

FRESS Concepts and Facilities

for the Layman

by

Andries van Dam

(Revised January 25, 1919)

1. INTRODUCTION

Writing is a slow process for all but the most talented, skillful, and prolific writers. Basically, writing consists of translating thought into words and phrases and then arranging them so that the original thought will be recreated in the mind of the reader. This is not an easy task and, therefore, the average person does very little true writing. Instead, he becomes accustomed to collecting and reordering familiar word patterns which he uses to express equally familiar thoughts, experiences, or facts relating to his daily life. In order to distinguish shopping lists and letters to one's relatives from more serious and professional writing, we will refer to the latter as document or manuscript preparation - a name which implies more than just jotting down ideas.

Document preparation is essentially a three-step process. The first is to gather pertinent ideas either by putting one's own thoughts into words or by collecting fragments from one's previous writing (or the writing of others). This step often begins with merely an idea of what one wishes to write about and with incomplete facts or insufficient understanding which require one to have access to additional information during the process of writing. The second step involves compiling a more or less cohesive stream of words and ideas which is easy to read and which clearly explains the author's original intention. Unless the first step was exceedingly well done, the second step will involve rearranging sections into logical units in the correct order (traditionally known as cutting and pasting), adding additional facts and explanations where needed, deleting unnecessary words and passages, and correcting typographical errors ("typos"). This version is the raw input to step three in which the manuscript must be put into the proper format (page layout, headings, footnotes, table of contents, index, etc.) and then either typed or printed. At this point, documents of any importance will go through the second step at least one more time to correct remaining mistakes and achieve better organization.

Anyone familiar with document preparation knows how time consuming and expensive (in man-hours) this iterative process is. Technology has, in recent years, made considerable advances in assisting with step three.¹ Automatic typesetters, computer assisted page layout, high speed printers, optical text scanners, and a multitude of copying machines have decreased the number of man-hours needed and have improved general document appearance. Yet, despite these conveniences, little was done until recently to mechanically assist the "evolution" of documents - that is, the creation of step one and the cutting, pasting, and correcting

¹A survey of these advances can be found in van Dam and Rice, "Computers and Publishing: Writing, Editing, and Printing", in Advances in Computers, Vol. 10, Academic Press, New York (1970).

of step two. The practice of wading through piles of drafts, old versions, new versions, references, loose pages of additions and corrections, of waiting for the latest update to be typed, and of dividing and recompiling work among several people, each of whom may do things a bit differently, is just too expensive and too slow to handle adequately the tremendous amount of document preparation which has become necessary in recent years. It is safe to say that fewer revision cycles take place than are necessary and, therefore, work of lower quality is being turned out for want of better methods.

While the typewriter marketplace in the last few years has been inundated by automatic typewriters and Word Processing systems, these are still primarily oriented towards the production of small scale documents (letters, memos) which undergo relatively few, small changes, typically typo corrections. FRESS (File Retrieval and Editing SyStem), in contrast, is a sophisticated large computer assisted text editing and information handling system designed to give the maximum mechanical assistance to all phases of document preparation. FRESS has been designed not only to perform all operations used in cutting up, rearranging, and putting together complex documents (anything traditionally done with scissors and paste -- ranging from Word Processing-type corrections to major rearrangements ("major surgery")), but also to actually assist in the information gathering and thought expression step of document preparation. FRESS allows the user to:

- 1) collect segments of written material (hereafter referred to as text), thoughts, ideas, and facts that he may have written himself or "cannibalized" from other documents and previous work;
- 2) separate or combine these segments in any way he desires;
- 3) "catalog" them in various ways to allow quick (online) access to specific items in his document; and
- 4) edit and format them for hardcopy production.

Text is more important for the ideas (information) it contains than for its physical representation on paper. Thus, while FRESS is well equipped to correct typos, to assist in formatting the layout, and to produce "finished" manuscripts from these directions, it should be understood that the primary value of the system is its ability to assist in the gathering, storing, accessing, and manipulation of textual information. It is of little value to have good-looking machine prepared documents if the content is poorly arranged or lacking because much of the energies of the writer(s) was spent grappling with piles of paper and index cards rather than with the thoughts and ideas themselves. Thus, FRESS should be considered not as a means of producing the same documents in less time (which it is),

but as a means of producing better documents in the same (or even shorter) amount of time. A final point to be remembered is that FRESS is not an "automatic document producer". It will not accept raw information and then magically convert it into finished documents while the user merely "turns a crank". However, by supplying the user with easily specified and powerful tools, FRESS enhances and extends his capabilities significantly.

2. THE SYSTEM ENVIRONMENT

FRESS is a program available to terminals (minimally typewriter consoles) located on the user's premises running in an interactive timesharing mode under an IBM system 360 or 370 computer to which they are connected via a telephone line, or directly "hard-wired". Interactive means that the user and the computer have a continuous "dialogue" of requests and answers, with successive user requests usually influenced by the computer's current answers. In the case of FRESS, the requests are retrieval and editing commands, and the answers are updated glimpses of the text or "prompts" by the system as to what is expected next from the user or what he did wrong. Timesharing is a method of more fully utilizing the power of a large computer by allowing many people to use it at the same time. Since FRESS is interactive, the computer would normally be waiting while the user decides upon his next request and types it in. The computer processes his request at tremendous speed, sends a return message which the user's terminal types out, and again waits for the user. Altogether, the computer works only a small fraction of the time while the user works at maximum efficiency.

The mechanism of timesharing allows the computer to handle simultaneous requests from a number of terminal users; while one user is selecting or typing in his next request, the computer is answering another user. The computer thus divides its time among users without a noticeable delay in response time. In fact, it takes 30 or 40 users to "saturate" (use most of the available time of) the machine. In the event that a user enters his request at the same time as another, he merely has to wait a second or so until the other's work is done by the computer - a time not noticeable compared to the time it takes to type out a line. Thus, many users can share the resources of a powerful computer simultaneously, while getting maximum computing at minimum cost. Furthermore, a user with modest demands need not purchase a computer to run FRESS; he need only dial into a central timesharing computer.

3. OTHER COMPUTER-ASSISTED TEXT EDITORS

With many computer-assisted Word Processing and Text Editing systems ("editors") on the market today,¹ what, if any, are the features which set FRESS apart? Most currently available time-sharing editors are "line editors". This means that they operate on text stored as collections of lines, with maximum lengths, just like lines on a printed page, and that one may edit only one line at a time (which is not necessarily a complete sentence). For example, if more text is to be inserted into a line than it will hold, the user must delete some or all of the original line and type in the two or more lines necessary to replace it; otherwise, the system will truncate the original line at its maximum length and discard any extra text. Operations such as deleting a sentence which runs from the middle of one line to the middle of another two lines down require a separate step for each of the three lines involved. More complex operations (such as reversing the order of two such sentences) require so many steps that it is often easier to delete all lines involved and retype them. A line editor, then, is far from ideal in serious, free-form document preparation and is useful only for editing line-oriented text such as computer programs and lists of names and addresses or for correcting misspellings.

FRESS is a "text editor" rather than a line editor such as Brown's CMS editor. It operates on text stored as a continuous "stream" of characters, not as lines of maximum length. Since there are no "line boundaries", the text may be spread apart (for inserts) or squeezed together (for deletions) by any amount desired. For such operations as reversing the order of two sentences, one merely indicates the beginning and end points of the sentence to be moved and the point to which it is to be moved, and the text is rearranged and compacted automatically.

FRESS has additional advantages over other (computer-assisted) text editors. Magnetic tape or card storage editors and editors run by small "local" computers are designed primarily for "office use" (letters and short reports). They have a limited number of editing capabilities (10 to 20 commands, as compared to more than 100 in FRESS) and often have limited storage capacity (tens of pages as opposed to hundreds or more for FRESS).

Word Processing systems, even those which, like FRESS, are not line oriented, still have limitations on the amount of text that can be manipulated in one command, typically do not support arbitrary rearrangements, especially between documents, and have many fewer editing and formatting facilities. The innovative

¹Timesharing editors available recently are described in detail in van Dam and Rice, "On-line editing: A Survey", Computing Surveys, 3, No. 3 (1971).

features of FRESS allow it to adapt to a greater range of document types, formats, editing styles, and demands of users. A description of FRESS's capabilities follows. Note that though numerous examples are included, this is not a user's guide (which exists as a separate document), but rather a guide to the many different ways FRESS can assist you in document preparation, storage, and retrieval.

4. THE FRESS SHELLS

FRESS is a very powerful system capable of performing numerous different operations on text structured in varying manners. A description of the complete system would be overwhelming to the uninitiated, so FRESS has been designed to be approached (learned and used) in a number of levels or shells of increasing complexity. The demarcation between each shell is often somewhat fuzzy; the description below is structured to aid in understanding the FRESS system organization. The beginning user or secretary transcribing editing changes would probably work primarily with only the first shell, since the other shells are oriented towards online browsing, creating, and retrieving for the sophisticated author.

In what follows, examples have been provided, whenever possible, to clarify the concepts presented. The examples appear in Section 7.

4.1 SHELL ONE: BASIC COMMANDS

Minimum knowledge for productive use of FRESS may be acquired in approximately one hour. The commands follow in the approximate logical order of their use.

LOGGING INTO THE SYSTEM:

The login procedure is dependent on the computer's operating system and also the user's terminal (Fig. 1).

READING, ACCESSING, AND STORING TEXT FILES:

Files are subdivisions of the computer's magnetic tape or disk memory. FRESS text files are files created for the specific purpose of storing text to be retrieved and edited by FRESS. Each text file has associated with it a name and one or more passwords. The first command after selecting FRESS must indicate

a text file for FRESS to edit, either by giving the name and password of an existing file (Fig. 2a) or by providing a (previously unused) name and a password to identify a new text file which FRESS will automatically create (Fig. 2b).

LINEAR TRAVELING:

All but the most trivial text files are too large to view in their entirety at once. Imagine that the text is printed on a long strip of paper (as if the pages of a document were connected end-to-end). Since such a piece of paper would be difficult to handle and since we could only look at small sections at a time, the most logical solution would be to roll up each end leaving a small unrolled portion in the middle to look at - in other words, to form a scroll. To read through such a document one would "travel" through the scroll by unrolling paper from the bottom and rolling it back on the top a few lines at a time. One might consider a document in this format more difficult to use manually than one with pages; however, if the rollers are moved mechanically, (e.g., as on microfilm viewers), this format is as easy, if not easier.

The way in which FRESS displays a file is conceptually (though not mechanically) equivalent to this scrolling method. The system types out a number of lines (called a "window") of the file. The window assumed by the system for a typewriter terminal is one line of 50 characters, but the user may reset this to any line width between 20 and 130 characters for any number of lines up to a total of 500 characters (Fig. 3a).

Window size is an important factor in almost all editing. The system will retype (or redisplay, if a TV-like display screen is used instead of a typewriter terminal) the full window after every traveling and editing command unless told otherwise, so that the user can see the results of the command. Since a typewriter terminal types much more slowly than the user can read (although much faster than he can type), the smaller the window, the faster the editing cycle (typeout-thought-response-typeout). However, the smaller the window, the less text visible for viewing, and thus the more traveling necessary. Experience will allow the user to determine (and specify to the system - see Fig. 3a) the best window size for the type of editing he is performing.

Linear traveling is the process of moving the window forward or backward in the file, either by a specified number of lines (called scrolling, since it is like using a scroll) (Fig. 3b) or to an occurrence of a specified piece of text (called pattern scanning) (Fig. 3c). The piece of text to be located can be specified explicitly (e.g., the character sequence "text editing") or may be indicated by a more complex "pattern" (e.g., "the quick...fox" with the ellipsis to match any intervening text). To scan for the same pattern several times in succession,

one simply types "l" which will scan for the last pattern specified; to scan backwards, one concatenates the "b" modifier to the abbreviation for the locate command, i.e., "lb". The user quickly becomes adept at linear traveling; the facilities a system provides for moving around within a file are key factors in determining the ability to do serious editing (other than correcting typos).

SIMPLE EDITING COMMANDS:

By "editing" we mean making changes in the actual text. In FRESS, one may edit any portion of the text in the current (most recently displayed) window, not just the top line. In fact, one may edit within a 2100 character "buffer" starting with the top line of the current window; thus one may edit text not even in view in the window, but "below" it. (With "deferred location pointers" mentioned below and discussed in the FRESS User's Guide, one may even go beyond that 2100-character limit.) In addition to this restriction-free editing flexibility, FRESS has one other enormous advantage (particularly to beginners) over other editors: editing mistakes may be undone (reverted) after the user sees the result simply by typing "REV"!

There are five main editing commands which reflect the usual editing operations done with pencil and scissors. In contrast, line editors usually have only "substitute within a line" and "move one (or more) whole line(s)".

Insert puts the text typed in (of any desired length) immediately after the point indicated by the user (Fig. 4).

Delete removes the text included between two indicated points (Fig. 5).

Substitute combines delete with insert and allows the user to replace one string of text of arbitrary length with another in a single function (Fig. 6).

Move rearranges text between two indicated points to a third point (Fig. 7). Indication of the new location may be delayed ("deferred") while traveling is done to allow text to be moved to remote parts of the file or another file; this operation is analogous to scissoring and pasting. Note that one may thus move any length character string from one location to anywhere else, in one operation. This is impossible with line editors.

Copy is similar to move, but the occurrence of the text in its original location is not deleted (Fig. 8).³

³Note that move and copy can be used to transfer arbitrary length material both within and between files.

SIMPLE FORMATING:

Formating commands allow the user to change the layout of the printed page, either as it is typed out on his console typewriter (for small files) or, for high-volume, better quality printouts, remotely, on the high-speed printer. He may specify lines to be skipped (user specifies number of lines), paragraphs, indentations and hanging indentations (user specifies number of spaces to indent), headings (user specifies level -- produces an outline format and a table of contents), left and right margin justification (margin alignment), and footnotes. Fig. 9 demonstrates a few of these formating commands.

SPECIFYING TEXT POINTS:

Editing operations are performed at specific points in the text. The user must indicate these points by supplying sufficient context so as to be unambiguous within the window he is currently viewing and within the rest of the 2100-character buffer below the window. Note that on a large display console the entire buffer could be displayed in the window, whereas on a typewriter or simple alphanumeric display console a printout from an earlier session would be used to show what is below the window. If the point is ambiguous, the first occurrence of the specified context will be taken as the desired one. Other methods of simplifying context specification beyond the scope of this discussion are detailed in the FRESS User's Guide. To aid in checking what the system has done in response to a given editing command, the results are immediately shown (and can be reverted; if needed).

"Transcription Mode" moves the display point in the file automatically down to the area where the editing takes place to allow the user to see the context of his edit directly. Transcription mode is in fact the default viewing mode on typewriter consoles where multi-line "display" is not practical. However, the examples in Section 6 are shown without Transcription Mode for simplicity; the display point stays where it was.

As you can see, the first shell of FRESS is already beyond the scope of commercial computer-assisted editors. We will now proceed to discuss more sophisticated functions in FRESS which contribute to its being such a powerful text manipulation tool.

4.2 SHELL TWO: DIRECT ACCESS AND RETRIEVAL

In large files it is cumbersome to travel linearly between widely separated points, especially when the location of the specific point being sought is not remembered. Often, in fact,

the user will want just to browse through a file, jumping from place to place, to examine various parts without doing any editing. Scrolling and long pattern scans are expensive in both computer time and real (user) time in these cases, and we describe here another method of traveling. The user of FRESS may assign unique names ("labels") to specific points in his file (Fig. 10a) and then access these points by reference to these names (Fig. 10b). In large files with many labels, the user can use either an alphabetized table or file sequence table of labels (displayed upon request) as an online table of contents to indicate important places (e.g., the beginning of each logical section) in the file.

In addition to providing the ability to access points in a file directly (this method of retrieval is called direct-access traveling, as opposed to linear traveling), FRESS automatically "remembers" the point of origin from which such traveling takes place. Thus one can take "side trips" for specific information and return to one's original location to continue editing or reading (Fig. 11).

With the introduction of direct-access traveling, we may now utilize another feature of FRESS. Originally, as mentioned before, the text in a file is one continuous stream (with embedded format codes such as paragraphs). The user may, however, wish to separate ideas or sections in his file so that they do not run together.

For example, the "work space", is a system-defined area (such system areas are called spaces), which may be used as a temporary storage and collection area. The functions move to work, copy to work, move from work, and copy from work allow pieces of text to be collectively added to the work area for viewing, editing, and formatting. The entire contents of the work area may then be moved or copied to some place in the user's file. The value of the work area is to allow "boilerplating", that is, constructing new documents from segments picked out of existing documents.

FRESS permits a user to structure a file with a decimal outline of sections, such as used in this article. FRESS will automatically renumber the section numbers as sections are inserted, deleted, and moved. It will also update references to sections (such as, "see Section 3.1.4") automatically.

Often repeated format codes or other text strings can be defined with special "format macros". Then a user can input the format macro in place of these text strings. Global changes can be effected by merely changing the definition of the format macro. Frequently repeated sequences of FRESS commands can be combined into "command macros". This greatly facilitates certain otherwise tedious editing tasks.

Other FRESS "shells" are oriented to online text creation and browsing by authors, using large screen display consoles with lightpens (a device connected to the computer which senses light given off by the display). Among the features currently implemented are:

- 1) keyword tagging of sections of text for subsequent Boolean retrieval, e.g., "tropical fish or reef fish and not barracuda",
- 2) linking areas and files into a "hypertext" with jumps and splices (useful for maintaining separate versions of the same manuscript, or for computer guided browsing via jump "menus" offering choices).

A special version of FRESS developed under grants from the EXXON Education Foundation and the National Endowment for the Humanities permits use of multiple windows on large screen graphics consoles and allows cross editing between these windows displaying different portions of the file, line drawing insertion and editing capability, and other advanced features. English 16 at Brown, "Critical Analysis of English and American Literature: Poetry," has twice used these features to aid instruction on a specially purchased display console (IMLAC PDS-1D). FRESS is also used by Brown's Placement Office to match job applicants to job lists, with the keyword retrieval facility.

5. AFTERWORD - WHO SHOULD USE FRESS?

FRESS was introduced as a means of assisting document preparation. By now, we hope an appreciation has been developed that document preparation consists of more than merely correcting typos and printing a nice looking copy. Rather it is the process of compiling, arranging, distilling, and properly referencing many "bits and pieces" of information. The speed with which one can enter (or collect) and subsequently retrieve such information, not the speed with which one can correct mistakes, is the limiting factor in document preparation. FRESS is unique in its ability to perform these functions quickly and efficiently, while retaining sufficient flexibility to allow each user to utilize the system to the fullest of his ability.

The current implementation of FRESS utilizes primarily typewriter consoles and small (980-1860 character) alphanumeric displays connected to a timesharing system. Typewriter terminals are used primarily for editing rather than browsing through files, since typing speed limits the amount of information that can be transferred back and forth in a reasonable amount of time. Because of this, FRESS on typewriter terminals is "secretary oriented"; that is, it is used more by people concerned with composition and corrections than those actively engaged in

"creative" writing (i.e., not more than they would use typewriters presently). While FRESS runs on such inexpensive, limited capability consoles, it is designed to utilize the more powerful display consoles like the IMLAC PDS-1B for which its predecessor, the Hypertext Editing System⁴, was designed. Such a terminal is very similar to a television console (in display capabilities, not in function). At a single time, the entire face of the terminal can be filled with text (20-30 lines), and any point in the text can be indicated to the system by pointing at it with a light pen.

Since the display can be regenerated almost instantly, this terminal is ideal for browsing (i.e., display text as fast as anyone could assimilate or even scan), and the light pen greatly increases editing speed. Using such a terminal, experience has shown that a person writing a document can compile and correct it for himself (except for minor typos, etc.) as fast, or faster than he could explain to someone else how to do it for him. On such a terminal, therefore, all the powerful information accessing facilities of FRESS can be put to use by the person needing them most, and the fullest capability of the system is realized. It is envisioned that such "softcopy" display terminals will become a regular fixture in the offices of professors, executives, lawyers, journalists, professional writers, etc., as a means of capturing thoughts and ideas, and of providing access to a wide "data base" at computer speeds.

FRESS's greatest value is for those who use the system early in the creative process. FRESS can be used effectively for these purposes on the typewriter and small display consoles. Typically an author will compose and then edit documents using a mix of online display and offline printouts. Secretaries are completely bypassed in this process. Others find it is efficient to dictate an initial draft to a secretary who inputs the material into FRESS files. Subsequently the author can edit these documents online.

Those who have documents near completion may find other editors or methods more efficient.

6. REFERENCES

FRESS User's Guide (August 6, 1975)

Describes the basic use of FRESS, including command specification, inputting, editing, and formatting. A

⁴Carmody, S.T., et al.: "A Hypertext Editing System for the /360, Pertinent Concepts in Computer Graphics, University of Illinois, Urbana, Illinois, 1969.

complete list of format codes and special characters is given; no structure is discussed.

FRESS Reference Manual -- Structure and Commands (2nd printing, January 15, 1976) [FRM]

Contains a complete list of FRESS commands and describes the creation and use of structure (labels, jumps, decimal blocks, etc.).

A FRESS Guide to Large Document and Thesis Preparation (in press)

A 20 page document with advice for those preparing large documents.

FRESS NEWSLETTER, Vol. 2, No. 1, March 3, 1976, FRESS Command Macros

Contains a complete description of the Command Macro facility.

FRESS NEWSLETTER, Vol. 2, No. 2, November 8, 1976, 1976, FRESS Display Keywords

Contains a complete description of the Display Keyword facility.

All of these documents are available in Computer Center document racks. The Reference Manual, and the User's Guide can be purchased in the Brown Bookstore. The Newsletters are available for printing with the FRESS command ".frsnews".

For further information about FRESS contact the User Services Staff at the Computer Center or the office of Prof. Andries van Dam, Program in Computer Science (401/863-3088).

7. EXAMPLES

In the examples below, command names are written out with their upper-case abbreviation followed by the rest of the name of the command in parentheses (upper-case is used only for clarity in these examples - the command may also be typed in lower case); generic names to be specified by the user are enclosed in angle brackets (<>). Machine (FRESS) response of more than one line is double spaced for legibility.

FIG. 1 - LOGGING IN

Syntax: (dependent on terminal and operating system)

Machine: FRESS READY

FIG. 2A - ACCESSING FILES

Syntax: G(et) F(ile) <filename> <password>

User: GF/usrguide/zebra

Machine: (keyboard is unlocked)

Explanation: GF is the command for "get file" (i.e. access an existing file). The file name is "usrguide" and its password is "zebra". The filename and password are correct, so FRESS returns with a green proceed light and an unlocked keyboard, positioning the user's display at the start of the file. When the user wants to read or edit the file, he will have to "scroll down" (Fig. 3b).

FIG. 2B - CREATING FILES

Syntax: M(ake) F(ile) <filename> <password>

User: MF/progrprt/smith

Machine: INPUT

Explanation: MF is the command for "make file" (i.e. create a new text file). The new (previously unused) file name is "progrprt", and the user has assigned a password "smith" to it. The machine acknowledges (by typing "INPUT") that it

has created the file and that it is ready to accept input to it.

FIG. 3A - SETTING THE DISPLAY WINDOW⁵

Syntax: S(et) D(isplay) <no. of lines> <no. of characters>

User: SD/4/55

Machine:

has called for a mass descent upon the Coachella Valley of California June 6 to completely halt grape picking at farms still resisting the five-year-old strike by the United Farm Workers. In a statement of solidarity

Explanation: The user had typed in a magazine news story and now wants to edit starting at this point in the file. Here he has chosen 4 lines of display of 55 characters each (one would typically, however, choose 1 on a typewriter console and 10 or more on a display console).

FIG. 3B - SCROLLING

Syntax: <number>

User: 2

Machine:

at farms still resisting the five-year-old strike by the United Farm Workers. In a statement of solidarity with the striking farm workers, MECHA declared, "We will stop those grapes of wrath from leaving Coachella

Explanation: Given the current window (shown in Fig. 3a), the user issues the command to scroll forward two lines (he

⁵To save space, the updated window in this and ensuing figures will be the starting window of the next figure unless otherwise stated.

need only type in "2", since scroll is the assumed command, and forward is the assumed direction. To scroll backwards 2 lines, say, the user would type in "-2".) The new display shows the window advanced by two lines.

FIG. 3C - PATTERN SCAN

Syntax: L(ocate) <modifiers>⁶ <pattern>

Simple Example: To locate "June" the user would indicate
"L/June".

More Powerful Example: LB/MECHA

Machine:

MECHA (Movimiento Estudiantil CHicano de Aztlan) has
called for a mass descent upon the Coachella Valley of
California[†] June 6 to completely halt grape picking at
farms still resisting the five-year-old strike by

Explanation: The user wishes to use the locate command (L) to locate the first occurrence of the word "MECHA" going backwards from the current window (LB). The locate is successful and the new window begins at the point at which the pattern is found (note that since there are no internal line boundaries, the lines are slightly different from those in the previous window). (Points in the text where editing will be performed in the next example are indicated by daggers.)

FIG. 4 - INSERT

Syntax: I(nsert) <point of insert> <text to be inserted>

User: I /fornia/ (located east of Los Angeles)

⁶Modifiers may be optionally concatenated to the "L" to indicate special forms of pattern scanning.

Machine:

MECHA (Movimiento Estudiantil CHicano de Aztlan) has
called for a mass descent[†] upon the Coachella Valley of
California (located east of Los Angeles) June 6[†] to
completely halt grape picking at farms still resisting

Explanation: The user indicates that he wishes to make an insert (I) after the word "California" ("fornia" is sufficient context and "nia" would also have sufficed; the last character of the context string is taken as the point desired). (In this example the user chose to use "/" to delimit his parameters; any other graphic character not occurring in the parameters could have been used as a delimiter.) The machine types the new window showing the updated version of the text. Note again that one may edit anywhere in the displayed window or below it, not merely in the top line (which is usually the only line "displayed" on a typewriter console).

FIG. 5 - DELETE

Syntax: D(elete) <text to be deleted>⁷

Simple Example: To delete a short string such as the word "completely" the user would type "D/completely".

More Powerful Example: D/upon...6þ

Machine:

MECHA (Movimiento Estudiantil CHicano de Aztlan) has
called for a mass descent[†] to completely halt[†] grape
picking at farms still resisting the five-year-old
strike by the United Farm Workers. In a statement of

⁷<text to be deleted> can be specified as a literal string or by two strings with an ellipsis in the middle.

Explanation: The user indicates that he wishes to delete (D) the text from the word "upon" (the u in the context string upon) to the blank (represented as ␣) following the six ("6") inclusive. He uses the "..." convention which has the usual meaning of defining a string starting with the leftmost character and ending with the rightmost character. Note how easy it is to delete large segments which may extend over more than one line - a great improvement over line-oriented editors.

FIG. 6 - SUBSTITUTE

Syntax: S(substitute) <old text string> <new text string>

Simple Example: To substitute "group" for "mass", the user types "S/mass/group".

More Powerful Example:

S /to...halt/involving students and workers to sabotage

Machine:

MECHA (Movimiento Estudiantil CHicano de Aztlan) has

called for a mass descent[†] involving students and

workers to sabotage grape picking[†] at farms still

resisting the five-year-old strike by the United

Explanation: The user does a substitute (S) and types in the string to be substituted. Note that FRESS automatically adjusts the text to accommodate the new text string regardless of its size, rather than truncating the text if it exceeds a certain maximum, as line editors do. Again the "..." is used to reduce the effort required to specify text strings.

FIG. 7 - MOVE

Syntax: M(ove) <text to be moved>⁸ <new location>

⁸As in delete, the text to be moved can be indicated by a simple literal string.

Simple Example: To move "beagle" after "This" in the sentence "This is a good family beagle dog.", specify:

"M /beagle /This ".

Machine: This beagle is a good family dog.

More Powerful Example: M / at...Workers/descent

Machine:

MECHA[†] (Movimiento Estudiantil CHicano de Aztlan)[†] has
called for a mass descent at farms still resisting
the five-year-old strike by the United Farm Workers
involving students and workers to sabotage grape

Explanation: The user indicates that he wishes to move the text running from the space before the word "at" (after the word "picking") to the end of the sentence (the "s" in "Workers"). The user indicates that this string of text is to be moved after the word "descent", and this change is shown in the ensuing FRESS reply. Note that in one operation the same thing has been accomplished (with far less typing) that would take three or more operations in a line editor.

FIG. 8 - COPY

Syntax: CO(py) <text to be copied> <new location>

Simple Example: "CO/MECHA/and ", would copy "MECHA" after the first occurrence of "and".

More Powerful Example: CO / (...)/2|MECHA

Machine:

MECHA (Movimiento Estudiantil CHicano de Aztlan) has
called for a mass descent at farms still resisting
the five-year-old strike by the United Farm Workers
involving students and workers to sabotage grape

Explanation: The user indicates that he wishes to copy (CO) the words in parentheses (including the blank before the

parenthetical expression) so that they will follow the first occurrence of "MECHA" starting at line two (i.e. the second occurrence of "MECHA" in the buffer. This is indicated by the "2|" preceding MECHA in the command). The window remains unchanged since copy does not affect the original text, and since the new copy of the text is not visible in the current four-line window. If the user were to scroll forward a few lines, the change would be seen.

FIG. 9 - SOME FORMATING

User: I

Machine: INPUT

User:

```
!-s0-Roll along asphalt, press down on the pedals,  
!-s0-speeding, fearless,!-i19-ring,!-i24-ring,  
!-i29-ring!-s0-get clear of Moscow, frighten a  
one-eyed cock!-s0-with a broken tail
```

Explanation: Since there is more than one line to type in, the user goes into Input Mode (I), and then types in the text. Format codes are inserted as if they were regular text, and are differentiated by an initial "!". On the printout, the text would skip 0 lines (-S-) (this in effect starts a new line) at the word "speeding". Next there would be three indents (-I-) of 19, 24, and 29 spaces respectively for each of the succeeding "rings" (note that each indent also starts a new line). Next there would be a new line at the word "get" and at the word "with". On final fully-formatted printout, the text would appear as:

```
Roll along asphalt, press down on the pedals,  
speeding, fearless,  
    ring,  
        ring,  
            ring  
get clear of Moscow, frighten a one-eyed cock  
with a broken tail
```

FIG. 10A - MAKING A LABEL

Syntax: M(ake) L(abel) <attachment point> <label name>

Previous Display: Here we show a label surrounded by text.

User: ML/Here/sample label

Machine:

Here

%L(sample label) we show a label surrounded by

Explanation: The user inserts, after the word "Here" in the line displayed, the label "sample label". The "%" in the machine response is the FRESS internal structure delimiter and "L" specifies that a label (enclosed in parentheses) follows. (Note, although the label starts a new line when it is displayed, when the file is printed offline, neither the label nor a line start will appear.)

FIG. 10B - GET LABEL

User: GL/sam

Machine:

%L(sample label) we show a label, surrounded by text.

The user can access any labeled point in his file instantaneously. He can use labels to make an online table of contents, say, one label per section.

Explanation: The display starts at the labeled point; only a unique initial substring of the label need be specified.

FIG. 11 - RETURN

Syntax: R(eturn)

User: GL/second

Machine:

%L(second shell)

!-h2-4.2 Shell Two: Direct Access and Retrieval

In large files it is cumbersome to travel linearly between widely separated points, especially when

User: R

Machine:

%L(sample label) we show a label, surrounded by text.

The user can access any labeled point in his file
instantaneously. He can use labels to make an online
index, say, one label per section.

Explanation: Looking at the display of Fig. 10b, the user wants to jump to the description of the FRESS second shell. After some reading and scrolling, perhaps, he returns to the display of Fig. 10b. (The "!"-h2-" is the format code for a heading.)

TABLE OF CONTENTS

1. Introduction.....	1
2. The System Environment.....	3
3. Other Computer-Assisted Text Editors.....	4
4. The FRESS Shells.....	5
4.1 Shell One: Basic Commands.....	5
Logging into the system:.....	5
Reading, accessing, and storing text files:.....	5
Linear traveling:.....	6
Simple editing commands:.....	7
Simple formatting:.....	8
Specifying text points:.....	8
4.2 Shell Two: Direct Access and Retrieval.....	8
5. Afterword - who should use FRESS?.....	10
6. References.....	11
7. Examples.....	13
Fig. 1 - Logging In.....	13
Fig. 2a - Accessing Files.....	13
Fig. 2b - Creating Files.....	13
Fig. 3a - Setting the Display Window.....	14
Fig. 3b - Scrolling.....	14
Fig. 3c - Pattern Scan.....	15
Fig. 4 - Insert.....	15
Fig. 5 - Delete.....	16
Fig. 6 - Substitute.....	17
Fig. 7 - Move.....	17
Fig. 8 - Copy.....	18
Fig. 9 - Some Formatting.....	19
Fig. 10a - Making a Label.....	19
Fig. 10b - Get Label.....	20
Fig. 11 - Return.....	20