

# Upward Migration

## Part 2: A Comparison of CP/M-86 and MS-DOS

*An end user and a systems programmer examine the two operating systems vying for dominance in the 16-bit arena.*

---

Roger Taylor and Phil Lemmons  
c/o BYTE Publications Inc.  
POB 372  
Hancock, NH 03449

---

Last month, we looked at programs that translate 8-bit CP/M-80 software to 16-bit MS-DOS or CP/M-86 software. In this part, we look at what the translated software will find when it reaches the new world of MS-DOS and CP/M-86. We first examine the two operating systems from a non-technical user's perspective and then turn to a close technical look at how they make basic functions available to programmers. Along the way, we will report the results of some benchmarks.

We have also invited other individuals to express their opinions about MS-DOS and CP/M-86. These opinions appear in text boxes accompanying this article and represent a variety of perspectives. One expresses the view of a systems integrator; one states the view of an assembly-language programmer at a large applications and languages software house; one comes from an applications programmer working in high-level languages; one comes from a software house that produced a BIOS (basic input/output system) for CP/M-86 on the IBM Personal Computer, as well

as applications programs for both CP/M-86 and MS-DOS; and one was written by the head of a company that manufactures 8086-based S-100 systems and processor boards.

If any pattern of opinion seems to be emerging, it is that MS-DOS is a better and faster single-user, single-tasking operating system for nontechnical users (consumers). CP/M-86 offers superior memory management; development into a multitasking, single-user operating system; and an easier upgrade path to a multi-user operating system. The issues, however, are complex, and many dissenting voices are heard. To complicate matters further, vendors of both operating systems are promising major changes this year. You will have to make the decision in light of your exact needs in an operating system.

### **The Nontechnical User's Perspective**

Most microcomputer users spend their time running off-the-shelf applications programs and, to a lesser degree, writing programs in high-

level languages like BASIC and Pascal. For these users, comparing Digital Research's CP/M-86 and Microsoft's MS-DOS (sold by IBM as PC-DOS and by Lifeboat Associates as SB-86) is a study in trade-offs. We will first examine some of the trade-offs in the current versions of MS-DOS and CP/M-86. Then, we will describe what additional trade-offs may appear in the enhanced versions promised for later this year.

### **Speed and Space**

The most obvious general advantages of MS-DOS are its greater speed in disk input/output, greater efficiency in use of disk space, and its ability to recover from errors.

MS-DOS is faster primarily because it buffers more data in each gulp and because it keeps the File Allocation Table in memory rather than on disk. CP/M-86 keeps information about file locations stored on the disk, and buffers data in small mouthfuls. Tables 1, 2, and 3 give some timings of MS-DOS and CP/M-86 on the same hardware. It should be noted, however, that the

Benchmark	System X	System X	System X
	CP/M-86 1.0 BASIC-86 Rev. 5.21	CP/M-86 1.0 BASIC-86 Rev. 5.22	Preliminary MS-DOS BASIC-86 Rev. 5.21
Empty do-loop	7.2	7.3	7.8
Division	25.0	20.6	21.8
Subroutine jump	15.5	15.4	16.9
MID\$ (substring)	23.6	23.6	24.6
Prime number program	199.2	189.9	197.0
Disk write program (64K-byte file)	60.8	60.8	50.3
Disk read program (64K-byte file)	20.6	20.3	21.3

**Table 1:** Timings of benchmark programs run in Microsoft BASIC-86 under CP/M-86 and MS-DOS. The three sets of timings differ on the computational benchmarks (the first five) because the BASICs themselves are different. BASIC-86 comes in two different versions for CP/M-86, as well as a version for MS-DOS. MS-DOS's superior speed in the disk-write benchmark is remarkable because the version run was a preliminary version that had not yet been optimized for disk input/output on system X.

	System X CP/M-86 1.0	System X Preliminary MS-DOS
Load BASIC-86 30K bytes	5.7	2.7

**Table 2:** Timings for loading BASIC-86 under CP/M-86 and MS-DOS. Again, note that this preliminary version of MS-DOS had not yet been optimized for disk input/output on system X.

	IBM-PC Compuview CP/M-86 PIP	IBM-PC MS-DOS 1.1 COPY
File copy: Copy three files 13K, 13K, 6K	36.9	19.3
Load 30K file using VEDIT	10.1	18.8

**Table 3:** Timings for file-copying and file-loading under CP/M-86 and MS-DOS on the IBM Personal Computer. MS-DOS's COPY command is faster, partly because it resides in system memory rather than on disk, and despite the fact that the BIOS in the Compuview CP/M-86 seems quite fast. The speed of the Compuview BIOS is reflected in the time required to load Compuview's editing program VEDIT with a 30K-byte text file. The CP/M-86 version of VEDIT is actually faster than the MS-DOS version. (BASIC-86 is unavailable for use with CP/M-86 on the IBM Personal Computer.)

differences in speed are proportionally less on large operations because CP/M-86 has to find a file only once, at the beginning. Different versions of CP/M-86 also vary greatly in speed. Hardware manufacturers write a CP/M-86 BIOS for their own systems, which vary in performance. Finally, on systems using hard disks, disk input/output under either MS-

DOS or CP/M-86 is so fast that any differences are negligible. The fact remains that MS-DOS is significantly faster on floppy-disk-based systems, as the benchmarks in table 1 show.

MS-DOS uses disk space more efficiently because it can have files ranging in size from tiny to huge. CP/M-86 allocates disk space in blocks of 2K bytes even when the ac-

tual data in the file is only a few bytes—like letting a mouse sleep in a separate queen-size bed. With larger data files, of course, CP/M-86's methods of allocation are less wasteful. But if your system's disk space is limited, wasting 2K bytes to store a few bytes can be important.

Error recovery in an operating system functions like the shoulder on a mountain road: in its absence, one wrong move can plunge you into the abyss. Error recovery is especially crucial to new and nontechnical computer users. MS-DOS provides error recovery as part of the disk operating system; as for CP/M-86, there is no provision for error recovery in the disk operating system, although there is nothing to stop a hardware manufacturer from writing error recovery into its own BIOS for CP/M-86. Manufacturers don't always do so.

With CP/M-86 (unless the manufacturer does add error recovery), if you try to read a disk in an empty drive, or forget to turn on the printer before trying to print a file, the system either "hangs" (no message at all) or presents a farewell message explaining why the system is shutting down. This is about as useful as having a note pinned to your chest to inform you that you are dead. If you're trying to save new data on disk, the data is lost. If the problem is a bad sector on the disk, you can't swap disks and try again. You have to reload the operating system and do again whatever work was wasted. MS-DOS responds to errors by offering you a chance to salvage the situation. A typical message is:

```
Data error reading drive B
Abort, Retry, Ignore?
```

If the problem is a bad disk, you can insert a new disk in B and retry. If the printer is off, you can turn it on and retry. You can also abort the operation and start over without reloading the operating system.

### Help

CP/M-86 comes with a large HELP file. Whenever you are puzzled about what to do, if you type HELP and the



# Limited High-Level Languages: Caveats for MS-DOS Software Developers

Camilo Wilson  
Lifetree Software Inc.  
117 Webster St.  
Monterey, CA 93940

[Mr. Wilson is the author of *Volkswriter*, a word processor written in Pascal for the IBM Personal Computer.]

While the MS-DOS operating-system environment has many useful features for program development, and indeed the first crop of IBM software is primarily designed to satisfy the needs of programmers, this software has severe shortcomings in large applications.

The drawbacks revolve around the usage of the computer's address space and the lack of documentation of the internal workings of MS-DOS, BASIC, and Pascal. In BASIC, your program cannot be larger than 64K bytes, nor can your variables take up more than 64K bytes of room. Although the DEF SEG statement lets you place variables in other memory segments, you are left with the problem of where they should be, and you have to become intimately familiar with the workings of machine language and BASIC before you can use this feature. On the other hand, the CHAIN statement lets you handle larger programs so that the limitation on program size is not severe.

But BASIC is not a language for writing complex programs, both because of its inefficiency and its cumbersome nature. BASIC is intended as a

language for solving small problems simply and quickly, not for systems implementation. For that there is Pascal, an excellent language, whose set of IBM extensions makes it quite powerful and easy to use.

## Drawbacks of Pascal

Unfortunately, Pascal suffers from the same problems. A Pascal program cannot be larger than 64K bytes. About one-third of this space is taken up by library routines that you cannot use as needed (if you need one routine out of a family, you must carry the whole family in your program). Similarly, MS-DOS takes 12K bytes regardless of how much of it you use. Most important, unlike in BASIC, you cannot chain programs so that they overlay each other. This alone makes a wonderful language almost useless for large applications.

In Pascal, you cannot have more than 64K bytes of combined variable space, stack space, and heap (dynamic storage allocation) space. However, like BASIC's DEF SEG, the ADS data type allows you to address any memory location, even outside the 64K-byte data segment. But once again, you have to do your own storage bookkeeping within the extra segments.

Where does this leave the software developer? Without extensive knowledge of the 8086 architecture and a great deal of snooping at the actual contents of memory, it is impossible to write a large program such as a word processor, a spread sheet, a database handler, or even a complex vertical application. While larger software companies may have the necessary talents in-house, many smaller companies do not. The obvious result: no software for the Personal Computer owner.

## Necessary Steps

What can be done? Lots. IBM should release a linkage editor, compiler, assembler, and loader that allow overlays; if not, CP/M-86 and UCSD Pascal will become the systems of choice for the larger applications where the Personal Computer can shine. Meanwhile, IBM should release comprehensive technical documentation for DOS, BASIC, and Pascal.

IBM has done an excellent job so far in supporting the software author, but key pieces are missing: overlays and, most important, comprehensive technical documentation. Mainframe users are provided with program logic manuals and extensive documentation. Why not Personal Computer users too?

name of a command, CP/M-86 will provide instructions for the command. If you can set aside 22K bytes of disk space for the HELP file, CP/M-86 is much friendlier than CP/M-80 used to be. Moreover, the new CP/M-86 user's guides are much better than their predecessors. These include a *CP/M-86 Operating System User's Guide*, *CP/M-86 Operating System Programmer's Guide*, and *CP/M-86 Operating System System Guide*. The user's guide and the HELP command make learning CP/M-86 a less forbidding challenge for novices.

MS-DOS's provision for error recovery makes it safe for novices to

learn by doing. It can therefore be argued that the Microsoft operating system doesn't need online HELP. But both error recovery and online HELP are necessary if a system is to be *really* friendly.

## Reconfiguration

This is where CP/M-86 shines and MS-DOS emphatically does not. CP/M-86's STAT program provides a way to change device assignments. If, for example, you want to use a serial printer as the LST (list) device or printer in your system, you can use STAT to make a change in the oper-

ating system resident in the computer's memory. You can also use CP/M-86's DDT-86 (dynamic debugging tool) to change device assignments lastingly in the operating system on the disk. This changes CP/M's famous IOBYTE—the byte in which the operating system keeps device assignments. Neither of these ways of changing device assignments is clear to novices, but you can often get computer dealers to change IOBYTE for you or give you written instructions for making the necessary changes.

If you have an MS-DOS system and buy some new hardware, you



may have to be a skilled assembly-language programmer to reconfigure your operating system to use the hardware. In this respect, MS-DOS is like a Ferrari that requires a mechanical engineer to change a flat tire.

Microsoft prides itself on MS-DOS's device-independent input/output. Promotional literature for MS-DOS explains the happy result: "There is no need to rewrite programs when a new device is added to the system." The concept of device-independence is attractive. The operating system can treat devices just as if they were files. But this is small solace to the many IBM Personal Computer owners who acquire a serial printer they would like to use. The only way for people with little technical knowledge to use a serial printer with this computer is to write a BASIC program that asks for a file specification. The file specification in the PC-DOS version of BASIC-86 can include such things as COM1 (the serial port), the data-transmission rate, etc. While this adaptability in file specification is commendable, who wants to load BASIC, run a BASIC program, and enter a long file specification just to print a text file? Or to hit the IBM's PRINTSCREEN key only to be told that the printer device is unavailable?

The glaring omission of a handy way for people with little technical knowledge to reconfigure MS-DOS has forced many to beg their dealers for software fixes or to spend hundreds of dollars for a redundant parallel printer. The dealers queried were either supporting a single model of serial printer or, more often, none at all. A call to Microsoft a few months ago brought the advice, "Don't expect support for serial printers on the Personal Computer until IBM is selling a serial printer." There is hope, however. Microsoft president Bill Gates promised Personal Computer followers at the West Coast Computer Faire that a future version of MS-DOS will include support for serial printers.

Now for comparisons of utilities that perform some of the most com-

mon operations in each operating system.

### **COPY Versus PIP**

MS-DOS keeps its file COPY program (and its File Allocation Table, as noted earlier) resident in memory. Before actually starting to copy a file, CP/M-86 must read from the disk both its Peripheral Interchange Program (PIP) and the location of the file named for copying. The result is that COPY transfers a file much faster than PIP does.

---

## **Both Microsoft and Digital Research are moving to correct deficiencies.**

---

In defense of PIP, it does more than just copy files. Placing single-character switches after the PIP command causes it to do such things as echo transferred data to the console (video display), filter formfeeds out of the file, check data for proper Intel hexadecimal format, translate uppercase characters to lowercase or vice versa, add line numbers to the destination file, set page length if the file is going to a printer, quit or start copying after finding a certain string of characters in the file, expand tabs in the file by adding spaces, verify that the file has been copied correctly, and mask the eighth bit in each byte transferred. PIP is larger because it does more. But most of PIP's extra functions have less appeal to nontechnical users than to assembly-language programmers. For most of us, the main thing is that COPY runs like lightning.

### **CHKDSK and DIR Versus STAT and DIR**

STAT, the CP/M-86 utility that reports statistics about the contents and status of a disk, also lets you change device assignments (as noted above). STAT will give detailed information about each file on a disk. The MS-DOS program CHKDSK (for Check Disk) reports only a summary of the status of a disk and the status of system RAM (random-access read/

write memory), i.e., the number of bytes free. Both STAT and CHKDSK reside on disk, and their speed of operation is similar.

Both MS-DOS and CP/M-86 have a directory command, DIR, residing in system memory. In CP/M-86, the DIR command just lists the names of files. In MS-DOS, DIR tells you the exact size of each file and the date that it was last written to disk. The DIR command is fast in both systems.

### **The Technical User's Perspective**

We will now make some observations about a few MS-DOS features that differ from CP/M-86. Then we will make a close comparison of the way the same basic operating-system functions are made available to the programmer under CP/M-80, CP/M-86, and MS-DOS. Finally, we will show the results of some simple benchmarks of CP/M-86, MS-DOS, and EM-86 running under MS-DOS, which lets MS-DOS emulate CP/M-86.

### **DEBUG Versus DDT-86**

With one important exception, DEBUG is the rough equivalent of DDT-86. While DDT-86 has an Assemble command, DEBUG does not. As a result, MS-DOS gives programmers no quick way to test ideas or hardware or to install patches by writing a small assembler program in memory.

### **Error Trapping**

MS-DOS permits an applications program to trap disk errors so as to invoke appropriate routines to keep the program running or to shut down gracefully. Since CP/M-86 doesn't permit such error trapping, premature exits from applications programs do happen.

### **The FORMAT Programs**

Under MS-DOS, you must specify when you format a disk that you want to create a "bootable" MS-DOS disk. You can't put a bootable system on an existing data disk. The MS-DOS approach permits a data disk to have more room than a bootable



# The Systems Integrator's View of MS-DOS and CP/M-86

Jerry Houston  
Vice-President  
G&G Engineering  
13708 Doolittle Dr.  
San Leandro, CA 94577

To a systems integration house, it is unfortunate when two respectable but incompatible operating systems contend for acceptance on the same equipment. A company that runs trains would no doubt prefer that tracks be standardized, and a company that makes cars would surely prefer that a single fuel type be universally available. Amtrak and Chrysler may have their problems, but it is not the dreaded compatibility problem. That problem is reserved for systems integrators trying to decide on a 16-bit operating system to run Intel's 8088 or 8086 processors.

The two popular operating systems for running 8088/8086-based microcomputers, CP/M-86 by Digital Research and MS-DOS by Microsoft, pose a special problem because they represent two different offshoots of a single 8-bit lineage.

The great problem is that software that once ran on CP/M-80 2.2 (for example, CBASIC by Compiler Systems and Microsoft BASIC by Microsoft) is being made available in 16-bit versions compatible either with CP/M-86 (CBASIC) or MS-DOS (Microsoft BASIC), but not with both. This means that the enormous libraries of source code written in both languages will no longer run under the same DOS, leaving users who wish to upgrade to 16 bits with three choices:

1. Choose one DOS and abandon all that is unique to the other.
2. Purchase and become familiar with two operating systems.
3. Choose one DOS and purchase and become familiar with special utility software designed to allow programs from the other camp to run in a hostile environment.

For those who do not want to put up with the headaches involved with choices 2 and 3, I offer some comparisons between CP/M-86 and MS-DOS.

## Four Comparisons

Comparisons will be arranged as four sets of advantages offered by each DOS:

- minor BDOS advantages
- major BDOS advantages
- other advantages
- major advantage

BDOS advantages are of importance only to those who wish to write assembly-level software (something more common to systems integrators and software developers than to end users). "Other Advantages" are still technical, but may affect an end-user's decision to commit to one operating system or the other. "Major Advantage" is the killer advantage that makes systems integrators cringe when forced to make a decision between CP/M-86 or MS-DOS.

## Minor BDOS Advantages of CP/M-86

CP/M-86, like CP/M-80 2.2, includes the IOBYTE function, which, among other things, allows users to toggle between output devices, e.g., between a fast line printer and a slower, letter-quality printer.

CP/M-86 files can be write-protected through the BDOS (as opposed to physically modifying the disk).

CP/M-86 allows the user to set file attributes, such as R/O (Read Only), SYS, DIR, and R/W (Read/Write).

The GET ADDRESS (DISK PARMS) function, which supplies disk-drive characteristics, is useful for hardware-dependent code.

CP/M-86 uses uniform calling conventions—all BDOS functions are channeled through interrupt 224.

## Major BDOS Advantages of CP/M-86

CP/M-86 allows directory partitioning, so that user numbers can be assigned to specific files.

Function 0 allows programs to exit (return control to CP/M), yet remain resident in RAM to intercept function calls, etc. This allows extensions to the operating system.

Memory-management handling under CP/M-86 is far superior to that of MS-DOS. Under CP/M-86, programs can dynamically set up code and data space, and can allocate code and data space for other programs, which

in turn can allocate space for other programs, etc.

## Other Advantages of CP/M-86

The debugger/monitor included with CP/M-86 (DDT-86) has an online assembler. DEBUG, the debugger/monitor that comes with MS-DOS, curiously lacks this feature.

High-level languages supported under CP/M-86 include several important commercial compilers, including:

CBASIC 86—Compiler Systems (now Digital Research)  
Pascal/MT—MT Microsystems (now Digital Research)  
Pascal M86—Sorcim  
ANSI COBOL 74—MBP  
ANSI FORTRAN 77—National Semiconductor

Due to their use of CP/M-86's memory-management capability, it is not guaranteed that these programs will run under MS-DOS even under an emulator.

## Biggest Advantage of CP/M-86

MP/M-86 provides a direct path for expansion into a multi-user environment, while CP/M-80 2.2 provides a broad base of file-compatible data and source code created on 8-bit systems. MP/M 8-16, developed by G&G Engineering for Compupro, is an example of the ease of the transition from 8 to 16 bits, and from single- to multi-user, that is made possible by remaining within the Digital Research family of operating systems.

## Minor BDOS Advantages of MS-DOS

The ability to parse file names is a great time-saver for software developers, and is a good example of the friendliness of the MS-DOS system.

## Major BDOS Advantages of MS-DOS

The ability to time-stamp files is available only under multi-user flavors of CP/M. MS-DOS allows time stamping—a tremendous aid to file maintenance, which is by no means a task unique to multi-user systems.

Continued on page 340



### Other Advantages of MS-DOS

File granularity in MS-DOS is down to the byte. A great deal of execution time and programming effort can potentially be saved by this feature, since it is not necessary to become mired down parsing through logical records to access physical bytes.

File size under MS-DOS is virtually unlimited. The size of mass-storage devices available for microcomputers has been increasing at a dizzying pace, and the 8-megabyte limit imposed by CP/M-86 already seems confining. Future database applications may actually approach a gigabyte. MP/M-86 has extended the logical drive and file

size of the Digital Research family, but single-user versions continue to strain under the 8-megabyte limitation.

### Biggest Advantage of MS-DOS

8086-compatible versions of Microsoft's popular line of high-level languages, although announced for CP/M-86 by Microsoft, are as yet available only for MS-DOS.

### Summary

CP/M-86 seems to be the professional continuation of the line of software begun with CP/M-80, while MS-DOS may well be the consumer line of

the family tree. IBM has only just released CP/M-86 for the Personal Computer, although it had announced its imminent availability since the first press releases on the Personal Computer. This gives MS-DOS a firm edge in the 16-bit consumer market. However, the lack of any expandability into the multi-user world imposes a considerable limitation on MS-DOS, and its poor memory-management capability may force high-level languages written for MS-DOS to perform just like their 8-bit predecessors. Both operating systems have advantages, which makes choosing between them a painful task.

disk, but it requires greater foresight. You must make the decision to have a bootable disk at the time of formatting.

### Access to Functions of CP/M-80, CP/M-86, and MS-DOS

We will now discuss how the programmer can make these three operating systems perform useful functions. Some familiarity with CP/M-80 is assumed.

To interact with the disk operating system, the programmer must first place in the appropriate register the function codes listed below. The appropriate registers are as follows: for CP/M-80, the C register; and for CP/M-86 and MS-DOS, the CL register. Any information that the operating system needs must be placed in another register or set of registers: for CP/M-80, the E register or DE register pair; and for CP/M-86 and MS-DOS, the DL register or DX register.

Finally, the programmer has to tell the operating system to perform the requested service. To do this, CP/M-80 requires a CALL 5 instruction, CP/M-86 requires an INT 224 instruction, and MS-DOS requires a CALL 5. (MS-DOS also allows another way to request service by placing the function in the AH register and then executing an INT 33, but this complicates the conversion of programs from CP/M-80.)

If the information to be returned is only a byte, CP/M-80 returns it in the A register, MS-DOS returns it in the AL register, and CP/M-86 returns it

in the AL register and, in some cases, also in the ES register. CP/M-86's use of the ES register in this way can complicate matters. The impact follows when CP/M-86 loads an executable file and gives control to the applications program. CP/M-86 sets the four segment registers, including the ES register, to the proper values when the program is loaded. But if the program makes a BDOS (basic disk operating system) call to CP/M-86, the value in the ES register may be changed. That means you have to take care to restore the ES register to the value it held when the applications program was loaded.

If the information to be returned by the operating system is more than one byte, CP/M-80 returns it in the HL register pair, and CP/M-86 and MS-DOS return it in the BX register.

### The List of Functions

We'll take the BDOS functions one by one in numerical order and compare how the operating systems respond to them. We're using the decimal numbers as the Digital Research documentation does, rather than the hexadecimal numbers used in the Microsoft documentation.

**Function 0—System Reset:** CP/M-80 and MS-DOS respond the same; CP/M-86 requires a value to be placed in the DL register to designate an abort code. The value 0 asks for a termination and return to CCP (the console command processor). The value 1 asks for the program to stay

in memory and for the memory allocation to remain as it is.

**Function 1—Console Input with Echo:** CP/M-80 and CP/M-86 expand tabs and look for the characters Control-S for scroll control and Control-P for printer copy; MS-DOS looks for Control-Break for scroll control.

**Function 2—Console Output:** Same as for function 1.

**Function 3—Reader Input:** CP/M-80 and CP/M-86 have device assignment through the IOBYTE facility if the OEM (original equipment manufacturer) chooses to implement it. MS-DOS supports only one auxiliary input device.

**Function 4—Punch Output:** Same as function 3.

**Function 5—List Output:** Same as function 3.

**Function 6—Direct Console I/O:** CP/M-80 and MS-DOS respond the same: if 0FF hexadecimal is passed, a status check on the console is performed; if there is no character waiting to be returned, a value of 0 is returned; otherwise, the character is returned. If a value other than 0FF hexadecimal is passed to the operating system, it is considered to be a legal ASCII character that must be printed on the console. CP/M-86 requires 0FE hexadecimal to be passed in order to do a status check; 0FF hexadecimal passed will cause a wait until a character is ready before returning to the user. Any value other than 0FF or 0FE will be treated as a valid output character. For all three



systems, no check for special characters is performed.

**Function 7—Get IOBYTE:** CP/M-80 and CP/M-86 both support this function the same way. MS-DOS, however, treats this function as direct console input without echo and with no Control-Break check.

**Function 8—Set IOBYTE:** CP/M-80 and CP/M-86 treat this function the same. MS-DOS treats this the same as function 1, console input without echo.

**Function 9—Print String:** CP/M-80, CP/M-86, and MS-DOS all treat this function call the same way.

**Function 10—Read Console Buffer:** CP/M-80 and CP/M-86 treat this the same way, with the regular CP/M editing characters. MS-DOS allows the buffer to be edited with its own

editing characters and implements the template scheme to make command editing easier. MS-DOS also places a Carriage Return after the last character in the buffer, but the Carriage Return is not counted in the character count, in order to retain compatibility with existing software.

**Function 11—Get Console Status:** CP/M-80 and MS-DOS return OFF hexadecimal if a character is ready, and 0 if it is not ready; CP/M-86 returns a 1 if a character is ready, and 0 if it is not ready.

**Function 12—Return Version Number:** CP/M-80 and CP/M-86 return the version number. MS-DOS treats this function as a clear keyboard buffer and invokes an input function, where the function code is in the AL register and is function 1, 6, 7, 8, or

10 (referring to the functions on this list). These functions result in waiting for a new character to be typed.

**Function 13—Reset Disk System:** This function is treated the same by all three systems. Disk A is selected as the default drive.

**Function 14—Select Disk:** CP/M-80 and CP/M-86 treat this the same way, naming the disk drive designated in the DL register as the default drive, with the value 0 designating drive A, 1 designating drive B, etc. MS-DOS uses the DL register in the same way and also returns the total number of disks in the AL register.

**Function 15—Open File:** CP/M-80 and CP/M-86 treat this function the same, returning in the A or AL register a directory code of 0, 1, 2, or 3, indicating success, or OFF hexa-

*Text continued on page 348*

## Strengths and Gaps in MS-DOS and CP/M-86

Rick Fortson  
Programmer/Analyst  
Compuview Products Inc.  
1955 Pauline Blvd.  
Ann Arbor, MI 48103

Timo J. Lock  
President, Microcraft Systems Inc.  
Systems Software Manager  
Compuview Products Inc.  
1955 Pauline Blvd.  
Ann Arbor, MI 48103

*CP/M-86 and MS-DOS are both similar to CP/M-80. Anyone with experience using CP/M-80 should have little trouble becoming proficient in either environment. With the exception of the specific names for various system utilities, and the order in which their parameters are passed, both CP/M-86 and MS-DOS have the same "feel" as CP/M-80. Of course, CP/M-86 bears a much stronger resemblance to its predecessor, having many features that are virtually identical to those of CP/M-80, such as the line-oriented text editor ED.*

*Perhaps this is also one of the strongest criticisms one can find with either CP/M-86 or MS-DOS: they are both too similar to CP/M-80. Much time has elapsed in which some of the shortcomings of CP/M-80 should have been resolved. Some highly desirable features for a single-user operating system are conspicuously absent from*

*either of the 8086/88 packages, including the ability to set up a default drive on which to find an invoked-command (.COM, .CMD) file, a fully functional SUBMIT utility that would support conditionals, and enhanced I/O capabilities.*

*The degree to which an operating system supports the variety of hardware under which it runs is a good measure of its usefulness to the programmer, and thus to the end user of the software so developed. No matter how user-friendly an operating system may be, it is next to useless if the programmer has to go to inordinate lengths in order to take advantage of the particular features the hardware has to offer. However, an operating system loaded with convenience features and utilities may also run very slowly; conversely, an operating system that is compact and fast may offer the programmer next to nothing*

*in the way of system utilities and the ability to create sophisticated yet portable programs.*

### MS-DOS Memory Management

*The IBM Personal Computer is capable of supporting more than 64K bytes of memory in a fully expanded hardware configuration. However, MS-DOS as supplied with the Personal Computer cannot directly support (through software system calls or an allocation table) the full memory complement inherent in the 8088. One wants to know both the exact hardware memory configuration and the current status of the memory available to an operating system. No dynamic-memory management is in MS-DOS beyond setting location 6 to the number of paragraphs available and being able to specify that an "EXE" file loads either at the top or the bottom of avail-*

*Continued on page 344*



able memory. The MS-DOS BIOS does maintain a memory-available location that is apparently used by COMMAND.COM to set up the initial program segment. However, this does not necessarily help determine if a region of memory is available for some specific function or reserved for some other function. An "EXE" format is provided that allows the loading of complex modules, through the use of a program header; however, these are not fully dynamic. Exactly how one is supposed to utilize all available memory is not made clear in the MS-DOS documentation. We are still studying the possibilities. More fully supported memory-management facilities are clearly desirable.

### CP/M-86 Memory Management

CP/M-86 supports an alterable, possibly noncontiguous, memory-available map in its BIOS in the form of the CP/M-86 System Memory Segment Table. CP/M-86 also maintains a memory allocation table so that a complex memory status is always available. CP/M-86 command files are preceded by a command file header that contains information that both the operating system and programs written under CP/M-86 can use to deter-

mine the minimum and maximum memory requirements for the command file to be executed, as well as other relevant memory use information. Under CP/M-86, programs can be loaded from disk and left intact and inviolable in system RAM, safe from interference from other programs that may also be resident in memory. The result of these features is to enable the system designer to use memory in a very sophisticated but well-supported way.

### The Missing Assembler

CP/M-86 is supplied with an 8086 assembler, while MS-DOS provides only a linker. The assembler provided with CP/M-86 has no macro capabilities, however, and although very useful for program development, it could probably be streamlined for faster and more efficient operation.

### Input/Output Support and Online Help

CP/M-86 supports both serial and parallel printer support, featuring the CP/M IOBYTE that facilitates software control of system I/O. Unfortunately, MS-DOS supports a parallel printer port only, without the benefit of software-selectable I/O. MS-DOS

could also benefit from a more efficient disk format and a BIOS implementation that could better handle the hardware features of the IBM Personal Computer. Also, CP/M-86 supports an online "HELP" command that provides a concise explanation of the CP/M-86 feature in question, as well as an example of its use. This is a particularly useful feature for those getting started with CP/M-86.

### Speed and Documentation

MS-DOS is, however, faster than CP/M-86 in its available system calls, and the documentation supplied is unsurpassed for those starting out with an IBM Personal Computer. There will eventually arise more supported methods for utilizing whatever memory is available, whether from an extension of the current DOS or from third-party efforts. One hopes that improvements will be somewhat CP/M-86-compatible.

We hope we have raised some valid points about these two operating systems. We will continue to support both and hope that both will improve greatly as the products mature.

[The authors collaborated on Compuview's fast and impressive CP/M-86 for the IBM Personal Computer.]

**ORDER: 1-800-547-2492**

IN OREGON CALL

FOR TECHNICAL ASSISTANCE  
(503) 479-4711 or (503) 479-4150

**ATARI®**

400 (16K) \$325  
800 (16K) \$675

EPSON w/GRAFTRAX

I.D.S.

MX-80 \$439 560G Printer \$999  
MX-80FT \$539 Prism 80 w/color \$1249  
MX-100 \$699 Prism 132 w/color \$1549

RANA DRIVES

VIC \$245

Controller - 4 Drives - \$119  
Elite 1 \$379

TRS-80  
(CALL FOR PRICES)

**TERMS:**

**SHIPPING:** Add 3% of total transaction for UPS brown (ground) or 5% for UPS blue (air), Parcel Post, or any special arrangements.  
**PAYMENT:** Cashier's checks, certified checks, money orders, and bank wires honored immediately. Wire transfer funds to U.S. National Bank of Oregon, South Grants Pass Branch. Credit RCE, account number 501-981, Add 3% for Visa and Master Charge. Allow 20 days for personal checks to clear.  
**REFUNDS:** 10% restocking charge on all returns or exchanges. No refunds on opened software. Call first.  
**GUARANTEE:** All products with full manufacturer's warranty. Sanyo and Apple warranty available. We have full repair and service facilities for all electronic repairs with HP, Dynacast, Pioneer, Sanyo and Apple trained and certified technicians. For any technical service call them for instant advice or questions right on their benches at (503) 479-4150.  
**REPAIRS:** Out of warranty guarantee: Labor 30 days from date of your receipt, 90 days on parts. Call for details on quality guaranteed discount repair and reconditioning service.  
We have been repairing electronic equipment for 12 years and love it!

**SPECIAL ON APPLE REPAIRS - ALL PRICES PLUS PARTS**

APPLE II \$50 DRIVES \$40 PRINTERS \$100  
APPLE III \$100 SILENTYPE \$50 PERIPHERAL CARDS \$45

**"A Unique Combination of  
Quality Products, Competitive Prices, and Service"**

### APPLE HARDWARE

MICRO SCSI DISK DRIVES (DIRECT APPLE REPLACEMENT)  
A-2 DISK DRIVE \$379  
A-70 \$489  
A-40 \$369  
CONTROLLER - 3.3 & 3.2 \$79  
CCS 7710A SERIAL INT \$129  
CCS 7710D SERIAL INT \$129  
VIDEX 80 COLUMN \$239  
VIDEX ENHANCER II \$119  
LOWER CASE ADAPTOR \$39  
16K RAM CARD \$149  
MICROSOFT OR MPC \$139  
ALS \$99  
MICROTEK BAM-1 \$275  
Z-80 CARD (MICROSOFT) \$269  
MICROMODEM (HAYES) \$209  
SMART MODEM (HAYES) \$209  
NOVATION MODEMS \$309  
APPLE CAT \$209  
AUTO CAT \$139  
CAT \$149  
D CAT \$239  
MOUNTAIN CLOCK \$309  
MOUNTAIN MUSIC SYSTEM \$269  
MOUNTAIN A/D CONVERTER \$169  
MOUNTAIN INTROL/X-10 CARD \$149  
MOUNTAIN C.P.S. \$139  
KBD 23 KEYPAD \$299  
SUP-R-TERM (80 COLUMN) \$24  
SUP-R-MOD R.F. MODULATOR \$45  
JOYSTICK (T.G.) \$145  
9" B&W MONITOR (SANYO) \$189  
12" B&W MONITOR (SANYO) \$229  
12" GREEN MONITOR (SANYO) \$119  
12" GREEN MONITOR (ZENITH) \$114  
12" GREEN MONITOR (BMC) \$383  
13" COLOR MONITOR (SANYO) \$79  
EPSON CABLE & INTERFACE \$70  
EPSON GRAFTRAX \$39  
POWERLINE PROTECTOR \$39

### ATARI HARDWARE

410 PROGRAM RECORDER \$75  
810 DISK DRIVE \$429  
820 PRINTER \$249  
822 PRINTER \$339  
825 PRINTER \$579  
830 MODEM \$149  
850 INTERFACE MODULE \$165  
853 16K MEMORY EXPANSION \$79  
MPC 32K \$149

### SOFTWARE

VISICALC (ATARI) \$159  
BASIC (ATARI) \$49  
VISICALC \$179  
VISIDEX \$179  
VISIPLLOT \$145  
VISITERM \$75  
VISITREND/PLOT \$214  
VISIFAC \$499  
DESK TOP PLAN #2 \$179  
DESK TOP PLAN #3 \$214  
MICROLAB DATA FACTORY \$129  
D & B MASTER (STONEWARE) \$179  
SUP-R-TEXT II \$109  
WORDSTAR \$349  
B.P.I. SOFTWARE (each) \$310  
STOCKFILE INVENTORY \$269  
ZORK \$35  
SARGON CHESS \$29  
GALAXIAN \$22  
VERBATIM DISKS \$29  
MAXELL DISKS \$32  
THE SOURCE \$85

MINIMUM ORDER—\$100.00

MINIMUM SHIPPING CHARGES—\$4.00

PRICES SUBJECT TO CHANGE WITHOUT NOTICE

WRITE FOR CATALOG



**RALSTON-CLEARWATERS  
ELECTRONICS**

530 N.E. 'E' Street • Grants Pass, Ore. 97526  
ALL BRAND NAMES ARE REGISTERED TRADE MARKS



Text continued from page 342:

decimal, indicating failure. MS-DOS returns a 0 to indicate success and OFF hexadecimal to indicate failure. In addition, MS-DOS will update the disk designator in the FCB (File Control Block) if the default disk is requested. This allows changing the default disk after the FCB is opened without getting into trouble.

**Function 16—Close File:** CP/M-80 and CP/M-86 both return a directory code for successful completion (0, 1, 2, or 3) and OFF hexadecimal for failure. MS-DOS returns a 0 for success and OFF hexadecimal for failure.

**Function 17—Search For First:** CP/M-80 and CP/M-86 return 0, 1, 2, or 3 in the A or AL register to indicate the file is present, or OFF hexadecimal to indicate failure. MS-DOS returns 0 to indicate success or OFF hexadecimal to indicate failure, as noted in function 15.

**Function 18—Search For Next:** Same comments as for function 16.

**Function 19—Delete File:** CP/M-80 returns a directory code (0, 1, 2, 3 for success, or OFF hexadecimal for fail-

ure). CP/M-86 and MS-DOS both return a 0 if the operation is successful or OFF hexadecimal if unsuccessful.

**Function 20—Read Sequential:** All three systems treat this function the same way.

**Function 21—Write Sequential:** All three systems treat this function the same way.

**Function 22—Make File:** CP/M-80 and CP/M-86 both return a directory code of 0, 1, 2, or 3 if successful and OFF hexadecimal if unsuccessful. MS-DOS returns a 0 if successful and OFF hexadecimal if unsuccessful.

**Function 23—Rename File:** All three systems treat this function the same way.

**Function 24—Return Login Vector:** CP/M-80 and CP/M-86 treat this the same way, but MS-DOS does not use this function at all.

**Function 25—Return Current Disk:** All three systems treat this function the same way.

**Function 26—Set DMA Address:** CP/M-80 and CP/M-86 treat this in

the same manner. CP/M-86 has an additional system call to set the DMA segment base. MS-DOS includes the segment in this function by using the DS segment register.

**Function 27—Get Address of Allocation Vector:** CP/M-86 returns the allocation vector offset in the BX register, as you would expect for compatibility with CP/M-80. But CP/M-86 also returns the segment base of the vector in the ES segment register. MS-DOS returns the same segment information in the DS segment register. In addition, it returns the number of allocation units in the DX register, the number of records per allocation unit in the AL register, and the size of the physical sector in the CX register. Note that the MS-DOS allocation vector is in a format different from that used in CP/M-80 and CP/M-86.

**Function 28—Write Protect Disk:** CP/M-80 and CP/M-86 treat this function the same way. MS-DOS does not use it at all.

**Function 29—Get Read/Only Vector:** Same comment as for function 28.

**Function 30—Set File Attributes:** Same comment as for function 28.

**Function 31—Get Address of Disk Parameters:** CP/M-86 returns the offset of the BIOS-resident DPB (disk parameter block) offset of the currently selected drive in the BX register, as expected, and the segment base of the BIOS-resident DPB in the ES segment register. MS-DOS does not use this function at all.

**Function 32—Set/Get User Code:** CP/M-80 and CP/M-86 both treat this the same way. MS-DOS does not support this function at all.

**Function 33—Read Random:** All three systems return the same code for success, 0. MS-DOS has different failure codes from CP/M-80 and CP/M-86.

**Function 34—Write Random:** Same comment as for function 33.

**Function 35—Compute File Size in Records:** CP/M-80 and CP/M-86 have no error-return codes, while MS-DOS returns 0 for success and OFF hexadecimal for failure.

**Function 36—Set Random Record:** Same comment as for function 35.

## START YOUR OWN COMPUTER CO.

**HOW TO START YOUR OWN SYSTEMS HOUSE** \$36.  
7th edition, November 1981

Written by the founder of a successful systems house, this fact-filled 220-page manual covers virtually all aspects of starting and operating a small systems company. It is abundant with useful real-life samples: contracts, proposals, agreements and a complete business plan are included in full, and may be used immediately by the reader. Proven, field-tested solutions to the many problems facing small turnkey vendors are presented.

**HOW TO BECOME A SUCCESSFUL COMPUTER CONSULTANT** \$28.  
by Leslie Nelson, 4th revised edition, December 1981

Independent consultants are becoming a vitally important factor in the micro-computer field, filling the gap between the computer vendors and commercial/industrial users. The rewards of the consultant can be high: freedom, more satisfying work and doubled or tripled income. This manual provides comprehensive background information and step-by-step directions for those interested to explore this lucrative field.

**FREE-LANCE SOFTWARE MARKETING** \$30.  
by B.J. Korites, 3rd edition, June 1980

Writing and selling computer programs as an independent is a business where • you can get started quickly, with little capital investment • you can do it full time or part time • the potential profits are almost limitless. This best-seller by Dr. Korites explains how to do it.

**HOW TO START YOUR OWN WORD PROCESSING SERVICE** \$39.50  
by Leslie Nelson, May 1982

Turn a small investment into a steady, money making business that adds \$10,000, \$50,000 or \$100,000 to your income. Detailed start-up, marketing and operations plans are included.

Send check, money order, VISA, Master Charge or American Express # and exp. date. Publisher pays 4th class shipping. Add \$1.00 per book for UPS shipping (USA only). NJ residents add 5% sales tax. For faster shipment on credit card orders call (201) 783-6940.

ESSEX PUBLISHING CO. Dept. 2  
285 Bloomfield Avenue • Caldwell, N.J. 07006





# Oryx software

What does the Oryx, an African antelope, have in common with reliable service? They are both almost extinct. Oryx Software wants to bring trustworthy service back to life. We're here to stay!

**GUARANTEED LOWEST PRICES!**  
We will match any advertised price.  
Just show us the ad.

**ORDER TOLL FREE - Outside WI**  
**1-800-826-1589**

## SOFTWARE

<b>APPLE</b>	
Mailing List Program	48
Supertext II	128
C. O. R. P. I	235
Estimator	295
Magic Window	79
Visiend/Plot	199
Visiterm	120
Supercalc	269
Locksmith 4.1	99
Tax Preparer '82	127
Real Estate Analyzer	153
Creative Financing	149
DB Master	174
Micropro Wordstar	248
Context Connector	180
Easy Writer (Pro)	195
Easy Mailer	160
Datadex	249
Nevada Cobol	159
Visicalc 3.3	159
Spellguard	239
PFS	110
Systems plus Acct'g. Module	425
Electronic 1, 2, 3	39
Wizardry	42
Apple Panic	25
Sublogic Flight Simulator	28
Micropainter	34
Zork II	33
Edu-ware	35
Raster Blaster	25
Intl. Grd. Prix	26
Graphic Software	19
Logic Simulator	135
<b>AND MANY MORE!</b>	
<b>CP/M</b>	
Word Processor I	425
Mail Management	250
Mailmerge	115
Spellstar	169
Supersort II	159
Microplan Basic	419
Micro Pro Wordstar	297

**AND MANY MORE!**

Others for  
CBM, NEC, ATARI, IBM, ETC. . .

Prices valid for stock on hand  
and subject to change without notice.

Write for our **FREE** catalog.

For **Peripherals**  
see our other ad, page 158

Please:  
• Wisconsin residents - add 5% sales tax  
• Add \$2.50 for shipping per software and small item. Call regarding others.  
We Welcome:  
• Visa, Mastercharge - (Add 5%)  
• Checks (Allow 1-2 weeks for clearing)  
• COD (Add \$1.50 per shipment)

**ORDER TOLL FREE - Outside WI**  
**1-800-826-1589**

For Technical Info & in Wisconsin:

**715-848-2322**

**Oryx Software**

205 Scott St., Dept. BB  
P.O. Box 1961 • Wausau, WI 54401



Benchmark	CP/M-86 1.0 BASIC-86 Rev. 5.21	MS-DOS emulating CP/M-86 (EM-86) BASIC-86 Rev. 5.21
Empty do-loop	6.04	6.04
Division	22.12	22.12
Subroutine jump	11.99	11.99
MID\$( substring)	20.94	20.94
Prime number program (only 2000 primes)	47.97	47.97
Disk write program (64K-byte file)	18.3	15
Disk read program (64K-byte file)	9.1	10.1

**Table 4:** Timings of benchmark programs run in Microsoft BASIC. The programs were run on a Compupro 8085/8088 machine running at 5 MHz with one wait state on input/output. In one case, BASIC was running under CP/M-86; in the other, under MS-DOS, which was itself running EM-86, a program that emulates CP/M-86. Since the first five benchmarks were computational, the BASIC was the same, and no file loading was involved, it is not surprising that there was no difference in the times. The times on the last two benchmarks, which involve disk input/output, are surprising. MS-DOS can support the emulation program and run the BASIC disk read program almost as fast as CP/M-86 can. In running the disk write program, MS-DOS emulating CP/M-86 is actually faster than CP/M-86 itself.

	CP/M-86 1.0 PIP	MS/DOS COPY	EM-86 PIP
File copy: Copy 22K-byte file	9.9	5.75	16.15

**Table 5:** Comparison of file-copying speed of CP/M-86, MS-DOS, and MS-DOS running EM-86. The copy utilities were run on a Compupro 8085/8088 machine running at 5 MHz with one wait state on input/output.

### Benchmarks

Tables 4 and 5 show the results of some benchmarks of MS-DOS and CP/M-86 running on a Compupro 8085/8088 system at 5 MHz. The most notable result in table 4 shows that an MS-DOS system emulating CP/M-86 performed the BASIC disk write benchmark faster than CP/M-86 itself.

### Summing Up the Current Version

Based on the comparisons made so far, the serious drawbacks of CP/M-86 would seem to outnumber those of MS-DOS. MS-DOS wins points for superior speed, more efficient use of disk space, and error recovery. CP/M-86 wins points for reconfigurability and online help.

From the assembly-language programmer's viewpoint, although CP/M-80, CP/M-86, and MS-DOS are quite similar, the programmer

must be alert to some significant differences.

### What About the Future?

But now we come to the question of promised revisions of the two operating systems and learn that Digital Research has a strategy for redressing the balance.

### Concurrent CP/M-86

The most important revision coming in CP/M-86 will mean that users can do more work with the same machine. Digital Research previewed Concurrent CP/M-86 at the West Coast Computer Faire in March. In an operating system, concurrency means that a computer can do more than one thing at a time. Under Concurrent CP/M-86, for example, you can simultaneously edit one file, run an automatic spelling checker on a second file, and print a third file. You can run one BASIC program while



# MS-DOS and CP/M-86: A System Manufacturer's View

Richard Lomas  
President  
Lomas Data Products Inc.  
729 Farm Rd.  
Marlborough, MA 01752

I support both MS-DOS and CP/M-86 on my 8088- and 8086-based microcomputer systems. After using both operating systems and looking for software to run under each, I would choose CP/M-86 if forced to support only one. Although MS-DOS is in many ways an equal or superior operating system, CP/M-86 offers many advantages.

In seeking languages and applications programs for the 8086/88 processor family, I have found more available for CP/M-86 than for MS-DOS. The most notable exceptions are the Microsoft languages: BASIC, BASCOM, FORTRAN, COBOL, and Pascal. For whatever reason, Microsoft has decided not to support the CP/M-86 operating system with its languages, at least not in the near future. While some sources for a CP/M-86 version of Microsoft's interpreted BASIC exist, the version is not generally available. Other software houses are supplying versions of all the languages concerned except for FORTRAN and interpreted BASIC. Furthermore, other languages are available for CP/M-86 that are not available for MS-DOS, including C, LISP, and APL. I believe that someone will provide a good FORTRAN 77 compiler for CP/M-86 shortly. I also hope that Microsoft will see the potential market for its languages under CP/M-86.

An additional plus for the Digital Research operating system is the rumor of an Intel Universal Development Interface to CP/M-86, which would allow the use of Intel's very efficient

compilers for FORTRAN, Pascal, and PL/M.

When purchasing an operating system and other software, I want to be assured that I can continue to use the same software as my needs and my system grow. With CP/M-86, I can easily move my software to an MP/M-86-based multi-user system. With MS-DOS, the upgrade path to a multi-user system is not so clear. While Microsoft describes Xenix as the upgrade path, most—if not all—software running under MS-DOS will not run under Xenix. MP/M-86 does have one major fault, lack of memory protection, but this operating system is desirable even in single-user applications, where the problem of memory protection is minimized. Now that Intel and Digital Research have announced that MP/M-86 will be available on the 80286, a processor that itself makes provisions for memory protection, it appears that MP/M-86's problem will go away.

The last deciding issue between CP/M-86 and MS-DOS is that of performance. MS-DOS does offer performance advantages in some areas, especially disk input/output. MS-DOS has internal disk buffering built into the operating system. This makes disk input/output faster, especially when using physical disk sectors larger than 128 bytes. MS-DOS's internal blocking and deblocking also simplifies writing disk drivers for the input/output system.

By allowing records of other than 128 bytes in length, MS-DOS makes writing assembly-language programs

easier. With high-level languages, the related problems are largely transparent. The Lomas Data Products system that I use provides a CP/M BIOS with a full-track buffer. The track buffering largely offsets MS-DOS's internal buffering mechanism. For example, the times listed below are for assembling the CBIOS that comes with the Lomas Data Products implementation of CP/M-86:

	Assembly Time
CP/M-86, no sector buffering, single-density	3:30
CP/M-86, track buffered, single-density	2:30
CP/M-86, track buffered, double-density	2:20

The track buffering makes CP/M-86 come close to the speed of MS-DOS. Unfortunately, I cannot offer a direct comparison between identical software running under the two operating systems.

In summary, CP/M-86 offers the advantages of the wider availability of software and the ability to upgrade to multi-user operation without losing programs that are running under the single-user system. The availability of Intel languages for CP/M-86 would be another advantage. From a performance standpoint, a BIOS that uses track buffering can make CP/M-86 run about as fast as MS-DOS, which offsets one of MS-DOS's greatest advantages.

compiling a second program, debugging a third program, and receiving electronic mail through a communications port. How do you keep track of all that's going on? Concurrent CP/M-86 constructs "virtual terminals," each of which is a window on one operation. On machines with integrated consoles (like the IBM Personal Computer), you will be able to change from one virtual terminal to

the next with a single keystroke.

Frank Hollsworth, head of the 8086-operating-system development group at Digital Research, also promises a doubling or tripling of the speed of CP/M-86. "Hashing"—transforming index keys into a shorter, more easily manipulated form—will be used to speed up file access. Speed will be increased further by moving the blocking and deblocking of data

during transfer, now controlled in CP/M-86's BIOS, to the BDOS. As things stand, the BDOS asks for an absolute track and sector number, and the BIOS has to translate that information before doing a read or write. Moving the blocking and deblocking to the BDOS will speed matters up by eliminating some calculations.

Promotional literature promises



that Concurrent CP/M-86 will also offer:

- extensive error-handling and error-reporting
- date and time stamping of files
- record and file locking
- password protection of files, so that you can share your computer without sharing your secrets (e.g., the payroll in your office)
- networking compatibility
- real-time capability (useful, for example, for leaving the computer to send a message over the telephone while you dine out)

Hollsworth also noted that Digital Research is developing an interactive BASIC to fill the void left by Microsoft's decision not to provide future versions of BASIC-86 for CP/M-86-based machines. Also in the works is CP/M-68K, for the Motorola 68000 processor, due sometime this year. The family of CP/M operating systems will then support networking a mixture of systems based on the 8080, 8085, 8088, 8086, and 68000.

Both Hollsworth and Dr. Gary Kildall, head of Digital Research, were at pains to emphasize that Digital Research is looking at ways of adding Unix-like functionality to its operating systems. The Digital Research staff has several members with Unix backgrounds.

## MS-DOS 2.0

Microsoft will be improving MS-DOS, too. At a seminar held in New York on March 10, Microsoft's co-founder and executive vice-president Paul Allen outlined the company's plans for MS-DOS version 2.0, due in the third quarter of 1982.

Perhaps most important, Microsoft will give MS-DOS a new "shell"—the face the operating system presents to the user. Instead of a simple command line, MS-DOS will give you a screen full of information divided into "windows" by category: files, utilities, applications programs, etc. You will be able to select files and programs by cursor movement. This means no more typing of file names and program names.

There will also be online help. You will be able to get help at any time by typing a question mark, and the help will be "context sensitive." And you will be able to customize the shell.

MS-DOS 2.0 will use program and driver interfaces that enable applications programs to talk to the screen and keyboard using ANSI (American National Standards Institute) standard Escape sequences. That should simplify the writing of applications programs for MS-DOS and therefore encourage more authors to write for the MS-DOS market. Furthermore, MS-DOS will support AT&T's Pre-

sentation Level Protocol for graphics and text, which will make it easier for MS-DOS systems to interface with cable networks and many database systems.

MS-DOS 2.0 will also support networking, both local networking of MS-DOS systems and hybrid networks of MS-DOS systems with systems running Microsoft's Xenix version of Unix. Microsoft is weighing the use of the Xerox networking protocols or the Department of Defense IP/TCP protocols. Microsoft will prototype its network on IBM Personal Computers.

An important new utility routine in MS-DOS 2.0 will be a print spooler. In other words, MS-DOS will let you print a file simultaneously with running a program.

An enhancement of the DEBUG utility will permit direct typing in of assembler mnemonics, a number of Xenix-derived utilities will permit filtering files in various ways, and EDLIN will become a more powerful line editor.

As noted earlier, Bill Gates has promised to make MS-DOS reconfigurable so that owners will be able to use serial printers.

Microsoft also plans to improve the already very fast performance of disk input/output by using multiple buffers.

## MS-DOS and CP/M-86 on the IBM Personal Computer: Not My Dream Operating Systems

---

Mark Tinsdale  
c/o BYTE Publications Inc.  
POB 372  
Hancock, NH 03449

---

*Comparing MS-DOS and CP/M-86 on the IBM Personal Computer would be a lot easier if I could begin by invoking apples and oranges. Actually, these two operating systems share far more strengths and weaknesses than the suppliers, Microsoft and Digital Research, would like to admit. There are, however, indications that these systems will be evolving in different directions.*

*When Digital Research observes that MS-DOS is a "CP/M derivative," Microsoft counters that what it has supplied (known variously as MS-DOS, PC-DOS, and SB-86) is the low-end member of a new generation of operating systems. In fact, Microsoft has devoted a lot of time and energy to its Unix-based operating system (MS-DOS was purchased), and claims MS-*

*DOS to be fully "source-compatible" with the higher-level operating system. This of course assumes an application written in a portable dialect of a portable language. Much of the promised power of the Microsoft line of operating systems doesn't yet surface on the IBM Personal Computer.*

*Available for more than a year, CP/M-86 is more of a known quantity.*

*Continued on page 355*



While early versions contained more than a few bugs, and some criticism has been leveled at the elegance of the translation of an 8-bit system to the 16-bit world, several vendors have been using CP/M-86, and some applications programs have been appearing on it. MS-DOS has a clear performance advantage over CP/M-86, but this is mainly due to some fundamental design decisions, and not lack of translation overhead. Neither system is my dream computer environment. In fact, neither system is entirely acceptable according to my criteria.

### **An Applications Programmer's Dream**

What is my dream operating system? First and foremost, it will allow the support of user-friendly software solutions for noncomputer people. These solutions must be able to fully exploit the host hardware, in this case, one of considerable flexibility and power. The dream operating system will allow my program full control of the user's environment. If that user has any problems, I know it's my fault. The perfect system will charge a minimum of overhead for these services, but will support full graphics, console, disk, and communications I/O, and a host of other facilities provided by a computer as advanced as the IBM Personal Computer. All these requirements must be met by any system purporting to be usable. In addition, my particular dream operating system will allow me to easily develop those applications. Keep in mind that I am not addressing these systems primarily from the viewpoint of software developers, but from that of the Personal Computer owners who will use my software.

### **Sacrificing Memory Management to Gain Speed?**

MS-DOS is the single-user member of a line of operating systems from Microsoft. To move to the world of multi-users, networking, or multitasking, Microsoft seems to suggest buying a DEC VAX. MS-DOS contains none of the memory-management and allocation facilities supplied by CP/M-86. One immediate benefit of this design decision is performance. Showing the effects of both a simpler operating-system design and a rethinking of file structure, MS-DOS can access a floppy disk with impressive speed. Using a heavily overlaid application and a hard disk, MS-DOS made the overlay loads imperceptible, while CP/M-86 introduced a noticeable lag of a second or so.

### **Superior User Interface?**

When it comes to the user interface, MS-DOS demonstrates a few improvements over Digital Research's line. Since the copy program is resident, there is no more fumbling for a system disk with PIP and enough room. Furthermore, the syntax of operations like copy is a bit more logical to the human mind (copy from-to). Not having to type Control-C after a floppy change is a step in the right direction, and an auto-execute feature allows the creation of "bootable" applications without having to patch the BIOS. The ability to trap system errors helps in the construction of programs that never confound the user. All in all though, the system looks too much like CP/M to offer much of an improvement in the human interface area. While Microsoft claims that all its operating systems retain certain high-level features, such as device-independent I/O, it must be somewhat of an embarrassment that the system doesn't support the IBM Personal Computer's serial interface (and thus most printers). The homogeneity of this line of operating systems seems to leave a few holes that the Personal Computer owner must negotiate.

### **A Better Family?**

CP/M-86 is also part of an operating-system family. This family, however, is much more interrelated and compatible than Microsoft's. Other members of this family include MP/M-86 for multiprocessing and the recently announced Concurrent CP/M for multitasking. While these systems are not transportable to non-8086 family computers, that is really Digital Research's problem—not the IBM Personal Computer owner's. Thus, a program developed under CP/M-86 stands an excellent chance of running, with no conversion effort, under the multi-user system accessing Digital Research's local network. While some may argue that the Personal Computer is not well suited to multi-user situations, the concept of concurrency (one user running several tasks simultaneously) is intriguing. Digital Research promises these derivative systems on the IBM Personal Computer soon, and they do exist on other 8086-based systems. Microsoft will be hard-pressed to expand the capabilities of MS-DOS significantly, at least in a timely fashion. And while at least one other vendor is offering a Unix-like system for the IBM Personal Computer (yes Virginia, there will be more than two), Microsoft has not been promising anything.

### **Terminal Support**

Another difference between the two systems on the IBM Personal Computer is that MS-DOS does not emulate any terminal, while CP/M-86 does (the IBM 3101). While this difference is not inherent, but brought about by BIOS (machine-dependent) implementation, it may be the most controversial. Since BIOS content was under the control of Microsoft and Digital Research, we must conclude that they are significant aspects of the plans of the respective vendors. The lack of a simple way to specify video controls (such as cursor position and display attributes) will prevent many applications from being brought over to MS-DOS quickly (and, perhaps, optimally). Many display-oriented programs (such as word processors) that grew up in the CP/M world of microprocessors connected to terminals are ported via display drivers using control sequences. Eventually, manufacturers of machines with integrated displays supplied BIOSs that intercepted the control sequence of some popular terminal in order to allow a profusion of products to be brought to their systems. Since these terminal emulators involve much more overhead than direct video memory access, the capabilities of these integrated display systems were often not exploited.

In an apparent attempt to avoid this profusion of maladaptations, MS-DOS forces an author to rewrite the code making use of more efficient methods of Personal Computer display access. Reflecting its heritage, CP/M-86 is inviting as much software to appear on the IBM Personal Computer as soon as possible.

### **Bugged by Debuggers**

And yes, I know I promised that I was addressing the concerns of users, not software developers, but I'll relent for one moment. The MS-DOS debugger is quite adequate, but suffers from the maddening omission of an Assemble command. Anyone at all familiar with the encoding of the 8086 instruction set understands the horror of hexadecimal programming, but unfortunately Microsoft's DEBUG forces a return to basics. On the other hand (the one that giveth), a Search command has been added to locate a byte pattern in memory. Although inserting that P-relative call in hexadecimal is so exasperating, at least you are able to locate the buggy code using last month's listing. This is perhaps a symbolic example of how combining features of these two systems can produce one heck of a computer. I hope someone does it soon!



## Conclusions

Despite its lack of an online HELP facility and a utility for reconfiguring the system to use new devices, MS-DOS is a better operating system for users with little technical knowledge than is CP/M-86. MS-DOS has significant advantages in speed of disk input/output, efficiency of use of disk

space, and, most important of all, recovery from errors.

CP/M-80, designed for systems with limited memory, had to be a small operating system. As a result, no space was available to devote to things like code for error recovery. CP/M-86 has inherited some of these deficiencies. The continuing fall in the

price of memory chips should encourage Digital Research to enlarge CP/M-86 as necessary to meet the higher standards of convenience that are sure to arise as microcomputer ownership becomes more common.

Throughout its seminar in March, Paul Allen and other Microsoft spokesmen described MS-DOS as "Microsoft's single-user, single-tasking operating system." That appears to leave a big opening for Digital Research's Concurrent CP/M-86. The importance of concurrency cannot be overstated. Two years from now, all 16-bit microcomputers will have concurrency. Failure to support concurrency is a waste of computing power. Computer users will not settle for doing one thing at a time when they know their computers can do three things at a time.

Up to now, Microsoft has advised people who want multitasking to buy Xenix. At Microsoft's reception during the West Coast Computer Faire, however, Allen would not rule out the development of a Microsoft operating system that supports multitasking but stops short of being full Xenix (XE-DOS?).

In short, both Microsoft and Digital Research are moving to correct deficiencies. MS-DOS will be providing online help and support for a greater variety of hardware; CP/M-86 will increase speed and add error-handling capabilities. CP/M-86 already provides an upgrade path to MP/M-86, and MS-DOS will provide an upgrade path to Xenix. Despite MS-DOS's present impressive superiority in speed and error-handling, Digital Research's Concurrent CP/M-86 will give people a means of getting more work out of their computers.

Competition between CP/M-86 and MS-DOS will bring dramatic improvements in both operating systems. Much depends on when the promised improvements actually become available to the average user. If Microsoft and Digital Research do not deliver the improvements in a timely fashion, Unix may take up residence in more microcomputers than CP/M-86 and MS-DOS combined. ■

## A Vote for MS-DOS

Neil J. Colvin  
President  
Phoenix Software Associates Ltd.  
POB 207  
North Easton, MA 02356

*[Editor's Note: The following statements have been excerpted from the transcript of a speech given by Mr. Colvin at Microsoft's recent "Inside 16-Bit Operating Systems" seminar.]*

My development staff has been designing and implementing CP/M-80-based software for five years and is now making the transition to the 8086. Since all members of my staff are now using MS-DOS for their development environment (a choice not imposed upon them), I must conclude that MS-DOS provides a more productive environment in which to work and a better foundation upon which to build new software products.

Among the important differences between MS-DOS and CP/M-86 are the following:

**Program management:** Both CP/M-86 and MS-DOS allow multiple segments of code to be included in a program, but only MS-DOS will automatically relocate intersegment linkages at the time of program loading. Under one of CP/M-86's program models, the segment bases of additional program segments are available in the memory image, but they must be explicitly managed by program code.

**Memory management during program execution:** MS-DOS uses an approach like that of CP/M-80 for memory

management. After a program is loaded, certain locations in its low data area contain the addresses of the top of the current data segment and the top of all available memory. The program has access to all memory from the base of its code segment through the top of memory, and can manage it as it desires. As a result, MS-DOS only supports machine environments that have contiguous user RAM.

CP/M-86 provides a set of system functions for allocating and releasing dynamic-memory space. Memory can be allocated from the free-memory pool or at absolute memory locations, providing control over access to memory-mapped devices or other non-shareable memory resources. CP/M-86 permits requests of memory segments of specific size or of the maximum memory segment available. As a consequence, available memory need not be contiguous.

**Reliability:** Who is to say when one new software system is more reliable than another? I can only relate our experience to date, which shows a vast reliability difference between the two operating systems. We have been using CP/M-86 for almost a year and are still finding bugs, including complete features that just do not work as documented. We have been using MS-DOS for only about four months, but have yet to find a single bug.