

Z-80/8080 TEXT EDITOR¹

INTRODUCTION

The text editor program is a strong line oriented program written in Z-80/8080 assembly language. The program is designed for use in the development of source programs to be processed by an assembler or compiler or for general purpose ASCII file generation. Twenty-seven separate commands are recognized by the program.

The editor does not require line numbers to be present in the ASCII file. It has the capability to search for and locate any string of valid ASCII characters in the file regardless of their location within a line. Lines can be added, deleted, replaced, modified, or printed with simple input commands. Once initialized, the program contains self-protection features so that it cannot overwrite itself.

HARDWARE REQUIREMENTS

The program occupies approximately 2.5K words of memory and 2K words for the Operating System and memory space for the file being edited. An additional 128 words of memory are used for the Z-80/8080 stack. Peripherals supported are a TV-Typewriter, Baudot teletype (output only), and a cassette tape. Several of the driver routines for the peripherals are contained in the system monitor ROM and must be supplied externally for conversion to other Z-80/8080 systems.

COMMAND FORMAT AND DESCRIPTION

All commands to the editor are input as ASCII data terminated by a carriage-return. The only non-printing ASCII characters recognized by the program are carriage-return (C/R; octal 015 or 215), end-of-file (EOF; octal 001 or 201), and Tab (Control T; octal 024 or 224). The program outputs a greater-than symbol, >, as a prompt indicating that it is waiting for a command to be input.

The commands recognized may be classified into three general categories: Initialization, Edit, and Utility. All commands must be followed by a space and/or terminated by a C/R. Additional parameters associated with a command (numerical or string data) must be separated from the command by one or more spaces.

Initialization Commands - The initialization commands set the file start address and define the end of file. All initialization commands request the file starting address which must be input from the keyboard.

<u>INITIALIZATION COMMAND</u>	<u>RESULTS</u>
NEWF	Defines a new file location starting at the input address and enters the input mode.
EDIT	Edit an existing file at the input address. Outputs the first line or page of the file as specified by the output mode.
LOAD	Loads a file from tape beginning at the starting address. Loading begins when a C/R is input following the address input to allow time for manual tape setup.

¹ Program written by Fred J. Greeb, Denver, CO; Documentation as published in June/July 1976 DR. DOBB'S JOURNAL OF COMPUTER CALISTHENICS & ORTHODONTIA, Box 310, Menlo Park, CA 94025 - Reproduced with the permission of People's Computer Company.

Edit Commands - All edit commands are used to display and/or edit lines within the file. All edit commands operate on or with respect to the current line. In most cases, the current line is defined as the last line displayed on the TVT screen. The program utilizes a line pointer which always contains the starting address of the current line. This address changes as different lines within the file are accessed.

In the following description, a string is defined as any sequence of any length of valid ASCII characters. Parameters contained within parentheses are optional parameters which may be included in the command line. ONLY the parameters and not the parentheses are included if the optional parameters are used.

<u>EDIT COMMAND</u>	<u>MEANING</u>
A string	Append the string to the end of the current line and display the result.
BOTM	Set the current line pointer to the end of file.
C %string1%string2	Find the first occurrence of string1 in the current line and change it to string2. The two string lengths need not be equal and the second string can be null (i.e., a C/R following the second delimiter). The delimiters (%) may be any printing ASCII character.
D (M)	Delete the current line (or M lines beginning with the current line) from the file. The file is moved in memory so that no empty space exists in the file. M is input as a decimal number - maximum value = 255.
F string	Find and display the first line in the file which begins with the string. The search begins with the line following the current line and continues until a match is found or the EOF is reached. The found line becomes the current line.
I string	Insert the string as a new line following the current line. The file is moved up in memory to make space for the new line. If no string is included or if only a C/R is input as a command, the editor enters the continuous input mode. In this mode, multiple lines may be entered in the file by typing in each line followed by a C/R. Exit from the continuous input mode is accomplished by inputting a null line (C/R only). When the continuous mode is entered, the message INPUT will be displayed. Upon exiting this mode, the message EDIT will be displayed. No prompt is issued between multiple input lines which indicates that the editor is in the input mode.
INSM M	Insert M lines from memory following the current line (M = 1 to 255). The file is moved in memory to accommodate the new lines. The location (starting address) of the new lines will be requested and must be input from the keyboard. This command is designed for merging together of the two files but may also be used to move lines within the same file if the destination is at a higher memory address than the source. If this is not the case, only one line at a time may be moved correctly within the file.

- LIST List the entire file on the output device (TVT or TTY).
- L string Locate and display the first line in the file which contains the string anywhere within the line. The search begins with the line following the current line and continues until a match is found or the EOF is reached. The located line becomes the current line.
- N (M) Move the current line pointer to the next line in the file (or move M lines) and display the new current line. M may be positive or negative (max. range = ± 255).
- P (M) Print the current line (or M lines). The last line printed becomes the new current line.
- PAGE List one page (15 lines) beginning with the current line. The current line is unchanged.
- R string Replace the current line with the input string and display the result.
- T Set the current line pointer to the top of the file and display the first line or page of the file.

Utility Commands - The utility commands allow displaying of the various pointers used by the program; specifying parameters to the program; and outputting files to tape. All addresses output by these commands are displayed in split octal, low order address first followed by the high order address. The utility commands interface with the TVT only and do not output to the TTY.

<u>UTILITY COMMANDS</u>	<u>FUNCTIONS</u>
DISP	Displays current line pointer. This command is useful for the INSM command to determine the starting address of the lines to be inserted.
DEOF	Display end of file address.
DISM	Display current setting of maximum memory size.
SETM	Set maximum memory address. This value is preset to 10K for use in a 10K system. This command requests an address input.
MODE L P	Sets the output to the line (L) or page (P) mode. In the line mode, only the current line is displayed following a command. In the page mode, 15 lines are displayed. The first line displayed is the current line.
RUBO X	Sets the rubout character to X. X (initialized to ") may be any printing ASCII character. The rubout character erases the previous input character in a command line. Multiple rubouts may be used to erase (back up) multiple characters.

KILL X Sets the kill character to X. X (initialized to ?) may be any printing ASCII character. The kill character deletes the entire input line. If the kill and rubout are set to the same character, the kill function will take precedence.

Q Quit. Exit to monitor.

TAPE Transmits the entire file to cassette tape. Two sub-commands are associated with this command and require responses to queries displayed on the TVT. The first TVT output is "REMOVE TABS?". An input of Y (yes) will cause tabs to be converted to spaces prior to transmission to the tape recorder. If N (no) is input, the file will be taped unmodified. The next output message is "FULL OR PARTIAL FILE?". If an F (full) is input, the file is terminated by a double end of file on the tape. If P (partial) is input, the file is terminated by a single end of file followed by an end of record (octal 003 or 203). These two tape end formats are not used directly by the editor program but are for use in an associated assembler, where they signal the assembler either that more data is required, or that the end of the source code has been reached. Transmission of data to tape begins when the C/R following the F or P response is input.

ERROR MESSAGES

The program will output the error message "WHAT?" in response to unrecognizable or improperly formatted commands. In addition to this general error message, several other error messages may be displayed.

On all commands which require an address input, the address is tested against the minimum useable file address. If the input address is less than the minimum, the error message "MIN ADDR (LH) = XXX YYY" will be displayed. This prevents overwriting of the editor program by the file being edited. The minimum octal address is L = 000; H = 024.

If a command is entered which increases the size of the file, the new end of file location is tested against the set maximum memory value. If the maximum would be exceeded by the command, the message "MEM OVERFLOW" is displayed and execution of the command is inhibited. During the LOAD command, the maximum is not tested until after the load from tape is complete and it can overwrite data stored above the maximum limit.

During execution of the INSM command, the data to be inserted is verified to be valid ASCII data. (Note: ASCII data, as defined in this program, is the 64 character upper case subset). If a non-ASCII character other than a control character recognized by the program is encountered, the message "BAD DATA XXX YYY" is displayed where XXX YYY is the address of the invalid data. Execution of the INSM command is terminated if this error is displayed.

If execution of a command such as Print M causes the end of file to become the current line, the message "BOTTOM" will be displayed. This message will also be displayed if a Find or Locate command fails to match the input string indicating that the string is not present in the portion of the file searched.

