DOS/65 BRINGING THE SYSTEM UP

VERSION 2.1

Table of Contents

VERSION 2.1

SECTION 1 - INTRODUCTION

This note discusses the procedure to be used to bring DOS/65 up on a new system. Related material can be found in the S-100 INTERFACE GUIDE, NON-STANDARD CONTROLLER NOTES, and in the SYSTEM INTERFACE GUIDE. It is strongly suggested that those documents be carefully read before any attempt is made to bring the system up.

SECTION 2 - PREPARATION

2.1 DOS/65 CONFIGURATION

Probably the most important step in the whole process of bringing DOS/65 up is to order the system in the correct configuration. The key parameters which must be set to ensure DOS/65 to user compatibility are:

Controller Location

TEA Start

Console I/O Locations and Characteristics

Drive Peculiar Software

Since the ORDERING INFORMATION does not explain these parameters, the following sections explain the critical aspects of each parameter.

2.1.1 CONTROLLER LOCATION

This parameter is one of the single most critical parameters.

RECOMMENDATIONS

1. Do not locate the controller in pages zero or one. No real advantage results if it is and the resulting conflicts with DOS/65 or transient memory usage will create enormous headaches.

2. Do locate the controller such that it need not move as more memory is added.

3. The logical outgrowth of the first two recommendations is that the controller should either be between page one and the TEA start (if any room exists) or above the highest RAM location ever contemplated for purchase or use.

4. Do check out the controller using the information in the DOS/65 documentation package as well as the data in the controller manual before trying to use the DOS/65 diskette.

2.1.2 TEA START

This parameter will normally be set to a very obvious value. The Ordering Guide lists
several values which are closely tied to specific systems as shown in the following lists:

TEA START	SYSTEMS
$200	CUSTOM, OSI, SYM-1, AIM65, TIM
$400	Special
$800	Commodore & Apple II
$1000	TIM
$1400	Rockwell
$2000	KIM-1

These relationships are based upon the beginning of "user RAM" in each system. The key
requirement is that the loading of a transient beginning at the TEA start and then
execution of that transient by JMPing to the TEA start will not wipe out the users system
(i.e., I/O).

One note of interest - the lower you set TEA the better off you are from several points.
Specifically the lower you set TEA start:

1. For a given memory size there will be more usable memory.

2. More software will be available for your use.

While the first item is obvious, the second may not be. While a KIM system is limited to a
TEA start of $2000 or more, other systems could use software originally designed for the
KIM even when using a much smaller TEA and thus need not use $2000 as TEA start. In
the simplest possible use of this approach, the following program could be used for a TEA
of $200 to execute KIM software.

```
*     =     $200
      JMP   $2000
      .END
```

A KIM program (Microsoft BASIC for example!) could be loaded via cassette at $2000 and
then saved using the SAVE command. At that point the following steps could be used to
create an executable program.

1. Using EDIT create the above patch.

2. Assemble the patch using ASM.

3. Run DEBUG using the following sequence of commands:

```
i PROGRAM.COM
r
i PATCH.KIM
r
(ctl-c)
```

4. SAVE the resulting file. Clearly for the 3 byte patch shown one need not actually create and assemble the code, it could just as easily be entered using the DEBUG memory modification features. If that patch is more complex (i.e., contains DOS/65 console and disk I/O interface routines and buffers) then the only practical approach is the four step (followed by test) process shown above.

2.1.3 CONSOLE I/O LOCATIONS

Specifying this portion of your DOS/65 system may be more difficult than initially imagined. The points to be remembered about the console I/O routines are:

1. All characters (input or output) are passed in the A register.

2. All registers (A, X, or Y) may be altered by the routines.

3. ASCII is the only allowable mode.

4. Upper or lower case can be used. (When operating in CCM any a to z input to CCM is converted to A to Z for internal use but is echoed as the lower case character.)

5. The MSB of each character must be zero for the input function.

6. The Console Ready function is absolutely necessary and must return a non-zero value if and only if an input is ready.

7. All inputs will be echoed by DOS/65 (control characters other than return ($D) or linefeed ($A) are not echoed). Thus the routine in (or called by) SIM for console input should not echo any inputs.

Definition of your console I/O routines while complying with all those conditions is not

necessarily all that difficult. For some systems the routines are either already there or are easily created. The SYM-1, for example, has three I/O vectors which provide, with some slight modifications for the READY function, the necessary functions.

CAUTION
For some commands PEM saves and restores any data in page zero locations $02 through $0F upon entry and exit respectively. Similarly, some standard SIMs save and restore the contents of some page zero locations during disk read and write operations. In addition, various standard transients use parts of page zero for data storage. Make sure that if at all possible your system I/O routines do not use page zero. If this is not possible, then use only the very highest page zero locations. See the SYSTEM INTERFACE GUIDE for a discussion of restricted locations in page zero and page zero usage by transients.

2.1.4 DRIVE PECULIAR SOFTWARE

The standard SIM disk I/O routines assume that the drives used are Shugart 800 or 801 devices or equivalent for SIF A and D and are "Industry Standard" drives (e.g., Shugart 400, TEAC FD50A, MPI B-51) for SIF B and C. No attempt has been made to include the provisions, if any, for other drives (such as PERSCI) or for double sided or double density operation which may require different drivers. For your use, I have included a listing from the Tarbell Public Domain disk of the 8080 source code for the Tarbell BIOS (equivalent to SIM). If you require such software, it is suggested that you follow the steps in the NON-STANDARD CONTROLLER manual in order to get your BOOT and SIM routines working and installed.

2.2 DISK SUBSYSTEM TESTING

Before inserting your DOS/65 master diskette into Drive A and attempting to boot the system, please - please test the drive using some small single or multiple sector read and write routines and a blank but formatted diskette.

Those routines can be extracted from the SIM listing or they can be your own creation. Regardless of how you got them, let me suggest the following sequence of tests:

1. Test for proper response to a system RESET.

2. Test for proper response to step-in and step-out commands.

3. Test for proper response to a seek command.

4. Test for proper response to a restore command.

5. Test for proper response to a read sector command.

6. Test for proper response to a write command then loop back to step 5 (or earlier) to verify that data was correctly written to the proper track and sector.

The message here is that one should not be too anxious to try booting the system.

SECTION 3 - EXECUTION AND TROUBLESHOOTING

3.1 EXECUTION

If everything is OK up to this point; now is the time to reset your system, insert the master diskette in Drive A and execute your LOADER. If all goes well, you should hear the drive engage for a couple of seconds and then see a message similar to following on your console output device (if not, go to Section 3.2).

```
16K DOS/65 V2.14
SIF A - TARBELL - STANDARD VERSION
HOW MANY DRIVES (1 TO 4)?
```

Type the number of drives (e.g., 2). The system should now type the following on the console output device (if not, go to Section 3.2).

```
A>
```

At this point you are operating in the CCM portion of DOS/65 and are up and running. It is suggested that you now take the following steps:

1. Make a backup copy of your system diskette by entering the command COPY ALL and then following the instructions printed on your console.

2. If necessary, modify SIM and BOOT using EDIT; assemble it using ASM; and then create your new system using SYSGEN. Until fully checked out, the new system should only be written to a copy of the master diskette, not to the original.

3.2 TROUBLESHOOTING

Obviously I cannot address every possible contingency but the following hints may help you solve any problems.

PROBLEM	POSSIBLE CAUSES
Drive does not engage when executing LOADER	a. No diskette b. Wrong drive c. Drive power not on d. Loader Not Correct
Drive Engages for Short (< .1 sec.) time but no console output	a. Diskette does not contain DOS/65 system b. BOOT not correct c. System does not have RAM at SIM address
Drive Engages for Long Time (> 1 sec.) but no console output	a. BOOT not correct b. Read error c. System does not have RAM at SIM address d. Incorrect console output routine e. Incorrect I/O initialization f. Incorrect interrupt handling
No Response to Input of Number of Drives	a. Incorrect console input routine b. Incorrect interrupt handling
Number of drives input is echoed but A> is not printed on console	a. Incorrect interrupt handling
No response to inputs after A> is printed	a. Incorrect interrupt handling
(ctl-c) does not cause successful system WARM BOOT	a. Console input routine does not allow control characters b. Incorrect SIM disk read routines
TYPE, DIR or transient execution do not work	a. Incorrect SIM disk read routines
REN, ERA or SAVE do not work	a. Incorrect SIM disk write routines