



DIGITAL
RESEARCH®

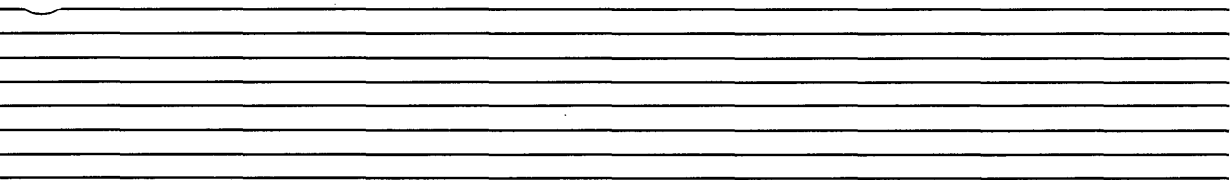
Concurrent™
Operating System

User's Reference Guide
with Tutorial



**DIGITAL
RESEARCH®**

Concurrent™
Operating System
User's Reference Guide
with Tutorial



COPYRIGHT

Copyright © 1984 by Digital Research Inc. All rights reserved. No part of this publication may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language or computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual or otherwise, without the prior written permission of Digital Research Inc., Post Office Box 579, Pacific Grove, California, 93950.

This manual is, however, tutorial in nature. Thus, readers are granted permission to include the example programs, either in whole or in part, in their own programs.

DISCLAIMER

Digital Research Inc. makes no representations or warranties with respect to the contents hereof and specifically disclaims any implied warranties of merchantability or fitness for any particular purpose. Further, Digital Research Inc. reserves the right to revise this publication and to make changes from time to time in the content hereof without obligation of Digital Research Inc. to notify any person of such revision or changes.

TRADEMARKS

CP/M and Digital Research and its logo are registered trademarks of Digital Research Inc. Concurrent, Concurrent CP/M, and TEX are trademarks of Digital Research Inc. IBM is a registered trademark of International Business Machines. Intel is a registered trademark of Intel Corporation.

The Concurrent Operating System User's Reference Guide with Tutorial was prepared using the Digital Research TEX™ Text Formatter and printed in the United States of America.

* First Edition: May 1984 *

Foreword

Welcome to the Concurrent™ operating system for CP/M® and PC DOS. This operating system runs both IBM® PC DOS and CP/M applications and reads and writes to PC DOS and CP/M files.

Throughout this manual we refer to the Concurrent operating system for CP/M and PC DOS as Concurrent. This manual contains initial instructions on how to use Concurrent. It also includes a summary of Concurrent's commands for easy reference. Additional information about Concurrent is contained in the Concurrent CP/M™ Operating System User's Guide (cited as Concurrent CP/M User's Guide), which is distributed with this system. Refer to it when needed.

Who Should Read this Manual?

The information in this manual is for the following three groups of people:

- users not familiar with the PC DOS operating system
- experienced PC DOS users who want to see how Concurrent compares to the operating system they have been using
- skeptics not really sure that one operating system can run both PC DOS and CP/M applications, and read and write to both PC DOS and CP/M data files

If you are a new PC DOS user, you will be learning concurrency features as you learn the entire Concurrent operating system.

If you have been using PC DOS for some time, you will want to know how this system compares to it. This manual tells you which program and utility commands have changed. It also describes the many features of Concurrent that are not part of the PC DOS operating system.

If you are skeptical, having been told that one operating system cannot run both PC DOS and CP/M applications, you will learn that Concurrent does this successfully. Skeptics are concerned about protecting their existing data files and confirming that their application programs run in this system.

All these needs and interests are addressed in this manual. Our goal is for you to feel comfortable with this new system. It is a powerful system and we want you to begin using it just as soon as possible.

How to Use this Manual

The Concurrent Operating System User's Reference Manual with Tutorial has two main parts. The first part includes Sections 1 through 5 and contains initial information about Concurrent and describes its many features. The second part includes Sections 6, 7, and 8 and is primarily a reference to be used once you become familiar with Concurrent.

Sections 1 and 2 introduce you to the Concurrent operating system and explain what it can do. Section 3 provides some specifics of Concurrent, including the use of virtual consoles.

What you need to know to use the tutorial is explained in Section 4. Section 5 describes Concurrent's features in detail.

Section 6 summarizes the PC DOS commands supported by Concurrent. Batch processing and the batch subcommands are described in Section 7. Section 8 gives a summary of the special commands that signal Concurrent.

In this manual, what you enter at the keyboard is shown in boldface type.

Go ahead now and leaf through the Contents. We at Digital Research believe you will find Concurrent to be a powerful and versatile operating system.

Contents

1	Introduction	
	What Your System Contains.	1-1
	What You Need to Run Concurrent	1-1
	Concurrent Is Easier and Faster	1-1
	What Is Exciting About Concurrent?	1-2
2	An Introduction to Operating Systems	
	CP/M and PC DOS Operating Systems	2-1
	Concurrent Operating System	2-1
3	Getting Acquainted with Concurrent	
	Concurrent on a Dual Floppy-disk System	3-1
	Concurrent on a Hard-disk System	3-1
	Starting Concurrent	3-2
	Using Virtual Consoles	3-2
4	Using The Tutorial	
	Access the Demo Disk	4-1
	Lessons in the Tutorial	4-2
5	Getting Comfortable With Concurrent	5-1
	Understanding More About Virtual Consoles	5-1
	Reviewing the Status Line	5-2
	Making Use of User Numbers	5-2
	User Numbers and PC DOS Disks	5-3
	Internal and External Operating System Commands	5-3

How Concurrent Uses Disk Drives and Files	5-4
The Default Drive	5-4
The System Drive	5-4
File Search Paths	5-5
Displaying the File Directory	5-7
The PC DOS Subdirectories	5-7
When to Use the SUSPEND Command	5-7
Using the MEM (Memory) Command	5-9
About the 8087 Command	5-11
Using COPY and PIP Commands	5-11
Tips on Using Concurrent with a Hard-disk System	5-11
Accessing the Hard Disk	5-11
Copying Concurrent Utilities to a Hard-disk System	5-12
Partitioning Your Hard Disk	5-12
6 Supported DOS Commands	
COPY Command	6-2
DATE Command	6-8
DIR Command	6-10
ERASE Command	6-13
RENAME Command	6-14
TIME Command	6-16
7 Batch Processing	
Creating Batch Files	7-1
Batch Subcommands	7-1
PAUSE	7-2
REM	7-3
SYS	7-3

Running Batch Files	7-4
Using Parameters with Batch Files	7-4
8 Programs that Signal Concurrent CP/M	
MEM	8-2
SUSPEND	8-2
SYSDISK	8-6
8087	8-8
Index	Index-1

Tables and Figures

Tables

5-1. External PC DOS Commands Under Concurrent . . .	5-3
6-1. COPY Command Options	6-3

Figures

2-1. Relationship Between Concurrent and PC DOS . . .	2-2
3-1. Using Concurrent's Virtual Consoles	3-3
5-1. Routing of Applications Under Concurrent and PC DOS	5-8
5-2. Allocation of Memory with the MEM Command . . .	5-10

Introduction

What Your System Contains

- Concurrent operating system disk
- demonstration disk containing a hands-on learning tutorial

What You Need to Run Concurrent

To run Concurrent, your microcomputer should have the following characteristics:

- 8088 or compatible microprocessor
- minimum of 256 kilobytes (256K) of memory, 512K is recommended
- hard-disk drive with one or more floppy-disk drives, or two floppy-disk drives

Although Concurrent runs successfully on a computer with dual floppy-disk drives, the full potential of concurrency can best be realized with a hard-disk drive. We recommend you use a microcomputer with this configuration.

Concurrent Is Easier and Faster

People buy computers to save time. After using a computer for a while, they wish it could save them even more time.

How many times have you been in the middle of writing a report on a word processor and realized that some of the information you needed was in a spread sheet you created earlier?

Have you ever faced a deadline and needed to print one document, access a data-base file, and write another document all at the same time? Concurrent helps solve these kinds of problems for you. It uses different screens running different programs at the same time. Each screen is referred to as a virtual console, and you see only one virtual console at a time. You can work on as many as four virtual consoles at a time, switching from one to the other as easily as changing channels on a television set.

Using Concurrent, you can search through a data base on one console, print a document on another, and even write a cover letter for the document on a third--all at the same time!

What Is Exciting About Concurrent?

No matter which operating system you used in the past--a CP/M-based one or PC DOS--you found you could only use programs that were written to run on that particular operating system.

Concurrent brings you a major new feature. Now you can run software applications designed for either PC DOS or CP/M operating systems at the same time. No longer will the operating system limit your choice of software applications.

In addition, you can run up to four programs at the same time. Within a few weeks, you will be amazed that you ever managed with only one program running at a time.

Because Concurrent reads all PC DOS files as well as the popular PC DOS programs, the PC DOS user does not have to worry about retyping or copying valuable information. Concurrent reads all those files just as PC DOS did. Remember it also reads all CP/M files.

End of Section 1

An Introduction to Operating Systems

One way to think about an operating system is through the analogy of a traffic officer. Envision a traffic officer, complete with white gloves and whistle, high up on a podium in the middle of a busy intersection. That officer directs the flow of activities, making sure that everyone gets their turn and that all traffic moves smoothly. The traffic officer is ultimately responsible for the whole system and everyone in it.

The operating system of a computer is very similar. It makes sure that what you enter at the keyboard is routed to the correct location and that material is stored so you can find it when you want it. It also directs all the peripherals that may be part of your computer system: printers, plotters, modems, and the like.

Operating systems make life easier for the everyday user. Without them, entering information and commands is cumbersome and highly technical. Operating systems let you enter simple terms, like "help", sparing you from the complexity of computer jargon.

CP/M and PC DOS Operating Systems

As the microcomputer market evolved, users found that they needed to learn details of the operating system to load and use the software applications that they had purchased.

Even more important, they found that not all software worked on all operating systems. It was necessary to consider the operating system, along with the hardware and software, when purchasing a computer.

One very popular operating system is CP/M. It runs most of the business applications and educational software sold today. Another popular operating system is PC DOS, the operating system found on the IBM Personal Computer.

Concurrent Operating System

Now the Concurrent operating system is available. It runs programs designed to run under PC DOS as well as those designed to run under CP/M.

Thus, Concurrent is an operating system that does all this:

- runs the popular PC DOS software applications
- reads PC DOS data files
- runs the 16-bit CP/M software applications
- reads CP/M data files

Note, however, that the PC DOS operating system cannot read or run CP/M applications or files. Figure 2-1 shows the relationship between the Concurrent and PC DOS operating systems.

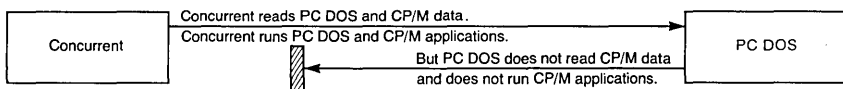


Figure 2-1. Relationship Between Concurrent and PC DOS

With Concurrent, you can use the operating system commands you are familiar with, either PC DOS or Concurrent. Details about the various commands are discussed in the following sections of this document. Sections 6, 7, and 8 give a complete summary of the commands supported by Concurrent.

The PC DOS user might find some CP/M commands more useful than their PC DOS equivalents. For example, the PIP (Peripheral Interchange Program) command is more powerful in Concurrent than the COPY command in PC DOS, even though they both copy files. So both commands have been included in the Concurrent operating system.

Because Concurrent has some unique features, there are new commands that control those features. For example, the MEM (memory) command allows you to allocate the memory available to a particular application program. The memory of the microcomputer is shared by programs running on each of the four virtual consoles. Some popular programs, like spread sheets, need to know how much memory is available. MEM prevents any one program from taking all the available memory. MEM, and the other commands unique to Concurrent, are discussed in more detail in Section 5, "Getting Comfortable with Concurrent."

End of Section 2

Getting Acquainted with Concurrent

First, as with all new programs, you need to make a backup copy of both the operating system disk and the demonstration disk. To do this, follow the directions that are supplied with your microcomputer.

There are differences between using Concurrent on a floppy-disk system and on a hard-disk system. Before you start using Concurrent on your microcomputer, read the following sections that describe these differences.

Concurrent on a Dual Floppy-disk System

You have the full power of the Concurrent operating system available to you on a dual floppy-disk system. However, because you have only two disk drives, the number of programs you can run at one time might be limited. You might encounter limitations if the program you are running has the following needs:

- It takes up an entire floppy disk, preventing you from putting two or more programs on one disk.
- The program you are running needs a large amount of memory to handle all the input and computations; for example, a large spread sheet.
- The program requires a separate disk for its data storage.

You need to note the disk space requirements of the programs you run, taking into consideration the above factors. If you are using many programs and they require large amounts of data, you will find Concurrent's support of hard-disk drives a valuable feature.

Concurrent on a Hard-disk System

While you are getting acquainted with Concurrent, we suggest you run it off floppy-disk drive A. After you have become more familiar with the system by completing the exercises in this manual and on the demonstration disk, transfer the commands and utilities to your hard disk. To do this see "Tips on Using Concurrent with a Hard-disk System" at the end of Section 5.

Starting Concurrent

Insert the Concurrent operating system disk into the A drive of your microcomputer. The A drive is the default drive, the one your computer accesses when no other drive is specified.

Close the drive door and turn the power on. After a short pause, during which time the system is loading into memory, you will see the system prompt. It looks like this:

A>

The prompt tells you Concurrent is ready and waiting to read a command from your keyboard. You will do a simple exercise now that shows you how to change virtual consoles.

Using Virtual Consoles

With Concurrent, you have four different consoles available to you at all times; they are as easy to access as changing channels on a television set. We call them consoles because they are like having four different stand-alone computers working for you at the same time.

At any one time, input from the keyboard is directed to one virtual console, called the active or foreground console. The other three virtual consoles are referred to as background consoles.

Figure 3-1 shows you the concept of four virtual consoles.

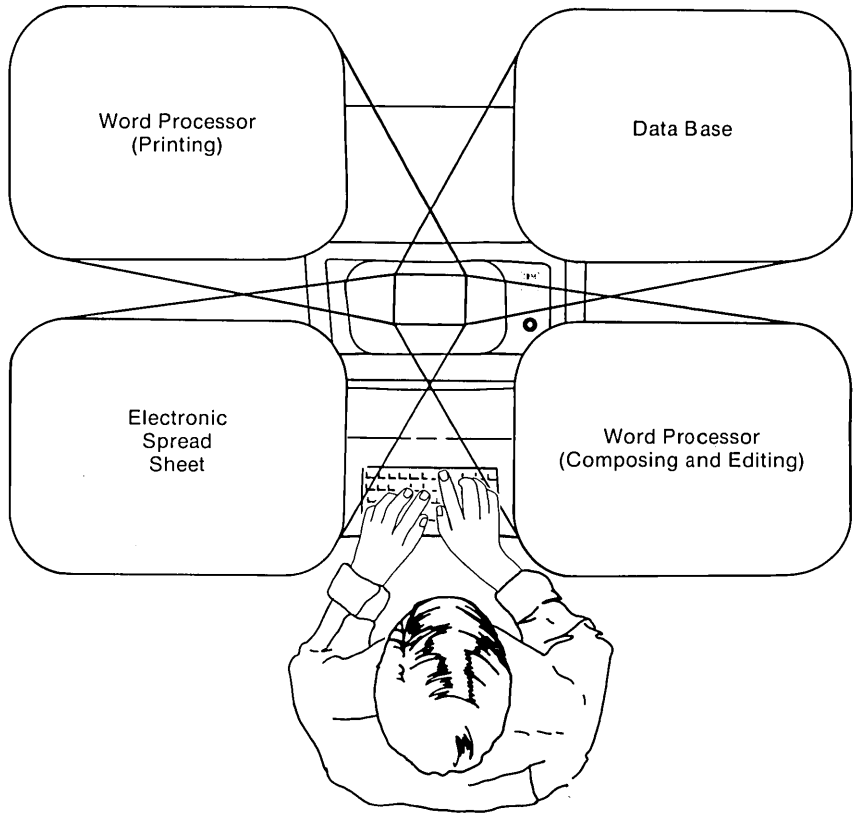


Figure 3-1. Using Concurrent's Virtual Consoles

The last line of your terminal is known as the status line. The status line displays certain information including which virtual console is currently in use.

The number of the active console is always displayed on the left side of the status line.

When you first start Concurrent, you are always on Console 0. The four consoles are numbered 0, 1, 2, and 3.

Moving from console to console is simple. All you do is hold down the CONTROL key with your left hand and press the number key of your choice on the keypad to the right side of the keyboard. On most keyboards the CONTROL key is spelled CTRL or Ctrl. In this manual, we use CTRL to refer to the CONTROL key.

Go ahead now and move from one console to the other. Consoles 1, 2, and 3 are empty at the moment. However, you can tell that you have actually moved by noting the number on the left side of the status line.

Hold down: the CTRL key

Press: 1, 2, 3, and 0 to move among all four consoles

End of Section 3

Using the Tutorial

Access the Demo Disk

Your Concurrent operating system package includes a demo disk. The demo disk contains a tutorial that demonstrates Concurrent's features in detail. Plan to spend about an hour going through the tutorial. The following information tell you how to use it.

1. Start Concurrent if you have not already done so. Directions for doing this are in the preceding section.
2. When you see the prompt, remove the operating system disk from drive A.
3. Place the demo disk in drive A.
4. Start the tutorial on the demo disk by typing the following command:

A>tutor

Then press the RETURN (or ENTER) key.

5. You can exit the tutorial on the demo disk at any time. Simply press the CTRL Key and the number 3 key simultaneously. This moves you to Console 3. Type "stop" and press RETURN. This exits you from the tutorial program.

Hold down: the CONTROL key

Press: 3

Type: Stop

Press: the RETURN key

Lessons in the Tutorial

The tutorial is in two parts.

Part 1 contains exercises that build confidence in using all four consoles. The exercises show how to do the following:

- move among all four virtual consoles
- move from one application program to another
- use the Concurrent HELP facility

Part 1 shows that programs on virtual consoles continue to work even when you are not viewing them. It will also show that you access your disk directory without exiting a program being worked on.

Part 2 of the tutorial demonstrates how Concurrent reads PC DOS files. You need to have a PC DOS-formatted disk for this part of the tutorial.

The tutorial contains a sample letter file which you will transfer to your PC DOS-formatted disk and display it on the terminal screen. After transferring the letter file with the COPY command, you will use the Concurrent operating system to read the file from the PC DOS disk.

After you are finished with the tutorial, go on to Section 5. Section 5 examines more closely the details of Concurrent and introduces you to several additional features.

End of Section 4

Getting Comfortable with Concurrent

This section outlines again some of the material contained in the tutorial on the demo disk. It also tells you more about the Concurrent operating system and some of its unique commands. It tells you when to refer to the Concurrent CP/M User's Guide that accompanies this document and when to refer to directions supplied by the manufacturer of your microcomputer.

Understanding More About Virtual Consoles

Remember that we call the four available "channels" the four virtual consoles.

Each time you start Concurrent, whether it is on a hard-disk or floppy-disk system, you are always on the first console, Console 0. The other available consoles are 1, 2, and 3.

You can always tell what console you are on by looking at the left-hand side of the status line at the bottom of the screen.

Each console acts independently. You can run different applications on each console. These applications can access data on any drive in your computer. Three examples follow that illustrate how programs run concurrently on the virtual consoles.

It is Monday morning and you need to finish a sales report by 5:00 PM, using a word processor to write the report. However, some of the data you need for the report must be developed in your spread sheet program. You need to go quickly back and forth between the word processor and the spread sheet to complete the report. This is one example of how concurrency helps you--when you need more than one program to complete a single task.

It is Tuesday morning and you are behind because you spent yesterday writing the sales report. You have a number of different tasks to catch up on. You must phone many customers and also update customer orders. You need constant access to both your name and address file and your customer account files. This is a second example of how concurrency helps--when you need to move back and forth between unrelated tasks.

It is Wednesday morning. You need to transmit the customer order information via a modem to a main-frame computer in your regional office. From experience you know it take 45 minutes to transmit the information. But you have a rush memo to get out in the next half hour, and so you also need to use your word processor. This is a third example of why you need concurrency--when you are doing one task that normally takes up all your computer's time and you have something else that must be done at the same time.

Reviewing the Status Line

You already know that the status line on the bottom of your screen tells you what virtual console you are working on. It also tells you other information, including the name of the program you are using, other files you may have open, what printer you are connected to, and if the printer is engaged. Refer to the Concurrent CP/M User's Guide for more information about the status line. At this point, being able to check what console you are on is enough information to use the operating system efficiently.

Making Use of User Numbers

User numbers are a feature of Concurrent that is supported by CP/M-formatted disks. User numbers help you keep better track of what is on your disk by allowing you to group together related files. This is especially helpful for hard-disk users who store a number of programs and many supporting data files on the hard disk. There are 16 user numbers available, starting with 0 and going through 15.

User numbers help organize your files into meaningful groups. For a very simple system, you might assign user numbers to files like this:

User 0	system utilities, command files, and application programs
User 1	word processor data files
User 2	spread sheet data files
	and so on

If you do not explicitly set the user number, all files are automatically stored on User 0. Refer to the Concurrent CP/M User's Guide for more information about user numbers.

User Numbers and PC DOS Disks

PC DOS disks and programs do not recognize user numbers. To take advantage of user numbers with PC DOS files, copy the PC DOS programs and data to a CP/M-formatted disk.

Note: PC DOS programs and data files are automatically stored on User 0. If you set a number other than zero to the program or file, that number appears at the system prompt, but the program or file will still be stored to User 0.

Internal and External Operating System Commands

PC DOS supports a number of commands that are referred to as internal commands. This means that they are always resident in memory and can be used at any time. DIR is an example of an internal command. When you type the command DIR, the computer shows you a directory of all your files.

External command are not resident in memory. They are stored as separate command files on your operating system disk and loaded in memory when you enter the command by typing it in at the system prompt. Keep external commands that you frequently use on each of your program and data disks. Some users make a point of routinely copying external commands to new disks immediately after they format them; they are then always accessible. The Concurrent CP/M User's Guide tells you how to format disks and copy command files to formatted disks. On the hard-disk drive, the external command are always available.

Some command files that are internal to the PC DOS operating system are now external commands in the Concurrent operating system. They are listed in Table 5-1.

Table 5-1. External PC DOS Commands Under Concurrent

Command	Description
COPY	copies files from one location to another
DATE	sets the date
DIR	looks at a directory of files
ERASE or DEL	erases a file
REN	renames a file
TIME	sets the time

These commands are described in detail in Section 6, "Supported PC DOS Commands."

How Concurrent Uses Disk Drives and Files

This section describes two features of Concurrent's support for PC DOS: access to PC DOS media and the filetype search order.

The Default Drive

When Concurrent is running, there is one drive whose letter is displayed at the operating system prompt. The system prompt is the letter you see with the prompt sign, >, displayed after it. This drive is called the default drive. Concurrent looks to the disk in the default drive to find the programs or data files that you ask for.

Note: Each virtual console acts independently and so a different default drive can be set for each virtual console. For example, A might be the default drive for virtual consoles 0 and 2, and B the default drive on virtual consoles 1 and 3.

At any time you can tell the Concurrent operating system to look at another drive for a given file by including the drive designation as part of the program name. For example, if your default drive is the A drive and you want to access a file called MEMO that is stored in the B drive, you do so by typing the drive letter, a colon, the file name, and then pressing the carriage return. Your entry would like this:

```
A>B:MEMO
```

You can change the default drive at any time by typing the drive letter, followed by a colon, and pressing the carriage return. For example, if your current drive is A and you wish to change it to B, you type B: at the A prompt. Your command looks like this:

```
A>B:
```

The default drive is now the B drive, and the system prompt is now displayed like this:

```
B>
```

The System Drive

Think of the system drive as the backup to the default drive. The Concurrent operating system looks at the system drive to find a program or command file after it first searches for the file on the default drive. The system disk is a useful place to store your frequently used commands. You might want to keep

all your operating system utilities and external commands on the system-drive disk.

For example, on a dual floppy-disk microcomputer, use the B drive as your system disk and keep frequently used commands and system utility commands on a disk in this drive. On a hard-disk microcomputer, specify your hard disk as your system disk and keep frequently used commands, system utility commands, and all your programs there.

Note: In order for the files on your system disk to be visible from other drives, they must be on User 0 and have the SYS attribute assigned to them. Refer to the Concurrent CP/M User's Guide for detailed information about the SYS attribute and how to set it.

Concurrent assigns one drive as the system disk whenever you start the operating system. To find out what drive is designated as your system disk, type SYSDISK at the system prompt and press the return key. If you are at the A prompt, the screen looks like this:

```
A>SYSDISK
```

The system responds by telling you which drive is the system disk.

You can change your system disk designation at any time by specifying the drive of your choice after you type SYSDISK. If you want to change the system disk to drive B, enter the following command:

```
A>SYSDISK B:
```

File Search Paths

Concurrent searches for files in a predetermined search path using both drive and file paths as criteria. As discussed above, when you type in a request for a command or program, Concurrent first looks for it on the default drive. If it does not find the program or command file on the default drive, Concurrent looks for it on the system drive. In both cases, the operating system also looks for the program or command file on the current user number and then on User 0.

Program and command files can be of several types. They are designated by a file extension and are part of the filename. The extensions are always separated from the rest of the filename by a period; for example, SORT.BAT. The file extensions, or filetypes, that Concurrent searches for are as follows:

CMD	a CP/M program
COM	a PC DOS program
EXE	a PC DOS command
BAT	a PC DOS batch file

If you have two files with the same name but different extensions, Concurrent uses the extensions as a tie breaker. It searches for the file in the pattern shown above, that is, first as a CMD, then as a COM, then an EXE, and finally as a BAT file. For example, if you have two files, one called SORT.EXE and one called SORT.BAT, and you type in the command SORT, Concurrent looks for the command file first as SORT.COM, then as SORT.COM, and then as SORT.EXE. It finds and executes the SORT.EXE command for you.

Concurrent uses both the drive and the filename when searching for files. For example, if you type SORT without specifying either the drive or the extension, the operating system searches for it in the following sequence:

CMD	on the default drive and then on the system drive
COM	on the default drive and then on the system drive
EXE	on the default drive and then on the system drive
BAT	on the default drive and then on the system drive

For the SORT example, the operating system tries to find an EXE command on the default drive before it searches for it on the system drive. It would try to find SORT.COM before it searches for SORT.BAT on either drive. The first file the operating system finds during the search will be the file it executes.

At any time you can override the search path by specifying both the drive and the command extension. For example, if you want to execute the SORT.BAT file that is on the B drive, enter it like this at the system prompt:

```
A>B:~SORT.BAT
```

Note: Be careful when you load and use programs that contain identical file names. Depending on their location, you might execute a program or command that is not the one of your choice.

Displaying the File Directory

With PC DOS, the command DIR lists the name and size of each file in the directory. Under Concurrent, DIR only shows you the name of each file.

To obtain complete file information, use DIR.EXE or the Concurrent command, SDIR. These are both external commands. SDIR gives you all the information you are used to seeing under PC DOS, plus additional information. Read about the SDIR command in the Concurrent CP/M User's Guide to find out more about what SDIR can tell you.

The PC DOS Subdirectories

With PC DOS versions 2.0 and higher, subdirectories are available. These subdirectories organize files into groups.

Concurrent can access all files in the root directory of PC DOS 2.0 or higher, but it can not access files stored in the PC DOS subdirectories. If you want to access files stored in a subdirectory, first copy those files to the root directory.

When to Use the SUSPEND Command

With the Concurrent operating system, several applications can run at the same time, each one on a separate virtual console. While each virtual console is totally independent, they all share common system resources, such as memory, the disk drives, and the terminal screen. Concurrent manages this entire process, part of which is ensuring that each application gets its share of the common resources.

Most CP/M-based applications are written in such a way that output from the application passes through the operating system before appearing on the terminal screen. Because the output passes through the operating system, Concurrent can coordinate the output from several programs at one time. This coordination ensures that while one program is active on your terminal screen, another program does not send information to that screen and write over the information being displayed by the active program.

A number of PC DOS applications bypass the operating system and write directly to the terminal screen. Figure 5-1 illustrates this difference between the two operating systems.

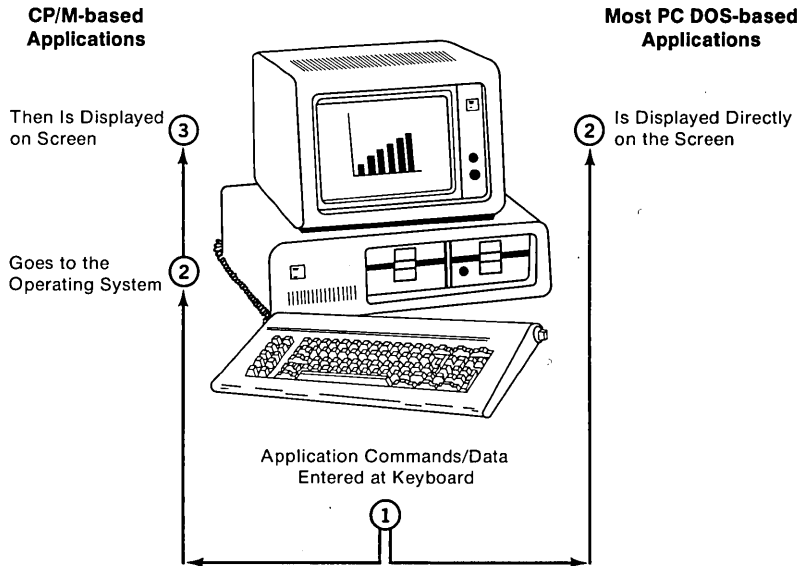


Figure 5-1. Routing of Applications Under Concurrent and PC DOS

Because some PC DOS programs do bypass the operating system, they might continue sending characters to the screen, even if the program is on a virtual console running in the background. When this happens you might find stray characters writing over the program you are using on the foreground virtual console.

Concurrent manages both types of programs and prevents stray writing to the terminal screen with a command called SUSPEND. This command suspends the operations of any program occupying a background virtual console. It does this by automatically recognizing PC DOS programs and sets SUSPEND=ON when the program is loaded.

If a program in a background virtual console continues to write information to the terminal screen, check the setting of the SUSPEND command. See Section 8, "Special Commands that Signal Concurrent," for more information about the SUSPEND command.

Using the MEM (Memory) Command

As already mentioned, each virtual console is independent, although they all share common system resources. One of the shared resources is memory.

The MEM command lets you tell the Concurrent operating system how much memory any particular PC DOS program needs. Doing this leaves the remainder of the memory available for other programs on other virtual consoles. You can set MEM for PC DOS programs that have a filetype of EXE or COM. Looking at the program name in the directory shows whether the PC DOS program has the EXE or COM filetype. The documentation that comes with the PC DOS program tell you how much memory the program needs.

Most of the time, set MEM to match the amount required to load the program. In some cases the amount of memory needed to load the program only runs that program at a minimum level and you might need to set memory to a higher value. An example of this is running a large spread sheet model that needs more memory than the minimum requirement.

When you load the Concurrent operating system, there is a default memory value set for each of the virtual consoles, usually this is 128K. Use the MEM command to check the default setting. Many popular programs need 128K of memory; however, some need less and some need more.

The MEM command lets you change the default setting for each console to match the memory needs of the various programs. It also ensures that the program does not occupy more memory than it needs. Figure 5-2 illustrates how memory is allocated under Concurrent. If your program requires 192K of memory, you would use the MEM command to override the default setting of 128K.

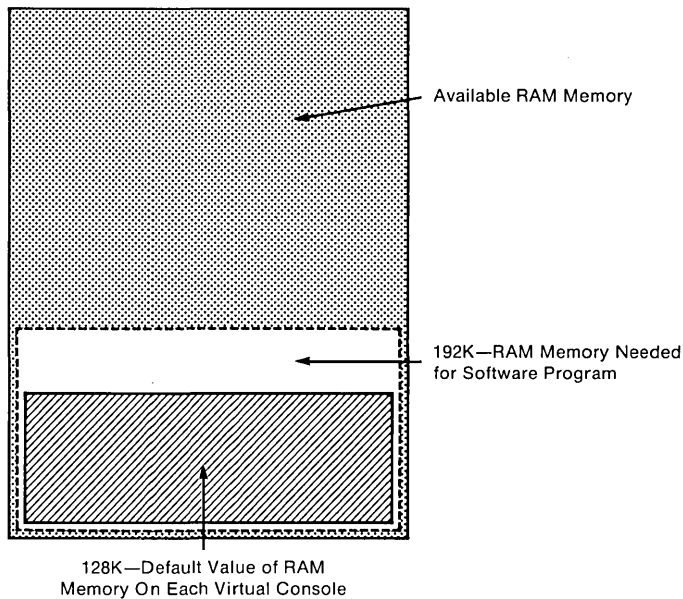


Figure 5-2. Allocation of Memory with the MEM Command

Set MEM by doing the following steps.

1. Move to the virtual console that will run your program.
2. At the system prompt, type in the following command:

```
A>MEM XXX
```

where XXX stands for the amount of memory you wish to allocate for that console.

3. Press the RETURN key; this sets the MEM value.

View MEM by doing the following steps.

1. Move to the virtual console that you want to view.
2. At the system prompt, type in the command

```
A>MEM
```

3. Press the RETURN key. The current MEM value for that console is displayed.

When you load a program, you might get an error message stating that memory needed to load the program is not available. If you get this message, use the MEM command to check the default setting. If you know that the program requires less memory than the default setting, use MEM to set the value to reflect actual memory requirements for the program. You can also get this message when the minimum memory required to load the program is larger than the default setting. When you know the program needs more memory, use MEM to set the value to equal the requirement.

About the 8087 Command

The 8087 microprocessor chip is customized to meet the needs of computer users who work with large batches of numbers and number-processing tasks. If your computer has an 8087 chip, this command lets you access it at will. Complete instructions for using this command are in Section 8, "Special Commands that Signal Concurrent."

Using COPY and PIP Commands

COPY and PIP both copy files from one place to another, but PIP is a more powerful command than COPY. It has a number of advanced features useful in programming. Go ahead and review its many uses in the Concurrent CP/M User's Guide; you will find there are times when you may prefer it over the PC DOS-based COPY command.

Tips on Using Concurrent with a Hard-disk System

Now that you are more familiar with concurrency, you will probably want to use the full potential of Concurrent on your hard-disk system.

Accessing the Hard Disk

Concurrent lets you access both CP/M and PC DOS media. This means that you can read and write to a PC DOS-formatted disk. It also means you can read and write to a PC DOS partition of a hard disk.

There are no special commands that you need to know to enable the Concurrent operating system to access PC DOS media. The same set of commands work with both CP/M and PC DOS media. You can access files on a PC DOS-formatted hard disk using the same commands that you did while running PC DOS.

Copying Concurrent Utilities to a Hard-disk System

When using Concurrent on a hard-disk system, copy all Concurrent commands (the CMD and EXE files on your operating system disk) to your hard disk using the PIP or COPY command. Then, set the hard disk as your system disk using the SYSDISK command. This ensures that Concurrent can access all the external commands and utilities that you need during a work session. PIP is described fully in the Concurrent CP/M User's Guide. For more information on the COPY command see Section 6 in this manual. See Section 8 for details on the SYSDISK command.

Note: Do not copy the CCPM.SYS file to your hard disk along with your utilities. Concurrent must always be started by booting from the floppy-disk drive.

Partitioning Your Hard Disk

Many hard disks can be partitioned so that the partitioned sections can be formatted and accessed by different operating systems. If your hard disk currently is partitioned for CP/M and PC DOS, Concurrent can read and write to both partitions.

If your hard disk is currently formatted entirely for PC DOS, consider installing a CP/M partition. CP/M formatted media support password protection and file and record locking capabilities. This level of data security is not available on PC DOS formatted disks.

To install a CP/M partition on your hard disk, take the following steps.

1. Back up the current contents of the hard disk.
2. Set the size of both the CP/M and PC DOS partitions.
3. Format the CP/M partition.
4. Restore the contents of your PC DOS partition.

Follow the instruction in the reference materials supplied by the manufacturer of your microcomputer to complete these steps.

End of Section 5

Supported PC DOS Commands

Concurrent supports six PC DOS commands. You can use these commands to maintain your files. The commands COPY, DATE, DIR, ERASE, RENAME, and TIME all perform simple housekeeping tasks for you.

To execute any of these six PC DOS commands, its corresponding EXE program file must be on the default drive or on your system drive.

If you want to access PC DOS programs from any drive or user in your system, you must use the Concurrent SET command to assign PC DOS programs the SYS attribute. Store these SYS files in User 0 of your system drive. See the Concurrent CP/M User's Guide for more information on how to set file attributes with the SET command.

COPY

Syntax

```
COPY{/A|/B|/n} src-filespec{/A|/B|/n}
dest-filespec{/A|/B|/V|/n}

COPY{/A|/B|/n} src-filespec{/A|/B|/n}+
src-filespec{/A|/B|/n} dest-filespec{/A|/B|/V|/n}
```

Explanation

The COPY command allows you to copy one or more files from one drive or device to another drive or device. COPY can rename a file after copying it and can combine two or more files into one file. COPY can also create a copy of a file within the same drive, but you must be sure to give the files different names.

COPY can transfer data from a logical input device to a logical output device. Concurrent uses logical input and output devices to communicate with the actual physical devices that are connected to your computer. For example, COPY can send a character (ASCII) file to a printer or to a logical output device.

Concurrent recognizes the following logical devices:

- CON:
- LPT1:
- PRN:
- NUL:

CON: is assigned to your console. Both LPT1: and PRN: are assigned to the current default printer. NUL: is a dummy output device used for testing programs. When you use NUL: as a device assignment, your programs's output is discarded.

You can also type input at your keyboard and create a character file using COPY. For example,

```
A>COPY CON: MYFILE.BAT
```

This is an example of how to create a file.

```
<Press CTRL-Z>
```

You now have a file in drive A called MYFILE.BAT that contains the preceding sentence. The CTRL-Z is the end-of-file character. After pressing a CTRL-Z and RETURN, you return to the Concurrent prompt.

Table 6-1 describes the COPY command options.

Table 6-1. COPY Command Options

Option	Meaning
/A	/A causes the file to be treated as an ASCII (text) file. If you include the /A option after the source filespec, the file's data is copied up to, but not including, the first end-of-file character (CTRL-Z, which is hexadecimal 1A) found in the file. The remaining data is not copied. Using a /A after the destination filespec adds a CTRL-Z as the last character in the file.
/B	Using a /B option with your source filespec causes the entire file to be copied regardless of any CTRL-Z character found in the file. If you put a /B at the end of your destination filespec, no end-of-file character (CTRL-Z) is added.
/V	The /V option causes Concurrent to verify that the data written to your destination diskette is recorded accurately. Using /V causes the COPY command to run slower than usual due to the additional time required for verification.
/n	The /n option denotes the user number for a particular file. n is a number from 0 to 15 that specifies which user number to copy to or from. If you include the /n option after the source filespec, the file is copied from user n. Using a /n after the destination filespec causes the file to be created under that user number. Note that user numbers exist only under Concurrent not PC DOS. Therefore, the /n option is ignored when reading from, or writing to, PC DOS media.

Note: When you concatenate, the default value is /A. When you are not concatenating files, the default is /B.

Examples:

A>COPY B:MYFILE.TYP

This example takes your source file and copies it to your default drive, in this case drive A. The filename does not change. Note that if you omit the destination file specification, COPY assumes you intend the default drive.

A>COPY MYFILE B:

This example copies your source file in the default drive A to the designated drive B.

A>COPY *.* B:

In this example, all files in drive A are copied to the diskette in drive B. You see each filename appear at your screen as it is copied.

A>COPY MYFILE.TYP B:* .TXT

This command copies MYFILE.TYP from drive A to drive B and renames it MYFILE.TXT

If you want to combine two files, copy them to a new drive, and rename your new file, type this command:

A>COPY A:MYFILE.TYP+B:YOURFILE.TYP C:OURFILE.TYP

The preceding command appends YOURFILE.TYP in drive B to MYFILE.TYP in drive A. Then the new file is copied to drive C where it is renamed OURFILE.TYP. If MYFILE and YOURFILE are the only two TYP files in drives A and B, you can shorten the command as follows:

A>COPY A:* .TYP+B:* .TYP C:OURFILE.TYP

Suppose you have the following list of files in your A and B drives:

Drive A:	Drive B:
● A.OUT	● A.OUT
● B.OUT	● C.OUT
● C.OUT	● A.LST
	● C.LST

If you enter the command,

```
A>COPY *.OUT+B:*.OUT C:ALL.OUT
```

COPY first concatenates all the files from the beginning to the end of the directory in drive A with a filetype of OUT. Then COPY appends the files from drive B with the filetype of OUT to the OUT files in drive A. The files concatenated in this sequence are then written to file ALL.OUT in drive C.

Use the same files listed previously and note how the previous example of concatenation differs from the following. If you entered the following command:

```
A>COPY *.OUT+B:*.LST C:*.ALL
```

COPY looks for the first file in drive A, in this case A.OUT, and then searches in drive B for a file with the corresponding filename, in this case A.LST. Once A.LST is found, it is appended to A.OUT. The new file called A.ALL is in drive C. Next, COPY finds B.OUT in drive A and searches in drive B for B.LST, which does not exist. B.OUT is copied directly to drive C and called B.ALL regardless of whether a match was made in drive B. Finally, COPY appends B:C.LST to A:C.OUT and copies the new file to drive C where it is called C.ALL.

You can combine ASCII and binary files by using the /A and /B options (see Table 6-1). If you specify any of the options in Table 6-1, they stay in effect until the next option is encountered. For example,

```
A>COPY LARRY/B+RAY+TOM+ALIX/A+CYNDI
```

The files LARRY, RAY, and TOM are concatenated with the /B option; the files ALIX and CYNDI are concatenated with the /A option. After concatenation, you will have one file, named Larry.

There are four error messages that you might encounter when using the COPY command. The messages and corresponding circumstances are listed below.

If you use the /V option to verify that all your files are being copied properly and a bad sector is encountered during the copy, you receive the following error message:

Verification error

This error occurs when the sectors are being written to the destination diskette or hard disk. The COPY program automatically aborts and you return to Concurrent.

When you use the /n option and fail to specify a user number between 0 and 15, you receive the following error message:

Invalid user number

If you enter a file specification that does not conform to Concurrent file naming conventions, COPY gives you the following error message and you are returned to Concurrent:

Invalid file specification

You receive a file creation error when you try to copy a file to a diskette that has a full directory. The COPY program aborts and you receive this message:

File creation error

When there is insufficient space on your destination diskette or hard disk to receive a copy of a file, the COPY program aborts, the following error message appears, and you return to Concurrent:

Insufficient disk space

You cannot copy a file that is currently open in a different virtual console. If you attempt to copy such a file, you receive the following error message:

File in use

When this message appears, COPY aborts and you return to Concurrent. Check your other virtual consoles and, whenever possible, close the open files.

Concurrent cannot execute two identical COPY commands from two different virtual consoles simultaneously. In this case, you would receive the preceding error message.

DATE

Syntax

DATE {MM-DD-YY}

Explanation

The DATE command lets you display, set, or change the date on your system. Concurrent uses the date and time you enter with the DATE and TIME commands to keep track of when you create, access, or update your files. You can use Concurrent's SDIR or DIR.EXE to display the date and time when your files were created or updated. If time and date stamping are an important feature of your files, be sure to invoke DATE and its companion command, TIME at the beginning of your terminal session to ensure that the date and time are current. See the Concurrent CP/M User's Guide for more detail on SDIR.

If you intend to use DIR.EXE to display date and time stamps for files stored on a CP/M-formatted diskette or hard disk, you must run Concurrent INITDIR program to initialize the directory area of the diskette or hard disk. Then use the SET command to enable date and time stamping for that diskette or hard disk. See the Concurrent CP/M User's Guide for more information on the INITDIR and SET commands.

You can invoke DATE by entering the command at your system prompt or by including the DATE command in a batch file.

The symbols used in the DATE syntax line are as follows:

- MM is a one- or two-digit number from 1 to 12 for the month.
- DD is a one- or two-digit number from 1 to 31 for the day.
- YY is a two-digit number from 80 to 99 or a four-digit number from 1980 to 2099 for the year.

Concurrent considers any date that falls within the preceding ranges valid. If you type an invalid date, you receive the following message:

```
Invalid date
Enter new date: _
```

If the date is correct and you do not want to alter it, press RETURN. You can use either hyphens, -, or slashes, /, as delimiters when entering the date.

Examples

When you enter the date command, you see the following message at your screen:

```
A>DATE
Current date is day MM-DD-YY
Enter new date: _
```

For example, you enter a DATE command and see that your system records today's date as Wednesday, November 30, 1983. You type the correct date, December 19, 1983 (12/19/83) and press RETURN. The date known to your system is now December 19, 1983. Your screen appears as follows:

```
A>DATE
Current date is Wed 11-30-83
Enter new date: 12-19-83
```

You can also enter the date directly after you type the DATE command. For example,

```
B>DATE 12-31-83
```

The date is set to December 31, 1983.

DIR

Syntax

```
DIR {d:} {/P}{/W}  
DIR filespec
```

Explanation

The DIR or DIRectory command lists all or a specified subset of the files contained in a specified or default drive. The information that DIR displays at your screen includes:

- filename
- filetype
- size (in bytes)
- date (when file was created or modified)
- time (when file was created or modified)
- number of files displayed

To accurately record dates and times, you must use the DATE and TIME commands after you turn on the power to your computer or perform a system reset. See DATE and TIME in this section for a discussion of these commands.

Note: For PC DOS-formatted diskettes, DIR can access files in the root directory only.

You can use the wildcard characters, ? and *, to substitute for filenames and filetypes in a DIR command line.

DIR has two options, /P (Pause) and /W (Wide). When you use the /P option, the display pauses at the end of each screenful of information. When you want the display to continue, press any key.

The /W option provides you with a wide screen display. The wide display lists only filenames, filetypes, and how many files are displayed. Each horizontal line contains five filenames.

Examples

```
A>DIR
```

This example displays all files stored in drive A. The file's size, the date and time of its creation, or the latest modification to it, appear on the screen. Note that creation

or modification times only appear if you have enabled time stamping or disk is a PC DOS disk. Also the amount of free space existing in that drive is displayed.

A>DIR B:

This command displays all files in the specified drive B.

A>DIR B:MYFILE.TYP

The preceding command causes DIR to display the file specification MYFILE.TYP in drive B if the file is located in that drive. If the file that you specified does not exist in the drive, DIR responds with this error message:

File not found

The next example displays all the files in drive A that have a filetype of TYP:

A>DIR *.TYP

The following example lists all the files in drive B that begin with MY and have a filetype beginning with T and ending with P:

A>DIR B:MY*.T?P

The preceding command might find any of the following files: MYFILE.TAP, MYNAME.TMP, MYNAME.TYP.

The inclusion or exclusion of a period in a filespec that you enter in a DIR command line has significance for the way DIR searches for files.

If you omit the period, DIR searches as if a period and an asterisk wildcard immediately followed the last character in the filename you enter. In such a case, DIR searches for files with filenames matching the characters you enter that have any filetype, including a blank filetype.

For example, in response to

A>DIR KAT*

DIR would find the files KAT, KATRINKA.LTR, KATHARS.SLP, KATMAN, and KATS.PJS. In response to the following command:

B>DIR SMUG

DIR would find the files SMUG, SMUG.LER, SMUG.GER, and SMUG.LY.

If you type,

B>DIR *

you see a directory display of every file in drive B.

If you include the period, DIR searches for files that have both filenames and filetypes that match the characters you enter. If a period is the last character in the filespec, DIR searches for matching filenames with a blank filetype.

For example, in response to

A>DIR CHAP*.

DIR would find the files CHAP, CHAPTER, CHAPPY, and CHAPOLA.

In response to

B>DIR SIM??.*

DIR would find SIMTO.DOC, SIM.D, SIMI.DI, and SIMPY.DIP.

ERASE

Syntax

```
ERASE filespec  
DEL filespec
```

Explanation

DEL, short for DELeTE, is an abbreviated form of the ERASE command. The ERASE command removes one or more files from the directory of a diskette or hard disk.

In the process of deleting a file, ERASE automatically reclaims directory and data space used by the file. Wildcard characters are accepted in the filename and filetype fields of the file specification.

Be careful when using wildcard file specifications in ERASE command lines. If you use wildcard characters, use DIR to look at a list of the files that are designated by the wildcard file specification to verify which files you will be erasing.

Examples

```
A>ERASE *.*
```

This command erases every file in the directory in drive A. Before ERASE actually erases all your files in drive A, the following prompt appears:

```
Are you sure (Y/N)?
```

Press y to remove all files, and n to avoid erasing any files.

```
A>ERASE B:MYFILE.TYP
```

The preceding command line deletes a file named MYFILE.TYP in drive B.

```
A>DEL B:*.TYP
```

This command line erases any file with a filetype of TYP from drive B. Be careful when using wildcard characters as in this example. Unlike Concurrent's ERA command, ERASE (or DEL) does not prompt you following a command like this to allow you to reconsider your action.

RENAME

Syntax

```
REN{AME} {d:}oldname{.typ} newname{.typ}
```

Explanation

The RENAME command lets you change the name of a file cataloged in the directory of a diskette or hard disk.

In the preceding syntax line, the filename oldname identifies an existing file. RENAME changes the file named by oldname to the new filename given in the command line.

Examples

```
A>REN MYFILE.TYP FILE1.TXT
```

In this example, MYFILE.TYP is renamed FILE1.TXT in drive A. If you omit the drive specification, Concurrent assumes that you want to rename a file in your default drive.

```
A>REN B:MYFILE.TYP CAM.TXT
```

The preceding example changes the name MYFILE.TYP, in drive B, to CAM.TXT. If your source file did not exist in drive B, RENAME would not find MYFILE.TYP. You would see the following error message:

```
File not found
```

```
A>REN B:CAM.TXT *.TYP
```

This command renames the file CAM.TXT, in drive B, to CAM.TYP.

```
B>REN *.IN *.OUT
```

This command renames all files with an IN filetype to a filetype of OUT. If for some reason you make an error, for example, you try to rename a file to a name that already exists, RENAME terminates and you are returned to Concurrent. All remaining files designated by *.IN are not renamed. You would receive the error message:

```
Duplicate file name or File not found
```

If you fail to specify the destination file specification when entering a RENAME command, you receive the following error message:

Missing file name

TIME

Syntax

TIME {HH:MM:SS.XX}

Explanation

The TIME command lets you display, set, or change the time on your system. Concurrent uses the time and date you enter with the TIME and DATE commands to keep track of when you create, access, or update your files. You must use TIME and its companion command DATE to ensure that the correct date and time is recorded with the file. You can use Concurrent SDIR to display the date and time that your files were created or updated. See the Concurrent CP/M User's Guide for more details on SDIR.

If you want to use DIR.EXE to display time stamps on a CP/M-formatted diskette or hard disk, you must run Concurrent's INITDIR program to initialize the directory area of a diskette or hard disk. Then use the SET command to enable time and date stamping for that diskette or hard disk. See the Concurrent CP/M User's Guide for more information on the INITDIR and SET commands.

Time is expressed in terms of a 24-hour clock. If your system has a status line clock, resetting the time with the TIME command also changes the time displayed on the status line clock.

You can invoke TIME by entering the command at your system prompt or by including a TIME command in a batch file.

The symbols used in the TIME syntax line are as follows:

- HH is a one- or two-digit number from 0 to 23 for the hour.
- MM is a one- or two-digit number from 0 to 59 for the minute.
- SS is a one- or two-digit number from 0 to 59 for the second.
- XX is a one- or two-digit number from 0 to 99 for the hundredths of a second.

Concurrent considers any time valid providing it falls within the preceding ranges. If you do not set all the fields, TIME sets the remaining fields to zero. If you type an invalid time, you receive the following message:

```
Invalid time
Enter new time: _
```

If you do not want to alter the time, press RETURN. Use colons, :, to delimit hours, minutes, and seconds, but use a period, ., to separate seconds from hundredths of a second.

Examples

When you enter the TIME command, you see the following message at your screen:

```
Current time is HH:MM:SS.XX
Enter new time: _
```

For example, after invoking the TIME command, you discover that your system records the time as 8:30 a.m. exactly. You type the correct time, 11:59 p.m. and 34 seconds (23:59:34), and press RETURN.

```
A>TIME
Current time is 08:30:00.00
Enter new time: 23:59:34.00
```

Your system now records the time as 23:59:34:00.

You do not have to include a leading zero for a.m. settings. TIME considers the following command valid:

```
A>TIME
Current time is 08:30:00.00
Enter new time: 9:23:46.87
```

If you set or change fewer than all four fields, working from left to right, TIME sets the remaining fields to zero. For example,

```
A>TIME
Current time is 08:30:15.00
Enter new time: 08:40
```

Your system now records the time as 8:40 a.m. exactly.

You can also enter the time directly after you type the TIME command. For example,

B>TIME 10:12:00

The time is set to 10:12 a.m.

End of Section 6

Batch Processing

Batch processing describes the facilities in Concurrent that execute the commands contained in a file, called a batch file. You enter the filename of the batch file at the system prompt just as you would enter a command. A batch file contains one or more commands that the operating system executes in the sequence in which the commands appear in the batch file. All batch files must have a BAT filetype.

Creating Batch Files

You can create a batch file by using a word-processing or text-editing program. Within the file, you must begin each command on its own line and conform to the correct command line syntax for that command.

Here is an example of the contents of a batch file:

```
COPY *.LTR A:  
ERASE *.BAK  
DIR A:
```

If you name this file BACKUP.BAT, you can enter

```
B>BACKUP
```

at the system prompt and press Return. Concurrent then opens the file named BACKUP.BAT and executes the commands that it contains in the following sequence:

1. All files with the filetype of LTR are copied to drive A with the COPY command.
2. The ERASE command deletes all files with the filetype BAK from the default drive.
3. DIR displays directory entries for drive A.

Batch Subcommands

Concurrent supports three batch subcommands so that existing batch files can function properly. These commands are PAUSE, REM, and SYS.

PAUSE

Syntax

```
PAUSE {remark}
```

Explanation

The PAUSE command lets you interrupt your batch processing and display messages within batch files. Using PAUSE gives you the opportunity to change diskettes or abort processing in the middle of executing a batch file. PAUSE allows you to display a remark that is up to 121 bytes (characters) long.

When you create your batch file, place PAUSE commands anywhere you might want to abort batch processing or change diskettes. When batch processing reaches a PAUSE command, and you do not want to continue, press a CTRL-C. If you want to continue with batch processing, press any other key.

Examples

Suppose you create a batch file where you copy all of the files with a TXT filetype from drive B to drive A. Once your files are in drive A, you usually want to rename the files in drive B to a filetype of OLD. You could insert a PAUSE command with a remark that queries you after your COPY command to continue your batch file or to return to Concurrent. Your batch file would appear as follows:

```
COPY A:*.TXT B:  
PAUSE Rename files or abort?  
RENAME B:*.TXT *.OLD
```

Each time you run this batch file, your TXT files are copied from drive A to drive B. You are then prompted:

```
A>PAUSE Rename files or abort?  
Strike any key when ready..._
```

The first line is your remark, the second line is the PAUSE prompt. If you want your batch file to continue processing, strike any key. If you do not want to continue, press CTRL-C.

REM

Syntax

```
REM {remark}
```

Explanation

Use the REM subcommand to make remarks within your batch file. The remark appears at your screen when your loaded batch file reaches the REM command. You can also use REM without a remark to create blank line spaces within your batch file. You can display a remark that is up to 123 bytes long.

Examples

You can use REM to make comments within your batch file, or you might want to insert important reminders to yourself or other people who use your batch file. Here is an example of a REM command:

```
REM Archive backup files for security.
```

SYS

Syntax

```
SYS d:
```

Explanation

Concurrent recognizes, but does not process, SYS commands. Batch files that contain SYS commands run as they usually do except that the SYS commands are ignored. You do not receive any error messages pertaining to the use of SYS.

Running Batch Files

When you want to execute the commands in a batch file, enter only the filename at the system prompt. You are not required to specify the filetype unless you want to override Concurrent's filetype search sequence. To stop execution of a batch file, press CTRL-C. The system prompts:

Terminate batch job (Y/N)?

If you press Y, the system aborts batch processing and the system prompt reappears. If you press N, only the currently running command is aborted and batch processing continues with the next command in the file.

The last command in a batch file can be the name of another batch file. If this is the case, the second batch file begins execution when the first is complete.

Using Parameters with Batch Files

You can create batch files that include the percent symbol, %, coupled with a number from 0 to 9. These symbols are called replaceable parameters. When you execute the batch file, you can include data such as file specifications or drive specifiers in the command line. This data substitutes for the parameters that you entered in the batch file.

For example, you can use a word-processing or text-editing program to create a file called MYFILE.BAT which consists of the following commands:

```
COPY %1 %2
ERASE *.*%3
DIR %2
```

When you enter MYFILE at the system prompt, you can include filespecs or drive specifiers that substitute for the parameters in MYFILE.BAT. Parameters are replaced in the sequence that items appear in the batch command line. That is, the first item replaces every occurrence of %1, the second item replaces every occurrence of %2, and so forth. For example, if you enter the command line:

```
A>MYFILE *.LTR A: BAK
```

Concurrent's batch-processing facilities replace %1 with *.LTR. Both occurrences of %2 are replaced by A:. The filetype BAK replaces %3.

The filespec of the batch file substitutes for %0.

End of Section 7

Special Commands that Signal Concurrent

There are four commands that signal Concurrent that you are going to run certain kinds of PC DOS programs. These commands are MEM, SUSPEND, SYSDISK, and 8087. This section explains the operation of these commands and the reasons for using them.

MEM

Syntax

```
MEM {= n}
```

Explanation

The MEM utility lets you display and define the amount of memory space allocated to PC DOS programs running in a given virtual console. This allocation applies only to PC DOS programs with COM or EXE filetypes. Invoke MEM from the virtual console in which you intend to change the memory allocation.

MEM can be set to any integer from 0 to 999, with n representing the number of kilobytes (K). After invoking MEM and specifying an amount of memory, your definition stays in effect until you reinvoke MEM, or begin a new terminal session. The default memory allocation for PC DOS programs in any virtual console is 128K.

When Do You Use MEM?

You will use MEM when you are running PC DOS programs (with a filetype of COM) and you need to change the amount of memory space allocated to the program for execution. Refer to the documentation supplied with the PC DOS application program to find how much memory space the program needs. Then use MEM to define the correct memory allocation size. If the documentation does not specify how much memory space the application program needs, experiment with different values for MEM.

If you receive a message telling you that there is insufficient memory to load a program, then the amount of memory allocated for PC DOS programs does not exist in your system. You can then use MEM to define smaller allocation sizes and test if the application will load and run in the smaller memory region.

Note: Allocating an excessive amount of memory in a virtual console limits the memory space available to programs running in other virtual consoles.

Examples

To find the amount of memory space allocated in a given virtual console, enter the following command at the system prompt:

```
A>MEM
```

```
Memory Allocation Size = 128K
```

You see a message similar to the one above depending on what the Memory Allocation Size actually is for your virtual console. Remember that 128K is the default setting.

If you have a PC DOS program that requires 192K in memory, enter the following setting:

```
B>MEM = 192
```

```
Memory Allocation Size = 192K
```

Concurrent responds with the preceding message indicating the new memory allocation.

SUSPEND

Syntax

SUSPEND=ON|OFF

Explanation

The SUSPEND utility causes Concurrent to suspend the operation of any PC DOS programs while those programs occupy a background virtual console. SUSPEND works on a per-virtual console basis. That is, SUSPEND is invoked from each virtual console to suspend the PC DOS program when it is in the background.

SUSPEND has two settings, ON and OFF. The default setting for SUSPEND is ON. The effect of the SUSPEND=ON command lasts for the remainder of your terminal session. You can invoke SUSPEND with the OFF option to end the effect of a previous SUSPEND=ON command.

Concurrent recognizes PC DOS programs and will automatically set SUSPEND to ON when that program is put into the background. If a program in a background console continues to write information to the terminal screen, check the setting of the SUSPEND command.

When Do You Use SUSPEND?

Most PC DOS programs do not follow the PC DOS conventions for sending characters to the video display. When these programs output characters while occupying a background virtual console, they can "write through" the foreground virtual console to display characters on your screen. This is why Concurrent automatically invokes SUSPEND when it recognizes a PC DOS program.

SUSPEND suspends the operation of any subsequently-run PC DOS programs while they occupy a background virtual console. You may want to experiment with your PC DOS application to see whether it does indeed "write through" by setting SUSPEND to OFF. This will not affect the program in the foreground console. SUSPEND has no effect on CP/M-based programs.

Examples

To signal Concurrent that you want the operating system to suspend the operation of a PC DOS program running in a background virtual console, enter this command:

```
B>SUSPEND=ON
```

Enter the following command to cancel a previous SUSPEND=ON command:

```
B>SUSPEND=OFF
```

If you mistype the SUSPEND command setting as in the following example,

```
A>SUSPEND=ONN
```

you would receive this error message:

```
Invalid Suspend State, IGNORED
```

SYSDISK

Syntax

```
SYSDISK  
SYSDISK D:
```

Explanation

The SYSDISK command tells you which drive is the current system drive and lets you specify a new system drive. The system drive is the drive where Concurrent looks for program files after searching the default drive.

The system drive is the drive whose letter appears in the system prompt when you first start Concurrent. There are two ways to use SYSDISK to change the assignment of the system drive to any other drive in your system. One way is to type SYSDISK at the system prompt, followed by the new drive specification, such as B:.

The second way is to include a SYSDISK command in a start-up file that assigns the system drive to the drive of your choice each time you start Concurrent. Refer to "Start-up Files" in the Concurrent CP/M User's Guide for an explanation on how to use start-up files.

When Do You Use SYSDISK?

It is convenient to store all of your program files, both applications and Concurrent commands and utilities, in user 0 of your system drive. This is because, regardless of which drive and/or user number you are logged into, if a program file exists with a SYS attribute in user 0 of the system drive, Concurrent finds it and runs it.

For information on how to assign the SYS attribute to files, see the Concurrent CP/M User's Guide.

Examples

Use the SYSDISK command to tell you which is the current system drive by typing the following command:

A>SYSDISK

System disk is A:

In the following example, SYSDISK is used to assign the system drive to drive B.

A>SYSDISK B:

System disk is B:

There are three error messages you can receive from SYSDISK. If you enter improper syntax, for example

A>SYSDISK B;

you receive the following error message:

Invalid drive code, use following syntax SYSDISK A:

If files are open on the current system disk and you try to designate another drive as the system drive, you receive this error message:

Unable to set new system disk because of open files

Specifying a drive that does not exist on your system, gets you the following error message:

Specified device does not exist

8087

Syntax

8087=ON|OFF

Explanation

The 8087 command signals Concurrent that any PC DOS programs you run following the invocation of the 8087 command use the Intel® 8087 co-processor chip. Like SUSPEND, 8087 works on a per-virtual console basis. You must invoke 8087 from each virtual console from which you intend to load a PC DOS program that uses the 8087 chip.

8087 has two settings: ON and OFF. The default setting is OFF.

To turn 8087 ON, type this command:

```
B>8087=ON
```

The ON setting lasts for the remainder of your terminal session. You can invoke 8087 with the OFF option to end the effect of a previous 8087=ON command.

When Do You Use 8087?

If you have PC DOS programs that use the 8087 chip and have an 8087 chip installed in your computer, you must use the 8087 command to enable your program to access the 8087 chip. A program's documentation makes it clear to you whether that program uses the 8087 chip.

Examples

To signal Concurrent that you intend to run a PC DOS program that uses an installed 8087 chip, enter

```
A>8087=ON
```

from the virtual console in which you intend to run the PC DOS program.

To cancel the effect of a previous 8087=ON command, enter the following:

```
A>8087=OFF
```

If you should mistype the 8087 command setting, for example:

A>8087=OON

you would receive this error message:

Invalid 8087 State, IGNORED

If you try to enable the 8087 command and your computer does not have a properly installed 8087 chip, you receive the following error message:

8087 not installed

End of Section 8

Index

- % parameter, 7-4
- * character, 6-10
- ? character, 6-10
- /A option, 6-3, 6-5
- /B option, 6-3, 6-5
- /n option, 6-3, 6-6
- /P option, 6-10
- /V option, 6-3, 6-6
- /W option, 6-10
- 8087 microprocessor, 5-11, 8-8
- 8087, 5-11, 8-8
 - default setting, 8-8
 - error messages, 8-9
 - when to use, 8-8
 - with virtual consoles, 8-8
- 8088 microprocessor, 1-1
- A**
 - active console, 3-3
 - application programs, 2-2,
 - 3-1, 5-8
 - ASCII, 6-2, 6-3, 6-5
- B**
 - BAT filetype, 5-6, 7-1
 - background consoles, 3-2, 5-8
 - batch files
 - creating, 7-1
 - batch processing, 7-1
 - filetype, 7-1
 - interrupting, 7-2
 - parameters, 7-4
 - PAUSE command, 7-2
 - REM command, 7-3
 - running files, 7-4
 - subcommands, 7-1
 - SYS command, 7-3
 - binary files, 6-5
 - booting, 3-2
- C**
 - CCPM.SYS, 5-12
 - CMD filetype, 5-6
 - COM file, 5-9
 - COM filetype, 5-6, 8-2
 - command files, 5-3, 5-4, 5-12
 - commands
 - see also individual commands
 - 8087, 5-11, 8-8
 - COPY, 6-1
 - DATE, 6-1, 6-8
 - DEL, 6-13
 - DIR, 5-7, 6-1, 6-10
 - DIR.EXE, 5-7, 6-8, 6-16
 - ERA, 6-13
 - ERASE, 6-1, 6-13
 - external, 5-3
 - INITDIR, 6-8, 6-16
 - internal, 5-3
 - MEM, 5-9, 8-2
 - PAUSE, 7-2
 - PC DOS supported, 5-3, 6-1
 - REM, 7-3
 - RENAME, 6-1, 6-14
 - SDIR, 5-7, 6-16
 - SET, 6-1, 6-8, 6-16
 - SUSPEND, 5-7, 5-8, 8-4
 - SYS, 5-5, 7-3, 8-6
 - SYSDISK, 5-5, 8-6
 - TIME, 6-1, 6-16
 - CON:, 6-2
 - concatenate files, 6-3,
 - 6-4, 6-5
 - consoles, 3-2, 5-1
 - active, 3-2
 - background, 3-2, 5-8
 - number, 3-4
 - CONTROL key, 3-3
 - COPY, 2-2, 5-3, 6-1, 6-2
 - combining files, 6-5
 - creating a file, 6-2
 - default drive, 6-4
 - error messages, 6-6
 - options, 6-3
 - renaming a file, 6-2
 - using, 5-11
 - wildcards, 6-5
 - CTRL C, 7-2, 7-4
 - CTRL key, 3-3
 - CTRL Z, 6-2

D

- DATE, 5-3, 6-1, 6-8
 - change, 6-8
 - delimiters, 6-9
 - display, 6-8
 - set, 6-8
 - stamping, 6-8, 6-16
- default drive, 3-2, 5-4, 5-5, 5-6, 6-4
 - changing, 5-4
 - virtual consoles, 5-4
- DEL, 5-3, 6-13
- demo disk, 4-1
- DIR, 5-3, 5-7, 6-1, 6-10
 - error messages, 6-11
 - filetypes, 6-11
 - options, 6-10
 - root directory, 6-10
 - wildcard characters, 6-10, 6-11
- DIR.EXE, 5-7, 6-8, 6-16
- directory, 5-7
 - displaying, 6-10
 - initialize, 6-8
 - root, 6-10
 - sub, 5-7
- disk drives, 3-1
 - see also drive
- disk
 - see hard disk or floppy disk
- disks
 - formatting, 5-3
- drive
 - default, 3-2, 5-4, 5-6, 6-4
 - file search order, 5-6
 - system, 5-4, 5-6, 8-6

E

- end-of-file character, 6-2
- ERA, 6-13
- ERASE, 5-3, 6-1, 6-13
 - how to use, 6-13
 - wildcards, 6-13
- error messages
 - 8087, 8-9
 - COPY, 6-6
 - DIR, 6-11
 - RENAME, 6-14
 - SUSPEND, 8-5
 - SYSDISK, 8-7
- EXE file, 5-9
- EXE filetype, 5-6, 6-1, 8-2

- external commands
 - see commands

F

- file extensions, 5-6
- file search paths, 5-5
- file search pattern
 - override, 5-6
- filename, 5-6
- files
 - accessing, 5-6, 5-7
 - appending, 6-4
 - ASCII, 6-2, 6-3, 6-5
 - batch, 5-6, 7-1
 - binary, 6-5
 - combining, 6-5
 - command, 5-3, 5-6
 - concatenation, 6-3, 6-4, 6-5
 - copy, 5-3
 - directory, 5-7
 - erase, 5-3, 6-13
 - executable, 5-6
 - open, 6-7
 - rename, 5-3
 - renaming, 6-14
 - security, 5-12
- filetype search order, 5-6
- filetypes, 5-6, 5-9
 - BAT, 5-6, 7-1
 - COM, 5-6, 8-2
 - CMD, 5-6
 - EXE, 5-6, 6-1, 8-2
- floppy disk, 3-1
 - stamping, 6-8
 - storing commands on, 5-5
- floppy-disk system, 3-1
- formatting disks, 5-3

H

- hard disk, 3-2, 5-11
 - accessing, 5-11
 - copying to, 5-12
 - partitioning, 5-12
 - stamping, 6-8
 - storing commands on, 5-5
 - using, 5-11
 - with user numbers, 5-2
- hard-disk system, 3-2, 5-11
- HELP, 4-2
- hexadecimal, 6-3

I

- INITDIR, 6-8, 6-16
- internal commands
 - see commands

L

- logical devices, 6-2
 - CON:, 6-2
 - LPT1:, 6-2
 - NUL:, 6-2
 - PRN:, 6-2
- LPT1:, 6-2

M

- MEM, 2-2, 5-9, 8-2
 - default setting, 5-9
 - filetypes, 5-9
 - when to use, 8-2
- memory, 2-2
 - see also MEM
 - allocating, 5-9, 5-10, 8-2
 - displaying, 8-2
 - minimum, 5-11
 - requirements, 1-1
- microprocessors
 - 8087, 5-11, 8-8
 - 8088, 1-1

N

- NUL:, 6-2

O

- options
 - COPY, 6-3
 - DIR, 6-10
 - /A, 6-3, 6-5
 - /B, 6-3, 6-5
 - /n, 6-3, 6-6
 - /P, 6-10
 - /V, 6-3, 6-6
 - /W, 6-10
- output
 - to screen, 5-7

P

- password protection, 5-12
- PAUSE, 7-2
- PC DOS commands, 5-3, 6-1
 - see also individual commands

PIP, 2-2

- using, 5-11

PRN:, 6-2

programs, 2-2, 3-1, 5-8

prompt, 3-2, 5-4

R

- REM, 7-3
- REN, 5-3
- RENAME, 6-1, 6-14
 - error messages, 6-14
 - using wildcards, 6-14
- root directory, 5-7, 6-10

S

- SDIR, 5-7, 6-8, 6-16
- search order for files,
 - 5-4, 5-5
 - override, 5-6
- SET, 6-1, 6-8, 6-16
- stamping, 6-8, 6-16
- start-up files, 8-6
- status line, 3-3,
 - 5-1, 5-2
- subdirectories, 5-7
- SUSPEND, 5-8, 8-4
 - default setting, 8-4
 - error messages, 8-5
 - when to use, 5-7, 8-4
- SYS, 5-5, 7-3, 8-6
- SYSDISK, 5-5, 8-6
 - error messages, 8-7
 - when to use, 8-6
 - with start-up file, 8-6
- system drive, 5-4,
 - 5-6, 8-6
 - changing, 8-6
 - setting, 5-5
- system prompt, 3-2, 5-4
- system
 - requirements, 1-1

T

- TIME, 5-3, 6-1, 6-8, 6-16
 - change, 6-16
 - delimiters, 6-17
 - display, 6-16
 - set, 6-16
 - stamping, 6-8, 6-16
 - with batch files,
 - 6-16
- tutor command, 4-1

- tutorial, 4-1
 - exiting from, 4-1
 - lessons in, 4-2
 - starting, 4-1

U

- User 0, 5-3, 5-5, 6-1, 8-6
- user numbers, 5-2, 5-5,
 - 6-3, 6-6
- and PC DOS, 5-3

V

- virtual consoles, 1-1, 3-2, 5-1
 - active 3-2
 - and SUSPEND, 5-8, 8-4
 - background, 3-2, 5-8
 - default drive, 5-4
 - example, 3-3
 - number, 3-4
 - using, 3-4

W

- wildcard characters, 6-10
 - with DIR, 6-10, 6-11
 - with COPY, 6-2
 - with ERA, 6-13
 - with ERASE, 6-13
 - with RENAME, 6-14

NOTES

Reader Comment Card

We welcome your comments and suggestions. They help us provide you with better product documentation.

Date _____

1. What sections of this manual are especially helpful?

2. What suggestions do you have for improving this manual? What information is missing or incomplete? Where are examples needed?

3. Did you find errors in this manual? (Specify section and page number.)

Concurrent™ Operating System User's Reference Guide with Tutorial
First Edition: May 1984
1035-2063-001

COMMENTS AND SUGGESTIONS BECOME THE PROPERTY OF DIGITAL RESEARCH.

From: _____



NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES



BUSINESS REPLY MAIL
FIRST CLASS / PERMIT NO. 182 / PACIFIC GROVE, CA

POSTAGE WILL BE PAID BY ADDRESSEE

 **DIGITAL RESEARCH®**

Attn: Publications Production

P.O. BOX 579

PACIFIC GROVE, CA 93950-9987

