To: Distribution

From: Suzanne L. Krupp

Date: October 1, 1980

Subject: A Basic Video-Terminal Editor for Menu Applications

INTRODUCTION

This MTB describes an experimental video-terminal editor which tries to provide basic editing capabilities along with an easy-to-use interface for novice users. An "easy-to-use interface" is defined here as being an interface where editor requests are built in much the same way as English sentences. In other words, editor requests consist of things like verbs, nouns, and modifiers where certain combinations of these elements have meaning. These combinations of verbs, nouns, and modifiers are specified through the use of the function keys on a VIP7801 terminal. This type of interface plus a very basic editing capability make up the editor.

The overall design of this editor is patterned after another system called Etude. Etude is a much more extensive and powerful text editing facility which was presented at the MIT Office Automation Symposium several months ago.

There are many applications of the menu manager that require a basic editing tool which is easy to use. The editor described in this document is proposed to be a tool which can fulfill these needs.

HOW TO INVOKE THE EDITOR

Even though the editor is an Emacs extension, it is invoked via a command called key_editor. The command line used to invoke the editor is as follows:

```
key_editor path
```

where path is the pathname of the file to be edited. The path argument is required.

The key_editor command was designed for use as a test interface to the editor for two reasons: 1) to make sure that no special knowledge of Emacs is needed in order to use the editor; 2) to provide a quick and easy way to use the editor outside of menu applications.

The command prepares the correct Emacs environment and reads in the file specified in the command line. The "correct" Emacs environment is where all regular Emacs functions are disabled. Only the special functions known to the editor described in this document are enabled. Even the screen format is different (as described below). Also, any personal Emacs start_ups are ignored.

THE SCREEN

After invoking the editor, the user must wait several seconds for it to get started. When it has started up, it takes control of the terminal display and sets up the screen in the following manner. First, it clears the screen. It then displays the message "Reading..." for a few seconds. This is to inform the user that the editor is busy reading in the file specified in the command line. Once the editor is finished reading, it displays the file placing the cursor at the upper left-hand corner of the screen (which is also the beginning of the file). Most of the screen, the upper part, is used to display the file. The bottom four lines are the exception. The two lines at the extreme bottom edge of the screen are initially blank. They are where errors, prompts and informative messages are displayed. The other two lines display the current request (as it has been specified so far), and the previous request. This particular screen format is used throughout the editor invocation.

WHAT'S IN AN EDITOR REQUEST

Editor requests resemble simple English sentences in that they are made up of combinations of nouns, modifiers and verbs. (See Table 1 at the end of this discussion for lists of nouns, modifiers and verbs. See also "Descriptions of Request Elements" below for descriptions of these elements.) Nouns specify a type of object such as "sentence" or "paragraph". Modifiers modify the meaning of the noun so that a specific object or point in the text of the file can be located. For example, "previous sentence", a combination of a modifier and a noun, designates a specific object in text. To locate a specific point in text, a combination of two modifiers and a noun such as "beginning of previous sentence" could be used. Verbs describe the action which will be performed upon an object, point, or region in text. Verbs which act on a point in text are "go to" and "mark". Other verbs which act on objects or regions in text are "delete" and "copy". These elements (verbs, modifiers and nouns) are combined to form requests.

There are rules which must be followed when making a request. They are:

1. A request can consist of, at most one verb, two modifiers, and one noun. An example of this is the request "go to beginning of previous sentence". Note that one item is taken from each column of Table 1 to make this request. (See the end of this discussion for Table 1.)

2. A request must contain at least a noun. The rest of the request is filled in by the editor. For example, the request "delete sentence" is expanded by the editor o mean "delete current sentence". The request "beginning of word" is expanded to "go to beginning of current word". The specification of a noun alone, such as "paragraph", will be expanded into the request "go to beginning of current paragraph". For a list of defaults, refer to Table 1.

3. There are some exceptions to rules 1 and 2. The requests which are exceptions are listed in Table 2, "Stand Alone Requests". These requests really do stand alone. They are not combined with anything else to form a complete request because they are complete by themselves.

4. Some modifiers exclude the use of other modifiers within the same request. When two modifiers are used in a request, they must come from separate groups. (Note that modifiers in Table 1 are divided into two groups.) For example, the specification of "next" and "previous" in the same request will not be accepted.

Even after faithfully following all of the above rules and regulations, there are still some combinations which are unacceptable, for example the request "delete beginning of sentence". This request is unacceptable because "delete" works on an object or region of text whereas "beginning of sentence" only specifies a point in text. Such unacceptable combinations are few. When an unacceptable request is specified, the editor will display an error message at the bottom of the screen.

HOW TO INVOKE A REQUEST

A request is invoked through the use of function keys, specifically those keys on a VIP7801 terminal labeled f1 through f12. The function keys have been assigned names which are communicated to the editor when they are pressed. To invoke a request, simply press the upper or lower case function keys which have the appropriate names as illustrated in the following examples. (See "Description of Request Elements" below.) The function key name assignments for a VIP7801 are diagrammed on the last page of this MTB.

Examples

Most requests act on text with respect to the current cursor position meaning, the text which immediately surrounds the cursor or whose position may be described using an editor request. In order to get to text which does not fit into either of the above categories, the cursor must be moved. Moving the cursor from one place to another can be achieved in a variety of ways. One such way is to use the arrow keys. The arrow keys move the cursor left, right, up or down. For instance, to move the cursor five characters to the right, press the right arrow five times. To move the cursor up five lines, press the up arrow five times. In order to move the cursor over an undetermined number of lines or characters, simply hold down the appropriate arrow key. Another way to achieve cursor movement is by building an editor request using the "go to" key. For instance, to move the cursor to the end of the file, press the keys "go to", "end of", "last" and "word". It is not always necessary to specify "go to" since that is the default verb. The same thing could have been achieved by pressing the keys "end of", "last" and "word". If the goal is to move the cursor to near the end of the file, the key combination "last" and "word" could have been specified. This would move the cursor to the beginning of the last word.

To invoke a request which will delete the previous sentence, press the "delete" key, the "previous" key and the "sentence" key. The sentence previous to the one which contains the cursor will disappear. If this was the wrong thing to do, simply press the "undo" key and the deleted text will reappear. If the intent was to move that sentence to another place, delete it, position the cursor to where the deleted text is to be moved and issue the "retrieve" request. This is done by pressing the "retrieve" key. The difference between using the retrieve key and the undo key is that the undo key undoes the previous request, no matter what it was. The retrieve key places the last deleted (or copied) text at the current cursor position.

If one wants to work on a region of text, the region must first be marked. If the region is to be an object such as the current sentence, mark the beginning and end of the region (sentence) by making the request "mark sentence". This request puts the beginning-of-region mark at the beginning of the current sentence and the end-of-region mark at the end of current the sentence. If a region is to contain an undetermined amount of words, sentences or paragraphs the region is marked in the following manner. Move the cursor to the place where the region starts. Press the keys "mark", "beginning of", and "region". The editor will respond with "marked". Then move the cursor to the end of the region and press the keys "mark", "end of" and "region". The region has been marked. Now suppose that, in either of the above cases, it is decided that the beginning-of-region mark is not in the correct place. It can be moved simply by moving the cursor to the correct place and making the request "mark beginning of region". However, the beginning-of-region mark cannot be moved past the end-of-region mark. The same can be done with the end-of-region mark. Once the region has been marked a request (such as "delete region") can be performed upon it.

In order to copy a region of text, for example the next sentence, the region must first be marked. To mark the region press the "mark", "next" and "sentence" keys. The editor will respond with "marked". Then make the "copy region" request. The editor will then respond with "copied". This response means that the text has been saved and may be retrieved using the retrieve key. Move the cursor to the place to where the text is to be copied and press the "retrieve" key. The copied text will appear where the cursor is. If this was the incorrect place to insert the copied text, press the "undo" key and the text will disappear. Then move the cursor to the correct place and insert the text again using the "retrieve" key. If the same sentence is to appear more than once throughout a file, simply move the cursor to the appropriate points, retrieving the sentence at each point.

An easier way to copy the next sentence is to simply make the request "copy next sentence". The editor will respond with "copied". Then move to the desired place and retrieve the text. When editing involves an object such as a word, sentence, paragraph or line, it is generally easier to not use the region facility. Regions come in handy when either a part of an object or a number of objects are involved in an editing operation.

As a final example, the use of the help and menu keys will be illustrated. The help key is useful if an explanation of a single key is required. Such a case might be when a description of the "delete" key is desired. In this case, press the "help" key and the "delete" key. A description of the delete key will appear at the top of the screen. To remove this display, press the clear/reset key. The help key may be used at any time. The menu key is useful when in the middle of specifying an editor request. When specifying a request and the menu key is pressed, a menu is displayed at the top of the screen. The menu contains options which may be selected via the function keys in order to complete the current request. For example, suppose the partial request "delete current" has been specified. Then the "menu" key is pressed. A menu is then displayed which contains the names of keys which can be used to complete this request. In this case the menu would contain: word, line, sentence, paragraph, and region. Any one of these keys may be used to complete the example request. Press the desired key to complete the request. The menu will disappear and the request will be executed.

DESCRIPTION OF REQUEST ELEMENTS

Below are detailed descriptions of the request elements listed in Table 1. These are intended for reference but should be read thoroughly at least once to obtain a full understanding of what each element (key) does.

The following terms are used throughout descriptions of the specific verbs, modifiers and nouns.

point
: a particular spot in text such as the beginning of a word or the end of a paragraph.

object
: a word, line, sentence or paragraph.

region
: an area of text delimited by two marks set by the user. A region can consist of one or more objects, or part of an object.

Verb Keys

go to
: goes to a specific point in text (example, "go to" moves the cursor to the "beginning of previous sentence" or "end of next sentence").

delete
: deletes an object or region of text (example, a word or a marked region). The deleted text is put into an

invisible save area. (The same save area which is used to keep copied text.) The deleted text is saved until another delete or copy is done, then it is overwritten with newly deleted or copied text. The text in the save area can be inserted into the file at any place by moving the cursor to the desired spot and using the retrieve key.

copy

copies an object or a region of text into a save area. (The same save area which is used to keep deleted text.) The text is saved until another copy or delete is done, then it is overwritten with the new text which has just been copied or deleted. The text in the save area can be retrieved and inserted into the file at another place using the retrieve key.

mark

sets off a point, object, or region of text. The mark key does this by putting a mark at a point in text or two marks around a specific object. When something is marked, it becomes special to the editor. For example, one can build the request "mark current sentence". This request puts one mark at the beginning of the current sentence and one mark at the end of the current sentence. Alternatively, the request "mark beginning of next sentence" puts a mark just _at_ _the_ _beginning_ of the next sentence.

An area of text delimited by marks becomes a region and so any further action on that area of text must be done as an action on a region (i.e., via the region key). In the previous example, marks were placed around the current sentence. That sentence has become the current region and should be deleted via a request such as "delete region".

There can be only one region at a time. The last thing marked by the user is the region. For example, if the request "mark next sentence" is issued, the region is the next sentence. If the next request is "mark previous sentence" there is still only _one_ region marked. It is the previous sentence. In other words the region consists of the thing which has been most recently marked.

The above keys (verbs) are used in combination with other keys to form editor requests. The following keys are complete requests in themselves. They are never combined with anything else.

undo
: undoes the previous request. For example, if the request just executed is "delete current paragraph", the deleted paragraph is retrieved. If in the middle of specifying a request, "undo" undoes the specification. In other words, it wipes the slate clean as if the partial request had never been specified.

retrieve
: retrieves text from the save area. A piece of text may be retrieved any number of times. See the description of the copy or delete key.

help
: provides a description of any specified key. When in need of a description for a particular key, press the "help" key. The editor will display the prompt "Key: . The user must the respond by pressing the key for which a description is desired. The name of this key will be printed at the bottom of the screen to complete the prompt. A description of that key will be displayed at the top of the screen. This display can be removed by using the clear/reset key.

menu
: aids in the completion of requests by displaying a menu of acceptable keys which may be specified to complete the request. For example, when in the middle of specifying a request and the "menu" key is pressed, a menu of keys which may be specified to complete the current request is displayed. Simply press the desired keys, the menu disappears and the request is executed.

clear/reset
: redisplays the screen when it has been messed up.

transmit
: writes the file out to the pathname specified in the command line.

arrow keys
: move the cursor one character in the direction of the arrow printed on the key.

#
: deletes the character to the left of the cursor.

@
: deletes all characters between the cursor and the beginning of the current line.

quit

used to leave the editor. If the file has not been written at the time of a quit, the editor asks whether or not the user wants to quit. This is to protect the user from inadvertently leaving the editor without writing the file.

Modifier Keys

The following modifier keys, when used in a request, help specify a point in text. Even though up to two modifiers may be used in a request, only one may be chosen from this group.

end of

used in combination with other modifier and noun keys to specify a point in text. For example, "end of current word" or "end of next sentence".

beginning of

used in combination with other modifiers and nouns to specify a point in text. For example, "beginning of current word" or "beginning of previous sentence".

The following modifier keys, when used in a request, specify the particular object with which the editor is to be concerned. Even though up to two modifiers may be used within a request, only one may be chosen from the following group.

current

used in combination with nouns to specify the object which, at present, contains the cursor.

next

used in combination with nouns to specify the object which, at present, is immediately past the object which contains the cursor. For example, the specification of "next word" means the word immediately past the <u>word</u> which currently contains the cursor. Also, "next sentence" means the sentence immediately past the <u>sentence</u> which presently contains the cursor.

previous

used in combination with nouns to specify the object which, at present, is immediately before the object which contains the cursor. For example, the specification of "previous word" means the word immediately before the <u>word</u> which currently contains the cursor. Also, "previous sentence" means the sentence immediately before the <u>current</u> <u>sentence</u>.

last
: used to specify the last of a particular type of object in the file being edited. The specification "last word" means the absolute last word in text.

first
: used to specify the first of a specific type of object in the file being edited. The specification of "first sentence" means the absolute first sentence in text.

NOUN KEYS

word
: an unbroken string of upper and lowercase alphabetics, numbers, underscores, and backspaces.

sentence
: begins: at the first printed character following the end of a sentence; the first letter following an empty line; or the first letter at the beginning of the file. A sentence ends: after a period, question mark or exclamation point followed by whitespace or the end of a line; a period, question mark or exclamation point followed by a double quote or right parentheses followed by whitespace or a newline; the last character preceding an empty line; the last letter of a file.

paragraph
: begins: at the beginning of a non-blank line preceded by an empty line; the beginning of the file. A paragraph ends: at the end of a non-blank line followed by an empty line; the end of the file.

region
: same as defined above.

CONCLUSION

This editor was built as an experiment, to see whether or not this kind of editor would be easy to comprehend and use.

Table 1. Request Elements

	Modifiers		
verbs	Group 1	Group 2	Nouns
*go to	*beginning of	next	word
delete	end of	previous	line
copy		first	sentence
mark		last	paragraph
		*current	region

* DEFAULTS

NOTE: There is no default noun since one must always be specified. Also, the modifier "beginning of" is used as a default only when the verb specified is not going to act on an object or region of text. For example, if the "delete sentence" request were specified the expanded request would be "delete current sentence" since this request works on an object (sentence). If the request "go to next sentence" were specified, the expanded form would be "go to beginning of next sentence" because this request acts on a point in text.

Table 2. Stand Alone Requests

undo
retrieve
help
menu
arrow keys
clear/reset
#
@
quit
transmit

de- lete	re- trieve	undo	help	end of	pre- vious	last		line	para- graph		quit
go to	copy	mark	menu	beg of	next	first	cur- rent	word	sen- tence	region	
1	2	3	4	5	6	7	8	9	10	11	12

Figure 1. Diagram of VIP7801 Function Keys