DDSC #1

THE DOS MOVER

Relocation Routines For Moving The Standard

NORTH STAR DISC OPERATING SYSTEM

I. Purpose

This software package is designed to allow relocation of the standard issue of North Star DOS (at 2000H), including I/O patches, to any new starting address. These routines will only work on those systems using the standard DOS at 2000H and the control ROM at E900H. These relocated versions of DOS are not replacements for the standard DOS. They will not work directly with the standard issue of North Star Basic (without patching) since it uses the I/O jump table of the DOS at 2000H. However, they are likely to be very useful for those who wish to use their disc system with software that would normally overlap the standard DOS at 2000H. Such programs include most of the versions of Basic available for 8080/Z-80 based systems as well as many machine language games and utilities.

II. Requirements

A. Hardware requirements

A 8080/Z-80 based computer system with a North Star Microdisc system or a Horizon system with control ROM at E900H configured for the standard DOS at 2000H. During relocation, there must be continuous RAM from 2000H to 3FFFH.

B. Software requirements

A recent release of the standard North Star DOS (at 2000H). The DOS MOVER routines are known to work for DOS version 2, release 3 and Horizon DOS version 3.1. A program is provided to test your DOS for compatibility with the relocation routines.

C. Human requirements

It is likely that any person that is familiar with the use of their computer and some simple 8080 machine coding can use this software package.

III. Introduction

The standard issue of North Star Dos (STDOS) is located at 2000H with a control ROM at E900H. This ROM contains the the main disc control routines as well as a bootstrap loader to load the standard DOS. This loader loads the stack at 2114H and then loads the sector at disc address 4 into memory starting at 2000H. It then jumps to 2004H, to a routine that loads the remaining sectors of STDOS. Some of the routines within the ROM use a 4 byte buffer starting at 2000H for keeping track of the

disc drive head positions of the various (up to 3) drives on the system. Thus, it is apparent that a relocated version of DOS must not use the ROM routines and it cannot be booted up directly with the ROM (without altering the ROM).

The DOS MOVER routines solve these problems by creating relocated versions of DOS that do not use the ROM. This is done by appending relocated versions of some of the ROM routines to the relocated DOS. Thus these relocated versions of DOS are 11 blocks long (instead of 10 blocks for STDOS). These relocated versions would normally be loaded from STDOS (or another relocated DOS). However, a short boot routine is provided that allows one to boot up indirectly from the ROM bootstrap with a relocated version of DOS.

The DOS MOVER diskette supplied contains 5 files:

1. DOS — A blank file for the user's DOS.
2. CHKR — A program to test your copy of STDOS for compatability with the relocation routines.
3. BOOT — A routine used to make a relocated version of DOS boot up with the ROM bootstrap.
4. CRDOS1 — The DOS relocator creation program.
5. CRDOS2 — Same as CRDOS1 but for the most recent version of STDOS such as used on the Horizon system.

The format of the remaining documentation consists of simple descriptions and examples of the use of the DOS MOVER routines. The appendix contains a more detailed description of the workings of the relocation process. It is suggested that the user read all of the material before using any of software provided. In the examples provided, the underlined commands indicate user inputs to program prompts.

IV. Program Descriptions and Examples

A. CHKR

This program is used to test your copy of STDOS for compatability with the relocation routines. Incompatability can result from very old versions of STDOS, extensively modified versions, or very new versions of STDOS released after your copy of the DOS MOVER was made.

To use the program, load your copy of STDOS and :

1. Run CHKR with the GO command.
2. Answer the prompt with the starting address of STDOS (see the appendix for extended capabilities).

CHKR will then examine selected portions of STDOS and respond

with a recommendation as to compatability. It then returns to the DOS that loaded it.

---------- Examole, use CHKR to test STDOS ----------

```
*GO CHKR↵

-- DOS CHECKER --
 START ADDRESS OF THE DOS TO BE TESTED : 2000↵

 THIS DOS CAN PROBABLY BE RELOCATED WITH CRDOS1

* ◄— return to calling DOS
```

CHKR is not infallible. It does not check all of the DOS. This was a deliberate compromise chosen because the relocation routines are usually capable of correctly relocating patches made to the main parts of DOS provided no calls to the ROM routines have been moved or added.

B. CRDOS (CRDOS1 and CRDOS2)

This program examines the standard DOS at 2000H (with your I/O patches) and the standard ROM at E900H and copys selected portions of these programs to RAM beginning at 2A00H and relocates the code to create a relocated version of DOS at 2A00H (it can be run at its present location). It also creates a pointer table driven relocation routine behind this DOS (starting at 3500H) which can be used to create relocated versions of DOS for any start address. This combined DOS and relocator at 2A00H is called RDOS and is to be save and used to generate relocated versions of DOS. This two step process was chosen to avoid copyright problems with North Star. It also provides a very reliable method of generating the relocated versions of DOS. Note that the relocated versions of DOS will automatically have all of the I/O routines and patches that were made in your copy of the standard DOS at 2000H.
To use CRDOS (1 or 2), load your STDOS and:

1. Run the recommended (by CHKR) version of CRDOS with the GO command.
2. Note the size of the RDOS that was created.
3. Create a file for RDOS and save it on the disc.
4. Test the relocated DOS at 2A00H (part of RDOS).

---------- Example, CHKR recommended using CRDOS1 -------

```
*GO CRDOS1↵

 RDOS : 2A00H - 38FCH ◄— the size of RDOS for a simple
                         port oriented I/O patch as
                         given in the North Star manual,
*CR RDOS 15↵
                         a more complex I/O patch may
*SF RDOS 2A00↵           give an RDOS 16 blocks long.
```

*JP 2A00↵ ← test the relocated DOS.

---- DOS AT 2A00H ----

*LI 0↵ ← test self identification feature.

---- DOS AT 2A00H ----

* ← other tests of all commands.

After you have verified that the DOS at 2A00H works correctly, reboot your standard DOS (*JP E900) and create a relocated DOS with RDOS. If the DOS did not work correctly, review the possible problems discussed in the appendix.

C. RDOS

This program was created from your copy of STDOS and the standard ROM. It can be used to generate relocated versions of DOS for any new starting address. You should never have to use CRDOS again unless you decide to add new patches to your STDOS that you want transferred to the relocated versions.

To use RDOS:

1. Boot up with STDOS.
2. Create a disc file 11 blocks long to copy the relocated version into.
3. Change the file type to a type 1 file at the specified load address.
4. Load the relocating DOS (RDOS) at 2A00H.
5. Jump to the relocation routine at 3500H.
6. After the prompt, give the desired new start address (the same as used in step 3) and hit carriage return. A control-C will abort the process and return to STDOS at 2028H. Any other non-hex characters will initiate a restart of the relocator. The relocation process is virtually instantaneous. When it is completed, the relocation routine is disabled and the standard DOS is restarted. The program from 2A00H to 34FFH has now been modified so that it will run at the new origin.
7. Save the relocated DOS on the file that you created.
8. Test it by loading it with the GO command.

---------- Example, create a DOS at 6B00H ----------
(this is the highest location allowed if you only have 32K of RAM starting at 0 and you want your DOS to use IN, DT, CF, and CD.)

```
*CR DOS6B 11
*TY DOS6B 1 6B00
*LF RDOS 2A00
*JP 3500
-- DOSXX RELOCATOR --
 NEW START ADDRESS : 6B00
*SF DOS6B 2A00
*GO DOS6B
---- DOS AT 6B00H ----
*LI 0                    <-------- test self identification feature.
---- DOS AT 6B00H ----
*
```

D. BOOT

This program provides a way of booting up with a relocated version of DOS by using the ROM bootstrap routine. This is done by placing the BOOT program at disc address 4 followed by a copy of the relocated DOS. The ROM bootstrap program loads the BOOT at 2000H and it loads the relocated DOS at its specified start address. Note, this operation overwrites the first 256 bytes (and stack area) beginning at 2000H. Thus, this is most likely to be useful when your system is first turned on and there is no useful information storred in the first 275 bytes starting at 2000H.

To use BOOT:

1. Initialize a blank disc and create files for BOOT and the relocated DOS starting at disc address 4.

2. Change the file type of the file for the relocated DOS to a type 1 file at the specified load address (this will allow you to use the DOS independently of the BOOT).

3. Load RDOS and run the relocator.

4. Load BOOT at some convenient location (say 1000H) and patch in the desired start address of the relocated DOS at relative address 37H (1037H). Example: a relocated DOS at 6B00H requires the patch:

```
1037:  00
1038:  6B   .
```

5. Save the patched BOOT on the disc beginning at disc address 4.

6. Save the relocated DOS on the disc beginning at disc address 5.

7. Test the program by jumping to the ROM at E900H. The relocated DOS should load and identify itself.

---------- Example, create a selfbooting DOS at 6B00H -------

```
*IN,                          <-- put in the blank disc
*CR BOOT 1,
*CR DOS6B 11,
*TY DOS6B 1 6B00,
*LF RDOS 2A00,                <-- put in the disc with RDOS and BOOT
*JP 3500,
-- DOSXX RELOCATOR --
 NEW START ADDRESS : 6B00,
*LF BOOT 1000,
                              <-- patch the copy of BOOT and
*SF BOOT 1000,                    put in the initialized disc
*SF DOS6B 2A00,
*JP E900,                     <-- test self boot
---- DOS AT 6B00H ----
*
```

APPENDIX

I. Similarities of STDOS and the relocated versions of DOS.

1. The jump table and read-after-write flag are in the same relative positions (relative address: 04H-2BH).

2. Most of the code and tables from relative address 82H to 09FFH is in the same place. This includes the 256 byte I/O area (relative address: 0900H-09FFH).

3. A modified version of the control ROM is loaded in the 256 byte area above the I/O area (rel. add.: 0A00H-0AFFH). Thus, this version takes up 11 blocks on the disc.

4. The 2560 byte area used for disc copy, test, and initialization was shifted up by 256 bytes to relative address: 0B00H-15FFH.

5. The head position buffer (4 bytes) usually found at rel. add. : 0-3 has been shifted up to relative add.: 5AH-5DH. This was done to put a jump at relative position 0. This allows the DOS G0 command to be used to load and start one of the relocated versions.

6. A sign on message has been added that identifies the start address of the relocated DOS. This feature will be invoked by a jump to its start address or an attempt to list the directory of drive 0 (*LI 0)- which gives the usual error message in the standard DOS. Also, every time that the sign on message is printed out, the head position buffer is initialized to 59H.

II. Comments on the Relocation Process

A. CRDOS

The relocation routines in CRDOS are similar to those found in a disassembler. They operate by scanning the code and comparing suspected instruction bytes to tables of 3 and 2 byte (8080) instructions. Those 3 byte instructions which appear to reference an address within the standard DOS are relocated and their positions are saved in a table starting at 35BCH. This process is not infalible. However, the relocation is done in sections and other routines make up for deficiencies that are known to occure when used on a copy of the standard DOS. This method is quite insensative to the nature of the I/O patches or even patches made to internal positions of the DOS. These patches will be handled correctly provided the following rules are observed:

1. The patches must not involve moving or adding calls or jumps to the control ROM at E900H.

2. They must be done cleanly by using NOPs (00H) in place of fragments of the previous code that are no longer used. Such fragments can throw the relocator "out of step" as it scans thru the DOS.

3. If the I/O routines (starting at 2900H) are separated by any extra space, insure that either all the extra space is filled by NOPs or that at least 3 NOPs preceed the beginning of each of these separated routines. This insures that either the relocator never gets out of step or that it at least gets back in step before it encounters any valid machine code.

4. Any 3 byte 8080 instructions used in your patches will be relocated if the 2 byte operand appears to reference an address within the DOS (2002H-29FFH). There is a way to correct errors made in this way by patching RDOS. However, this is usually unnecessary. Note, this method does not relocate (re-address) calls or jumps to external monitor ROMS (which is the desired response).

5. The patch should not involve moving any part of the DOS command symbol table. However, the command names and jump addresses can be changed without any problems with the relocator.

6. The use of some Z-80 codes not valid for the 8080A is likely to get the relocator out of step. However, errors made in this way can often be corrected by patching RDOS. Horizon DOS does not usually contain any of these instructions.

Most people will probably not have to worry about the above details if they made their own I/O patches from the original copy of STDOS supplied with their system. This copy usually has the I/O area filled with NOPs and no unusual or "unclean" patches in the DOS. The major problem is that some people may never have received a copy of STDOS. Our initial testing of this product uncovered that some computer stores were copying their own patched versions of STDOS over the original versions on the disc shipped with the system. This is often done without telling the customer that it was done or the precise nature of the patches made- even though the label on the disc suggests that it is the original version from North Star. If THE DOS MOVER does not work on your DOS, even though you believe that the above rules were obeyed, we suggest that you ask for some help at the computer store that sold you your system. They may know what part of their patches to fix or supply you with a copy of the original STDOS.

B. RDOS

The relocator part of RDOS works by using a table of over 400 addresses (2 byte pointers) that point to the positions in the relocated DOS (at 2A00H) that require relocation. These points are the operands of 3 byte instructions like LHLD, SHLD, STA, LDA, JMP, CALL, and LXI. This table is at the end of the relocator starting at 35BCH. It was filled by CRDOS as it scanned the code in STDOS. In the event that CRDOS was unable to accomplish the entire relocation correctly (for reasons discussed above), you may be able to fix RDOS. This will require locating the incorrectly relocated code, patching it so that the DOS at 2A00H will work correctly, and adding or disabling pointers in the relocation table. The errors are probably involved with your patches to the code. The pointers associated with the I/O patches typically start at 38F8H unless patches have been made to the main part of DOS. The position of the first empty position in the pointer table is given when CRDOS is finished (38FCH in the examples given earlier). A pointer can be disabled by replacing it by a new pointer to a buffer area in RDOS (whose data is unimportant). Such an address is 3200H. If you must add a new pointer to the table, insert the new pointer at the end of the table and increment the table length word at 351AH:

```
3519:  LXI B, LEN  .
```

As an example of how the table is addressed, note that the first pointer in the table (at 35BCH) points to the operand of the of the JMP instruction at 2A00H:

```
35BC: 01
35BD: 2A

2A00: C3 5E 2A    JMP 2A5EH  .
```

III. Extended features of CHKR and CRDOS

Both CHKR and CRDOS have an unusual feature that may be of some interest. They can both be used from a relocated version of DOS, although CRDOS must still have an example of STDOS at 2000H. They both must be started from a copy of DOS (possibly a relocated version of STDOS). This can be done either by the G0 command or a LF command followed by a JP 2A00. In either of these cases, DOS puts a return address on the stack (DOS origin+82H). CHKR and CRDOS use this address to set their return address as well as to compute the locations of the I/O jump table positions. These features may be of use in a computer store situation where the customer's copy of STDOS (with unusual I/O patches) is located at 2000H. Thus both CHKR and CRDOS can be run from a relocated version of DOS (with the store's I/O) to test and configure a relocating DOS from the customers STDOS. Also, note that CHKR can also be used to test the customer STDOS by loading it at some other location (say 1000H) and then running CHKR from the store STDOS (at 2000H) and giving CHKR the address of the STDOS at 1000H.

IV. Warning on jumping between different versions of DOS

Jumping between different versions of DOS can cause problems because the hardware does not keep track of the drive head positions. This is done by each individual version of DOS in software. When a new version of DOS is first loaded, its head position buffer is filled with 59H which causes an initialization to track 0 (a Home operation) before any other disc operations. If you then jump to another version of DOS that last thought the head was over track 2, it will become confused. You can demonstrate this most directly (and non destructively) by trying to list the directory in such a situation (you get a sort of random dump of characters). Thus, the safe thing to do is reboot between jumps to versions of DOS that have already accessed the disc at least once. Another option is to jump to the system monitor and reinitialize the head buffer with 59H. The relocated versions of DOS do this automatically when the sign on message is printed out.
