SOFTWARE

Getting compatible with CP/M

WILLIAM R. VAUGHN, Mostek Corp.

Manufacturers reshape a generally accepted operating system to broaden its application

In an attempt to broaden the applications for CP/M, several hardware manufacturers, including Cromemco, Inc., SD Systems and Mostek Corp., have molded a generally accepted μc operating system to their own use. Such support should improve the attitude of users toward μc systems and, more importantly, result in higher productivity. These newer systems improve both the efficiency of the hardware and the productivity of the application programmer, who does not have to write a special routine to perform a simple multiplication or to get at the system clock.

CP/M is a disk operating system designed to run on 8080- or Z80-μp-based μcs. Originally developed by Digital Research, Pacific Grove, Calif., in the mid-1970s, CP/M was one of the first hardware-independent disk operating systems for μcs, providing a plug for a previous market void. Before CP/M, a μc user had to rely on software supplied by the μc's manufacturer, which was often limited in quality and quantity and usually could not be transported to other manufacturers' systems. As a result, users were virtually tied to a particular manufacturer's products. CP/M freed users from dependence on vendor software and gave them access to a growing base of pre-written and tested application-software packages.

No operating system is without its faults, however, and CP/M is no exception. The original CP/M, for example, was intended for use with single-density diskettes, thus restricting the use of double-sided and quad-density floppy- and hard-disk systems. Similarly, CP/M was designed for use with a printing terminal and did not easily accommodate CRT terminals. These deficiencies, many of which have been corrected in subsequent CP/M releases from Digital Research, spurred several μc hardware and software vendors to create their own CP/M-compatible systems. These include I/OS (or TSA/OS) and Multi-OS sold by InfoSoft Systems; CDOS sold by Cromemco; SDOS and COSMOS sold by SD Systems, Garland, Texas; and the newest system, M/OS-80 sold by Mostek. All these systems can run CP/M application programs without change. Their operational syntax is easier to use and more forgiving than the original CP/M's, and they offer better user file protection and system performance. CP/M-compatible systems differ from their progenitor, however, in that

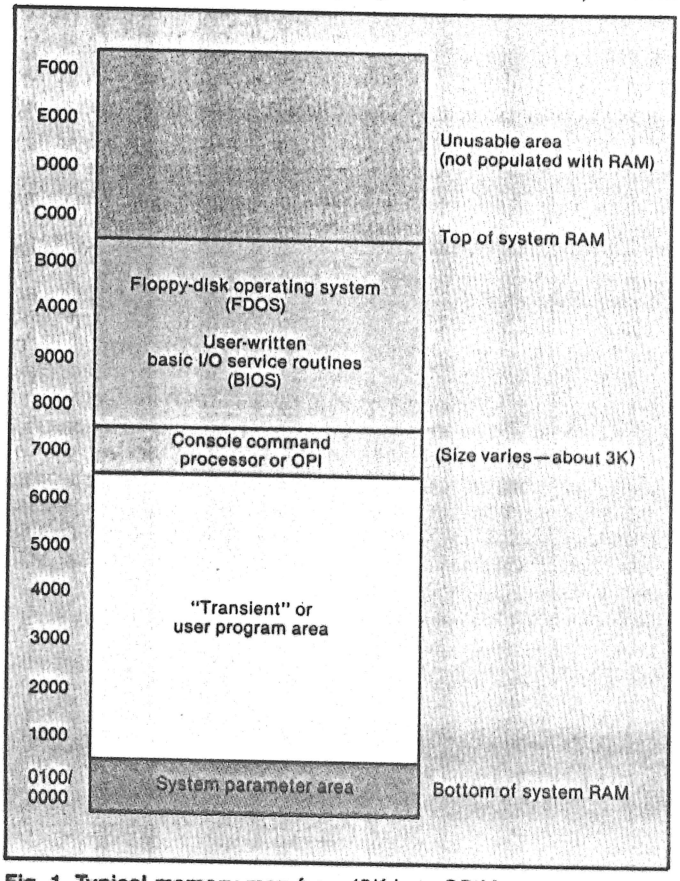

Fig. 1. Typical memory map *for a 48K-byte CP/M system.*

(Editor's note: As this article suggests, Digital Research hasn't been standing by idly as hardware vendors developed CP/M-compatible operating systems. The originators of the popular system have been improving on the product themselves, and will detail some of those developments in an article in a later issue.)

CP/M-compatible systems differ from their progenitor in that they are tailored to operate on a pre-defined set of hardware, yielding systems that perform better than the original CP/M but have less flexibility.

they are tailored to operate on a pre-defined set of hardware, yielding systems that perform better than the original CP/M but have less flexibility. This article compares CP/M-compatible systems to CP/M to show how differences in memory and disk structure affect system performance.

In CP/M-compatible systems, the basic memory structure is usually very similar to that of CP/M systems. The transient program area (TPA) must start at location 100 Hex (Fig. 1). The system implied-run directive assumes this and loads all transient programs at that address. Those programs can subsequently move themselves to other RAM locations to make room for other transient programs.

Many transient programs, especially the newer compilers, require the maximum amount of TPA RAM space (56K bytes) to function. As a result, they cannot work properly on smaller systems. To circumvent this problem, some older programs, such as Digital Research's DDT and some BASIC interpreters, overlaid the console command processor (CCP). They could do this because they handled the operator-computer interface themselves.

CP/M-compatible operating systems improve on this space-saving technique by allowing the CCP operator interface (OPI) to be made disk-resident at system-generation time or at operator request. This increases the amount of available memory space by about 3K bytes.

Improvements in the system parameter area

CP/M-compatible systems have also made improvements in the system parameter area (Fig. 2), which is used to store vectors and data required by user programs and the system itself. For example, starting at location 0, the system that branches to the "warm boot" or restart vector (location 0) causes some CP/M systems to reread portions of the DOS from disk. M/OS-80 and several other CP/M-compatible systems do not need to perform this disk access unless the OPI (console processor) is overlaid.

CP/M-compatible systems define a few areas (Fig. 2), such as locations 28H through 2FH, that were designated as unused or undefined in the CP/M documentation. Also, M/OS-80 and several other emulators provide the user with another level of security by trapping jumps to locations that contain Hex FF bytes. This feature protects the user from programs that branch into non-allocated areas of RAM. CP/M does not support this feature and would permit the user to call location 38, for example, using whatever random value has been placed there as an executable instruction.

CP/M also does not support user interrupts. Most CP/M implementations disable interrupts upon entry, thus defeating the entire interrupt protocol set up by a user application. Virtually all CP/M-compatible systems can and often do use interrupts for a number of user and system functions.

Disk structure and sector size

CP/M's disk structure is not very efficient or flexible. For example, the CP/M home-disk operation requires a

HOW COMPATIBLE ARE CP/M-COMPATIBLE SYSTEMS?

The CP/M-compatible systems discussed in this article are not totally compatible with all versions of CP/M because of the ambiguity of CP/M documentation and the existence of variations from the documentation. For example, some application authors have replaced areas of CP/M code with code that is more to their liking or better suited to their individual application. The response by the creators of CP/M-compatible systems is to disclaim compatibility with such programs because it would be impossible to maintain compatibility with these rogue applications.

Almost all CP/M-compatible systems were originally conceived in response to shortcomings in earlier versions of CP/M (up to 1.4). The compatibility question was considerably less complex then because of the lack of movement by Digital Research and the resulting stagnancy of the "standard." CP/M-compatible authors had several years to eliminate incompatibilities and had at one point corrected virtually all known bugs.

The arrival of CP/M 2.2, however, caused the compatibility question to rear its ugly head again. Because Digital Research chose to use a different means to achieve larger file sizes, those programs written for CP/M version 2.0 and later versions lost a level of compatibility with the emulative systems. The authors of the CP/M-compatible systems have enhanced their systems to emulate the new CP/M functions. M/OS-80, for example, supports all CP/M version 2 calls except those that Digital Research has labeled as not required for use by application programmers, or those that conflict with M/OS-80 standards.

The best solution to the compatibility question is to address each problem as it arises. Mostek has done so by buying and testing as many application programs and systems support languages as possible. Mostek has found that all programs that work on CP/M version 1.4 now work on M/OS-80. Programs that are specifically designed to run on CP/M version 2.2 are few and generally include new versions of existing compilers. These are currently being tested with results yet to be determined. Customers who are worried about compatibility are encouraged to let Mostek test their CP/M application under M/OS-80.

As for programs written expressly for CP/M-compatible systems, many use system calls not contained in the standard CP/M. This prohibits their use on a normal CP/M system. For example, many programs written by Cromemco for use with CDOS will not run on a non-CDOS system.

Location:	Contents:
0000-0002	Warm boot entry point; reinitializes operating system and passes control from user to DOS.
0003	IOBYTE; 8-bit switch to indicate various levels of I/O device hierarchy.
0004	(Reserved)
0005-0007	Jump vector to BDOS; used by transient programs to request supervisor services.
(0006-0007)	Address portion of above vector; used to determine lowest address in memory used by DOS.
0008-0027H	Interrupt locations 1-5 (not supported in normal CP/M implementations); used by interrupt-driven routines as required in CP/M-compatible systems.
(0028H-002FH)	(Not mentioned in CP/M documentation.) Reserved for interrupt vectors in CP/M-compatible systems.
0030H-0037H	Interrupt location 6 (not supported in normal CP/M implementations); used by interrupt-driven routines as required in CP/M-compataible systems.
0030H-0037H	CP/M-compatible debugger breakpoint.
0038H-003AH	Illegal address trap (M/OS-80) or CP/M DDT debugger breakpoint.
003BH-003FH	(Not used—reserved.)
0040H-004FH	Scratch area reserved for DOS.
0050H-005BH	(Not used—reserved.)
005CH-007CH	File control block (FCB) formatting area. For use by transient (user) programs).
007DH-007FH	(Not used—reserved.)
0080H-00FFH	128-byte buffer; default area used by disk I/O and console command line upon entry to a transient program.

Fig. 2. Contents *of page zero.*

long trip back to track two for a directory search or update. CP/M operations are substantially hobbled by this structure.

In most CP/M implementations, disk sectors are 128 bytes long (thus the 128-byte buffer area at location 080H). In other versions, especially the higher density systems, the sector size can extend to 1024 bytes. In these higher density systems, the DOS performs special handling and deblocking of these sectors. Some new disk-controller boards handle this deblocking with built-in CPUs and buffers in a manner totally transparent to the DOS. Although CP/M can be adapted to either soft or hard sectoring, most systems using 8-in. diskettes employ the soft-sectoring technique. There is no real standard for 5¼-in. diskettes.

The system allocates blocks (virtually synonymous with sectors) in groups of eight or more, called clusters, which usually have a length of 1024 (1K) bytes (128*8). A bit map of the used and unused clusters is kept in memory for each disk drive and is re-initialized every time the system is warm booted (whenever the operator types control-C) or the system is restarted upon power-up. The 1K cluster-allocation scheme significantly reduces the size of the bit map but requires that a file's size be MOD 1K, that is, no file can be smaller than 1K byte with additional increments of 1K for larger files.

Allocation of clusters is made as the user program demands them. When using CP/M, a serious problem can occur if a user fails to type control-C after changing a diskette. Because the bit map in memory would not necessarily match the newly inserted diskette, the system can allocate sectors that the bit map erroneously shows to be vacant. The result is a directory that has

Block*	Purpose
0	Bootstrap program
1 . . . 51	DOS system
52 . . . 67	System directory
68 . . .	User file area

*Blocks in most CP/M systems are one sector long (128 bytes)

Fig. 3. Disk structure *allocation.*

clusters assigned to more than one file, or files whose clusters are overlaid by incoming data from another file. Ordinarily there is little recourse if this happens.

Newer CP/M-compatible systems, however, like M/OS-80, provide more protection against such problems. Unlike CP/M, M/OS-80 makes a new bit map whenever a file is opened. This takes the system an additional moment to perform but provides the security needed to protect valuable files. In addition, a user is not locked out when a write-protect error occurs in an application. A user often can recover from disk errors by correcting the problem and instructing the operating system to continue despite the error, or to retry the procedure. A user can defeat this process by changing diskettes after files have been opened, or in the middle of an application that does not expect the operator to change disks. But the chances of a self-inflicted file wound are considerably lessened in M/OS-80 and some other emulators.

CP/M allocates the outer sectors of the system diskette (0-1) to the bootstrap program and operating-system-resident image—or at least a program that will give the system enough intelligence to load the DOS image. The full operating system is then loaded from

CP/M-compatible operating systems improve on the space-saving technique by allowing the CCP operator interface to be made disk-resident at system-generation time or at operator request.

the file named SYSTEM.COM. Then the operator interface is loaded from a file called OPI.COM. Because this file thrashing takes about 10 sec., CP/M-compatible operating systems do not reload the resident image into memory when a warm boot (control-C) is performed, saving a user several seconds of waiting. In addition, because the DOS is not reloaded, a system disk need not be kept on the master drive at all times as in some CP/M implementations.

The system file directory, which starts at track two or block 52, contains the names and locations of files on the disks and, in the case of CP/M-compatible systems, or CP/M version 2.n, it contains file-security attribute bits. These attribute bits enable a user to mark a file as write-protected, read-protected, erase-protected or system (to make the file invisible to an operator). Some multi-user systems also permit a user password or user number to be assigned to a file, adding another level of protection.

Increasing file sizes

CP/M version 1.4 systems limit a file to 256K bytes, minus the system boot and directory blocks, thus 240K bytes. CP/M-compatible systems increase the blocks-per-cluster to 16 or more to accommodate larger disks. This increases the cluster size to 2K and increases the minimum file size to 2K with 2K increments. Expanded clusters provide the user with files that can extend beyond 512K bytes.

CP/M version 2.n provides for larger files by allowing as many as 512 logical file extents, where each logical

Relative byte	Purpose
0	Entry type (not used by CP/M 1.4); disk specifier in CP/M 2.2 and CP/M-C.
1-8	File name (left-justified, blank-filled).
9-11	File extension.
12-13	Current file extent; beginning at 1, extent is incremented by 1 for every 16 clusters added to file. Essentially, this is the FCB number. Byte 13 used for larger files in CP/M-compatible systems.
14	CP/M-defined as for internal use.
15	Record count; number of blocks written to file in current extent (0-80 Hex).
16-31	Cluster allocation map; bytes contain the cluster numbers for each cluster assigned to this file.
32	Next sequential record to be read; relative to top of file with first record = 0.
33-35	Random record pointer (24-bit value) to permit fetching or writing records nonsequentially anywhere in a file.

Fig. 4. File control *block (FCB) structure.*

THE FUTURE OF CP/M

CP/M is one of the most widely used operating systems for the 8080/Z80 in existence. Although M/OS-80, CDOS and the other CP/M-compatible operating system vendors have not sold as many copies as Digital Research has of the original CP/M, they have provided healthy competition. Because of that competition, Digital Research has improved its systems support and has made several important improvements to the CP/M package. In short, CP/M-compatible systems have helped to bolster the performance, features and stability of the entire "CP/M standard" concept.

The future of CP/M or CP/M-compatible systems is tied to the future of the processors on which they are designed to run—the 8080 and Z80. As 16-bit processors evolve, operating systems will evolve around them, and users will again choose one system that best meets their needs. Some of these new processors may be able to emulate the 8080 or Z80 chips, and thus support CP/M. But, as was seen in the transition from the IBM 360 DOS to the IBM 370 OS/MFT, "antiquated" operating systems can quickly lose favor with customers who, for reasons ranging from vanity to performance demands, want to migrate to faster operating systems.

However, 8080/Z80 systems are likely to be around for decades. As the technology gets cheaper, the tendency to produce one large, powerful machine will be offset by the concept of having a dozen or a hundred small ones to share the work. In these multiprocessor systems, single or multiple sets of CP/M (or a derivative of it) could be preeminent.

The future will also see the maturing of multitasking versions of CP/M that are already emerging. These include MP/M (Digital Research, Pacific Grove, Calif.), COSMOS (SD Systems, Garland, Texas) and Multi-OS (InfoSoft Inc., Westport, Conn.). These systems enable a processor to be shared among several competing tasks. These systems, however, do not take full advantage of the declining costs of 8-bit μps by using dedicated processors to handle the various tasks. Some systems do use slave processors to perform smart console overhead and disk front-end processing. But this dispersal of system intelligence to sub-processors is about as far as most mass-produced multiprocessor systems have gone on the 8-bit level. These multiprocessor systems are expected to gain favor as time passes because they are cheaper to produce.

Virtually all CP/M-compatible systems can and often do use interrupts for a number of user and system functions.

extent contains 16K bytes of data. CP/M 2.0 is structured, however, so that as much as 128K bytes of data are addressed by a single physical extent (corresponding to a single directory entry), maintaining compatibility with previous versions.

Additional schemes involving entirely new directory structures are now in place in the newer M/OS-80 and CDOS versions as well as in the newest CP/M version 2.2. In the case of M/OS-80 (hard-disk version) and CP/M 2.2, these extended-directory allocation formulas expand the user file size to more than 64M bytes. These file sizes are intended for use on hard-disk versions where a considerably larger bit map is kept on the disk itself to conserve system RAM space.

The primary way to delineate files is the file control block (FCB). Although the FCB varies somewhat from system to system because of eccentricities of the implementation, the basic format is unchanged (Fig. 4). Newer hard-disk systems have altered the format to accommodate larger files, but virtually all CP/M systems will recognize the file structure of another. In addition, CP/M-compatible systems employ several unused bytes to define attribute bytes, user IDs and expanded file sizes.

There are several areas of inconsistency between CP/M and its emulative cousins (Fig. 4). For example, byte 14 is reserved by CP/M for internal use. In many implementations of CP/M, it is set by the DOS to some random value; in others it is set to zero. M/OS-80 and several other CP/M-compatible systems expect this number to be the secondary record count. This counter is used only when the record count in byte 15 exceeds

Program name CP/M	M/OS-80	Function
ASM		8080 absolute assembler.
	COPY	Image copy entire disk.
	DISKRD	Disk diagnostic. Read all sectors of disk requested.
DDT	(DDT)	CP/M debugger. This program is PROM-based in M/OS-80 with many of the same functions.
	DSKDUMP	Dump a disk or file by block for inspection or alteration. Permits alteration of any sector in any file including the diskette directory.
DUMP	DUMP	Print files in Hex and ASCII format.
ED	EDIT	Text editor.
	ERASE	Conditionally erase files with operator prompting.
	FORMAT	Initialize diskettes.
	GTOD	Initialize or read system clock.
	LABEL	Examine or alter disk labels. Alter directory or cluster size.
LOAD		Intel Hex file loader. Translates Intel Hex format into memory image.
	MEMTEST	Test RAM memory.
MOVCPM		Generate a new CP/M system for a particular memory size.
PIP	XFER	Transfer datasets to other datasets or devices.
	PRINT	Print disk files.
	SPLIT	Segment large files.
	SPOOL	Initiate print spooler. Permits printing of files as a separate task while system is functioning elsewhere.
STAT	XSTAT	Determine directory status. In CP/M, list files by size, show or alter IOBYTE assignments. List-size function performed by DIR (built-in function) in M/OS-80.
	STARTUP	User-written auto-start program for custom systems.
	STARTUP.CMD	User-written auto-start batch (submit) file.
	SXFER	Similar to XFER but requires only one disk.
SUBMIT	@	Submit a command stream from disk.
SYSGEN	WRTSYS	Write system image to tracks 0 and 1. M/OS-80 reads from either a file or the system area of any mounted disk and writes to a file or the system area of any disk, even in single-disk systems.
	XDIR	Extended directory. Provides a sorted listing in multiple columns for CRTs. Entire directory is shown on the screen at once.
XSUB		Extended submit utility provided in CP/M 2.2 versions. Permits programs run under SUBMIT to accept input data from files.

Fig. 5. CP/M and M/OS-80 *utilities compared.*

Call:	Function or feature description
30	Set file attributes. Permits marking a file as read-protect, erase-protect, etc.
33-36	Supports random I/O to files of more than 65M bytes.
130	Set user control-C handling. This permits setting up a one-use system that cannot be user-interrupted.
132-133, 152-154	Logical block I/O without regard to files.
133	Control printer spooling (not supported by CP/M in any version).
136	Chain to user program. Future versions will permit this call to support an overlay technique.
137-138	Built-in multiply and divide routines.
140	Eject disk. Useful on systems that have hardware facilities to control disk ejection. Can be set up to prohibit removal of disks from a system until the software permits it.
142	Set terminal functions. On some systems, permits a certain CRT protocol to be built into the system so that user programs can converse with an application-transparent terminal.
143-146	Set/read date and time.
149	Read disk label. Disks under M/OS-80 may be user-labeled.
150	Turn disk-drive motors off. Some implementations turn the motors off if the disks are not used after more than 15 sec.
151	Set system bottom. When loading user RAM- (or ROM-) resident subroutines, the system RAM top may be altered to reflect the presence of these programs.
159	Gets the disk assigned as master. In M/OS-80, one of the disks may be designated the master library disk. The disk is searched when searched-for files are unsatisfied on the current or requested disk.

Fig. 6. Enhanced CP/M *functions available with M/OS-80.*

$255_{10}(FF_{16})$. The result is that when listing the directory for some CP/M disks on M/OS-80 systems, the size shown is extremely large. However, once copied under M/OS-80, the fields are corrected.

Basically, the new CP/M-compatible systems can read old CP/M format directories. Not all new-format directory structures are inter-compatible, but this has not been a significant problem so far. The widest divergence in incompatibility lies in the structure of the various hard-disk implementations. Because virtually all of these are of the fixed-media type, and lack the ability to be removed and moved to another dissimilar system, the problem will not be significant until removable-media disks are more commonplace.

A more serious problem that has hampered transporting disks between systems concerns double-sided diskettes. Because IBM set the standard for 8-in. diskette formats (3740 format), a problem arose when IBM changed the optionally 00 or FF filler bytes to mandatory FF when using dual-density or dual-sided disks. This meant that single-sided, single-density diskettes created on new machines could not be read on older machines. Fortunately, new controller chips from Western Digital Corp. allow software to select the filler bytes to permit downward compatibility. M/OS-80 in the dual-density, dual-sided mode allows compatibility.

Almost all CP/M implementations include utility programs to manipulate system files, create new systems and system disks, examine file directories or perform other housekeeping functions. Some CP/M-compatible systems, such as M/OS-80 have considerably more support programs than CP/M (Fig. 5). Many of these programs take full advantage of the additional system calls provided by M/OS-80. Others are similar to programs provided by CP/M, but, in most cases, provide more features and flexibility than their CP/M equivalents. It would be impossible to determine what features CP/M 2.2 would have at this time if it were not for constant competition by CP/M-compatible systems.

M/OS-80, CDOS and other CP/M-compatible operating systems are intended to run on hardware of the manufacturer's choosing, enabling vendors to increase system efficiency and reduce system memory requirements.

But pre-packaged systems do not address the problem faced by system configurators who want to create systems around a custom card or a special front-end software handler. To alleviate this problem, Mostek is building a system that allows a user to write his own drivers and link them into the system as required. This method is considerably cleaner than the method used by Digital Research in the earlier versions of CP/M, which did not allow a user to boot up a hardware configuration different from the system on the boot disk.

In addition to supporting standard CP/M functions, CP/M-compatible systems provide additional features that enable systems and applications programmers to use system resources more efficiently (Fig. 6). For example, a system spool operation enables a system to use the abundant wait time during periods of I/O to spool print files onto disks for later printing. ■

William R. Vaughn is an applications engineer with Mostek Corp., Carrollton, Texas.