iCOM - CP/M

VERSION 2.0

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

SOFTWARE AGREEMENT

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 disk 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 iCOM Frugal Floppy, 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 signon. You may use this facility to automatically load a monitor or driver package which is too large to fit into the limited diskette 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 new PROM supplied with CP/M, at address p+3CC hex, where 'p' is the starting address of the PROM in your system.

The Level 1 Bootstrap is a program which loads CP/M. It is stored in track 0, sector 1 of a CP/M diskette. The Level 0 Bootstrap loads this program at address 0 in memory, and then jumps to it. The Level 1 Bootstrap loads CP/M into the top 7K of the configured system, and jumps to the CBIOS. Note that the Level 1 Bootstrap must be aware of the size of the configured system, in order to know where to load CP/M. This means that the Level 1 Bootstrap is altered whenever a different sized CP/M system is generated.

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 (^).

Installing CP/M

The first step in bringing up CP/M is to install the new interface PROM which has been provided in this package. This PROM contains, in addition to all the FDOS support routines, the level 0 bootstrap routine for CP/M. Note that the level 0 bootstrap is the only portion of CP/M which makes use of the iCOM PROM. This means that the user is free to make use of whatever shadow or phantom features his system hardware may provide, and may elect to provide his own level 0 bootstrap. The only requirements on this routine are that it read track 0 sector 1 into memory locations 0000 thru 007F hex, and then jump to location 0000.

The PROM is installed as follows:

Either work on a metallic, conductive surface or cover your work surface with aluminum foil or a metal tray. This will decrease the likelihood of static electricity, which can damage the PROM when it is handled.

With your computer turned off, remove the iCOM interface card from your computer system and lay it, component side up, on your work surface. Lay the new PROM, still in its protective black foam, next to the interface card.

Locate the existing type 2708 PROM on the interface card. It should be the 24-pin IC on the lower right corner of the interface card. Observe the orientation of the PROM in its socket, making note of which direction the marked corner of the PROM is pointing in. This marking will either consist of a dot, a notch, or a numeral '1'. Carefully unplug the PROM from its socket, and place it in aluminum foil or on black conductive foam for safekeeping.

Following the same orientation as the PROM you just removed, carefully remove the new PROM from its conductive foam and insert it into the socket on the interface card.

No further hardware modifications are required. You may now return the iCOM interface card to your computer, reconnect the disk drive cable, and power up your system.

Booting up CP/M

The procedure for booting CP/M is very simple. With your computer, terminal, and drives powered up, insert the iCOM-CP/M release diskette in drive 0 and execute at the iCOM-CP/M boot address of C3CC hex (BBCC for Sol systems). The disk drive head will load with an audible click and the ready light will remain lit for one to three 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 4A00 hex. CP/M should then sign on:

```
20k CP/M vers 2.0
A>
```

That's all it takes - you're now running CP/M !

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 DCOPY utility to make a backup system diskette as follows:

```
A>DCOPY ALL                      (this loads the DCOPY program)
SOURCE ON A                      (leave the release diskette on drive A)
OBJECT ON B                      (put a blank diskette in drive B)
THEN TYPE RETURN <ret>           (type a carriage return)
FUNCTION COMPLETE                (appears when finished)
SOURCE ON A
OBJECT ON B                      (program loops back to beginning)
THEN TYPE RETURN ^C              (control-C will break the loop)
REBOOTING, TYPE RETURN <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.

iCOM-CP/M Console Codes

Number	Letter	Configuration
		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:

		Configuration						
Parameter (hex)	Addr	A	B	C	D	E	F	G
Input Status port	4CC2	10	03	00	00	00	43	10
Input Data port	4CDA	11	02	01	01	01	42	11
Output Status port	4CE9	10	03	00	00	00	43	12
Output Data port	4CF9	11	02	01	01	01	42	13
Input AND mask	4CC7	01	02	01	20	40	02	80
Input XOR mask	4CC9	01	02	00	20	40	02	80
Output AND mask	4CEB	02	01	80	02	80	01	80
Output XOR mask	4CED	02	01	00	02	80	01	80
Initialization	4D36	Y	Y	N	N	N	Y	Y

The addresses given above are for the 20K 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 2A80 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 further edification.

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

```
A>MOVCPM ?:20 *                    (where '?' is the configuration
Constructing 20k CP/M vers 2.0     letter for your console from the
ready for "SYSGEN" or              table on page 6)
"SAVE 35 CPM20.COM"
A>SYSGEN                      (run SYSGEN to install system)
Sysgen vers 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:

```
20k CP/M vers 2.0
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 20K 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 iCOM drives contains the ability to insert the correct console routines from the list on page 6. 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 35 CPM32.COM"
A>SAVE 35 CPM32.COM
```

The CBIOS and bootstrap routines have been supplied for a 20K system. This matches the CP/M release diskette, which is also initially a 20K system. It is always necessary to operate a 20K 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 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 6-1/2K, to address 22FF 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:

Address	Track/Sector	Program
0900	0/1	BOOT
0980	0/2 - 1/19	CP/M (less CBIOS)
1F80	1/20 - 1/26	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>MOVCPM ?:xx *

Where 'xx' is the desired new system size in decimal K, and '?' is the configuration letter chosen from the table on page 6. The following messages should appear:

```
Constructing xxk CP/M vers 2.0
ready for "SYSGEN" or
"SAVE 35 CPMxx.COM"
```

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

A>SAVE 35 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:

Purpose	DDT address	20K sys address	
Printer status port	227F	4CFF	01
Printer AND mask	2281	4D01	01
Printer XOR mask	2283	4D03	01
Printer data port	2289	4D09	00
Punch status port	228C	4D0C	
Punch AND mask	228E	4D0E	
Punch XOR mask	2290	4D10	
Punch data port	2299	4D19	
Reader status port	229C	4D1C	
Reader AND mask	229E	4D1E	
Reader XOR mask	22A0	4D20	
Reader data port	22A7	4D27	

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 patches to BOOT is 900 hex. This offset is fixed, since BOOT will always have an origin of 0.

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 (1F80)
- s is the total size of CBIOS (0600)
- 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 0380 hex bytes long.

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

```
-H2580 6000
8580 C580
-
```

The second number (C580), is the result of (1F80+0600)-6000, and is the correct load offset for a 24K CBIOS. Continuing this 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)
-RC580                   (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 35 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. CP/M normally resides in the top 7K 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 iCOM/CPM, 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 you 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.

You may enter ZAPPLE from CP/M at any time by typing:

```
A>DDT
-GF01E
```

You may return from ZAPPLE to CP/M by typing:

```
>G0
```

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. A different version of the PROM is provided, which is addressed at B800 instead of the usual C000, so that it does not conflict with the SOLOS PROMs at address C000. 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.

Additional CP/M Utilities

FDOS-CPM and FDOS-DIR

These two utilities are used to transfer programs from an iCOM FDOS-II or FDOS-III diskette to a CP/M diskette. FDOS-DIR will display the directory of the FDOS diskette on drive B: by typing:

```
A>FDOS-DIR B:
```

Having used FDOS-DIR to identify the files you wish to transfer to CP/M, you may copy them onto the CP/M diskette in drive A: from the FDOS diskette on drive B: by typing:

```
A>FDOS-CPM destname.typ B:srnam
```

Where 'destname.typ' is the CP/M destination filename and 'srnam' is the FDOS source filename.

DCOPY

This is a whole-diskette copying program which serves the same purpose as the 'COPY' command in FDOS. The user has the option of copying the entire diskette, just the system tracks, or just the data tracks. The commands are DCOPY ALL, DCOPY SYSTEM, and DCOPY DATA, respectively. The copy is always from drive A: to drive B:. A sample dialogue follows:

```
A>DCOPY ALL
+SOURCE ON A
+OBJECT ON B
+THEN TYPE RETURN (mount desired diskettes and type return)
+FUNCTION COMPLETE
+SOURCE ON A
+OBJECT ON B
+THEN TYPE RETURN (type control-C to break this loop)
+REBOOTING, TYPE RETURN (mount a system diskette and type return)
A>
```

BASIC-E and RUN-E

This is a fairly powerful BASIC compiler/interpreter combination with diskette 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 24K CP/M system, and therefore will produce the CP/M message 'LOAD ERROR' on the 20K starter system provided. This error will vanish when a larger system (24K or greater) has been generated and installed.

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