

MICROPOLIS - CP/M

VERSION 2.0

By Bruce Ratoff
and
Computer Mart of New Jersey's
Software Division

SOFTWARE AGREEMENT

This software is copyrighted and has been supplied subject to the following restrictions: It shall not be reproduced or copied without express written permission. No attempt shall be made to reconstruct the source code of any program for which only object code was supplied.

To do any of the above without prior written approval may subject you to liability.

This agreement shall be considered accepted and binding upon your receipt of this and any software.

CP/M is the registered trademark of Digital Research Corp.

Introduction

The programming effort which results in a complete CP/M system comes from two sources: Digital Research (the inventors of CP/M) and your computer store. CP/M was originally designed to operate with any brand of floppy diskette drives utilizing what is commonly referred to as 'IBM format'. This most recent release has been extended to accomodate almost any floppy or hard disk system. Therefore, the specific routines and tables required to control a particular brand of disk hardware are isolated from the rest of the operating system. Provisions have been made in the design of CP/M for the addition of these routines. Your release diskette contains a copy of CP/M to which have been added routines that control the Micropolis diskette subsystem, resulting in a complete and very powerful disk operating system. These routines are known in CP/M terminology as the Customized Basic Input/Output System (CBIOS).

CBIOS also contains, in addition to the disk routines, routines to handle a terminal, a reader, a punch, and a listing device. On the diskette supplied, the terminal handler has been designed to readily accomodate a variety of console interfaces. The remaining routines are supplied in skeletal form, with sample arbitrary 8080 port numbers used. After bringing up your first CP/M system using the automatic console configurator supplied, you will probably want to edit the source code of CBIOS to include the appropriate routines to create a totally customized CP/M system.

You will also find that this implementation of CBIOS contains the ability to automatically execute a program prior to system sign-on. You may use this facility to automatically load a monitor or driver package which is too large to fit into the limited space allotted to CBIOS. Another possible use is to automatically force the user of the system into a special application program of your choosing, so that he need not be aware of CP/M at all. If you do not need this feature, you may remove it when you edit your custom modifications into CBIOS.

Besides CBIOS, several other programs are necessary to load and start CP/M. This operation is commonly known as bootstrapping, and for this system is accomplished in two steps or levels.

The Level 0 Bootstrap is a very short program whose sole purpose is to load and start the Level 1 Bootstrap. The Level 0 Bootstrap is located in the Micropolis PROM, at the beginning of the 1K block you have jumpered the Micropolis controller for.

The Level 1 Bootstrap is a program which loads CP/M and the initial program. It is stored in track 0, logical sectors 1 and 2 of a CP/M diskette. The Level 0 Bootstrap loads this program 200 hex below the address where CP/M will load. The Level 1 Bootstrap loads CP/M into the top 9K of the configured system.

Note that load address of the Level 1 Bootstrap changes with the size of the configured system. This means that the Level 1 Bootstrap is relocated whenever a different sized CP/M system is generated.

The Level 1 Bootstrap also implements the automatic program execution feature of Micropolis CP/M. It initializes CP/M, and then uses it to load a program into the Transient Program Area (TPA) of the system. The TPA starts at address 100 hex, and is the area in which all programs are normally executed under CP/M. Before passing control to the loaded program, page 0 of memory is set up to contain the vectors which normally link a transient program to the operating system. The argument blocks which are normally set up by CP/M when a program is invoked are cleared, so that it will appear to the program as if it were invoked in the normal manner, with no arguments.

The name of the program which is loaded on startup is INITIAL.COM. If this program is not found, no error is generated. The Level 1 Bootstrap simply jumps to CP/M, which then signs on and prompts for a command. On your Micropolis-CP/M release diskette, INITIAL.COM is a program which is used to insert the correct console routines in your CP/M system the first time it is booted up. This routine may then be renamed or erased, enabling you to set up a different program for automatic execution thereafter, simply by renaming it to INITIAL.COM.

Please note that in the instructions which follow, those portions of a line which you are supposed to type will be underlined, carriage return will be represented by '<ret>', and control characters will be preceded by an up-arrow (^).

Booting up CP/M

The procedure for booting CP/M is very simple. With your computer, terminal, and drives powered up, insert the Micropolis CP/M release diskette in drive 0 and execute at the Micropolis boot address for which you have jumpered your card. The drive head will load with an audible click and the select light will remain lit for three to five seconds, with disk activity taking place. The system will then hang in a loop near address 100 hex. Regain control of the system via your front panel or monitor program. Select the configuration **number** corresponding to your terminal I/O configuration from the table on the following page, and store it at address 5C hex. Then execute address 100 hex. If your console interface is not listed, patch the correct ports and masks into the addresses shown and execute at address 4B00 hex. CP/M should then sign on:

22k CP/M vers 2.0 on Micropolis disc
 Copyright 1979, by Bruce R Ratoff
 A>

Before proceeding any further, it would be wise to use your running system to make a backup copy of the CP/M release diskette. The idea of backup is vital to successful use of any computer system, large or small. You are advised in the strongest possible terms to copy your release diskette and put the original away in a safe place, doing all further work on the copy rather than the original. In this way, you always remain able to return to the beginning of the installation procedure in case a human mistake or machine problem wipes out part of the diskette on which you are working. If you have a single-drive system, ask a friend or your computer store to make you a spare copy of your release diskette. If you have two or more drives, you may use the MCOPY utility to make a backup system diskette as follows:

A>MCOPY ALL	(this loads the MCOPY program)
MCOPY version 2.0 Mod II	
Source on A	(leave the release diskette on drive A)
Destination on B	(put a blank diskette in drive B)
then type <ret> <ret>	(type a carriage return)
Function complete	(appears when finished)
Source on A	
Destination on B	(program loops back to beginning)
then type ^C	(control-C will break the loop)
Rebooting, type <ret> <ret>	(type return to reload CP/M)
A>	

You may now remove the release diskette from drive A and put it away. Place the backup copy in drive A and type control-C (you should type control-C whenever you change diskettes in any drive). In the steps which follow, we will deactivate the automatic configuration program, and install a copy of CP/M which permanently contains your console driver.

Micropolis-CP/M Console Codes

<u>Number</u>	<u>Letter</u>	<u>Configuration</u>
Port Drivers		
1	A	Altair 88-2SIO
2	B	IMSAI SIO2
3	C	Altair SIO A,B,C (not rev 0)
4	D	Altair SIO A,B,C (rev 0)
5	E	Processor Technology 3P+S
6	F	IMSAI MIO
7	G	Altair 88-4PIO
Monitor Drivers		
13 (hex)	S	Processor Tech SOL-20 with SOLOS 1.3
18 (hex)	X	Exidy Sorcerer
1A (hex)	Z	TDL System Monitor Board (ZAPPLE)

Note: Port drivers select a particular set of I/O ports to drive directly. Monitor drivers call the appropriate routines in a system monitor, which must already be resident in the user's system (usually in PROM).

All port drivers operate as follows: The "status" port is read, and the resulting byte is ANDed with the AND Mask and exclusive-ORed with the XOR Mask. The device is considered ready when the result of these two operations is zero (this is just like the WAIT instruction in Microsoft Basic). The data is then input or output to the "data" port. The ports and masks for each configuration are as follows:

<u>Parameter (hex)</u>	<u>Addr</u>	<u>Configuration</u>							
		<u>A</u>	<u>B</u>	<u>C</u>	<u>D</u>	<u>E</u>	<u>F</u>	<u>G</u>	
Input Status port	4F0D	10	03	00	00	00	43	10	
Input Data port	4F25	11	02	01	01	01	42	11	
Output Status port	4F34	10	03	00	00	00	43	12	
Output Data port	4F44	11	02	01	01	01	42	13	
Input AND mask	4F12	01	02	01	20	40	02	80	
Input XOR mask	4F14	01	02	00	20	40	02	80	
Output AND mask	4F36	02	01	80	02	80	01	80	
Output XOR mask	4F38	02	01	00	02	80	01	80	
Initialization	4F81	Y	Y	N	N	N	Y	Y	

The addresses given above are for the 22K system provided on the release diskette, and should enable you to do your own patches if you do not have one of the supported configurations. Addresses for other system sizes may be obtained by simple addition. Addresses for a SYSGEN image under DDT may be obtained by subtracting 2B00 from the execution address.

Making a Permanent Console Selection

On the release diskette provided, there is a special program which patches your console I/O when you boot the system. This program is the reason the system hangs until you have patched your configuration number into location 5C. The program exists in .COM form as INITIAL.COM. INITIAL.COM is a special filename which CBIOS looks for when CP/M is first booted up. Whenever a program with this name is found on drive A at boot-up time, it will be automatically loaded and executed before the system signs on. You will not want to have to go through the configuration procedure every time you boot CP/M, so we will now make your console selection permanent and rename INITIAL.COM, so that it is no longer automatically executed. Note that the configurator is also provided in source form as CONFIG.ASM for your edification.

To install CP/M with your console driver permanently set, perform the following steps:

```
A>MOVCPM ?:22 * (where '?' is the configuration
CONSTRUCTING 22k CP/M vers 2.0 letter for your console from the
READY FOR "SYSGEN" OR table on page 5)
"SAVE 40 CPM20.COM"
A>SYSGEN (run SYSGEN to install system)
SYSGEN VER 2.0
SOURCE DRIVE NAME (OR RETURN TO SKIP) <ret>
DESTINATION DRIVE NAME (OR RETURN TO REBOOT) A
FUNCTION COMPLETE
DESTINATION DRIVE NAME (OR RETURN TO REBOOT) <ret>
```

At this point, you have written a configured system to the boot tracks of your work diskette. You should now disable the automatic configurator by typing:

```
A>REN CONFIG.COM=INITIAL.COM
```

At this point, you should try to cold-boot from the work diskette. It should come up directly and sign on:

```
22k CP/M vers 2.0 on Micropolis Disc
Copyright 1979, by Bruce R Ratoff
A>
```

If this does not happen, you have made some error and should re-check all of the above steps. In order to start over, you will have to make another working copy of the release diskette, since the above steps will have altered this one.

If CP/M signs on, you have a 22K system which will properly drive your console. If you have selected one of the "monitor" drivers, you probably do not need any custom patches. See the special sections for SOLOS and SMB users to verify this. If you have selected a "port" driver, you will want to use the procedure which follows to modify CBIOS.ASM to include printer, reader and punch drivers.

Second-Level System Generation

The program 'MOVCPM' which is described in the Digital Research documentation will generate all parts of a new CP/M system. In addition to the features documented in the Digital Research documentation, the version of MOVCPM supplied for your Micropolis drives contains the ability to insert the correct console routines from the list on page 5. This is done by preceding the memory size argument by the correct configuration **letter**, followed by a colon (:), followed by the memory size with no intervening spaces, followed by a space, followed by an asterisk (*). For example, the following commands will generate and save a 32K system with console I/O for a Processor Tech 3P+S:

```
A>MOVCPM E:32 *
Constructing 32k CP/M vers 2.0
ready for "SYSGEN" or
"SAVE 40 CPM32.COM"
A>SAVE 40 CPM32.COM
```

The CBIOS and bootstrap routines have been supplied for a 22K system. This matches the CP/M release diskette, which is also initially a 22K system. It is always necessary to operate a 22K system when CP/M is first received. This system is then used to generate larger systems.

Modifying Your System

If you need to make any special patches, you should use MOVCPM to generate a standard system of the right memory size, and then use DDT to modify that system. The patched system can then be written into the system area of a diskette by using SYSGEN. The complete process is described below:

The main idea in creating a new CP/M system is to lay out the exact contents of diskette tracks 0 and 1 of the new system diskette in the transient program area. By convention, this layout starts at address 900 hex, and extends for 8K, to address 28FF hex. This leaves enough room in the TPA for the program SYSGEN, which will eventually be used to write the new system onto tracks 0 and 1 of a diskette, to load under the new system image. The following list relates the important memory addresses and diskette locations:

<u>Address</u>	<u>Track/Sector</u>	<u>Program</u>
0900	0/1 - 0/2	BOOT
0A00	0/3 - 1/14	CP/M (less CBIOS)
2000	1/15 - 1/32	CBIOS

Note that these addresses have nothing to do with the actual RAM addresses at which these programs execute, but simply establish the relative locations at which they are stored on the diskette.

Start the generation procedure by using the program MOVCPM.COM to relocate a fresh copy of CP/M. To do this, bring up your original system and type:

A><u>MOVCPM ?<u>:xx *

Where 'xx' is the desired new system size in decimal K, and '?' is the configuration **letter** chosen from the table on page 5. The following messages should appear:

```
Constructing xxk CP/M vers 2.0
ready for "SYSGEN" or
"SAVE 40 CPMxx.COM"
```

At this point, save the partially generated system by typing:

A><u>SAVE 40 CPMxx.COM

Where, once again, 'xx' is the new CP/M system size.

Any necessary patches are added by using DDT to read in CPMxx.COM and modify it, and then re-saving the modified memory image. The only 'catch' is that in each case, an offset or bias must be used to locate each patch in its correct place in the map described above, rather than inserting it at its eventual execution address. A method of determining the correct offset will be described below. Note that you should only run DDT if you need to make patches to the standard system. Otherwise, you may skip the next few steps, and go directly to the SYSGEN program.

Simple patches may be made directly on the system image. The patch addresses for simple printer, reader and punch drivers are provided in the following table:

<u>Purpose</u>	<u>DDT address</u>	<u>22K sys address</u>
Printer status port	244A	4F4A
Printer AND mask	244C	4F4C
Printer XOR mask	244E	4F4E
Printer data port	2454	4F54
Punch status port	2457	4F57
Punch AND mask	2459	4F59
Punch XOR mask	245B	4F5B
Punch data port	2464	4F64
Reader status port	2467	4F67
Reader AND mask	2469	4F69
Reader XOR mask	246B	4F6B
Reader data port	2472	4F72

More complex patches will be easier to do by editing the changes into CBIOS.ASM, assembling it, and using DDT to read the altered CBIOS in on top of CPMxx.COM. The procedure for doing this is described both here and in the CP/M Alteration Guide.

Start by invoking DDT with the newly relocated copy of CP/M by typing:

A>>DDT CPMxx.COM

The offset for CBIOS may be calculated using the hex calculation feature of DDT, and is given by the formula:

$$(i + s) - m,$$

where: i is the base of CBIOS in the DDT system image (2000)
s is the total size of CBIOS (0D00)
m is the memory size of the system being patched

Note: Part of CBIOS is uninitialized data areas which are not booted, but which must be considered in the total system size. The booted portion of CBIOS is 0900 hex bytes long. Over 1K of bootable space is unused on the release diskette and may be used to incorporate any special drivers you wish to have in your CP/M.

For example, for a 24K system (6000 hex bytes of memory), type:

-H2D00 6000
8D00 CD00

The second number (CD00), is the result of (2000+0D00)-6000, and is the correct load offset for a 24K CBIOS. The offset for BOOT is always 100 hex greater than the offset for CBIOS. In this case it is CE00. You usually will not need to change BOOT, since it contains no configuration-dependant code. Continuing the example, to read a re-assembled 24K CBIOS into a CP/M system currently in memory under DDT, type the following:

-ICBIOS.HEX (this sets up CBIOS.HEX for input)
-RCD00 (read CBIOS.HEX with offset C580)

Using the same offset, enter any machine-language patches required and return to CP/M command mode by typing control-C. The complete new system image should be saved for future use:

A>>SAVE 40 CPMxxC.COM ('xx' is memory size, as before)
A>

The following procedure may now be used to call the system image back into memory and write it onto the system area of a diskette:

A>>SYSGEN CPMxxC.COM
Sysgen version 2.0
Destination drive name (or return to reboot)B

Mount a diskette on your second drive, and type a carriage return. The new system will be written from the TPA into the system area of the diskette. Note that any files on the diskette will remain intact, since SYSGEN does not affect anything but the bootstrap area (system area) of the diskette. No data files are ever lost by doing a SYSGEN. After writing the new system out, SYSGEN will again ask:

Destination drive name (or return to reboot)

A carriage return at this point will reboot the system from drive A. To execute the newly-created system, mount the newly-created system diskette on your first drive and execute the Level 0 Bootstrap, as described previously under the heading "Booting Up CP/M". If the system fails to come up as expected, go back and carefully check all the preceding steps. It is usually a good idea to retain a working copy of your existing system until you are certain that your new one works correctly.

The final step is to use PIP to copy those utility programs which you expect to need onto the new diskette. You may find it useful to write a SUBMIT file which does this for you. This is explained in the Digital Research manuals.

Bringing Up CP/M Using ZAPPLE and/or the TDL SMB

There are advantages to using ZAPPLE with CP/M. These are found mainly in the device-handling capabilities of ZAPPLE. There is, however an additional consideration in generating your CP/M system. ZAPPLE requires that the top page of your lowest contiguous block of memory be untouched by all other programs in your system. This area contains a routine used by the 'G' command, as well as storage for all registers and ZAPPLE's stack. Micropolis CP/M normally resides in the top 9K of its nominal memory size. If a CP/M system as large as your lowest contiguous block of memory were created, it would overlap the page of memory required by ZAPPLE. Thus, if ZAPPLE and CP/M are to be used in the same system, the largest possible CP/M system is decreased by 1K. For example, if you have RAM in the first 32K of your machine (not counting the RAM used for ZAPPLE, if any) then the largest CP/M system you can generate is 31K.

If you specify configuration Z (1A) when configuring Micropolis CP/M, then all non-disk I/O will be performed by calling the generic routines contained within ZAPPLE. You may use the 'A' command within ZAPPLE to change your device assignments at any time, and CP/M will use the newly-assigned devices. If you modify ZAPPLE to store the IOBYTE at location 3 in memory, you may use STAT to perform the assignment function, as described in the CP/M Features and Facilities manual.

Bringing Up CP/M Using SOLOS

If you specify configuration S (13), CP/M will use SOLOS for all non-disk I/O, as would be required on a Processor Technology SOL computer. For each CP/M logical device, the correct two bits of the CP/M IOBYTE are picked up and the SOLOS routines AINP or AOUT are called. This permits the equivalent of the SOLOS 'SET' command to be performed using STAT, as described in the CP/M Features and Facilities manual.

Bringing up CP/M on the Sorcerer

Unfortunately, the Sorcerer monitor provides no simple means to decode the CP/M IOBYTE which would be useful in all systems. Therefore, if you specify configuration X (18), CP/M will simply use the Sorcerer keyboard and CRT drivers for console I/O. The initialize routine will call the Sorcerer CRT init routine. All other CP/M devices such as a printer must be patched in using the previously given system modification procedure.

Additional CP/M Utilities

MFORMAT

The design of Micropolis CP/M requires that each sector of a new diskette be written by CP/M at least once before it may be read. When you use MCOPY, this happens automatically. When you wish to start with a 'clean' diskette and write to it file-by-file (such as with PIP or BASIC), MFORMAT provides the means of doing this initial write. Mount a scratch diskette in either drive and type:

A>MFORMAT

MFORMAT vers 2.0

Enter drive to format (return to reboot) B

Mount disk to be formatted on drive B (Mod II)

then type return <ret>

Format complete

MFORMAT vers 2.0

Enter drive to format (return to reboot) <ret>

A>

Note that MFORMAT automatically detects whether you have a Mod I or Mod II drive, and indicates this in its prompt message.

MCOPY

This is a whole-track diskette copying program. The copy is always from drive A: to drive B:. The destination diskette is re-read after each track is written, to verify a good copy. In the event of errors, the user is given the choice of continuing or aborting the copy operation. The user has the choice of a SYSTEM copy, which copies only tracks 0 and 1 (the system area), a DATA copy, which copies from track 2 to the end of the diskette (the directory and file area), or ALL, which copies the entire diskette. The program automatically determines whether you have a Mod I or Mod II system, sets the maximum track number for DATA and ALL accordingly, and indicates the selection in the program's sign-on. A sample dialogue follows:

A>MCOPY ALL

MCOPY version 2.0 Mod II

Source on A

Destination on B

then type <ret> <ret>

Successful copy

MCOPY version 2.0 Mod II

Source on A

Destination on B

then type <ret> ^C

Rebooting, type <ret> <ret>

A>

BASIC-E and RUN-E

This is a fairly powerful BASIC compiler/interpreter combination with disk data file handling capability. The complete users' manual for it is available from several sources, including Computer Mart of New Jersey and Digital Research, for a nominal charge. Please note that the BASIC-E compiler requires at least a 26K CP/M system, and therefore will produce the CP/M message 'LOAD ERROR' on the 22K starter system provided. This error will vanish when a larger system (26K or greater) has been generated and installed.

Note: Due to insufficient diskette space, BASIC-E and RUN-E have been omitted from Mod I release diskettes. These programs may be obtained separately at a nominal charge from your computer store.

Any questions, comments, or suggestions relating to the use of this software should be directed to:

Bruce R. Ratoff
80 Gill Lane, Apt 1B
Iselin, NJ 08830
201-283-2724

Post Office Box 579, Pacific Grove, California 93950, (408) 649-3896

CP/M 2.0 has arrived! It's a brand-new upward compatible operating system built to match the low-cost high-capacity disk subsystems which are appearing on the market. We have a deluxe multiprogramming version, called **MP/M**, available as well. CP/M 2.0 provides unprecedented facilities for manufacturer and end-user reconfiguration, and provides data file management up to 128 megabytes. The new CP/M BDOS is completely table driven, so you can define as many as 16 logical drives of up to 8 megabytes each, containing any (reasonable) number of directory entries on each drive. Any file can be as large as 8 megabytes, and we've reserved enough room for eventual expansion up to 32 megabytes in future 2.0 releases. In addition to powerful new random access features, CP/M 2.0 provides separation of files by a user number tag and read-only protection of individual files. We've been using hard disk versions of CP/M for over a year now and know you'll be amazed at the increased performance and capabilities of these disk subsystems under CP/M 2.0. Best of all, CP/M 2.0 is upward compatible from release 1 for existing application programs which use sequential access, and provides random access compatibility with existing programs to 512K bytes per file.

Field reconfiguration of CP/M is one of its major strongpoints. Similar to earlier releases of CP/M, version 2.0 uses a BIOS (Basic I/O System) which defines the exact manner in which sectors are read and written. Version 2.0, however, includes a "disk parameter block" which encodes your particular selection of track sizes, reserved tracks, directory sizes, data block sizes, and sector skew factors. The disk parameter block can be constructed by hand, using the formulas given in our documentation, or it can be built automatically using the DISKDEF macro library, provided with CP/M 2.0, and processed using our MAC macro assembler. In fact, the DISKDEF macro library includes algorithms which produce sector skew tables so that you can easily experiment with various parameters to optimize the speed of your particular CPU, memory, and disk subsystem.

The new random access facility of CP/M 2.0 lets your programs instantaneously access any one of the 65536 records of an 8 megabyte file, with data blocks allocated only where data is written. Additional system calls allow you to read a particular record, write a particular record, determine the virtual size of a file, and compute a random record position from a file which is being sequentially read or written. The directory "extents" are transparent in random access and can individually address up to 128K bytes, removing the previous limitation of 16K bytes per extent. These new facilities, coupled with the increased speed of 2.0 directory and data access algorithms, should result in some very interesting applications!

Various enhancements appear in the CP/M 2.0 modules and utilities. STAT and PIP contain commands which display or manipulate disk characteristics, file attributes, and user numbers. Further, the CCP lists multiple directory entries on the same line (due to the distinct possibility of having many more active entries!), while console line editing additionally supports CRT devices which allow a backspace-overwrite operation so that command lines and operator data input is more legibly displayed.

New documentation for CP/M 2.0. includes the "CP/M 2.0 System Alteration Guide," and the "CP/M 2.0 Interface Guide" which provide customizing and system call information. Registered CP/M 1.4 users receive these two additional manuals when they order the "CP/M 2.0 Update Kit" listed in the order form below, with shipment commencing in mid-August. Conversion from CP/M 1.4 to 2.0 requires familiarization with assembly language coding, and thus you may wish to contact your original CP/M supplier for a customized version. It is important to emphasize that we continue to distribute and support our standard 1.4 release, due to its widespread availability and acceptance.

A new multiprogramming monitor for 8080, Z80, and 8085 processors, called MP/M 1.0, is also being introduced at this time. MP/M is largely upward compatible from CP/M, and uses the file structures of the 2.0 release. MP/M's advanced features include multiterminal support, multiprogramming at each terminal, and bank-switched memory support. Through its concurrency, MP/M can serve as the nucleus of a variety of loosely-coupled peripherally-connected processors.

The MP/M multiprogramming nucleus implements a general-purpose message queuing mechanism for synchronizing prioritized cooperating sequential processes. I/O devices can operate either interrupt driven or polled mode, with a fundamental requirement for timer interrupt generation at 1/60th of a second intervals. MP/M occupies at least 16K of the high portion of a memory bank, with the remainder of memory divided into absolute and relocatable segments. Absolute segments run standard CP/M "COM" files which have sufficient stack space for interrupt processing and which do not make unauthorized direct BIOS calls. Relocatable segments run "page relocatable" programs with the "PRL" file type. A simplified system generation procedure combines the MP/M modules and selected system processes, such as the spooler, real-time scheduler, and user-defined processes into a single relocatable module which is loaded upon cold start.

Due to MP/M's advanced features, it is inherently more complicated to adapt to a particular hardware configuration. Unless you are a rather adventurous individual with assembly language experience, you should contact your hardware supplier for prices and availability of the MP/M system for your particular computer system. MP/M shipment commences on September 1, 1979, and may be ordered using the form given below.

We have some products in the works which will be of interest to many of you. The PL/I compiler is in its final stages of development. The I/O system is being added, and we are commencing the linking loader implementation. We'll keep you informed on our progress, and announce its availability in a future newsletter. We have a new version of our powerful SID symbolic debugger, called ZSID which uses the Zilog Z80 mnemonics, rather than the 8080 instruction set. We are in the process of converting MAC to process a ZSID compatible instruction set in order to provide a complete program development environment. ZSID will be available for shipment in September - please call Digital Research for price and availability if you are interested.

O R D E R F O R M

Please send me the following CP/M or MP/M packages (all software shipped on standard 8" diskettes):

- | | | |
|--------------------------|---|----------|
| <input type="checkbox"/> | CP/M 2.0 System Diskette and Documentation | \$150.00 |
| <input type="checkbox"/> | CP/M 2.0 Update Kit, System Diskette, Two Manuals
(available only to registered 1.4 owners: serial# _____) | \$100.00 |
| <input type="checkbox"/> | MP/M 1.0 System Diskette and Documentation | \$300.00 |
| <input type="checkbox"/> | MP/M Documentation only | \$ 25.00 |

Total Purchase _____

California residents please add 6%, Canada and Mexico add \$5.00 shipping, and overseas customers please add \$10.00 shipping.

- I have enclosed a check or money order in the amount \$ _____
- Please bill the amount \$ _____ to my
- Mastercharge Charge Card # _____
- Bankamericard Card Expires _____/____

Name _____ Address _____

City _____ State _____ Zip _____