DEMONSTRATION INSTRUCTIONS
for
DISK ASSEMBLER and DISK EDITOR

Knowing how eager you probably are to see the Disk Assembler and Disk Editor operate, we have included some demonstration files on the diskette. After running the demonstrations, you should read the enclosed documentation before attempting to edit and assemble any of your own programs.

The diskette supplied by CSD does not have a DOS or BASIC file on it, but empty files have been created to permit you to copy your own personalized DOS and BASIC onto it.

ASSEMBLER DEMONSTRATION

1. Boot up with the North Star Disk Operating System.
2. Insert the diskette supplied by CSD.
3. Execute the assembler by entering "GO ASM".
4. The assembler will display the following.
   User responses are underlined.

```
8080 DISK ASSEMBLER, VERSION 1.2
FROM COMPUTER SYSTEMS DESIGN, WICHITA, KS.
OWNED BY (Owner's name)
SOURCE FILE NAME? S02↵
SOURCE FILE NAME? ↵                    ↵ means carriage return.
LIST DEVICE NUMBER? ↵
OBJECT FILE NAME? ↵
0000 S02                               First pass of assembler

0000 S02                               Second pass
     (Assembly listing)
```

The assembler will assemble the source file and display the assembly listing showing two undefined labels, CHOUT and CR. Now repeat the steps above, but enter S01 and S02 as source file names. CHOUT and CR are defined in S01, so there are no errors this time. You may temporarily suspend the assembly listing by hitting the space bar. Listing is resumed when you hit any other key.

EDITOR DEMONSTRATION

1. Boot up with the North Star Disk Operating System.
2. Load North Star BASIC by entering "GO BASIC".
3. Insert the diskette supplied by CSD.
4. Enter "LOAD EDITOR" and then "RUN".
5. Editing instructions are on the next page.

EDITOR DEMONSTRATION (continued)

User responses are underlined.

```
TEXT EDITOR, VERSION 1.1
BY COMPUTER SYSTEMS DESIGN, WICHITA, KS.
OLD FILENAME? S01
NEW FILENAME? S03
>L30
```

List 30 lines from the current location of the line pointer. The lp is initially set at the top of text. There are only 25 lines in the file, so only 25 lines are listed. Listing does not move the lp.

```
  (25 lines of text listed)
>SDISIM:
DISIM: LXI H,INPBUF
```

Search forward for DISIM: When found, display the line. The lp is now moved to this line.

```
>D
  CALL DISPMS  ;DISPLAY (etc.)
```

Delete this line. The lp is now located at the line following the deleted line. The line at the current location of the lp is displayed.

```
>I
?DISIM: LXI H,INPBUF ;PASS PNTR TO BUFFER
?
```

Insert a new line.

The new line will be in the same place as the deleted line.

```
>M-1
DISIM: LXI H,INPBUF ;PASS (etc.)
```

Move the lp back one line. The line at the current location of the lp is displayed. This is the line just entered.

```
>E
```

End. Saves the edited file in file S03 and returns to BASIC.

```
READY
```

If you now use the editor again, this time specifying S03 as the old file, and S01 as the new file, you can list S03 and verify that the editing carried out above was indeed saved in the new file. Now that you have seen the Disk Assembler and Disk Editor work, you should proceed to read the detailed documentation so that you can learn of the many other features available in the assembler and editor.

COMPUTER SYSTEMS DESIGN's

DISK EDITOR
for the

NORTH STAR MICRODISK SYSTEM

I. PROGRAM DESCRIPTION

The Disk Editor is a program which permits the user to create a new, edited text file on disk from an old text file on disk. The text files can contain any kind of text, but the Disk Editor was primarily written to facilitate the writing of program source files to be assembled by the Disk Assembler.

II. OPERATIONAL REQUIREMENTS

The Disk Editor operates on a North Star MicroDisk System consisting of an 8080 CPU with at least 16K of memory, the North Star minifloppy disk drive and controller, North Star BASIC and the North Star Disk Operating System linked to your terminal I/O routines.

III. PROGRAM OPERATION

The Disk Editor edits an input disk file and when editing is finished, the edited file is saved in an output disk file. The user must create these two files using the CREATE command in the Disk Operating System before using the Editor. The input file must be large enough to contain all of the text which the user wishes to enter into the file. The output file must be as large as the input file since the input file is written to the output file at the end of editing. In addition, the Editor requires a work file to keep track of the line locations during editing. This file must be named "EDITWF" and its size in sectors should be equal to

$$\frac{(\ 5\ X\ \text{No. of lines in largest text file})}{256}$$

All three files must be type 3 (data) files. There is always an input file even if it is initially empty because you are entering text for the first time.

The Disk Editor may be executed by following these steps.

1. Boot up the North Star Disk Operating System.
2. Load BASIC by entering "GO BASIC".
3. Insert the diskette containing the Disk Editor.
4. Enter "LOAD EDITOR" and then "RUN".
5. The Disk Editor will ask for OLD and NEW filenames.
   (User responses are underlined.)

   OLD FILENAME? S01

   NEW FILENAME? S03

   >
6. The ">" indicates that the Editor is ready to receive editing commands.

III. PROGRAM OPERATION (continued)

The Editor operates with an imaginary line pointer which is always located at the beginning of a line. Generally speaking, the Editor commands permit moving this pointer, inserting and deleting lines at the pointer location, and displaying lines from the pointer location. The Editor commands and their effect is shown below.

DESIRED ACTION	EDITOR COMMAND	EXPLANATION
Display lines	Ln	Display n lines forward from the current location of the line pointer. If n is omitted, display one line.
Move line pointer	M±n	Move the line pointer forward or backward n lines. If n is omitted, move forward one line.
	S(string)	Search forward from the current location of the line pointer for the string. If not found, display "NOT FOUND" and set line pointer at top of text.
	T	Move line pointer to top of text.
	B	Move line pointer to the empty line immediately past the end of text.
Insert lines	I ?(new line) ?(new line) : : . . ? >	Insert new lines in the text before the line where the line pointer is currently located. The "?" indicates the Editor is ready for a new line. The Editor will continue to accept new lines until the user responds to the "?" with a carriage return, at which time the Editor is ready for another edit command.
Delete lines	Dn	Delete n lines in the text beginning with the line where the line pointer is currently pointing. If n is omitted, delete one line.
Miscellaneous	W	Displays current line number (Where).
	N	Displays a count of the number of lines currently in the text file.
	E	Ends editing and writes the edited file to the output file.

It is possible to recover should you leave the Editor without having entered "E". In the immediate mode of BASIC, CLOSE #1 and #2. Then enter "RUN" and when the Editor asks for the old file name, respond with "RECOVER". The Editor will then ask for the old file name again and then the new file name. The user should respond with the same file names entered previously.

COMPUTER SYSTEMS DESIGN's

DISK ASSEMBLER
for the

NORTH STAR MICRODISK SYSTEM

I. PROGRAM DESCRIPTION

The Disk Assembler is a program which permits the user to translate up to ten 8080 assembly language source programs in disk files into machine (object) code. An assembly listing showing source statements and object code may be displayed, and the machine code may be saved in a disk file. With the Disk Assembler, the user may write programs in modules and relocate them merely by reassembling.

II. OPERATIONAL REQUIREMENTS

The Disk Assembler operates on a North Star MicroDisk System consisting of an 8080 CPU with at least 8K of memory, the North Star minifloppy disk drive and controller, and the North Star Disk Operating System linked to the user's terminal I/O routines. The Disk Assembler is about 4.9K in size and loads at 2A00. With 8K of memory the assembler will assemble programs with up to 200 labels. With 16K of memory, the assembler can handle up to 1200 labels.

III. PROGRAM OPERATION

The Disk Assembler is a two pass assembler. It assembles source files in the order in which they are entered by the user. On the first pass, the assembler reads the source files and builds the symbol (label) table. On the second pass, errors are detected, the assembly listing is displayed, if requested, and the object code is written to a disk file, if requested. The source files must be sequential and contain only strings. The strings must be less than 80 characters in length and terminated in a carriage return. The source files must be type 3 (data) and there must be an end of file indicator at the end of the source text. The CSD Disk Editor produces files with these characteristics. The object file must be large enough to contain all of the object code generated by the assembly. The object file must be a type 1 (object) file.

The Disk Assembler may be executed by following these steps.

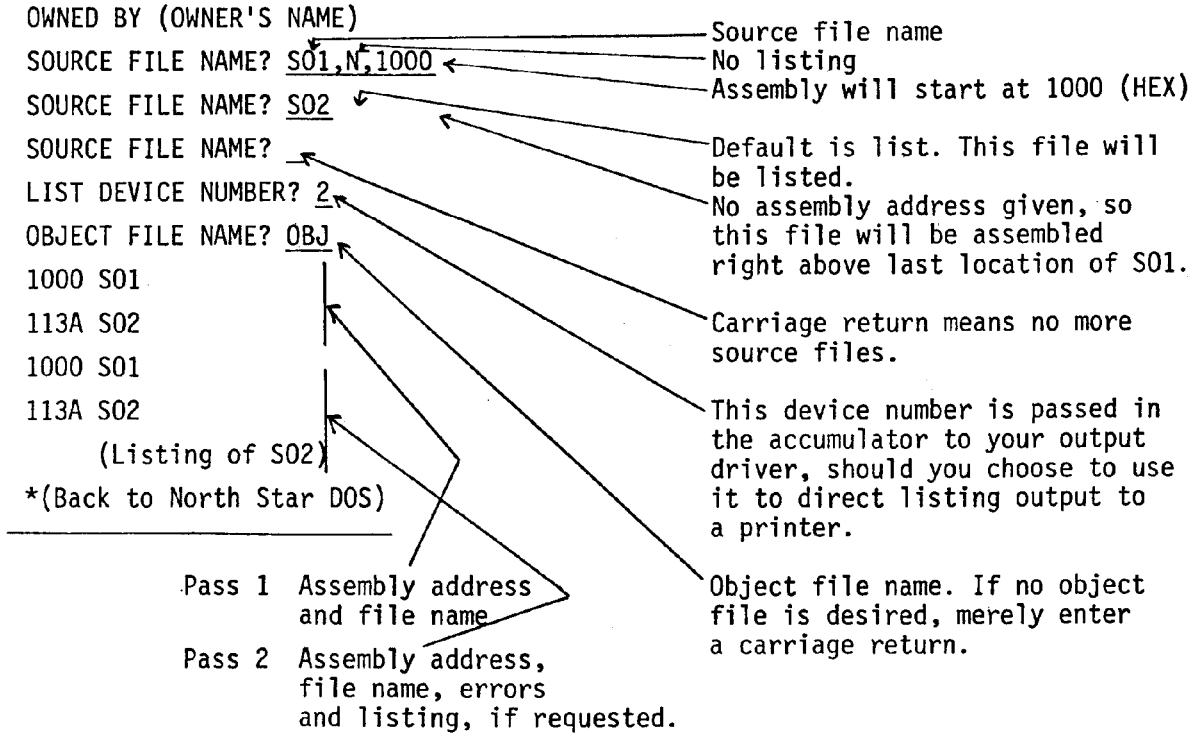
1. Boot up the North Star Disk Operating System.
2. Insert the diskette containing the Disk Assembler.
3. Execute the Disk Assembler by entering "GO ASM".
4. The Disk Assembler will ask for SOURCE and OBJECT file names, and list device number if a listing is requested.

The user response to the assembler questions is shown on the following page.

III. PROGRAM OPERATION (continued)

ASSEMBLER-USER INTERACTION

(User responses are underlined.)



When responding to the assembler questions the user may enter a delete (7F) to cancel the preceding character or an "@" to cancel the entire entry and repeat the question. Assembler operation may be cancelled with a return to the DOS by entering an escape character. Listing may be suspended by hitting the space bar, and resumed by hitting any other key.

The object file written to the disk is a memory image file. If the user enters an assembly address for a source file which would cause it to overlay a preceding source file, then the assembler will ignore the entry. When the user enters assembly addresses, he should attempt to keep the gaps between assembly addresses is not great or a rather large disk file will be required to save the object code. When the assembly address is advanced, the assembler fills with 00s.

An object file produced by the assembler may be loaded into memory and executed two different ways. It may be loaded into memory using the DOS load file command (LF) and then executed by using the DOS jump command (JP). Of course, the user must load the file into memory at the same address at which it was assembled.

If the user desires the object file to be loaded and executed with one command, he may use the DOS "GO" command. The "go address" associated with this file is both a load address and the address at which execution begins.

III. PROGRAM OPERATION (continued)

ASSEMBLER FEATURES

1. Up to ten source files may be assembled at one time
2. Source code may be free format.
3. Uses INTEL mnemonics.
4. Labels may be up to six characters in length.
5. Decimal, hexadecimal and ASCII string constants allowed.
6. Seven assembler directives:

DB	Define byte
DW	Define word
DS	Define storage
EQU	Equate
LST	Enable listing, if listing requested.
NLST	Disable listing
END	End of source code

IV. ERROR MESSAGES

ERRORS IN SOURCE CODE

Code	Meaning
A	Argument error
D	Duplicate label
L	Label error
M	Missing label
O	Opcode error
R	Register error
S	Syntax error
U	Undefined label
V	Value error