001	17	11	DC1 (CTRL-Q)
0010010	18	12	DC2 (CTRL-R)
0010011	19	13	DC3 (CTRL-S)
0010100	20	14	DC4 (CTRL-T)
0010101	21	15	NAK (CTRL-U)
0010110	22	16	SYN (CTRL-V)
0010111	23	17	ETB (CTRL-W)
0011000	24	18	CAN (CTRL-X)
0011001	25	19	EM (CTRL-Y)
0011010	26	1A	SUB (CTRL-Z)
0011011	27	1B	ESC (CTRL- [)
0011100	28	1C	FS (CTRL-\)
0011101	29	1D	GS (CTRL-])
0011110	30	1E	RS (CTRL--)
0011111	31	1F	US (CTRL-_)
0100000	32	20	(SPACE)
0100001	33	21	!
0100010	34	22	"
0100011	35	23	#
0100100	36	24	$
0100101	37	25	%

Table B-2. (continued)

Binary	Decimal	Hexadecimal	ASCII
0100110	38	26	&
0100111	39	27	'
0101000	40	28	(
0101001	41	29	)
0101010	42	2A	*
0101011	43	2B	+
0101100	44	2C	,
0101101	45	2D	-
0101110	46	2E	.
0101111	47	2F	/
0110000	48	30	0
0110001	49	31	1
0110010	50	32	2
0110011	51	33	3
0110100	52	34	4
0110101	53	35	5
0110110	54	36	6
0110111	55	37	7
0111000	56	38	8
0111001	57	39	9
0111010	58	3A	:
0111011	59	3B	;
0111100	60	3C	<
0111101	61	3D	=
0111110	62	3E	>
0111111	63	3F	?
1000000	64	40	@
1000001	65	41	A
1000010	66	42	B
1000011	67	43	c
1000100	68	44	D
1000101	69	45	E
1000110	70	46	F
1000111	71	47	G
1001000	72	48	H
1001001	73	49	I
1001010	74	4A	J
1001011	75	4B	K

Table B-2. (continued)

Binary	Decimal	Hexadecimal	ASCII
1001100	76	4C	L
1001101	77	4D	M
1001110	78	4E	N
1001111	79	4F	O
1010000	80	50	P
1010001	81	51	Q
1010010	82	52	R
1010011	83	53	S
1010100	84	54	T
1010101	85	55	U
1010110	86	56	V
1010111	87	57	W
1011000	88	58	X
1011001	89	59	Y
1011010	90	5A	Z
1011011	91	5B	[
1011100	92	5C	\
1011101	93	5D	]
1011110	94	5E	^
1011111	95	5F	<
1100000	96	60	'
1100001	97	61	a
1100010	98	62	b
1100011	99	63	c
1100100	100	64	d
1100101	101	65	e
1100110	102	66	f
1100111	103	67	g
1101000	104	68	h
1101001	105	69	i
1101010	106	6A	j
1101011	107	6B	k
1101100	108	6C	l
1101101	109	6D	m
1101110	110	6E	n
1101111	1ll	6F	o
1110000	112	70	p
1110001	113	71	q

Table B-2. (continued)

Binary	Decimal	Hexadecimal	ASCII
1110010	114	72	r
1110011	115	73	s
1110100	116	74	t
1110101	117	75	u
1110110	118	76	v
1110111	119	77	w
1111000	120	78	x
1111001	121	79	y
1111010	122	7A	z
1111011	123	7B	{
1111100	124	7C	|
1111101	125	7D	}
1111110	126	7E	-
1111111	127	7F	DEL

End of Appendix B

Appendix C

Filetypes

CP/M 3 identifies every file by a unique file specification, which consists of a drive specification, a filename, a filetype, and an optional password. The filetype is an optional three-character ending separated from the filename by a period. The filetype generally indicates a special kind of file. The following table lists common filetypes and their meanings.

Table C-1. Common Filetypes

Type	Meaning
ASM	Assembly language source file; the CP/M 3 assemblers assemble or translate a type ASM file into machine language.
BAK	Back-up file created by text editor; the editor renames the source file with this filetype to indicate that the original file has been processed. The original file stays on disk as the back-up file, so you can refer to it.
BAS	CBASIC program source file.
COM	8080 executable file.
ERL	Pascal/MT + " relocatable file.
HEX	Program file in hexadecimal format.
INT	CBASIC program intermediate language file.
IRL	Indexed REL file produced by LIB.
LIB	Used by MAC and RMAC for macro libraries. The ED R command reads files of type LIB. The ED X command writes files of type LIB. Printable file displayable on console or printer.

Table C-1. (continued)

	Meaning
OVL	Program overlay file. PL/1-80 compiler overlays files; you can create overlay files with LINK-80.
PAS	Pascal/MT + source program filetype.
PLI	PL/1-80 source program filetype.
PRL	Page Relocatable file; a file that does not require an absolute segment. It can be relocated in any Page Boundary (256 Bytes).
PRN	Printable file displayable on console or printer.
REL	Relocatable file produced by RMAC and PL/1-80 that can be linked by LINK-80.
SPR	System Page Relocatable file; system files required to generate CP/M 3, such as BNKBDOS.SPRI, BDOS.SPR, BIOS.SPR, and RESBDOS.SPR.
SUB	Filetype required for submit file containing one or more CP/M 3 commands. The SUBMIT program executes commands in files of type SUB, providing a batch execution mode for CP/M 3.
SYM	Symbol table file. MAC, RMAC, and LINK-80 output files of type SYM. SID and ZSID read files of type SYM.
SYS	System file for CP/M 3.
TEX	Source file for TEX-80", the Digital Research text formatter.
TOK	Pascal/MT+ intermediate language file.
XRF	Cross-reference file produced by XREF.
$$$	Temporary file.

End of Appendix C

Appendix D

CP/M 3 Control Character Summary

Table D-1. Nonbanked CP/M 3 Control Characters

Character	Meaning
CTRL-C	Terminates the executing program and redisplays the system prompt, provided the cursor is at the beginning of the command line. Also, if you halt scrolling with CTRL-S, you can terminate the program with a CTRL-C.
CTRL-E	Forces a physical carriage return but does not send the command line to CP/M 3. Moves the cursor to the beginning of the next line without erasing your previous input.
CTRL-H	Deletes a character and moves the cursor left one character position.
CTRL-I	Moves the cursor to the next tab stop. Tab stops are automatically set at each eighth column. Has the same effect as pressing the TAB key.
CTRL-J	Sends the command line to CP/M 3 and returns the cursor to the left of the current line. Has the same effect as a RETURN or a CTRL-M.
CTRL-M	Sends the command line to CP/M 3 and returns the cursor to the left of the current line. Has the same effect as a RETURN or a CTRL-J.
CTRL-P	Echoes all console activity to the printer. The first time you type CTRL-P, CP/M 3 rings a bell at your console. You can use CTRL-P after you halt scrolling with CTRL-S. A second CTRL-P ends printer echo; no bell rings. CTRL-P has no effect if your system does not include a printer.

Table D-1. (continued)

Character	Meaning
CTRL-R	Places a # at the current cursor location, moves the cursor to the next line, and displays any partial command you typed so far.
CTRL-S	Stops screen scrolling. If a display scrolls by too fast for you to read it, type CTRL-S. CTRL-Q restarts screen scrolling.
CTRL-U	Discards all the characters in the command line, places a # at the current cursor position, and moves the cursor to the next command line.
CTRL-X	Discards all the characters in the command line, and moves the cursor to the beginning of the current line.

Table D-2. Banked CP/M 3 Line-editing Control Characters

Character	Meaning
CTRL-A	Moves the cursor one character to the left.
CTRL-B	Moves the cursor to the beginning of the command line without having any effect on the contents of the line. If the cursor is at the beginning, CTRL-B moves it to the end of the line.
CTRL-C	Terminates the executing program and redisplays the system prompt, provided the cursor is at the beginning of the command line. Also, if you halt scrolling with CTRL-S, you can terminate the program with a CTRL-C.
CTRL-E	Forces a physical carriage return but does not send the command line to CP/M 3. Moves the cursor to the beginning of the next line without erasing the previous input.
CTRL-F	Moves the cursor one character to the right.
CTRL-G	Deletes the character indicated by the cursor. The cursor does not move.
CTRL-H	Deletes a character and moves the cursor left one character position.
CTRL-1	Moves the cursor to the next tab stop. Tab stops are automatically set at each eighth column. Has the same effect as pressing the TAB key.
CTRL-J	Sends the command line to CP/M 3 and returns the cursor to the beginning of a new line. Has the same effect as a RETURN or a CTRL-M keystroke.
CTRL-K	Deletes to th'e end of the line from the cursor.
CTRL-M	Sends the command line to CP/M 3 and returns the cursor to the beginning of a new line. Has the same effect as a RETURN or a CTRL-J keystroke.

D CP/M 3 Control Character Summary CP/M 3 User's Guide

Table D-2. (continued)

Character	Meaning
CTRL-P	Echoes all console activity to the printer. The first time you type CTRL-P, CP/M 3 rings a bell at your console. You can use CTRL-P after you halt scrolling with CTRL-S. A second CTRL-P ends printer echo; no bell rings. CTRL-P has no effect if your system does not include a printer.
CTRL-R	Retypes the command line. Places a # at the current cursor location, moves the cursor to the next line, and retypes any partial command you typed so far.
CTRL-S	Stops screen scrolling. If a display scrolls by too fast for you to read it, type CTRL-S. CTRL-Q restarts screen scrolling.
CTRL-U	Discards all the characters in the command line, places a # at the current cursor position, and moves the cursor to the next line. However, you can use a CTRL-W to recall any characters that were to the left of the cursor when you pressed CTRL-U.
CTRL-W	Recalls and displays previously entered command line both at the operating system level and in executing programs, if the CTRL-W is the first character entered after the prompt. CTRL-J, CTRL-M, CTRL-UL, and RETURN define the command line you can recall. If the command line contains characters, CTRL-W moves the cursor to the end of the command line. If you press RETURN, CP/M 3 executes the recalled command.
CTRL-X	Discards all the characters left of the cursor and moves the cursor to the beginning of the current line. CTRL-X saves any characters right of the cursor.

End of Appendix D

Appendix E

User's Glossary

ambiguous filename: Filename that contains either of the CP/M 3 wildcard characters, ? or *, in the primary filename or the filetype or both. When you use wildcard characters, you create an ambiguous filespec and can easily reference more than one CP/M 3 file. See Section 2 of this manual.

applications program: Program that solves a specific problem. Typical applications programs are business accounting packages, word processing (editing) programs, and mailing list programs.

argument: Symbol indicating a place into which you can substitute a number, letter, or name to give an appropriate meaning to a command line.

ASCII: The American Standard Code for Information Interchange is a standard code for representation of numbers, letters, and symbols. An ASCII text file Is a file that can be intelligibly displayed on the video screen or printed on paper. See Appendix B.

attribute: File characteristic that can be set to on or off.

back-up: Copy of a disk or file made for safe keeping, or the creation of the back-up disk or file.

bit: Switch in memory that can be set to on (1) or off (0). Bits are grouped into bytes.

block: Area of disk.

bootstrap: Process of loading an operating system into memory. Bootstrap procedures vary from system to system. The boot for an operating system must be customized for the memory size and hardware environment that the operating system mana es. Typically, the boot is loaded automatically and executed at power up or when the computer is reset. Sometimes called a "cold start."

buffer: Area of memory that temporarily stores data during the transfer of information.

built-in commands: Commands that permanently reside in memory. They respond quickly because they are not accessed from a disk.

byte: Unit of memory or disk storage containing eight bits.

character string: Any combination of letters, numbers, or special characters on your keyboard.

command: Elements of a CP/M 3 command line. In general, a CP/M 3 command has three parts: the command keyword, the command tall, and a carriage return keystroke.

command file: Series of coded machine executable instructions stored on disk as a program file, invoked in CP/M 3 by typing the command keyword next to the system prompt on the console. CP/M 3 command files generally have a filetype of COM. Files are either command files or data files. Same as a command program.

command keyword: Name that identifies an CP/M 3 command, usually the primary filename of a file of type COM, or a built-in command. The command keyword precedes the command tall and the carriage return in the command line.

command syntax: Statement that defines the correct way to enter a command. The correct structure generally includes the command keyword, the command tall, and a carriage return. A syntax line usualtv contains symbols that you should replace with actual values when you enter the command.

command tail: Part of a command that follows the command keyword in the command line. The command tall can include a drive specification, a filename and/or filetype, and options or parameters, but cannot exceed 128 characters. Some commands do not require a command tall.

concatenate: Term that describes one of PIP's operations that combines two or more separate files into one new file in the specified sequence.

console: Primary input/output device. The console consists of a listing device such as a screen and a keyboard through which the user communicates with the operating system or applications program.

control character: Nonprinting character combination that sends a simple corn- mand to CP/M 3. Some control characters perform line editing functions. To enter a control character, hold down the CTRL key on your terminal and strike the charac- ter key specified. See Appendix D.

cursor: One-character symbol that can appear anywhere on the console screen. The cursor indicates the position where the next keystroke at the console will have an effect.

data file: Nonexecutable collection of similar information that generally requires a command file to manipulate it.

default: Currentlv selected disk drive and/or user number. Any command that does not specify a c;isk drive or a user number references the default disk drive and user number. When CP/M 3 Is first invoked, the default disk drive is drive A, and the default user number is 0, until changed with the USER command.

delimiter: Special characters that separate different items in a command line. For example, in CP/M 3, a colon separates the drive spec from the filename. A period separates the filename from the filetype. Brackets separate any options from their command or filespec. Cominas separate one item in an option list from another. All of the preceding special characters are delimiters.

directory: Portion of a disk that contains descriptions of each file on the disk. In response to the DIR command, CP/M 3 displays the filenames stored in the directory.

DIR attribute: File attribute. A file with the DIR attribute can be displayed b'Y a DIR command. The file can be accessed from the default user number only.

disk, diskette: Magnetic media used to store information. Programs and data are recorded on the disk in the same way that music is recorded on a cassette tape. The term diskette refers to smaller capacity removable floppy diskettes. Disk can refer to a diskette, a removable cartridge disk, or a fixed hard disk.

disk drive: Peripheral device that reads and writes on hard or floppy disks. CP/M 3 assigns a letter to each drive under its control. For example, CP/M 3 can refer to the drives in a four-drive system as A, B, C, and D.

editor: Utility program that creates and modifies text files. An editor can be used for creation of documents or creation of code for computer programs. The CP/M 3 editor is invoked by typing the command ED next to the system prompt on the console. (See ED in Section 6 of this manual).

executable: Ready to be run by the computer. Executable code is a series of instructions that can be carried out by the computer. For example, the computer cannot execute names and addresses, hut it can execute a program that prints all those names and addresses on mailing labels.

execute a program: Start a program executing. When a program is running, the computer is executing a sequence of instructions.

FCB: See **File Control Block**.

file: Collection of characters, instructions or data stored on a disk. The user can create files on a disk.

File Control Block: Structure used for accessing files on disk. Contains the drive, filename, filetype and other information describing a file to be accessed or created on the disk.

filename: Name assigned to a file. A filename can include a primary filename of 1-8 characters and a filetype of 0-3 characters. A period separates the primary filename from the filetype.

file specification: Unique file identifier. A complete CP/M 3 file specification includes a disk drive specification followed by a colon (d:), a primary filename of 1 to 8 characters, a period, and a filetype of 0 to 3 characters. For example, b:example.tex is a complete CP/M 3 file specification.

filetype: Extension to a filename. A filetype can be from 0 to 3 characters and must be separated from the primary filename by a period. A filetype can tell something about the file. Certain programs require that files to be processed have certain filetypes (see Appendix C).

floppy disk: Flexible magnetic disk used to store information. Floppy disks come in 5 1/4- and 8-inch diameters.

hard disk: Rigid, platter--like, magnetic disk sealed in a container. A hard disk stores more information than a floppy disk.

hardware: Physical components of a computer.

hex file: ASCII-printable representation of a command (machine language) file.

hexadecimal notation: Notation for the base 16 number system using the symbols 0, 1, 2, 31 4, 5, 6, 7, 8, 9, A, B, C, D, E, and F to. represent the sixteen digits. Machine code is often converted to hexadecimal notation because it can be easily represented by ASCII characters and therefore printed on the console screen or on paper (see Appendix B).

input: Data going into the system, usually from an operator typing at the terminal or by a program reading from the disk.

interface: Object that allows two independent systems to communicate with each other, as an interface between hardware and software in a microcomputer.

I/0: Abbreviation for input/output.

keyword: See **command keyword.**

kilobyte: 1024 bytes denoted as 1K. 32 kilobytes equal 32K. 1024 kilobytes equal one megabyte, or over one million bytes.

list device: Device such as a printer onto which data can be listed or printed.

logical: Representation of something that might or might not be the same in its actual physical form. For example, a hard disk can occupy one physical drive, and yet you can divide the available storage on it to appear to the user as if it were in several different drives. These apparent drives are the logical drives.

megabyte: Over one million bytes; 1024 kilobytes. See byte and kilobyte.

microprocessor: Silicon chip that is the Central Processing Unit (CPU) of the microcomputer.

operating system: Collection of programs that supervises the running of other programs and the management of computer resources. An operating system provides an orderly input/output environment between the computer and its peripheral devices.

option: One of many parameters that can be part of a command tail. Use options to specifiy additional conditions for a command's execution.

output: Data that the svstem sends to the console or disk.

parameter: Value in the command tail that provides additional information for the command. Technically, a parameter is a required element of a command.

peripheral devices: Devices external to the CPU. For example, terminals, printers,, and disk drives are common peripheral devices that are not part of the processor, but are used in conjunction with it.

physical: Actual hardware of a computer. The physical environment varies from computer to computer.

primary filename: First 8 characters of a filename. The primary filename is a unique name that helps the user identify the file contents. A primary filename contains 1 to 8 characters and can include any letter or number and some special characters. The primary filcname follows the optional drive specification and precedes the optional filetype.

program: Series of specially coded instructions that performs specific tasks when executed by a computer.

prompt: Characters displayed on the screen to help the user decide what the next appropriate action is. A system prompt is a special prompt displayed by the operating system. The system prompt indicates to the user that the operating system is ready to accept input. The CP/M 3 system prompt is an alphabetic character followed by an angle bracket. The alphabetic character indicates the default drive. Some applica-tions programs have their own special system prompts.

Read-Only: Attribute that can be assigned to a disk file or a disk drive. When assigned to a file, the Read-Only attribute allows you to read from that file but not change it. When assigned to a drive, the Read-Only attribute allows you to read any file on the disk, but prevents you from adding a new file, erasing or changing a file, renaming a file, or writing on the disk. The SET command can set a file or a drive to Read-Only. Every file and drive is either Read-Only or Read-Write. The default setting for drives and files is Read-Wr'tte, but an error in resetting the disk or chang-ing media automatically sets the drive to Read-Only until the error is corrected. Files and disk drives can be set to either Read-Only or Read-Wr ite.

Read-Write: Attribute that can be assigned to a disk file or a disk drive. The Read-Write attribute allows you to read from and write to a specific Read-Write file or to any file on a disk that is in a drive set to Read-Write. A file or drive can be set to either Read-Only or Read-Write.

record: Collection of data. A file consists of one or more records stored on disk. An CP/M 3 record is 128 bytes long.

RO: See **Read-Only.**

RW: See **Read-Write.**

sector: Portion of a disk track. There are a specified number of sectors on each track.

software: Specially coded programs that transmit machine-readable instructions to the computer, as opposed to hardware, which is the actual physical components of a computer.

source file: ASCII text file that is an input file for a processing program, such as an editor, text formatter, or assembler.

string: See character string

syntax: Format for entering a given command.

system attribute: File attribute. You can give a file the system attribute by using the SYS option in the SET command. A file with the SYS attribute is not displayed in response to a DIR command; you must use DIRS (see Section 5). If you give a file with user number 0 the SYS attribute, you can read and execute that file from anv user number on the same drive. Use this feature to make your commonly use@ programs avalilable under any user number.

system prompt: Symbol displayed by the operating system indicating that the system is ready to receive input. See prompt.

terminal: See **console.**

track: Concentric rings dividing a disk. There are 77 tracks on a typical eight-inch floppy disk.

turn-key application: Application designed for the noncomputer-oriented user. For example, a typical turn-key application is designed so that the operator needs only to turn on the computer, insert the proper program disk, and select the desired procedure from a selection of functions (menu) displayed on the screen.

upward-compatible: Term meaning that a program created for the previously released operating system (or compiler, etc.) runs under the newly released version of the same operating system.

user number: Number from 0 to 15 assigned to a file when it is created. User numbers can organize files into sixteen file groups.

utility: Tool. Program that enables the user to perform certain operations, such as copying files, erasing files, and editing files. Utilities are created for the convenience of programmers and users.

wildcard characters: Special characters that give CP/M 3 a pattern to match when it searches the directory for a file. CP/M 3 recognizes two wildcard characters, ? and *. The ? can be substituted for any single character in a filespec, and the * an be substituted for the primary filename or the filetype or both. By placing wildcard characters in a filespec, you create an ambiguous filespec and can quickly reference one or more files.

End of Appendix E

Index

B

C

Index