DEFT Edit

Edit

1 Introduction

DEFT Edit is a program that allows you to create and modify PASCAL and Assembler source programs as well as any type of ASCII text file. Its features include:

- Text is maintained in memory to provide excellent command response.
- Files can be read and merged from either cassette or disk. They may be written to cassette, disk or printer.
- The user interface is a screen-mode "window" into the text with automatic up/down and left/right scrolling.
- All keys are auto-repeat.
- The *FIND* command allows you to search for specific patterns, *CHANGE* provides for changing the pattern in 1 or more instances.
- *MARK* and *APPEND* commands allow copying and moving of portions of text to either other places in the working text or to a file.

2 Basic Operation

After *LOADMing* and *EXECing* **DEFT Edit** you will see **DEFT Edit's** copyright screen which has the *INITIALIZE? (Y)* prompt. The editor uses the answer to this question to determine whether to initialize its in-memory text buffer. When you have just loaded the editor, you must answer this question yes. This can be done by entering anything other than *N* or *n* (including nothing) and then depressing the *ENTER* key. The only times that you would answer this question with a *N* or *n* is when you have previously used the editor, exited and did nothing to alter the computer's memory, and then re-entered **DEFT Edit**. See the *QUIT* command for more information.

Once the editor is loaded and initialized, you are now ready to enter text. The following sections will describe and explain what you see and what you can do.

2.1 Text Screen

Once you have answered the *INITALIZE?(Y)* question, you will see the text screen. This screen will be green with a blue square at the top left-hand corner and some numbers and letters on the bottom line in reverse video. The blue square is blinking and if you type some characters, they appear on the top line followed by a blinking orange square. The blue square has moved down to the second line. If you hold down a key, you see the corresponding character repeat. Each element on this screen is discussed in detail in the following subsections:

Blinking Square

There is always one square on the top 15 lines of the screen which blinks. This may be either a colored square, a character or a blank. The place on the screen which is blinking is the cursor. This is the point at which any text that you type in will appear. In addition, many commands that you can enter will affect text relative to the position of the cursor.

Blue Square

The blue square indicates the end of the text held in memory. Anytime the cursor is on a line which is within 14 lines of the end of the text, the blue square will appear at the left hand side of the screen on the line following the last line of text.

Orange Square

The orange square indicates the end of the line. It appears on the screen in the position that a carriage return is stored in memory. Every line, including the last line, always has a carriage return at the end.

Status Line

The line in reverse video at the bottom of the screen is the status line. This line provides information about the current status of your editting session. The information provided (in order) is:

1. The three characters at the left-hand side of the screen indicate the mode that the editor is in. INS (for insert) is the mode that the editor initially comes up in and causes each character typed to be inserted before the character pointed to by the cursor. The other modes are REP (for replace) and MARK which are discussed in later sections.
2. The number followed by the character *L* is the line number on which the cursor is currently positioned. The first line is numbered zero.
3. The number followed by the character *C* is the column number at which the cursor is currently positioned. The first column on a line is zero.
4. The number followed by the characters *LS* is the line size of the line on which the cursor is currently positioned. This count includes the carriage return at the end of the line.
5. The number followed by the character *T* is the number of remaining characters of text which can still be entered in memory. This number is updated each time the cursor is positioned to a new line.

Auto-Repeat

The auto-repeat feature allows you repeat the entry of any key on the keyboard by merely holding the key down for a full second. After this, the key will repeat at about 6 characters per second.

ENTER Key

The ENTER key is used to enter a carriage return into the text. This effectively splits the line at the cursor position and so creates two new lines.

SHIFT-0 Keys

The SHIFT-0 combination of keying, toggles the TRS-80 Color Computer from UPPER CASE into UPPER/lower case and from UPPER/lower case into UPPER CASE depending on what state the computer was in prior to the simultaneous entry of the SHIFT and 0 keys.

3 Cursor Positioning

As noted above, each character entered at the keyboard is displayed on the screen at the position of the cursor. The cursor then moves one column to the right. If the cursor is not currently positioned where you want it, you can use the four arrow keys to move the cursor. By depressing the appropriate up, down, left or right arrow key, the cursor will move in the same direction.

The cursor will always be positioned within the text of some line. This has the following side-effects:

1. When moving the cursor up or down, if the cursor moves from a long line to a short line such that it would be positioned beyond the end of the short line then the cursor will be positioned at the end of that line.
2. When moving the cursor to the right, if the cursor is at the end of the line then it will be positioned to the beginning of the next line.
3. When moving the cursor to the left, if the cursor is at the beginning of the line then it will be positioned to the end of the previous line.
4. When the cursor is positioned at the end of the text (blue square), the right and down arrows will not move it.
5. When the cursor is positioned at the beginning of the text (line 0, column 0) then the left and up arrows will not move it.

4 Scrolling

DEFT Edit lets you enter lines up to 255 characters long. This is considerably more than can be displayed on a 32 column by 15 line screen. The way that you view all of this text is by *scrolling* it past the screen. The screen becomes a window into the text.

This scrolling occurs automatically as you position the cursor by either entering text or by using the arrow keys. If the cursor is at the bottom of the screen and you force the cursor down to the next line, then all the lines on the screen move up 1 line with the top line disappearing and a new line appearing at the bottom of the screen. The reverse occurs when the cursor is positioned at the top of the screen and you force it to move up.

DEFT Edit also provides left and right scrolling in a similar manner. When the cursor is positioned at the rightmost column on the screen and you force it to move right, all the text on the screen shifts to the left by 12 columns. This prevents eye fatigue when entering data and having the text contantly scrolling to the left. The text will scroll to the right by 12 columns when the cursor is at the leftmost side of the screen and you force it to the left.

1 Functions

In addition to entering text, **DEFT Edit** provides many powerful functions that speed text editing. The general purpose functions are described in this section.

1.1 The CLEAR Key

The *CLEAR* key is used to invoke editor functions. When the *CLEAR* key is depressed, the cursor changes from a reverse video of the character that it is over to a white square. When the cursor changes to this white square, the next key entered is interpreted as a function rather than as a character to be entered into the text. Once the function is performed, the cursor returns to its normal reverse video state.

The *CLEAR* key itself becomes an *unCLEAR* function when it is depressed a second time, which returns the cursor to its normal mode without performing any function.

1.2 Major Cursor Positioning

By using the *CLEAR* key in conjunction with the arrow keys you can quickly position to a specific area of text. The CLEAR-arrow functions are as follows:

1. *CLEAR-Up Arrow* makes the cursor go UP by 15 lines to the beginning of that line. In addition, the line that the cursor is positioned to will be at the top of the screen.
2. *CLEAR-Down Arrow* makes the cursor go DOWN by 15 lines to the beginning of that line. In addition, the line that the cursor is positioned to will be at the top of the screen.
3. *CLEAR-Left Arrow* makes the cursor go to the beginning of the line that it is currently positioned on.
4. *CLEAR-Right Arrow* makes the cursor go the end of the line that it is currently positioned on.
5. *CLEAR-B* makes the cursor go to the beginning of the text.
6. *CLEAR-E* makes the cursor go to 15 lines before the end of the text. This line is positioned at the top of the screen with the cursor at the beginning of the line. This allows you to see the last 15 lines in the text. This command may take a couple of seconds on large files due to counting carriage returns in the text in order to maintain the line number.

1.3 Up Arrow Character Entry

The *Up Arrow* character is used in Pascal to denote pointer and file dereferencing. It is also used for cursor positioning by **DEFT Edit.** By first typing the *CLEAR* key and then depressing the *SHIFT* key while typing the *Up Arrow*, the *Up Arrow* character will be entered into the text.

1.4 Deleting Characters

There are two ways of deleting characters. The first is with the *CLEAR-D* function. When you use this function the character that the cursor is positioned over is deleted and all the characters to the right of the cursor are shifted to the left one character.

If you delete the carriage return at the end of the line, the line following will be appended to the end of the line. You cannot delete the last carriage return in the text.

A second way to delete characters is with the shifted left arrow key. In this case the cursor is moved one position to the left and the character there is deleted as previously described.

A third way is to delete all characters from the position of the cursor (inclusive) to the end of the line. First you position the cursor over the first character in the line from where you wish to *hack-off* the rest of the line, then you enter *CLEAR-H*. This function will *hack* that section of the line away and delete those characters.

1.5 Deleting Lines

A complete line can be deleted by positioning the cursor to any character on the line to be deleted (including the carriage return) and entering *CLEAR-L*. This function allows you to delete the last carriage return (as well as the last line) in the text.

1.6 Replace/Insert Modes

When **DEFT Edit** is first executed, it is in the *insert* mode of text entry. In this mode, when a character is entered at the keyboard it is inserted in front of the character that the cursor is positioned over. A second mode that the editor can be placed in is the *replace* mode. In this mode, when a character is entered at the keyboard it replaces the character that the cursor is positioned over. However, if the cursor is positioned over a carriage return then the character is inserted in front of it.

You can switch between these modes with the *CLEAR-I* and *CLEAR-R* functions. *CLEAR-I* puts you in the insert mode and *CLEAR-R* puts you in replace mode. The mode is always displayed on the status line.

2 Files

DEFT Edit allows you to load text from ASCII files on tape or disk, edit the text and then write back to cassette or disk. In addition, you can use the write function to write to the printer.

2.1 Getting A File

The *CLEAR-G* (Get) function allows you to insert the contents of a file into the current text in front of the character that the cursor is currently positioned over. This allows you to both initially load a file and to merge several files in memory.

When you enter *CLEAR-G* you are prompted for a file name on the status line. When typing on the status line the only thing that you can do is enter characters, the left-arrow to backspace and the ENTER key to terminate the entry. The default suffix used by the editor is blanks. If you enter no file name then the function is aborted and you return to the editing session.

2.2 Writing A File

The *CLEAR-W* function is used to write the in-memory text to a file. Like the *CLEAR-G* you are prompted for a file name. However, you are given the default of the file name used in the last *CLEAR-G* operation. If you enter any key other than ENTER then the default entry is erased and the character you entered is processed. Like the *CLEAR-G* function, if you enter a null file name the function is aborted.

2.3 Quitting and Reentering

The *CLEAR-Q* function is used to quit the editor and return to BASIC. After entering the function, you should immediately get the OK prompt.

When leaving **DEFT Edit**, the contents of the text area are not changed (unless you forgot to protect memory from BASIC with BASIC's CLEAR statement). You can reenter the editor and answer the *INITIALIZE? (Y)* question with either an *N* or *n* and return to the point in your edit session that you left. This is convenient when you wish to do a DIR to determine which files are on the disk before saving off the text in memory.

2.4 Exiting

The *CLEAR-X* function allows you to combine the *CLEAR-W* and *CLEAR-Q* functions with a single function. The write function is performed followed by the quit function. The text in memory is left un-changed.

2.5 File Errors

When reading from or writing to a file, a number of errors can occur. Whenever an I/O error occurs the message *FILE ERROR ...* is displayed on the status line and the editor waits for you to acknowledge seeing the message by depressing any key on the keyboard. This means that the first key depressed after the display of an error message will yield nothing more than the re-establishment of a normal status line presentation. Normal operation is then resumed. The possible error numbers are as follows:

- *-1, End of File* - You should not get this error number since an end of file is an expected occurence for **DEFT Edit.**
- *-2, I/O Error* - This indicates that some hardware oriented problem occured.
- *-3, File Not Found* - The file specified was not found.
- *-4, Illegal Operation* - This may occur if you try to read from the printer.
- *-5, Device Full* - There is no more space available on the specified device.

3 Pattern Processing

DEFT Edit contains commands for finding and changing text patterns.

3.1 Finding a Text Pattern

The *CLEAR-F* function is used to find a specific pattern in the text. After entering the *CLEAR-F* you are prompted on the status line for the string, of up to 24 characters, that you want to find. When typing on the status line the only things that you can type are characters, the left-arrow to backspace, and the *ENTER* key to terminate the entry.

When you depress the *ENTER* key the search will begin at the point in the text where the cursor was when you entered the *CLEAR-F* and will continue down to the end of the text. If a matching string is found then the line containing the string will be positioned at the top of the screen and the cursor will be positioned on the next character following the matching characters.

If you invoke the *CLEAR-F* function again, you will see that the prompt for the desired string defaults to the string that you entered on the last *CLEAR-F* or *CLEAR-C* function. You can just depress the *ENTER* key to find the next instance of the string in the text. If you type anything other than the *ENTER* key the old string will erase and you will be able to enter a new string.

If you enter no characters at all, no search will be made. If a search is made and the string is not found, the cursor will return to the point at which you entered the *CLEAR-F*.

3.2 Changing Text Patterns

In addition to finding a specific character pattern, you can change 1 or more occurances of one pattern to a second pattern. You use the *CLEAR-C* function to invoke this capability.

After entering the *CLEAR-C* you are prompted for the string to be searched for. After entering the string to be searched for, you are then prompted for the string that the first string is to be changed to. This can be 0 to 24 characters long. Finally, you are prompted for the number of occurances that are to be changed. If you don't enter any number, then the editor defaults to 1 occurance.

As each occurance is found and changed, it is displayed on the screen. When no more of the first string can be found, the function stops at the point where the last change was made. As in the

CLEAR-F function, if no first strings are found, the cursor will return to the point where it was when you entered the *CLEAR-C*. If you don't enter a first string, no changes are made.

4 Copying and Moving Text

There are 3 functions and a separate editor mode used to copy and/or move portions of text.

4.1 Marking and Saving Text

Before a portion of text can be copied and/or moved it must first be marked off and saved. This is done by positioning the cursor at either the first character or on the character following the last character of the text area to be saved. You then use the *CLEAR-M* function to mark that end of the area.

When you mark one end of a text area, two things happen. First, the mode changes to MARK to indicate that you are now marking an area of text rather than entering it. Second, the character that you marked is changed to a solid white square on the screen. This character will remain marked until you mark the other end of the text area.

Once you are in the mark mode, you cannot enter text. However, you can position the cursor with the arrows, CLEAR-Arrows, CLEAR-B, CLEAR-E and CLEAR-F functions. Once you have positioned the cursor to the other end of the text, you can mark it with the *CLEAR-M* function. The text that is saved starts with the mark that is closest to the beginning of the text and includes all characters down to but not including the mark closest to the end of text.

The mark function allows you to save up to 1.5K bytes of text in a separate in-memory *mark* buffer. If the marked area of text is greater than 1.5K bytes, then **DEFT Edit** prompts you for a name to give the file which it will create to save the marked text. This file name prompt occurs, provided the marked area is greater than 1.5K, immediately after the entry of the last *CLEAR-M* function. If a blank file name is entered then no action is taken and normal editing may be resumed.

4.2 Appending The Saved Text

Of course just saving the text away in a separate *mark* buffer or file doesn't do you much good unless you can do something with it. The *CLEAR-A* function allows you to append the text in the *mark* buffer into the screen text beginning in front of the current cursor position. The *CLEAR-A* function is not used to append text saved in a file, the *CLEAR-G* function is used instead when the text was saved in a file.

The contents of the *mark* buffer remain unchanged after this operation.

A typical copy operation would involve marking off the area of text to be copied, and then positioning the cursor to the point that it was to be copied to and invoking the *CLEAR-A* function.

If a section of text larger than 1.5K bytes needs to be copied into another area of a document, then the *CLEAR-M* function would be used to mark the text for copy. This would then yield a file name prompt for the file into which the saved text would be stored. Once the marked text is saved away, then the user would positon the cursor at the point in the document where the saved text was to be copied. The saved text would then be brought in with a *CLEAR-G* function followed by the name of the file containing the saved text.

4.3 Additional Mark Functions

When marking off a text area you can terminate the mark operation in 3 additional ways:

1. *CLEAR-D* may be used to mark the end of a text area. When used in this manner *CLEAR-D* is exactly like the *CLEAR-M* except that after saving away the text in either the mark buffer or a file, the area marked is deleted from the text. This provides the first half of a move operation rather than a copy. It can also be used to just delete areas of text.

2. *CLEAR-Q* terminates the mark operation without saving away any text. When the *CLEAR-Q* function is entered while the editor is in the mark mode, the mark operation is terminated with no action taken. The previous contents of the mark buffer are retained.

3. *CLEAR-W* allows you to save areas of text on a separate file or to print them on the printer. In this case the *CLEAR-W* function is entered to mark the ending point of a text area. After entering *CLEAR-W* you are prompted for a file name to which the marked off text is to be written. The contents of the mark buffer are not affected. This function allows the user to save any size of text to be filed, whereas the normal mark operation will only put text into a file if the text area being saved is larger than 1.5K bytes.